AH 2003 : workshop on adaptive hypermedia and adaptive web-based systems

AH2003: Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems

Twelfth International World Wide Web Conference,
Budapest, Hungary, May 20, 2003
Ninth International Conference on User Modeling,
Johnstown, Pennsylvania, USA, June 22, 2003
Fourteenth Conference on Hypertext and Hypermedia
Nottingham, UK, August 26, 2003

Background and Motivation

The research area of *adaptive hypermedia* originates from the coming together of the areas of *hypertext and hypermedia*, *user modeling* and *Web-based information systems*, all of which have embraced the topic of adaptation.

Under different names workshops have been organized, mainly at the world Wide Web Conference (1999), the user Modeling Conference (1994, 1996, 1997, 1999, 2001) and the ACM Hypertext Conference (1997, 1998, 2001). The Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems provides a forum for researchers to discuss their on-going work, in preparation of the next Adaptive Hypermedia Conference in 2004.

At the AH2003 workshop paper presentations are combined with discussions on the presented topics. The presentations are explicitly intended to provoke discussions. The papers included in the workshop are not finished research papers, but papers that present research challenges and directions. The papers at this workshop have been selected by a program committee consisting of leading adaptive hypermedia researchers.

Program

AH2003 Program Committee

The AH2003 workshop is organized by a committee consisting of a general chair, a co-chair for each of the three sessions, and members that helped select the papers to be presented at the conference. After the WWW2003 session two more workshop sessions will be held: one at the User Modeling Conference in June 2003 and one at the ACM Hypertext Conference in August 2003.

General chair:

Paul De Bra, Eindhoven University of Technology.

Session co-chairs:

- WWW2003 session: **Hugh Davis**, University of Southampton
- UM2003 session: **Judy Kay**, University of Sydney
- HT'03 session: **monica schraefel**, University of Southampton (formerly at the University of Toronto)

Members:

Lora Aroyo (Eindhoven University of Technology)
Peter Brusilovsky (University of Pittsburgh)
Ricardo Conejo (University of Malaga)
Alexandra Cristea (Eindhoven University of Technology)
Fiorella De Rosis (University of Bari)
Serge Garlatti (ENST, Brest)
Franca Garzotto (Politecnico di Milano)
Nicola Henze (Institut für Technische Informatik, Hannover)
Paul Maglio (IBM Almaden Research)
Tanja Mitrovic (University of Canterbury, New Zealand)
Wolfgang Nejdl (University of Hannover)
Vittorio Scarano (University of Salerno)
Daniel Schwabe (Catholic University of Rio de Janeiro)
Barry Smyth (Changingworlds and University College Dublin)
Julita Vassileva (University of Saskatchewan)
Gerhard Weber (Pedagogical University of Freiburg)
Ross Wilkinson (RMIT University, Australia)

A Semantic Web Approach for Adaptive Hypermedia

Serge Garlatti and Sébastien Iksal
Department of Artificial Intelligence and Cognitive Sciences
ENST Bretagne - Technopôle de Brest Iroise
B.P. 832, 29285 Brest Cedex, France
{serge.garlatti, sebastien.iksal}@enst-bretagne.fr

Abstract

Adaptation/personalization is one of the main issues for web services. Adaptive web applications have the ability to deal with different users' needs for enhancing usability and comprehension and for dealing with large repositories. Indeed, adaptive web services - also often called Adaptive Hypermedia Systems - can provide different kinds of information, different layouts and different navigation tools according to users' needs. We propose an open-ended adaptive hypermedia environment which is based on the virtual document and semantic web approaches and which is able to manage adaptive techniques at knowledge level. The aim is to simplify the creation and the management of adaptive web services by using ontologies and semantic properties for adaptation. Indeed, they are declarative parameters for computing on the fly services. Indeed, the specification of the adaptive mechanisms is defined by semantic properties associated to a hypermedia document by an author. These properties have the following roles: define how to evaluate the links/content for grouping them together in different classes according to a user model, determine how to manage these classes for each adaptive technique and assign user stereotypes to adaptive techniques. Then, an author can determine the relevant adaptive techniques for a given user group.

Keywords

Adaptive navigation and presentation, Virtual Document, Composition Engine, Semantic Web

1. Introduction

Nowadays, numerous services are available on the Web, for instance, portals, e-learning, digital libraries, on-line information systems, virtual museums, e-business and digital newspapers are current services. Adaptation/personalization is one of the main issues for web services. Adaptive web services have the ability to deal with different users' needs for enhancing usability and comprehension and for dealing with large repositories. Indeed, adaptive web services - also often called Adaptive Hypermedia Systems - can provide different kinds of resources, different navigation tools and different layouts according to users' needs [14]. The creation and/or maintenance of adaptive web services from these repositories require the following features: i) methods to facilitate web service creation and management and ii) reuse, sharing and exchange of resources through the internet/intranet iii) selection of the relevant resources and their organization according to user's needs.

The new age of Internet is the Internet of meanings and provides some of these features. The Semantic Web is a vision for making the contents of the web understandable to the machines. It would be a basis for creating intelligent web services in the future. Then, this third web generation has the ability to enhance information retrieval, the reuse, sharing and exchange of resources through the internet/intranet and to deal with automatic or semiautomatic services. Indeed, it is well known that keyword-based information access presents severe limitations concerning precision and recall. On the contrary, intelligent search engines, relying on semantic web initiative [14] and semantic metadata, overcome these limitations [14]. Nevertheless, information space is so huge that it is not sufficient to have a precise search engine. It is necessary to take into account user interests and to have an accurate metadata schema to be sure to focus on relevant pieces of information. Adaptive Hypermedia systems can be viewed as automatic or semiautomatic services dealing with different users' needs and distributed resources. This kind of services can generate dynamically a service adapted to the users.

It is not sufficient to rely on the semantic web initiative. Indeed, flexible hypermedia and more particularly that of virtual documents can lead to methods facilitating web service design and maintenance. According to Watters, "A virtual document is a document for which no persistent state exists and for which some or all each instance is generated automatically at run time" [14]. They have the ability to select the relevant resources and their organization according to user's needs. A virtual document, as a service, is composed of an information space – resources - and mechanisms – at least a composition engine - to compute on the fly numerous different real documents from a specification. The service maintenance may be done by changing the specification. Moreover, a composition engine may also be used for creating new services as soon as these services are compatible with the principles underlying the composition engine. These principles are implemented in the composition rules for selecting resources according to a metadata schema and organizing them. For instance, if you have a composition engine able to compute a concert summary of ten minutes, you could reuse this composition engine for a TV show summary. It is possible whether the video segmentation, selection and organization principles are suitable for these shows.

We have designed an open-ended adaptive hypermedia environment which is based on the virtual document and semantic web approaches and which is able to manage selection, organization and adaptation at knowledge level. Virtual document and adaptive hypermedia are closely related – they can be viewed as the two faces of the same coin. At present, we have focused our study on five adaptive navigation techniques (direct guidance, annotation, hiding, sorting and partial hiding) from which adaptive content can be deduced. The specification of the adaptation mechanisms is defined by semantic properties associated to an adaptive document by an author. These properties have the following roles: define how to evaluate the links/content for grouping them together in different classes according to a user model, determine how to manage these classes for each adaptive technique and assign user stereotypes to adaptive techniques. Then, an author can determine the relevant adaptive techniques for a given user group. Indeed, some experiments have shown that it is necessary to provide the relevant adaptive techniques to the current user [14]. For instance, annotation technique is advised for expert and hiding technique for novice.

Adaptive documents rely on some principles which are firstly presented. Secondly, we define the different views of digital document and the corresponding architecture for our adaptive hypermedia environment. Thirdly, the adaptation will be analyzed via our adaptive semantic composition engine. Fourthly, the composition of the delivered document is presented. Finally, we conclude by some perspectives.

2. Design Principles

In our framework, we consider an adaptive hypermedia as an adaptive virtual document. We define it as follows: an adaptive virtual document consists of a set of information fragments, their corresponding metadata, different ontologies and a composition engine. The latter selects and/or filters the relevant information fragments[1] – resources -, organizes and assembles them according to an author/designer specification. Composition and specification are the two stages of an adaptive virtual document. The **composition** consists of three functions which define the composition rules: the **selection** which retrieves the relevant set of fragments, the **organization** which provides an overall document structure and the **assembly** which defines the layout of the document. The organization determines how to access the relevant set of fragments and can be computed on the fly or specified by an author/designer. We are interested in author-oriented adaptive virtual documents. An author-oriented document has the following characteristics: authors have know-how which enables them to give coherence to a document. This coherence depends on the content and its organization according to user's needs. In such a framework, an author has to specify the content by means of the metadata and one or more organizations for this potential content. He also has to specify the different "rules" for adaptation. We call the outcome of this specification a **generic document** from which several adapted documents will be generated on the fly.

In a digital document, three different views may coexist: semantic, logical and layout [14]. These views are closely related to the semantic web architecture: i) semantic: logic, ontologies, RDFS/RDF, ii) logical: XML, iii) layout: XSL/XSLT. First of all the three views are presented. Secondly, our adaptive composition engine architecture based on these three views is analyzed.

2.1 Different views of a document

The three views have a specific structure organizing them. The semantic structure of a document conveys the organization of the meaning of the document content. As resources are distributed through Internet, there is no content at all in the semantic structure, but only a specification of the potential content. The semantic structure - as an overall document structure - plays the role of a site map in a static hypermedia document or the role of the "knowledge tree" in the approach of P. Brusilovsky [14].

The logical structure reflects the syntactic organization of a document. A document (for example books and magazines) can be broken down into components (chapters and articles). The logical view fits the syntactic level of the semantic web architecture. A logical structure is encoded in XML and represents a web page [14]. The layout view describes how the documents appear on a device and a layout structure describes it, (e.g. size and color of headings, texts, etc). An adaptive composition engine relying on these three views has been designed. In this paper, we focus on the semantic view and the corresponding adaptation mechanisms.

2.2 Adaptive Composition Engine

The adaptive composition engine architecture is based on two different studies: ICCARS[2] Project and CANDLE[3] Project which is a European project. Our composition engine is divided into three engines: semantic composition, logical composition and layout composition (cf. figure 1). They are sequential processes according to the three views described previously. In a virtual document framework, the three functions are distributed in these processes as follows: selection and organization are achieved in the semantic composition; assembly is divided into logical and layout compositions. The aim of the semantic composition engine is

1 Fragments are reusable units. Then, they have their own metadata. Fragments can be atomic or abstract. Atomic fragments are information units and cannot be decomposed. Abstract fragments are composed of atomic fragments and/or abstract fragments and one or more semantic structures.

2 ICCARS : **I**ntegrated and **C**ollaborative**C**omputer **A**ssisted **R**eporting**S**ystem (http://iccars.enst-bretagne.fr)

3 CANDLE : **C**ollaborative**A**nd**N**etwork **D**istributed **L**earning **E**nvironment (http://www.candle.eu.org)

to compute an adapted document from an information space, a user model and a generic document. The latter is defined by an author and is a narrative structure for presenting a particular viewpoint on a set of articles or course elements. It is a directed graph in which nodes have a content specification according to a metadata schema and edges are semantic relationships. These relationships belong to those analyzed by Rhetorical Structure Theory (RST)[4] [14]. The adapted document is an instance of the generic document in which content specification is replaced by fragments matching it. In our framework, the information space is reduced to a small subset of fragments to ensure the document coherence. This subset is defined by an author and is associated to the generic document. Nevertheless, it is possible to release this constraint and to provide access to internet/intranet on demand.

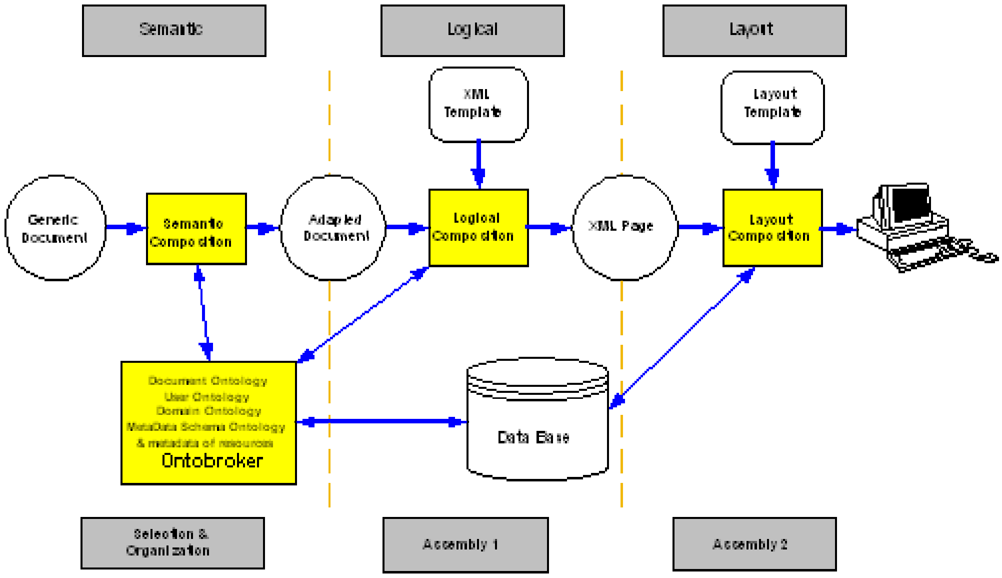


Fig. 1. The Composition Engine Architecture

The logical composition engine browses the adapted document according to user interactions and computes for each node an XML web page with content and navigation tools. An XML template and a user model are used to get this XML web page. In navigation tools, links have properties for managing adaptation [14]. The layout composition engine generates an HTML page from the XML web page in applying the layout rules given by the layout template and using an XSLT processor [14]. Adaptation processes can take place in the three engines. But we focus on the first one: the semantic composition engine.

For generating documents, the composition engine requires different types of models: the document model, the user model, the domain model and the metadata model. All these models are represented at a knowledge level by means of ontologies which are always organized – related – in the same way. They are separated to simplify their maintenance. As these ontologies are loosely coupled, they can be modified without effort. Consequently, it is easier to maintain adaptive documents and to design new adaptive documents.

The document model formalizes the specification of the generic document and its properties. It represents the author’s competences and know-how for creating adaptive document. It defines all the different categories of fragments which may be specified in a generic document (for instance, news in-brief, chronicle, press review, editorial, record, research, poll, news items, etc.), their relationships and the RST relationships. The generic document can be considered as an instance of the document model. It also defines the different adaptive techniques and their properties. The domain model defines all the concepts and their relationships in a particular domain (for instance, for special reports about fishing, the

4 The main goal of the Rhetorical Structure Theory is to give coherence to a document;

different categories of fish, vessels and their equipments and tools, people, etc. are defined). The user model is composed of **personal data** (identity, age, town, professional activity, role (reader, author, etc.), **history**, **session**,**preferences** information about interests and preferred adaptive techniques, **knowledge level** about the domain concepts – it is an overlay model. The metadata schema is composed of six parts : **General**: identification of the resource : title, authors, etc. ; **Lifecycle**: information about the management of versioning, number, date, authors, etc.; **Meta Metadata**: Information about the metadata, creator, validator, language, date, etc.; **Rights**: Information about use, cost, license, access restrictions, etc.; **Technical**: Information about format, size, software version, etc.; **Classification**: Description of the content : domain concepts, necessary knowledge level, etc.. The metadata schema is related to the document model for specifying the type of fragments, the domain model for defining the main concepts of a fragment. For designing a generic document an author as to use a subset of the metadata entries for specifying the potential content. In the ICCARS project, an author chooses the relevant fragments for each node of the generic document and the authoring tool will deduce the corresponding specification in a limited information space to ensure semantic coherence. The metadata used to specify the content are as follows: MD.6.1.1and MD.6.2.1 in table 1.

MD.6	Classification		Domain value
MD.6.1	Domain	List	
MD.6.1.1	Concept	Unique value	Domain ontology
MD.6.1.2	Level	Unique value	User Ontology : knowledge level
MD.6.2	Application	Unique value	
MD.6.2.1	Type of fragment	Unique value	Document Ontology: types of fragments

Table 1. Some metadata specifying the content

Potentially, a subset of fragments – a small one – can match the specification. These fragments are called variants of fragments. Some of their other metadata entries differ. They will be used by the adaptation process for evaluating them. Then, another subset of the metadata entries will be reserved for variants of fragments. These two subsets of metadata entries are mutually exclusive and depend on the application.

The semantic and logical composition engines rely on OntoBroker. It is used to browse the different models – ontologies – and to retrieve fragments which match the content specifications by means of the fragment metadata included in [14]. OntoBroker contains four ontologies [14] – one per model - and metadata closely related to them. These ontologies are: a domain ontology for representing contents; a metadata ontology at the information level which describes the indexing structure of fragments; a user ontology which may define different stereotypes and individual features; a document ontology which represents the author’s competences and know-how for creating adaptive document [14]. Ontobroker also contains resources’ metadata for information retrieval. These ontologies are parameters for the composition engine and their organization and relationships have to be stable across services for ensuring the composition engine reusability.

3. Adaptation Specification

The main goal of adaptation in a hypermedia document is to provide the relevant navigation tools, information units and organization. The specification of the adaptation mechanisms is defined by semantic properties associated with a generic document by an author. Then, it can be used in various contexts which require different adaptation mechanisms.

3.1 Principles

Five adaptive navigation techniques are managed by our system: annotation, direct guidance, hiding, partial hiding and sorting [14]. Adaptive content is deduced from them. The properties specifying the adaptation mechanism have the following roles: define how to evaluate the links/content for grouping them together in different classes, determine how to use these classes in the different adaptive techniques and assign user stereotypes to adaptive techniques. Then, an author can determine the relevant techniques for a given user group. First of all the evaluation method is presented. Secondly, the use of the evaluation method for adaptive navigation techniques is analyzed. Thirdly, the association of stereotypes to adaptive techniques is presented.

3.2 Evaluation method

For adaptive navigation techniques, one can evaluate the relevance of the links or the relevance of the destination of links – fragments. In our framework, we propose to evaluate the fragments by means of a uniform evaluation method for all adaptive techniques. Indeed, we propose to define up to five disjoint and totally ordered classes of fragments – at least two classes. They are classified by an evaluation of the destination fragments of the links, that is to say the variants of fragments matching the content specification of each generic document node. These classes are called: very good, good, average, bad, very bad. An author defines the necessary and sufficient conditions for class membership. We have chosen to have up to five classes for user comprehension. Indeed, it could be difficult for a user to deal with too many classes. For instance, several studies have proposed up to five different classes for managing annotation [14], but not more.

The rules for class membership are defined by Boolean expressions composed of unary and binary operators (not, and, or) and comparators (= | < | > | <= | >= | <> | in) between objects. These objects can take into account a user stereotype – user model features coming from the user ontology -, a subset of metadata dedicated to variants of fragments – metadata ontology – and a knowledge level about domain concepts – computed from the metadata and the user model – percentage of known concepts or percentage of concepts having a sufficient level of knowledge. A rule is a condition which has to be satisfied by the fragment and the user model to be a member of the corresponding class. Four examples and an explanation:

- Very Good = "(JOB = Fish wholesaler) AND (PCTAGEKNOWN >= 100)"
- Good = "((JOB = Fisherman) AND (PCTAGEKNOWN >= 75)"
- The fragment is a member of the class “Good” if the user is a fisherman and he knows at least 75% of the domain concepts which defined the semantic content of the fragment.
- Average = "(JOB = Fisherman) AND (PCTAGEKNOWN < 50)"
- Bad = "(NOT (JOB = Sea Job))"

In this example, four different classes have been defined which are mutually exclusive and there are always rank in the same way (Very Good > Good > Average > Bad). The first type of conditions - (JOB = Fish wholesaler) - are compared to the current user model and the second type of conditions – (PCTAGEKNOWN < 50) - are used to compare the knowledge property – overlay model - of the current user model and the domain concepts representing the fragments in metadata. Each fragment matches the content specification of the current node of the generic document is classified in one of these classes according to the current user model.

For the knowledge level, two cases are distinguished: the known concept and the concept having a required knowledge level. Indeed, an author may specify a minimal knowledge level to each relevant concept describing the content. The computed elements are:

- PCTAGEKNOWN: percentage of known concepts.
- PCTAGEGOODLEVEL: percentage of concepts having a sufficient level of knowledge.

3.3 Evaluation management for adaptive navigation techniques

For managing the different adaptive navigation techniques, it is necessary to define how to manage the different classes of fragments. It is sufficient to decide which classes are kept or suppressed for a given technique. For instance, direct guidance, annotation and sorting are managed as follows: i) direct guidance: the best class is kept and the others are suppressed, ii) annotation: all classes are kept and links will be annotated according to the class relevance, iii) sorting: all classes are kept and links will be ordered according to the class order. For hiding and partial hiding, an author has to specify which classes are kept.

3.4 Stereotypes associated to adaptive navigation techniques

A user may have an adaptive navigation technique whether its stereotype matches the user model. All user model features can be used to define a stereotype. A stereotype is defined by Boolean expressions composed of unary and binary operators (not, and, or) and comparators (= | < | > | <= | >= | <> | in) between user model features. An example and its meaning:

- ANNOTATION = "((Age > 18) OR (JOB = Student)) AND (Location In Brittany)"
- Annotation is for user being 18 years old or being students in Brittany

The adaptive presentation is deduced from the applied adaptive navigation technique. Indeed, they are applied on links which are evaluated by the destination content, that is to say fragments. According to the current adaptive navigation technique, each evaluated fragment is presented to the user or not. When there is link filtering – hiding, partial hiding and direct guidance – some fragments are not presented because the corresponding links are suppressed. Then, there is a content adaptation – content adaptation - and a modification of the semantic structure of the document. For each generic document, the following semantic features are defined for adaptation management: the number of classes, membership rules for each class and the management of classes for hiding and partial hiding techniques and for each technique, a user stereotype.

4. Adaptive Composition

The main goal of the semantic composition engine is to compute on the fly an adapted document which is an instanciation of the generic document by selecting content and modifying the semantic structure. The semantic composition engine consists of three processes: the first one selects the relevant fragments for the current node, the second one evaluates the fragments and classifies them in the different classes specified by an author and the third one determines the allowed adaptive navigation techniques and applies them – that is to say kept or not some or all fragment variants.

1. First of all, the content specification is used to query the search engine – Ontobroker – and selects the relevant fragments from the information space associated to the generic document. The outcome of this process is a set of fragment variants. These fragments differ in a subset of the metadata schema, for instance, the knowledge level for some domain concepts, the technical properties (format, size, etc.), etc.

2. Secondly, all fragment variants are evaluated, that is to say each fragment metadata subset dedicated to variants is compared to the different fragment classes. Then, each fragment variant belongs to one class and it has a semantic property called “class member” whose value is “very good” or “good” or “average” or “bad” or “very bad”. This property will be used to manage adaptive techniques.
3. Thirdly, the user model is compared to all stereotypes associated with the different adaptive techniques (annotation, hiding, direct guidance, etc.). Those which fit the user model are kept. For instance, whether the current user model only matches the stereotype of the hiding and direct guidance techniques. they are available to the user. According to the enabled adaptive navigation technique, a fragment variant is kept or not in the adapted document. For instance, if a user is allowed to have annotation, hiding and direct guidance techniques, all fragment variants are kept whatever their class. Indeed, it is necessary to keep all fragments to be able to annotate them. On the contrary, if a user is only allowed to have hiding and direct guidance, the fragment variants belonging to the best class are only kept - class “Very Good”. The others are deleted because the user is not allowed to use them. Consequently, this kind of deletion leads to content adaptation. Some adaptive navigation techniques, like hiding, partial hiding or direct guidance, allow the removal of the irrelevant fragment variants whether they are the only techniques available. They have direct consequences on the document content and structure and then on content adaptation. Whether some fragment variants are deleted, the document content and structure may be modified.

The logical composition aims at computing the current web page structure – XML - with a content and navigation tools for accessing the rest of the adapted document. The navigation tools are the local and global guides for navigation [14] resulting from the available adaptive navigation techniques. Among the enabled adaptive navigation techniques, the user has to choose by means of his preferences one of them otherwise the default one is chosen. For the adaptive navigation techniques, author constraints have priority over user preferences. An XML web page is generated from an XML template. A template describes the logical structure of a web page but without any content or navigation tools. It contains queries for computing navigation tools and for loading the content. The content is given by the selected fragment in the current node of the adapted document. The navigation tools depend on the selected adaptive navigation technique. For defining the links in navigation tools, the logical composition engine has to browse the adapted document and then “translates” semantic relationships into hyperlinks. Let A, B be nodes and R1 be a relationship from A to B. As soon as B has several variants, the relationship R1 is considered as several relationships of the same type, one per fragment from the source A to each destination fragment in B. It has also to use the “class member” property of fragment variants to manage the current adaptive navigation technique, that is to say it has to decide which links are kept or not.

5. Conclusion and Perspectives

In this paper, we have presented an open-ended adaptive hypermedia environment which is able to manage selection, organization and adaptation at knowledge level. It is based on the virtual document and semantic web approaches. The composition engine uses four ontologies: document ontology, user ontology, domain ontology and metadata ontology. The first one defines the author know-how – how to select and to organize a document at knowledge level – and the available adaptation techniques and their properties. They are always organized – related – in the same way and are separated to simplify their maintenance. As these ontologies are loosely coupled, they can be modified without effort. Consequently, it is easier to maintain adaptive documents and to design new adaptive documents. These ontologies are parameters for the composition engine and their organization and relationships have to be stable across services for ensuring the composition engine reusability. They can also be reused, shared and exchanged through internet. The composition engine has been implemented with the adaptive mechanisms. At present, the authoring tool is only able to deal with metadata tagging, searching of fragments according to the metadata schema and non adaptive generic document. It is possible to define adaptive generic document but without the authoring tool.

Moreover, the specification of the adaptive techniques is defined by semantic properties associated to a generic document by an author. The semantic properties specifying the adaptation mechanism have the following roles: define how to evaluate the fragments for grouping them together in different classes, determine how to use these classes in the different adaptive techniques and assign user stereotypes to adaptive techniques. Then, an author can determine the relevant techniques for a given user group. The management of adaptive techniques is made at an abstract level for the entire document. Consequently, it is easier to generate new adapted documents because it is sufficient to modify these semantic properties. Then, an author can deal with new adaptive documents and new user behaviors without changing the generic document a lot. Nevertheless, the management of adaptation at document level does not enable an author to change the adaptation mechanism at a very fine level of granularity. For instance, it is not possible to specify for each link or content how to manage adaptation like in AHA.

Our evaluation can be used for adaptive information retrieval. Indeed, a user can specify the adaptation criteria for adaptive information retrieval on the basis of combination of the following features: his interests and knowledge according to two different adaptive techniques. For example, a user can ask the search engine to sort the outcome by interests - up to five subsets - and to annotate each subset by knowledge. This possibility has been applied in the ICCARS project.

6. References

1. Brusilovsky, P., Methods and techniques of adaptive hypermedia. User Modeling and User-Adapted Interaction, 1996. 6(2-3): p. 87-129.
2. Watters, C. and M. Shepherd. Research issues for virtual documents. in Workshop on Virtual Document, Hypertext Functionality and the Web. 1999. Toronto.
3. Specht, M. and A. Kobsa. Interaction of domain expertise and interface design in adaptive educational hypermedia in Second Workshop on Adaptive Systems and User Modeling on the World Wide Web. 1999. Toronto & Banff: Eindhoven University of Technology.
4. Christophides, V., Electronic Document Management Systems. 1998, UCH/FORTH: http://www.ics.forth.gr/~christop/.
5. Brusilovsky, P. and H. Nijhavan. A Framework for Adaptive E-Learning Based on Distributed Re-usable Learning Activities. in Proceedings of World Conference on E-Learning, E-Learn 2002. 2002. Montreal, Canada.
6. Mann, W.C. and S.A. Thompson, Rhetorical Structure Theory: Toward a functional theory of text organization. Text, 1988. 8(3): p. 243-281.
7. Bray, T., et al., Extensible Markup language (XML) 1.0, (Second Edition). 1998, W3C: http://www.w3.org/TR/2000/REC-xml-20001006.
8. Adler, S., et al., Extensible Stylesheet Language (XSL) Version 1.0. 2000, W3C: http://www.w3.org/TR/xsl/.
9. Brusilovsky, P., E. Schwarz, and G. Weber. ELM-ART: An intelligent tutoring system on World Wide Web. in Third International Conference on Intelligent Tutoring Systems, ITS-96. 1996. Montreal.
10. Fensel, D., et al. On2broker in a Nutshell. in the 8th World Wide Web Conference. 1999. Toronto.

11. Iksal, S. and S. Garlatti, Documents Virtuels et Composition Sémantique: Une Architecture Fondée sur des Ontologies. NimeTic'01. 2001, Ecole des Mines d'Alès: Nimes. 91-96.
12. Iksal, S., et al., Semantic composition of special reports on the Web: A cognitive approach, in Hypertextes et Hypermédia H2PTM'01, P. Balpe, et al., Editors. 2001, Hermès. p. 363-378.
13. De Bra, P. and L. Calvi, AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, 1998. 4: p. 115-139.
14. Garlatti, S., S. Iksal, and P. Kervella. Adaptive On-line Information System by means of a Task Model and Spatial Views. in Second Workshop on Adaptive Systems and user Modeling on the World Wide Web. 1999. Toronto and Banff: Eindhoven University of Technology.

Logically Characterizing Adaptive Educational Hypermedia Systems

Nicola Henze and Wolfgang Nejdl
University of Hannover
ISI - Knowledge Based Systems
Appelstr. 4 · D-30167 Hannover
http://www.kbs.uni-hannover.de/~{henze,nejdl}
{henze,nejdl}@kbs.uni-hannover.de

Abstract

Currently, adaptive educational hypermedia systems (AEHS) are described with nonuniform methods, depending on the specific view on the system, the application, or other parameters. There is no common language for expressing functionality of AEHS, hence these systems are difficult to compare and analyze. In this paper we investigate how a logical description can be employed to characterize adaptive educational hypermedia. We propose a definition of AEHS based on first-order logic, characterize some AEHS due to this formalism, and discuss the applicability of this approach.

1. Motivation

This paper aims at developing a logical characterization of adaptive educational hypermedia and web-based systems (AEHS). AEHS have been developed and tested in various disciplines and have proven their usefulness for improved and goal-oriented learning and teaching. However, these systems normally come along as stand-alone systems - proprietary solutions have been investigated, tested and improved to fulfill specific, often domain-dependent requirements. So far, there has been no attempt to define a common language for describing AEHS. We claim that such a shared language will support the analysis and comparison of AEHS, and, in addition, a comprehensible description of AEHS will encourage an extended use of adaptive functionalities in e-Learning. This is especially important with respect to the Semantic Web [13], and, associated, the Adaptive Web [4] which knows like a personal agent the specific requirements of a user, takes goals, preferences or the actual context into account in order to optimize the access to electronic information.

Bringing personalization to the Web requires an analysis of existing adaptive systems, and of course this holds for the special case of e-learning and education. In this paper, we propose a component-based definition of adaptive educational hypermedia systems. A functionality-oriented definition of adaptive hypermedia has been given by Brusilovsky, 1996 [1].:

Definition 1 (Adaptive hypermedia system) "By adaptive hypermedia systems we mean all hypertext and hypermedia systems which reflect some features of the user in the user model and apply this model to adapt various visible aspects of the system to the user."

The component-based definition proposed in this paper is motivated by Reiter`s theory of diagnosis [10] which settles on characterizing systems, observations, and diagnosis in first-order logic (FOL). We decompose adaptive educational hypermedia systems into basic components, according to their different roles in the system: Each adaptive (educational) hypermedia system is obviously a hypermedia system, therefor it makes assumptions about documents and their relations in a *document space*. It uses a *user model* to model various characteristics of individual users or user groups. During runtime, it collects *observations* about the user's interactions. Based on the organization of the underlying document space, the information from user model and from the system's observation, *adaptive functionality* is provided.

This paper is organized as follows: In the next section we give a first description of the components of an AEHS and explain their roles and functionality with examples. We then give a definition of AEHS based on FOL. Due to this formalization, an artificial AEHS with few adaptive functionalities is described in section 3, and four examples of existing AEHS. Due to space constraints, we can not present the complete descriptions in this paper but will only exemplary show the applicability of our definition. The detailed descriptions can be found in a Technical Report [6]. A synopsis of the results is given in section 4. We conclude with a discussion about the results of our logic-based characterization of AEHS.

2. Towards a Logic-Based Definition of AEHS

In this section we will first give a description of the components in AEHS and their roles. Afterwards we will give a formal definition of adaptive educational hypermedia systems based on first-order logic. We claim that an Adaptive Educational Hypermedia System (AEHS) is a Quadruple

(DOCS, UM, OBS, AC)

with

DOCS:

Document **S**pace belonging to the hypermedia system in question as well as information associated to this document space. This associated information might be *annotations* (e.g. metadata attributes, usage attributes, etc.), *domain graphs* that model the document structure (e.g. a part-of structure between documents, comparable to a chapter - section - subsection - hierarchy), or *knowledge graphs* that describe the knowledge contained in the document collections (e.g. domain ontologies).

UM:

User **M**odel: stores, describes and infers information, knowledge, preferences etc. about an individual user (might share some models with DOCS). The observations OBS are used for updating the user model UM. Examples of user models are overlay models where the user's state of knowledge is described as a subset of an expert's knowledge of the domain. Student's lack of knowledge is derived by comparing it to the expert's knowledge. A stereotype user modeling approach classifies users into stereotypes: Users belonging to a certain class are assumed to have the same characteristics.

OBS:

Observations about user interactions with the AEHS. Here, everything about the runtime behavior of the system concerning user interactions is contained. Examples are observations whether a user has visited a document, or visited document for some amount of time, etc. Other examples are rules for compiling e.g. quizzes for testing a user's knowledge on some subject, etc.

AC:

Adaptation **C**omponent: rules for *adaptive functionality* (e.g. whether to suggest a document for learning, or for generating reasonable learning paths, etc.), rules for *adaptive treatment* (e.g. sorting the links leading to further documents according to their usefulness for a particular user, etc.), etc.

To formalize this above definition let's gain a deeper insight into these components:

2.1 DOCS: The Document Space

The objects of discourse in the document space are the *documents*, and, if applicable, the knowledge *topics*. Their equivalent in the logical description are the atoms: the *document identifier (doc_id)* or *topic identifier (topic_id)* respectively.

Domain graphs (or knowledge graphs) are expressed as predicates that state the relations between the documents (or topics). For formalizing the part-of domain graph mentioned as an example in the previous section, we define predicates like

part_of(doc_id_1, doc_id_2) .

Another example is the *prerequisite* relation between documents stating which documents need to be learned before a certain document can be studied:

preq(doc_id_1, doc_id_2).

Some AEHS use a separate knowledge graph to express relations about knowledge topics. These topics normally do not correspond one-to-one to the documents. If a separate knowledge graph exists, this graph will be expressed by several predicates as well. E.g., a taxonomy on topics will be expressed by predicates like

is_a($topic\_id_1$, $topic\_id_2$) .

A further example are learning dependencies modeled on topics:

is_dependent($topic\_id_1$, $topic\_id_2$) .

2.2 UM: The User Model

The user model expresses, derives and draws conclusions about the characteristics of users. This might be done by modeling each individual user or by modeling typical groups that represent users with similar behavior, requirements, etc. (so called *stereotypes*). Objects of discourse in the user model are the *user* which are logically expressed by atoms, the *user identifier (user_id)*, and the various *characteristics* which can be assigned to this user in this AEHS. The characteristics of a user are expressed by predicates:

has_property(user_id, characteristic_x) or
has_property(user_id, characteristic_x, value), etc.

A prominent characteristic in AEHS is the knowledge a user has on documents (or knowledge topics). The first of the following examples uses a binary value for the knowledge, the second example allows different grades of knowledge:

has_property(doc_id, user_id, know) or
has_property(doc_id, user_id, know, value), etc.

The characteristic "knowledge" is very prominent for educational adaptive hypermedia systems, so we can abbreviate the above predicates by:

knows(doc_id, user_id) or
knows(doc_id, user_id, value), etc.

2.3 OBS: The Observations

Observations are the result of monitoring a user's interactions with the AEHS at runtime. Therefor, the objects for modeling observations are the users (as in the case of the UM) and the observations.

Typical observations in AEHS are whether a user has studied some document. The corresponding predicate is

obs(doc_id, user_id, visited) or
obs(doc_id, user_id, visited, value), etc.

If the document is a test and the user has worked on this test by answering the corresponding questions, predicates like

obs(doc_id, user_id, worked_on) or
obs(doc_id, user_id, worked_on, value), etc.,

are used.

2.4 AC: The Adaptation Component

Finally, the adaptation component contains rules for describing the *adaptive functionality* of the system. An example for adaptive functionality is to decide whether a user has sufficient knowledge to study a document (recommended for learning). This functionality belongs to the group of functionalities which determine the "learning state" of a document. A simple rule might be to recommend a document for learning if all documents that are "prerequisites", e.g. that need to be studied before this document can be learned, have been visited:

$\forall$user_id $\forall$doc_id$_1$

($\forall$doc_id$_2$ preq(doc_id$_1$,doc_id$_2$) $\Longrightarrow$ obs(doc_id$_2$, user_id, visited))

$\Longrightarrow$ learning_state(doc_id$_1$, user_id, recommended_for_reading).

The *adaptive treatment* is a set of rules describing the runtime behavior of the system. An often used adaptive treatment is the traffic light metaphor [1] to annotate links: Icons with different colors are used to show whether a document corresponding to a link is recommended for reading (green color), might be too difficult to study (yellow color), or is not recommended for reading (red color). The rule defining this adaptive treatment "document annotation" is:

$\forall$doc_id $\forall$user_id
learning_state(doc_id, user_id, recommended_for_learning)
$\Longrightarrow$ document_annotation(doc_id, user_id, green_icon).

2.5 Definition of Adaptive Educational Hypermedia Systems

In this section, we will give a logic-based definition for AEHS. We have chosen first order logic (FOL) as it allows us to provide an abstract, generalized formalization. The notation chosen in this paper refers to [11]. The aim of this logic-based definition is to accentuate the main characteristics and aspects of adaptive educational hypermedia.

Definition 2 (Adaptive Educational Hypermedia System (AEHS)) An Adaptive Educational Hypermedia System (AEHS) is a Quadruple

(DOCS, UM, OBS, AC)

with

DOCS:
Document **S**pace: A finite set of first order logic (FOL) sentences with atoms for describing documents (and knowledge topics), and predicates for defining relations between these atoms.

UM:
User **M**odel: A finite set of FOL sentences with atoms for describing individual users (user groups), and user characteristics, as well as predicates and rules for expressing whether a characteristic applies to a user.

OBS:
Observations: A finite set of FOL sentences with atoms for describing observations and predicates for relating users, documents / topics, and observations.

AC:
Adaptation **C**omponent: A finite set of FOL sentences with rules for describing adaptive functionality.

The components "document space" and "observations" describe basic data (DOCS) and run-time data (OBS). User model and adaptation component process this data, e.g. for estimating a user's preferences (UM), or for deciding about beneficial adaptive treatments for a user (AC).

3. Examples

This section provides two examples: The first example is a prototypical (artificial) AEHS whose purpose is to illustrate the applicability of the above proposed framework. The second example shows excerpts of the logical description of an existing AEHS: the NetCoach system [14]. Due to space constraints, we can not provide the full description of NetCoach in this paper. We have characterized three further AEHS, the Interbook system [3], the ELM-ART II system [15], and the KBS hyperbook system [7]. The complete logic-based description of these systems can be found in a technical report [6]. In this paper, we only summarize the results of the descriptions (section 4).

3.1 A very simple AEHS

We describe a simple AEHS, called *Simple* with the following functionality: Simple can annotate hypertext-links to documents by using the traffic light metaphor with two colors: red for non recommended, green for recommended pages.

3.1.1 Simple: Document Space

A set of n atoms (n corresponds to the number of documents in the document space) which name the documents:

$D_1, D_2, \ldots, D_n$.

Plus a finite set of predicates stating the documents that need to be studied before a document can be learned, e.g. D_j is a prerequisite for D_i:

preq(D_i, D_j) for certain $D_i \neq D_j$.

(N.B.: This AEHS does not employ an additional knowledge model).

3.1.2 Simple: User Model

A set of m axioms, one for each individual user:

$U_1, U_2, \ldots, U_m$.

3.1.3 Simple: Observations

One atom for the observation whether a document has been visited:

Visited.

And a set of predicates

obs(D_i, U_j, Visited) for certain D_i, U_j.

3.1.4 Simple: Adaptation Component

One atom for describing the values of the adaptive functionality "learning_state":

Recommended_for_reading,

and two atoms representing values of the adaptive treatment:

Green_Icon, Red_Icon.

Rules for describing the learning state of a document

$\forall U_i \forall D_j$

$(\forall D_k \, preq(D_j, D_k) \Longrightarrow obs(D_k, U_i, Visited))$

$\Longrightarrow$ learning_state(D_j, U_i, Recommended_for_reading).

And rules for describing the adaptive link annotation with traffic lights:

$\forall U_i \forall D_j$

learning_state(D_j, U_i, Recommended_for_reading)

$\Longrightarrow$ document_annotation(D_j, U_i, Green_Icon),

$\forall U_i \forall D_j$

$\neg$ learning_state(D_j, U_i, Recommended_for_reading)

$\Longrightarrow$ document_annotation(D_j, U_i, Red_Icon).

3.2 NetCoach

NetCoach [14] is the successor of ELM-ART II and provides a framework for building adaptive educational hypermedia systems. NetCoach uses a knowledge base which consists of concepts. Due to space constraints, we can only describe a part of NetCoach, the complete description can be seen in a technical report ([6]).

3.2.1 NetCoach: Document Space

The document space consists of documents, test-groups and test-items.

$D_1, \ldots, D_n, TG_1, \ldots, TG_n, TI_1, \ldots, TI_n.$

Relation between documents are e.g.:

preq(D_i, D_j) for certain $D_i \neq D_j$.

infer(D_i, D_j) for certain $D_i \neq D_j$.

part_of(D_i, D_j) for certain $D_i \neq D_j$.

succ(D_i, D_j) for certain D_i and one $D_j \neq D_i$.

Further, NetCoach assigns certain test-items or test-groups to a document (see [6]).

3.2.2 NetCoach: Observations

NetCoach takes the following observations about the interactions of a user U_i with the system into account:

obs(D_j,U_i, Visited) for certain D_j, U_i

obs(TI_k, U_i, Worked_testitem) for certain TI_k, U_i,

obs(TI_k, U_i, Solved_testitem) for certain TI_k, U_i, and

obs(D_j, U_i, Marked) for certain D_j, U_i.

3.2.3 NetCoach: User Model

Among others, NetCoach derives whether a document D_j has been learned by a user U_i. A document has been learned, if it is either tested, inferred from other learned documents, or marked by the user. If there are no test items assigned to the document D_j or the tests are treated as voluntary exercises (i.e. criterion(D_j, Value) for Value=0), then D_j is assumed to be learned if it has been visited, or it can be inferred from other learned concepts, or marked by the user (for details, e.g. how rules like p_obs(D_j, U_i, Tested) are derived, compare [6]):

$\forall D_j \forall U_i$

p_obs(D_j, U_i, Tested)

$\vee$ (criterion(D_j,0) $\wedge$ (obs(D_j, U_i, Visited) $\vee$ p_obs(D_j, U_i,

Inferred_Known)

$\vee$ obs(D_j, U_i, Marked))

$\Longrightarrow$ p_obs(D_j, U_i, Learned).

3.2.4 NetCoach: Adaptation Component

NetCoach provides adaptive link annotation. E.g a link to a document D_j is marked with a green ball (a sign that this document is recommended for reading) for a user U_i, if all prerequisites of this page haven been learned by this user:

$\forall D_j \forall U_i$

($\forall D_k$ preq(D_j,D_k) $\Longrightarrow$ p_obs(D_k,U_i,Learned))

$\Longrightarrow$ document_annotation(D_j, U_i, Green_Ball)

It supports learning goals (which are defined as a set of documents) and provides learning sequences to reach these learning goals (see [6]).

4. Synopsis of some logically characterized AEHS

This chapter provides synoptical tables of the logic-based characterization of the adaptive educational hypermedia systems NetCoach [14], Interbook [3], ELM-ART II [15], and KBS hyperbook [7] (the complete characterizations can be found in [6]). The atoms used in the four systems in the components DOCS, UM, OBS, and AC are summarized in table 1. Table 2 shows the used predicates. An overview on the rules is given in table 3.

Table 1: Atoms used in NetCoach, ELM-ART II, Interbook and KBS Hyperbook.

System	DOCS	UM	OBS
NetCoach	$D_1, \ldots, D_n$,	$U_1, \ldots, U_m$, Learned,	Visited,
	$TG_1, \ldots, TG_k$,	Inferred_Known, Tested.	Solved_Testitem,
	$TI_1, \ldots, TI_\ell$.		Marked.
ELM-ART II	$D_1, \ldots, D_n$,	$U_1, \ldots, U_m$, Tested,	Visited,
	$TI_1, \ldots, TI_\ell$.	Inferred_known.	Solved_Testitem.
InterBook	$D_1, \ldots, D_n$,	$U_1, \ldots, U_m$, Learned,	Visited,
	$TI_1, \ldots, TI_\ell$,	Beginner, Intermediate,	Solved.
	$C_1, \ldots, C_s$.	Expert, No_knowledge.	
KBS Hyperbook	$D_1, \ldots D_n$,	$U_1, \ldots, U_m$, Learned,	Marked,
	$C_1, \ldots, C_s$.	Known, Well_known,	Expert,
		Excellently_known, Partly_known,	Advanced,
		Not_known, Child_known,	Beginner,
		Parent_known.	Novice.

System	AC-Adaptive Link Annotation	AC - Others
NetCoach	Green_Ball, Red_Ball, Yellow_Ball, Orange_Ball.	-
ELM-Art II	Green_Ball, Red_Ball, Yellow_Ball, Orange_Ball.	-
Interbook	Small_Checkmark, Normal_Checkmark, Big_Checkmark,	-
	Green_Ball, White_Ball, Red_Ball.	-
KBS Hyperbook	Green_Ball, White_Ball, Red_Ball.	-

Table 2: Predicates used in NetCoach, ELM-ART II, Interbook and KBS Hyperbook.

System	DOCS
NetCoach	preq(D_i, D_j) (prerequisite knowledge)
	infer(D_i, D_j) (documents inferred to be learned by studying D_i)
	succ(D_i, D_j) (reading order)
	part_of(D_i, D_j) (chapter structure)
	terminal_flag(D_i) (whether a document has no further sub-documents)
	criterion(D_i, Value) (defines number of testitems
	necessary for mastering D_i)
	test_assignment(D_i, X), X $\in$ {Testgroup, Testitem},
	(relates documents with testgroups and testitems)
ELM-ART II	preq(D_i,D_j) (prerequisite knowledge)
	out(D_i, D_j) (documents inferred to be learned by studying D_i)
	related(D_i, D_j) (author-defined relation between documents)
	successor(D_i, D_j) (reading order)
	part_of(D_i, D_j) (chapter structure)
	terminal_flag(D_i) (whether a document has no further sub-documents)
	test_assignment(D_i, X), X $\in$ {Testslot, Testitem}
	(relates documents with testslots and testitems)
InterBook	preq(D_i, C_j) (prerequisite knowledge)
	out(D_i, C_j) (concepts inferred to be learned by studying D_i)
	succ(D_i, D_j) (reading order)
	terminal_flag(D_i) (whether a document has no further sub-documents)
	part_of(D_i, D_j) (chapter structure)

KBS Hyperbook	keyword(D_i, C_j) assigns some concepts each document
	depends(C_i, C_j) learning dependencies between concepts
	role(D_i, X), X ∈ {Course, Goal, Lecture, Example, etc.}
	role of the document D_i
	role(C_i, X), X ∈ {Introduction, Concept}
	role of the concept C_i

System	UM	OBS	AC
NetCoach	-	obs(D_i, U_j, X), X ∈ {Visited, Solved_Testitem, Marked}	
ELM-ART II	-	obs(D_i, U_j, X), X ∈ {Visited, Solved_Testitem}	-
InterBook	-	obs(D_i, U_j, X), X ∈ {Visited, Solved}	-
KBS Hyperbook	-	obs(C_i, U_j, Marked, Value),	-
		Value ∈ {Expert, Advanced, Beginner, Novice}	

Table 3: Rules used in NetCoach, ELM-ART II, Interbook and KBS Hyperbook.

System	DOCS	UM	OBS
NetCoach	-	Rules to infer p_obs(D_i, U_j, X), X ∈	-
		{Inferred_Known, Learned, Tested}	
ELM-ART II	-	Rules to infer p_obs(D_i, U_j, X), X ∈	-
		{Inferred_Known, Tested}	
InterBook	-	Rules to infer p_obs(C_i, U_j, Learned, X), X ∈	-
		{Expert, Intermediate, Beginner, No_knowledge}.	
KBS Hyperbook	-	Rules to infer p_obs(C_i, U_j, Learned, X), X ∈	-
		{Known, Well_known, Excellently_known, Partly_known,	
		Not_known, Child_known, Parent_known}.	

System	AC - Adaptive Link Annotation
NetCoach	Rules to infer document_annotation(D_i, U_j, X),
	X $\in$ {Green_Ball, Red_Ball, Yellow_Ball, Orange_Ball}.
ELM-ART II	Rules to infer document_annotation(D_i, U_j, X),
	X $\in$ {Green_Ball, Red_Ball, Yellow_Ball, Orange_Ball}.
InterBook	Rules to infer document_annotation(D_i, U_j, X),
	X $\in$ {Green_Ball, White_Ball, Red_Ball,
	Small_Checkmark, Normal_Checkmark, Big_Checkmark}.
KBS Hyperbook	Rules to infer document_annotation(D_i, U_j, X),
	X $\in$ {Green_Ball, White_Ball, Red_Ball}.

System	AC-Adaptive Link Generation
NetCoach	Rules to infer next_best_page(D_i, U_j), learning_goal(X),
	curriculum_sequencing($D_1, \ldots, D_l$)
ELM-ART II	Rules to infer next_best_page(D_i, U_j) (+ a tutoring component
	for the lisp domain)
InterBook	Rules to infer prerequisite_based_help(D_i, U_j), learning_goal(D_i),
	reading_sequence(D_i, U_j), teach_me(D_i).
KBS Hyperbook	Rules to infer learning_sequence([$C_1 \ldots$, C_n], U_j), glossary(D_i)
	learning_goal([$C_1 \ldots$, C_n]), next_reasonable_goal(U_j)
	information_index([$C_1 \ldots$, C_n])

5. Discussion

In this paper, we have proposed a component-based definition of adaptive educational hypermedia systems that uses first-order logic to characterize AEHS. With this approach

- we can easily compare the adaptive functionality of the AEHS: we can e.g. derive that the above characterized systems are very similar in their way of employing adaptive functionality (all make adaptive navigation support, no adaptive presentation support (with respect to Brusilovsky's taxonomy of adaptive hypermedia technologies [2]));
- we hide a lot of functionality behind the rules, e.g. KBS Hyperbook uses a Baysian Network to calculate the Inferred_known characteristic. This is technically very different to calculating this characteristic by compiling the transitive closure of prerequisites. But, logically, it has the same functionality: to estimate the user's current knowledge state based on some input parameters (the observations);
- we can describe the taxonomy of concepts used by the systems in document spaces, the user models, the observations, and the adaptation component. E.g. in case of the document space, we can derive that Interbook uses documents, testitems and knowledge concepts, ELM-ART II uses documents and testitems, etc.;
- the rules in the adaptation component show which data is processed by the system for making decisions about particular adaptive treatment; decisions;
- thus we can encapsulate adaptive functionality in order to support transfer of functionality between AEHS,
- and to support the more wide-spread use of adaptation in web-based educational systems, e.g. by employing Web-services that provide adaptive features.

During the application of the proposed characterization of AEHS, it turned out that the documents and their relations play a decisive role for the way how adaptation components draw conclusions. We have seen, that, in contrary to our intentions motivated by the transfer of Reiter's approach [10] to educational hypermedia, we were not able to generalize the diversity of rules for adaptation for a meta-description of adaptation. However, we claim that a logical characterization of adaptive educational hypermedia is a way to find solutions of current open questions in this area. E.g. currently, there is no catalogue of "metadata for adaptation" which could be used in LOM [9], SCORM [12] or other catalogues of metadata for education. The main objection is that adaptive educational hypermedia systems are "too different" to generalize for a meta-data driven description. From the above characterizations we can derive which meta-data is needed by the characterized AEHS: We can derive which sources for input data are used in the different systems from DOCS and OBS. These sources can now be used as a candidate set for meta-data for adaptation.

Further, our approach contributes to solutions for the open document space problem [8,2]: If we consider adaptive functionality as a query in open environments (as it has been done e.g. in [5]) it turns out that a decisive task is to determine the characteristics of adaptive functionality in order to define useful queries.

6. Conclusion

This paper proposes a component-based definition of adaptive educational hypermedia based on first-order logic. We have shown the applicability of such a formal description language for adaptive educational hypermedia in various examples. We claim that this logical characterization of adaptive educational hypermedia enables comparison of adaptive functionality in a well-grounded way, promotes the transfer of adaptive functionality to other educational hypermedia and web-based systems, defines rule-based descriptions of adaptivity, and supports the understanding of the role of metadata for adaptation.

Acknowledgment

We would like to thank Peter Brusilovsky and Gerhard Weber for the discussions on their adaptive educational hypermedia systems.

References

1. BRUSILOVSKY, P., Methods and techniques of adaptive hypermedia. *User Modeling and User Adapted Interaction 6*, 2-3 (1996), 87-129.
2. BRUSILOVSKY,P., Adaptive hypermedia. *User Modeling and User-Adapted Interaction 11*(2001), 87-110.
3. BRUSILOVSKY,P., EKLUND, J.,AND SCHWARZ, E., Web-based Educations for All: A Tool for Development Adaptive Courseware. In *Proceedings of the Sevenths International World Wide Web Conference, WWW'98*(1998).
4. BRUSILOVSKY,P.,AND MAYBURY, M., *The Adaptive Web*. Communications of the ACM, 2002.
5. DOLOG,P., GAVRILOAIE, R., NEJDL, W.,AND BRASE, J., Integrating adaptive hypermedia techniques and open rdf-based environments. In *International World Wide Web Conference*(Budapest, Ungarn, May 2003).
6. HENZE, N., AND NEJDL, W., Logically characterizing adaptive educational hypermedia systems. Tech. rep., University of Hannover, April 2003. http://www.kbs.uni-hannover.de/Arbeiten/Publikationen/2003/TechReportHenzeNejdl.pdf.
7. HENZE,N.,AND NEJDL, W., Adaptation in open corpus hypermedia. *IJAIED Special Issue on Adaptive and Intelligent Web-Based Systems 12*(2001).
8. HENZE,N.,AND NEJDL, W., Knowledge modeling for open adaptive hypermedia. In *Proccedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems (AH 2002)*(Malaga, Spain, 2002).
9. LOM: Draft Standard for Learning Object Metadata. http://ltsc.ieee.org/wg12/doc.html.
10. REITER,R., A theory of diagnosis from first principles. *Artifical Intelligence 32*(1987).
11. RUSSELL, S., AND NORVIG, P., *Artificial Intelligence: A Modern Approach.* Prentice Hall, 1995.
12. SCORM: The sharable content object reference model, 2001. http://www.adlnet.org/Scorm/scorm.cfm.
13. T. BERNERS-LEE, HENDLER,J.,AND LASSILA, O., The semantic web. *Scientific American*(May 2001).
14. WEBER,G., KUHL, H.-C.,AND WEIBELZAHL, S., Developing adaptive internet based courses with the authoring system NetCoach. In *Proceedings of the Third Workshop on Adaptive Hypermedia, AH2001*(2001).
15. WEBER,G.,AND SPECHT, M., User modeling and adaptive navigation support in WWW-based tutoring systems. In *Proceedings of the Sixth International Conference on User Modeling, UM97*(Sardinia, Italy, 1997).

An Integrated Model for the authoring of Web-based Adaptive Educational Applications

Andreas Papasalouros[1], Simos Retalis[2], Paris Avgeriou[2], Manolis Skordalakis[1]

[1]Software Engineering Laboratory
National Technical University of Athens
15780 Zografou, Athens, Greece
Tel. +302 10 7722486, Fax. +302 10 7722519
{andpapas,skordala}@softlab.ntua.gr

[2]Computer Science Department
University of Cyprus
P.O. Box 20537 Nicosia, Cyprus
Tel: +357 22 892246, Fax: +357 22 339062
{retal,pavger}@cs.ucy.ac.cy

Abstract

In this paper a design model for Web-based Adaptive Educational Applications (WAEA) is presented. A model-based approach is proposed as an answering to the problem of the difficulty of authoring such applications. This approach is based on the use of object oriented modeling techniques and the specification of WAEA by means of an XML binding.

1. Introduction

The authoring of Web–based Adaptive Educational Applications (WAEA) is a complex task. People with different background are involved in this kind of development, such as software developers, web application experts, content developers, domain experts, instructional designers, etc. Furthermore, the WAEA are complex dynamic web-based applications with presentational, behavioral and architectural aspects. In order to effectively capture and specify the various aspects of a WAEA, to document the decisions concerning such applications, from the implementation of pedagogic and instructional design to subtle technical decisions of such applications and to facilitate the communication between the members of usually heterogeneous development teams there is a need for a *design model.* This model will incorporate an engineering approach [8] in the authoring of WAEA. Experience from traditional software engineering has shown that the adoption of such a model is beneficial for the quality of the software products and the efficient management of the resources, time and effort. Although this model can be used as a Reference Model, that provides common understanding and communications of the various components of a WAEA, its main purpose is to facilitate the process of development of Adaptive Educational Applications. A design model like this can be used as a framework [9] for authors of hypertext applications to develop and apply methodologies in order to create adaptive applications in a disciplined and controlled fashion. It incorporates the principle of separation of concerns in the design of hypermedia applications, dividing the design of the application in three stages: conceptual, navigational and presentational. We also claim that this separation of concerns aligns with the three types of adaptation, navigation and presentation. Beyond a design model, if the development of open, portable, maintainable WAEA is to be facilitated, there is a need for a formally specified description of the WAEA. This description must be automatically generated from the aforementioned design model, at least to an extend, and must be easily ported to specific run-time environments that will deliver the specific WAEA.

In this paper, we define a Web based Adaptive Educational Application (WAEA) as a dynamic web application, i.e. a set of dynamically generated web content, which provides a learning environment to its users. This environment comprises of electronic content for study as well as a set of tools that facilitate the study of a learner such as web – based questionnaires, glossaries, communication tools, etc. This model focuses on content, which is considered as hierarchically structured, usually dynamically created, personalized assembly of predefined learning resources, either created from scratch or reused. These resources can be available in any form such as files, database entries, etc. We propose a model for the design and the construction of this kind of learning content. This model is considered to have two equivalent views: One view consists of diagrams in the UML notation language [19], which facilitates the design of the application in an intuitive, human understandable manner. The other consists of the description of the information that describes the WAEA in a formal, machine consumable language, by means of an XML binding.

The structure of the paper is as follows: In Section 2 we provide the description of our model. In Section 3 an exemplar application is given while in Section 4 follows an analysis of existing approaches in adaptive application modeling. In Section 5 we evaluate our approach and describe feature work based on the proposed model.

2. The Model of a Web based Adaptive Educational Application

In this section the first component of our approach, the design model, is described. The model is based on the Unified Modeling Language. It is an extension of the UML, formulated using the standard extension mechanisms of the language, i.e. Stereotypes, and specified by means of a UML Profile, 'a coherent set of such extensions, defined for specific purposes' [18].

The decomposition of this model into sub-models is based upon and extends the AHAM reference model for Adaptive Hypermedia Applications [2]. The following sub-models comprise an extended set of the models defined in AHAM:

2.1 The Conceptual Model

The *Conceptual Model* defines the concepts of the subject that is going to be taught with their semantic interrelationships. It can be considered as the Ontology [22] of the subject to be learned by the students. The Conceptual Model provides an objective definition of the knowledge subject. This definition is provided by the author of the educational application who is considered as a subject matter expert. The actual hypertextual content delivered to learners comprises a personalized, dynamic view over this conceptual realm. The main entity of the conceptual model is the C*oncept,* which depicts a main idea or topic of interest into the educational application. Concepts are abstract entities that do not carry actual content by themselves. They can contain meta-data or other descriptions, but the actual content is defined in the associated Resources. The *Resources* are the actual fragments of content that compose the WAEA, text, images, sounds, videos, simulations, forms, etc, which are static, reusable components or dynamic components such as multiple choice questions or glossary terms, dynamically created and delivered by appropriate web-based tools whose operation is not specified by our model. These tools can sometimes be considered as resources themselves.

Note that the granularity of a resource can vary from the content of a whole chapter to a single picture or paragraph of text. Two (or more) concepts can be associated with *Relationships,* which capture the semantic links between these concepts. Both concepts and relationships in the Conceptual Model are described as attribute-value pairs. The elements of the Conceptual Model are unchangeable during in the whole life time of the WAEA.

2.2 The Navigation Model

The Navigation Model captures the decisions about how Concepts, Relationships and Resources of the Conceptual Model are mapped to actual hypertext elements Pages and Links, and how the conceptual relationships defined in the Conceptual Model are driving the structuring of the learning content. The *Navigation Model* is composed by two sub-models:

2.2.1 The Navigation Structure Model

This model defines the structure of the WAEA and defines the actual web pages and the resources contained in these pages.

This structure is composed of the following elements:

- *Content,* which is the top-level container in the hierarchy of an electronic content organization.
- C*omposite* entities that are used as containers, thus composing the hierarchical structure of learning content. The chapters and subtopics in which an electronic tutorial or book are organized are examples of composite entities.
- *Access structures* elements, namely *indexes* and *guided tours,* which are related to Content or Composite components
- *ContentNodes*, which are the actual pages of the learning content. Content, Composite and ContentNodes are associated with Concept elements, or directly with Resources, in the Conceptual Model.
- *Fragments* that are contained into the ContentNodes. Fragments correspond to Resource elements in the Conceptual Model.
- *Links* between ContentNodes as well as between Fragments. Note that these links are associative links [9, 17] implementing domain specific relationships of the conceptual model. They are not structural links denoting, for example, the transition from a page in the learning content to the next one.
- Composite, ContentNodes, Fragments and Links have a predefined attribute of Boolean type named *included*. This denotes whether or not a specific element (and all its descendants in the hierarchy) is included in the created hypertext or not, as a result of adaptation.

2.2.2 The Navigation Behavior Model

The *Navigation Behavior Model* defines the run-time behavior of the WAEA. Earlier research [6, 14, 24] has proposed the use of statecharts for the modeling of hypertext and web based applications. The Navigation Behavior model uses statecharts, as they are incorporated in the UML in order to specify the dynamic transitions of the hypertext structures as the user interacts with the WAEA. Every containing element of the Navigation Structure Model (Content, or Composite) is associated to a composite state in the Navigation Behavior Model, while every ContentNode corresponds to a simple state. Thus, the hierarchy of the navigational elements defined in the Navigation Structure Model corresponds to the hierarchy of nested states in the Navigation Behavior Model. The events that fire the transitions in the Navigation Behavior Model correspond to structure links into the ContentNodes: next, previous, up level, etc. In addition, guard conditions in these transitions can define alternative navigational transitions, which correspond to conditional behavior of the WAEA, thus implementing content sequencing and adaptive navigation.

2.3 The Presentation Model

The *Presentation Model* deals with the presentation aspects of the elements defined in the Navigation Model.

The presentation model is by itself separated in two additional sub-models: *Presentation Structure* Model, which defines the allocation of the navigational elements to actual user interface web elements: Web pages, frames, framesets, etc. Elements of this model, which is a variation of the synonymous model proposed in [14], are the following: *frameset*, *frame*, *window*. The aforementioned elements are associated with one or more elements of the Navigation Model.

User Interface Model, that captures the layout, colors, styles, etc of the entire web pages as well of atomic elements of the pages. This model consists of *Presentation* elements, which define the layout and style of associated elements of the navigation model.

2.4 The User Model

The User Model consists of two different parts, each one containing two types of elements: The *Overlay Model*, which is the domain specific part of the user model and defines the status of the learner's knowledge of the specific concepts covered by the learning material. The state of this model is frequently updated as a result of the learner's interaction with the learning content, for example the reading of learning material, the taking of an on-line test, the interaction with simulations, etc. The knowledge is defined as a structure of concepts (schema) and this structure is built during the user's learning activities. The Overlay Model depicts the system's awareness of the current status of the user's knowledge about the domain of the specific application as it is stated in the Conceptual Model. The elements of this sub-model are called *UserScheme* [2], and there can be one *UserScheme* element for each class of the Conceptual Model.

The second part of the User Model defines elements that are used to represent the usually predefined user knowledge profile either concerning the knowledge of the particular domain (novice, intermediate, expert, etc) or corresponding to the user's preferences or learning style. According to [4] this constitutes the Stereotyped User Model. The elements of this submodel are called *User*.

2.5 The Rules Model

The adaptive behavior of the application is specified with appropriate rules. The rules are applied in two ways:

- As Object Constraint Language Expressions (OCL) [23]. OCL is a formal language for applying constraints to UML models. Constraints are conditions that must hold for a specific model they are applied. The rules defined in the Rules Model are applied as two types of constraints:

    - *Invariants*, that is conditions that must *always* be true in the context they are applied (concept components, concept relationships).
    - *Postconditions*, that is conditions that must be met *after* the execution of a method or operation of a specific class.

  The constraints are applied to specific model elements, defined by the keyword **context,** as will be shown in the following examples.

- As *guard-conditions* in the transitions defined in the Navigation Behavior Model.

OCL rules can be applied to elements of every one of the aforementioned models. For example, a rule in the User Model provides a mechanism for updating the knowledge of the user on a particular concept as a result to his/her navigation.

2.6 The XML definition

For this purpose we developed an XML binding for the model described above. The XML bindings are defined as a Document Type Declaration (DTD). The DTD definition was preferred to XML Schema [http://www.w3c.org/XML/Schema], as the former is supported by a much wider range of current XML tools such as validators, XSL processors, etc. In the following the listing of the whole DTD is omitted due to space restrictions.

Each of the elements in the previous sections is defined as an element in the DTD. For example, the root element, i.e. *waea*, is defined as follows:

```
<?xml version="1.0" encoding="UTF-8"?>
<!ELEMENT waea (ConceptualModel, NavigationModel,
   PresentationModel, UserModel, RulesModel)>
```

Each element has a unique identifier through which it can be referred by its associated other elements. For example

```
<!ELEMENT Resource ANY>
<!ATTLIST Resource
   id ID #REQUIRED
   mime-type CDATA #IMPLIED
   uri CDATA #IMPLIED
>
```

Certain elements have arbitrary sets of attributes. The attributes are defined as follows:

```
<!ELEMENT Attribute EMPTY>
<!ATTLIST Attribute
   name CDATA #REQUIRED
   value CDATA #REQUIRED
   type CDATA #REQUIRED
>
```

3. An Example Application

We provide an example application in order to demonstrate the use of our model in a simple WAEA This example is not aiming to depict how a whole educational application can be modeled and authored according to the proposed approach, but to illustrate the elements and their use in constructing a model for an WAEA.

Figure 1 shows the conceptual model of an application for the teaching of Java Swing Basics. Java Swing is a framework for the development of user interfaces in the Java programming language. It shows the basic concepts that are going to be taught together with their semantic interrelationships. In order to keep the diagram simple, not all the resources associated with corresponding Concepts are depicted. Note that the model primitive elements (Concept, Resource, etc) are shown as stereotypes of UML classes.

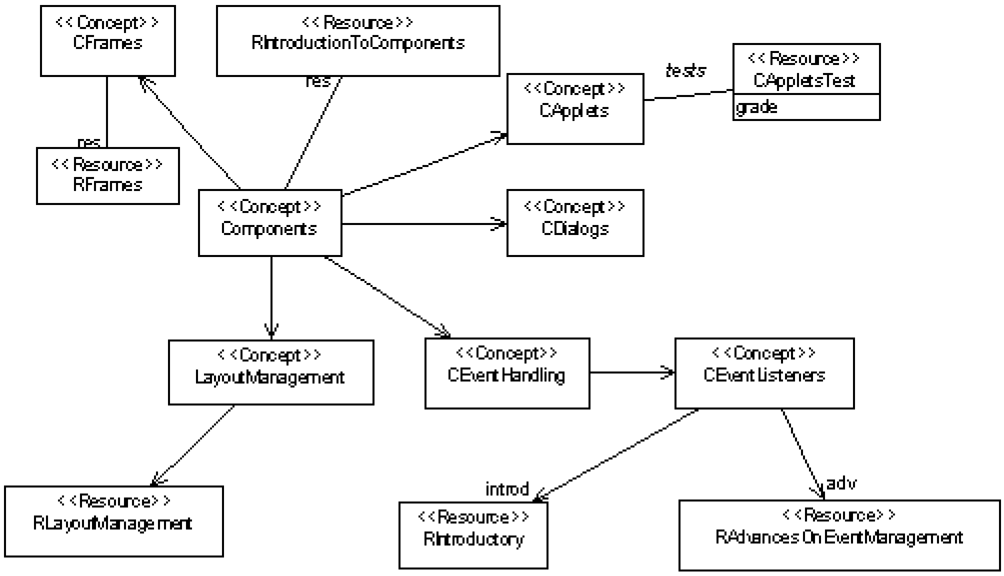


Fig. 1. Example of Conceptual Model

Figure2 displays the Navigation Model. Each element of this model is associated with corresponding elements of the Conceptual Model, though these UML associations are not displayed for the sake of clarity.

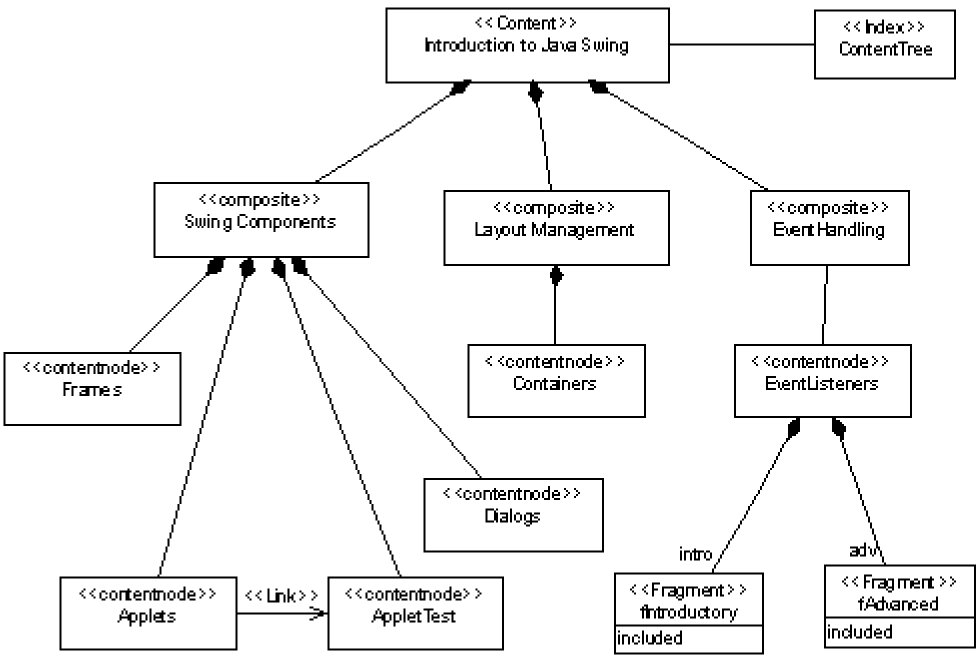


Fig. 2. Example of Navigational Structure Model

In Figure 3 the Navigation Behavior model for the application is shown. Note that the hierarchy of nested states in this example corresponds to the hierarchy of Composite and ContentNode elements in Figure 2.

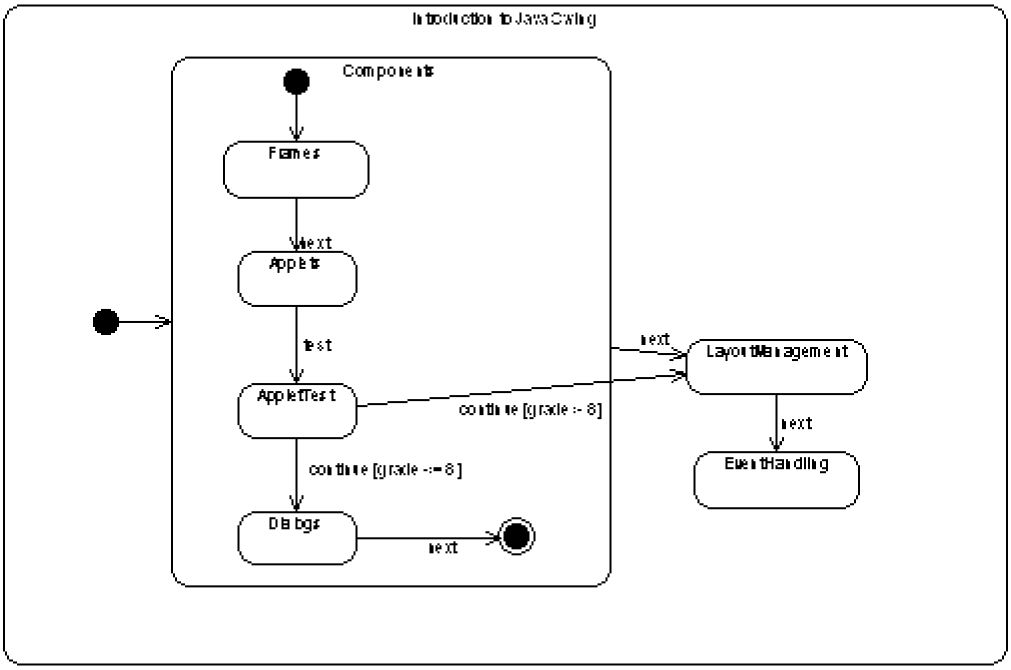


Fig. 3. Example of Navigation Behavior Model

The arcs in Figure 3 denote state transitions that correspond to link traverse in the navigational space of the applications. As shown in the same figure, when the user follows the *next* link while he or she was in the Applets page, the AppletTest page is presented to the user. The user takes the test and if the grade is high enough, the Dialogs section is omitted and the system presents the user the LayoutManagement section, as he follows the *continue* link. Note that in guard condition

[grade > 8]

grade is an attribute of the AppletTest class of the Navigation Model (see Figure 2).

As an example of content adaptation, according to the User Profile, we define a very simple user model as a User stereotyped class. This user class has an attribute named 'type'. We assume that there are two types of users, namely 'Novice' and 'Advanced'. We want to include different content (resources) related to the EventListeners concept, dependent on the type of the user. In the first case the RIntroductory resource is included, while in the second case the RadvancesInEventManagement is included (see Figure 2). The fragments correspond to the two available resources, namely fIntroductory and fAdvanced. The control over which of the two resources is finally displayed is through the *included* attribute applying an appropriate OCL expression:

```
context EventListeners
inv:    (user.type = 'Novice' implies (intro.included = true and
                adv.included = false))
        and
        (user.type = 'Advanced' implies (intro.included = false and
                adv.included = true))
```

where ‘user’ is the user type associated with the ‘EventListeners’ navigational class. These associations, which constitute the UML diagrams syntactically correct, are omitted for the sake of clarity. Note that the keyword **implies** means that when the expression is evaluated then if the condition left to the **implies** keyword is true then the condition to the right must also be true, in order for the whole expression to be true.

4. Related work

Up to our knowledge, the following approaches exist in modeling of Adaptive Hypermedia Applications, in general. In these approaches we have included the information models proposed by some learning technology standards specifications, namely the IMS Learning Design and Simple Sequencing Standards.

The meta–model defined in [7] provides a four layer architecture for the definition of hypertext applications, with provision for user modeling. This approach for modeling of hypermedia applications utilizes adaptive presentation and engages a variety of formal mechanisms and visualization techniques. It cannot be easily applied in different contexts, due to its tight relationship with specific systems, like the ConceptBase system, while the diversity of applied formalisms is an obstacle in applying it in general. Although it is domain independent it cannot be used for educational applications, because it does not address issues like adaptive sequencing, adaptive structuring of the learning content, etc which are important in this kind of applications.

AHAM [2] is a reference model for Adaptive Hypermedia Applications. Our modeling approach is based on primitives of the AHAM using them in the direction of a Design Model for a specific category of Adaptive Applications, i.e. Educational ones. AHA [1] is a system based on AHAM. The data model for the description of the Adaptive application is much similar to the one we propose. The main difference in our approach is the separation of concepts from resources, navigation and presentation, which gives the ability for conditional text, or other type of resource, inclusion, conditional page creation and presentation.

In [15] a software engineering approach based on UML (UWE) is proposed in order to facilitate the developing Adaptive Web based applications. In this approach specific views of an Adaptive Web Applications are defined, expressed as different models. This approach is very similar to ours in the separation of concerns in the hypertext design defining different models, the use of the UML and the use of OCL for the applications of rules. However, it is not appropriate for the domain of educational applications since it focuses on the development of web based systems with adaptive features which present highly structured content. On the contrary, educational applications are semi-structured, in the sense that they lay between highly structured systems, or applications, like hypermedia front ends for library systems or electronic market places, and unstructured applications that derive as a result of a purely creative artistic task, like electronic presentation of novels and literary work [11].

In [21] a meta-model for courseware design is defined. According to this abstract meta-model, specific models can be derived for certain domains of teaching, or disciplines. It resembles our approach in that it has both a UML design and an XML part. However it cannot be considered as an adaptive educational hypermedia model, since it does not provide a specific user model neither proposes a specific formalism for the definition of rules for adaptation.

The IMS Learning Design (IMS LD) Standard defined in [13] is not a hypermedia design model, but rather a framework for formally specifying educational activities in the context of a learning system that incorporates traditional learning methods with learning technologies. Not being a hypermedia model, IMS LD does not cope with the details of structuring of the hypertext, hyperlinks, presentation, etc, but rather focuses on the dynamic aspects of electronic content delivery viewing them as the implementation of pedagogic practices and specific learning design. It proposes an XML schema and does not cope with facilitating the design of Educational Applications.

IMS Simple Sequencing (IMS SS) [12] is a standard proposed by the IMS consortium that provides the basis for sequencing of Learning Activities in the context of a Learning Technology System according to specific rules. IMS SS deals with the sequencing issues of predefined, pre-structured learning content. It does not deal with adaptive content or adaptive presentation of the hypermedia content, but only with adaptive navigation, in the sense of the automatic selection of next resource to be presented to a learner according to her/his history of interaction with the learning material.

In [5] a layer approach for the modeling of Adaptive Educational Applications is provided, together with a method for the design of such applications. This approach is similar to ours in the distinction of three views of Adaptive Educational Application depicted as layers: A conceptual Layer, a lesson layer and a student adaptation and presentation layers, which resemble our separation in three sub-models, i.e. conceptual, navigational and presentational. A second main similarity is that both approaches recognize that the authoring of WAEA is driven by an initial mapping of the available resources in a high level conceptual model. The main differences from this approach are in the way of mapping of the initially defined concepts into specific navigation and presentation elements, as well as the specific formalism used in our approach, namely UML.

In [20] we define an Object-oriented Model for Adaptive Hypermedia Educational Applications. This model is an extension of AHAM for the modeling of adaptive educational hypermedia using UML and object-oriented principles.

5. Conclusions –Future Work

A number of solutions exist for modeling and representation of Web-based Adaptive Hypermedia for Educational purposes. The one presented here combines many of the features from the above in the aim of providing the means for an adequate description of WAEA and the facilitation of the author of such applications for a disciplined and effective application authoring. This approach is adequate in that it can describe various types of adaptation, useful in educational applications: Conditional content inclusion, sequencing of content according to the user's interaction, adaptation of the user interface. It provides a design model for WAEA that facilitates the process of authoring and maintaining of such applications through a consistent visual formalism facilitated with the use of UML which is a widely adopted modeling language.

Future work according to this model includes the development of a case tool that will facilitate the process of model authoring and the automatic production of XML-based descriptions of WAEA, the specification of the run-time system that complies with the proposed model and the implementation of an instance of such a system for further testing of the model. Also the compatibility of this model with the existing learning standards described above is to be investigated.

References

1. P. De Bra and L. Calvi. AHA! An open Adaptive Hypermedia Architecture, *The New Review of Hypermedia and Multimedia*, vol. 4, pp 106-112 (2000).
2. P. De Bra, G.J. Houben and H. Wu. AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In *Proceedings of the tenth ACM Conference on Hypertext and Hypermedia,* Darmstadt, Germany, February 1999. ACM Press.
3. Peter Brusilovsky. Adaptive educational systems on the world-wide-web: A review of available technologies. In *Proceedings of Workshop WWW-Based Tutoring,* 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX, 1998.
4. Brusilovsky. Methods and Techniqes of adaptive hypermedia. *User Modeling and User Adapted Interaction*, vol. 6, pp 87-129, 1996.
5. A. Cristea and L. Aroyo. Adaptive Authoring of Adaptive Educational Hypermedia. In *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web System,* Malaga, Spain, May 2002.
6. M. C. Ferreira de Oliveira , M. A. Santos Turine and P. C. Masiero. A Statechart-Based Model for Hypermedia Applications, *ACM Transactions on Information Systems*, Vol. 19, Issue 1, 2001.
7. P. Fröhlich, N. Henze, and W. Nejdl. Meta Modeling for Hypermedia Design. In *Proceedings of IEEE Metadata Conference,* Maryland, USA, Sept. 1997.
8. D. Garlan and M. Shaw. Software Architecture: Perspectives of an Emerging Discipline. Prentice Hall, New Jersey, 1996.
9. F. Garzotto, D. Schwabe and P. Paolini. HDM- A Model Based Approach to Hypermedia Application Design. *ACM Transactions on Information Systems*, vol. 11, n. 1, pp. 1-26, 1993.
10. F. Halasz and M. Schwartz. The Dexter Reference Model. *Communications of the ACM*, vol. 37, n. 2, pp. 30-39, 1994.
11. T. Isakowitz, E. Stohr and P. Balasubramanian. RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, vol. 38, n. 4, 1995.
12. IMS Global Learning Consortium Inc. *IMS Simple Sequencing Information and Behavior Model*. Available at http://www.imsproject.org/simplesequencing/
13. IMS Global Learning Consortium Inc. *IMS Learning Design Information Model*. Available at http://www.imsproject.org/learningdesign.
14. N. Koch and A. Kraus. The Expressive Power of UML-based Web Engineering. In *Proceedings of the 2nd International Conference Workshop on Web Oriented Software Technologies,* Malaga, Spain, June 2002.
15. N. P. de Koch. Software Engineering for Adaptive Hypermedia Systems, PhD Thesis, Verlag Uni-Druck, Munich 2001.
16. Rod Koper. Modeling units of study from a Pedagogical Perspective: the pedagogical meta-model behind EML. Available at http://eml.ou.nl.
17. D. Lowe and W. Hall. *Hypermedia & the Web, an Engineering Approach*, John Wiley & Sons, 1999.
18. OMG. UML Profile for CORBA Specification. Available at http://www.omg.org/uml.
19. OMG. Unified Modeling Language Specification. Available at http://www.omg.org/uml. September 2001.
20. A. Papasalouros and S. Retalis. Ob-AHEM: A UML-enabled model for Adaptive Educational Hypermedia Applications. *Interactive educational Multimedia*, vol. 4, 2002.

21. C. Süß, B. Freitag and P. Brössler. Meta-Modeling for Web-Based Teachware Management. In *Advances in Conceptual Modeling*, ER'99 Workshop on the World-Wide Web and Conceptual Modeling , Paris, France 1999. Springer.
22. M. Uschold and M. Grunninger. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review,* vol.1, n. 2, pp. 93-155, 1996.
23. J. Warmer and A. Kleppe. *The Object Constraint Language: Precise Modeling with UML.* Addison Wesley, Reading Mass., 1999.
24. Y. Zheng and M. Pong. Using statecharts to model hypertext. In *Proceedings of the ACM conference on Hypertext,* Milan, Italy, 1992, ACM Press.

Incorporating learning styles in hypermedia environment: Empirical evaluation

Namira Bajraktarevic[1], Wendy Hall[1] and Patrick Fullick[2]

[1]Intelligence, Agents, Multimedia Group
Dept of Electronics and Computer Science
University of Southampton
{nb99r, wh} @ecs.soton.ac.uk

[2]Research and Graduate School of Education, Southampton
plf@soton.ac.uk

Abstract

In this paper a hypermedia system that incorporates global and sequential learning styles is described. Elements of learning styles such as the sequencing of pages and structure have been incorporated in the courseware. Two different presentation style user interface templates were employed. The system provides an alternative to the "one size fits all" approach to development of web-based educational courseware, by creating learning materials to cater for individual learner preferences. The Felder-Solomon Learning Style Questionnaire was used to measure the learning style preferences of students. Information recall tests were created and scores that were obtained after matched[1] and mismatched[2] sessions have been compared. Significantly higher results were obtained for the matched session compared with the mismatched session. The findings are discussed with respect to their implication for further development of adaptive web based courseware.

1. INTRODUCTION

This document presents the results of the empirical-analysis of the experimental study relating to the application of learning styles in educational hypermedia and their effects on learning outcomes. The analysis intends to compare the post-test achievement score differences between matched and mismatched sessions of courseware and evaluate the correlation between the increase in post-test achievement scores and the session browsing times.

2. LITERATURE REVIEW

When designing instructional material, it is important to accommodate elements that reflect individual differences in learning. One such element addressed in this study is a learning style. There are a variety of definitions and interpretations of learning styles available in literature but in this paper the term learning style refers to "adopting a habitual and distinct mode of acquiring knowledge" as summarised by McLoughlin (1999). There are many models of learning style but this study focuses on the two learning styles that fall into the cognitive and information-processing model. Individual differences are observed by the way students interact with hypermedia based learning materials. It is important to study how the structure and organisation of information may contribute to a better learning in relation to individual differences.

2.1 Learning styles

Pask (1976) describes two particular learning styles: holist and serialist. He describes them in the following way: "*holist learners learn in layers. They prefer an overview of where they are going first before learning a complex process. They like having a map, knowing where they are headed and what they are working toward. They enjoy having examples shown to them even if they are not capable of imitating the skill yet. Holist learners sometimes get confused by step-by-step instructions, especially if the steps are numerous and complex. They pick up bits and pieces within a broad framework. They may leave gaps, repeat themselves and over generalise. They may also be more comfortable with "topic" based learning*". In addition, for serialist learners Pask said that such learners "*find introductory overviews distracting and confusing. They expect to learn whatever they are shown immediately or they become frustrated because they don't have the ability of the global learner to see 'the big picture'. They prefer to proceed step-by-step, in an orderly way, to the end result. They build their knowledge sequentially, they are impatient with 'jumping around' and they may lose sight of the broader picture. They may be more comfortable with inherently 'linear' subjects*".

Serialist learners are in the majority, and most educational materials are laid out in a sequential rather than a global way. Pask has demonstrated that a mismatch between the teacher and learner's styles seriously hinders the learning process. Riding and Cheema (1991) use wholist/analytics terminology to describe how individuals process information. Wholists, according to them, organise information into loosely clustered wholes, in order to construct an overall understanding of the given information. Analytics, in contrast, process information in clear-cut conceptual groupings. Felder and Silverman (1988) describe different individuals as "global" and "sequential". Global individuals are classified as holistic, systems thinkers, learning in large leaps. Sequential individuals are classified as linear, orderly and learning in small incremental steps. Clarke (1993) suggests that all these styles 'differ more in name than nature' and that they 'can be classified into either a preference for a reasonable degree of structure and guidance [serialist] and a preference for considerable freedom to explore [holistic]'. This paper seeks to explore the relationship between matching and mismatching learning style preferences in hypermedia material based mainly on the theoretical work of the above-mentioned researchers.

2.2 Previous attempts at applying learning styles in hypermedia

A number of researchers have promoted constructivist approaches to the design of computer based learning materials, whereby courseware is designed with the priority on individual student requirements and styles, as opposed to the tutor-led factors inherent in an instructionist approach. Additionally, researchers have addressed the issue of tailoring the design of learning activities to match an individual's learning preferences, in the context of computer based learning materials. This section presents some of the previous attempts at incorporating the wholist/analyst (global/sequential, holist/serialist) dimensions of learning styles into computer and hypermedia assisted instruction.

Kwok and Jones (1995) carried out an experimental study with a computerised 'front-end' study preference questionnaire (based on Ford, 1985) in order to suggest to the user a suitable navigation method through the system. They found that students at the far extremes of the learning style spectrum needed the navigational guidance, and it helped raise their interest in the material. Pillay and Willss (1993) reported the study where students with wholist/analytic cognitive style received instruction that matched and mismatched their cognitive style. The matched group performed better in the explanation and problem solving tasks. Riding and

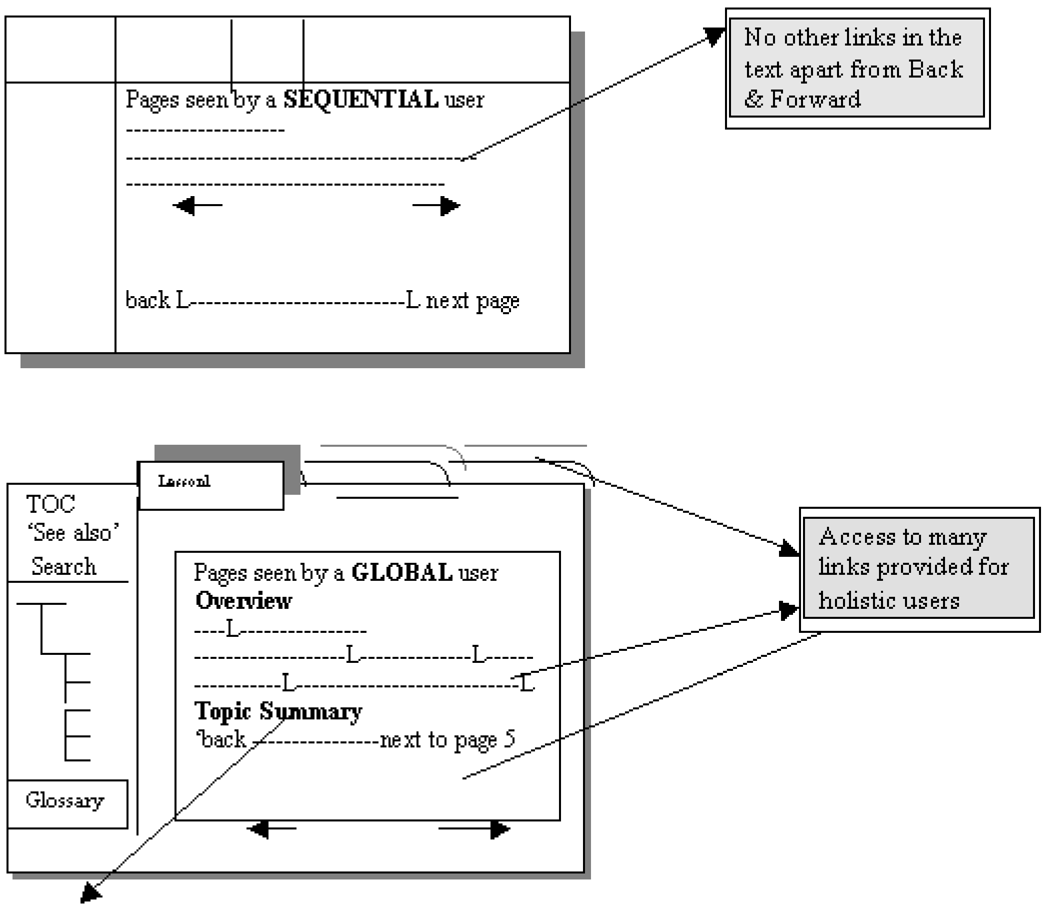
Sadler-Smith (1992) investigated an interaction between mode of presentation and style and their effect on learning performance. They believed that structure and organisation of the contents might interact with the wholist/analytical dimension of style. Their conclusion was that the mode of presentation has important effects on learning performance. Ford and Chen (2001) explored relationship between match and mismatch of instruction presentation style with student's cognitive style (field dependent and field independent). They have found significant differences in performances on conceptual knowledge for students under two different conditions. Graff (1999) tested the relationship between three different hypertext structures (linear, hierarchical and relational) and the performance of the students with *wholist-analyst* cognitive styles. He suggested that providing different linking structure to individuals of different cognitive styles would make the learning from hypermedia more effective. No significant differences on recall of information were found. The experiment conducted as part of this study is most similar to Graff's paper, in that an attempt has been made to incorporate two types of linking structure in the hypermedia material (liner and hierarchical), that will match sequential and global learning preferences. These linking structures were imposed on students by providing a different sequence for browsing the courseware, so that sequential students were provided with a linear sequence, while global students were allowed to 'globetrot' and had much more navigational freedom in the system.

3. ADAPTIVITY AND APPLICATION OF LEARNING STYLES

With application of individual differences in hypermedia such as the cognitive and learning style a few adaptive hypermedia systems were created recently, such as INSPIRE (Grigoriadou *et al*, 2001), CS388 (Carver, 1996), RAPITS (Woods and Warren, 1996) and AEC-ES (Triantafillou *et al.*, 2002). In INSPIRE the authors adopted Kolb's theory of experiential learning (activists, pragmatists, reflectors and theorist learning styles). Their user model consists of two parts: general information about user (age, sex) and current knowledge level unit. As the users progress through the system, they are monitored. Lessons are divided into layers and generated dynamically. There is also a presentation module responsible for modifying the appearance of knowledge modules. The authors used a*daptive presentation* techniques for different order of knowledge-modules. Learning style elements appear or do not appear inside a lesson. A*daptive navigation* techniques such as annotation of the links in the 'navigational' area of the system are used. Students' progress is indicated in metaphoric form (filling glass). The links recommended by the system change from black and white to coloured. Carver (1996) created a system (CS388) that consisted of a range of learning style tools. The learning styles were assessed using the Felder-Silverman learning style model. Students were allowed to traverse the courseware according to their own learning style. Carver's paper couples the Felder-Silverman learning styles with hypermedia to provide tailored lessons. In this approach the key was to determine what type of media are applicable and appropriate to different learning styles (such as graphs, movies, text, slideshows etc.). RAPITS (Woods and Warren, 1996) is an adaptive teaching system that compares student model to domain knowledge and automatically changes presentation style. The student can proceed through the lessons sequentially or switch to a non-linear hypermedia mode through the courseware. Student knowledge is assessed at the end of a topic. Triantafillou *et al.* (2002) created an adaptive educational system (AEC-ES) based on field dependent/field independent cognitive styles. Their system uses navigational support tools (concept map, graphic path, advance organiser) and adaptive presentation techniques. Students are provided with instructional strategies that suit their cognitive preferred style with an option to switch it to a non-preferred version.

3.1 Implications for the design of web-based learning materials in this study

Based on the learning style description by Pask, the following learning style representation in a hypermedia environment was compiled. The majority of these elements apply to the layout, sequencing and structure as well as the navigation of the user interface. The two principal considerations in designing hypermedia courseware to accommodate preferred learning styles are: the way in which the information is formatted and structured and how individuals process the given information. Hypermedia can be put to an advantage or disadvantage for the users depending on whether the material is matched or mismatched with the students' preferences. The way that global and sequential students process information would appear to be directly relevant to effective learning from information presented as hypermedia. Figure 1 depicts an example of the global and sequential page layouts. The difference in presentation of the two types of formats is apparent. For the students with a global learning style preference, pages comprised elements such as a table of contents, summary, diagrams, overview of information etc. For sequential students, the pages contained small chunks of information, text-only pages with 'forward' and 'back' buttons.



Where L represents a link

Fig. 1. The user interface templates (sequential and global)

4. OVERVIEW OF THE EXPERIMENT

As learning styles are a significant contributing factor in terms of student progress, we can attempt to represent some of the characteristics of the learning preferences within hypermedia environment and help students in different parts of the learning cycle. To examine whether differently formatted hypermedia courseware has an impact on information acquisition during the browsing process, an empirical evaluation was conducted. Students were asked to complete a number of tasks. These tasks included the following: browsing, reading, searching for and memorising information displayed on the computer screen. At the end of each round of the tasks, the students were asked to answer a series of questions to test their recall of the supplied information. In this study, the achievement scores from two post-tests were compared. Test questions were tied to lesson objectives and levels of Bloom's taxonomy (1956).

4.1 Procedure

Sample

The sample of students involved in this study comprised twenty-two, year-10 students (they are 14 year old students attending National Curriculum) in the first year of a two-year GCSE[3] geography course. They browsed through two different versions of web-based courseware; one version matched their preferred learning style and the other did not. The learning styles incorporated into the hypermedia courseware were global and sequential learning styles.

Courseware choice

Both sets of courseware were part of the GCSE syllabus. First of the courseware sets ("Countries of the world") was developed and adapted from a GCSE geography book (Pallister, 2001). The instructional material for the second courseware set ("Ozone layer depletion") was a mixture of web resources. The courseware was new to the students, so interference from prior knowledge was not a concern.

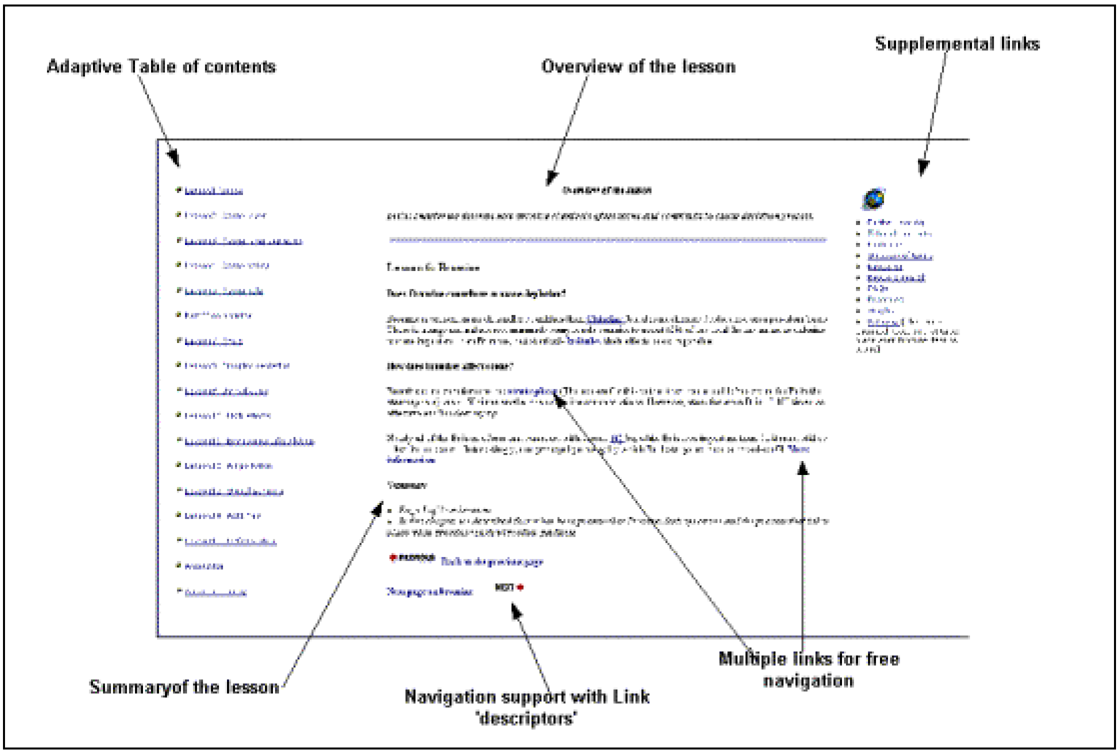


Fig. 2. Global session layout

Learning style instrument

Before the start of the experiment the positions of the students on the learning styles dimensions were evaluated using a web-based, self-administered Felder-Soloman Index of Learning Styles Questionnaire (ILSQ) that assessed four learning style dimensions: introvert/extravert, verbal/visual, sensitive/intuitive and global/sequential dimension. This Index of Learning Styles (ILS) has been developed by Soloman and Felder (1988). Attention in this study was focused on the bi-polar nature of the LSQ scales, particularly the global-sequential dimension. Students can have from mild to moderate and strong learning preference on that scale.

Procedure

At the start of the study, students read a short explanation concerning the use of the system. They then logged onto the system and their learning styles were recorded. The students then completed pre-test on the first learning subject and then proceeded to browse and study the material that matched their learning preferences. Having completed that, the students were presented with a recall-type post-test. The questions were knowledge questions as they tested recalling of facts, terms and concepts as suggested in Bloom's taxonomy. In the second part of the study, students logged in again, and completed pre-test about the second subject. They proceeded to browse and study material, which was in the adapted so that did not match their learning preference. Upon completion of the second browsing session the students completed the post-test. The two main dependent variables in this study were the achievement scores obtained in the two post–tests and the session-browsing times. For this study significance testing or hypothesis testing was used.

5. RESULTS

This section presents the performance results for the knowledge attainment and browsing times differences between matched and mismatched sessions. It also shows any dependency between the length of browsing times and the increase in scores achieved after browsing both sessions.

5.1 Data collection and analysis

To compare performance between students the times for both browsing tasks and the answers to all four sets of questions (pre-tests and post-tests for matched and mismatched sessions) were recorded. The population consisted of 21 students, where 9 were with sequential preferences (6 balanced, 2 moderate and 1 with strong preference) and 12 with global preferences (6 mild and 6 with moderate preferences). The results were analysed to determine whether significant differences between score means occurred for different session types. T tests were used to compare the mean scores between matched and mismatched conditions and a significance level of $p<0.05$ was adopted. Q_2 denotes the number of correctly answered questions after the 1st post-test (after the session in preferred style) and Q_4 denotes the number of correctly answered questions after the 2nd post-test (after the session in non-preferred style)

5.1.1 Difference between score means for matched and mismatched sessions

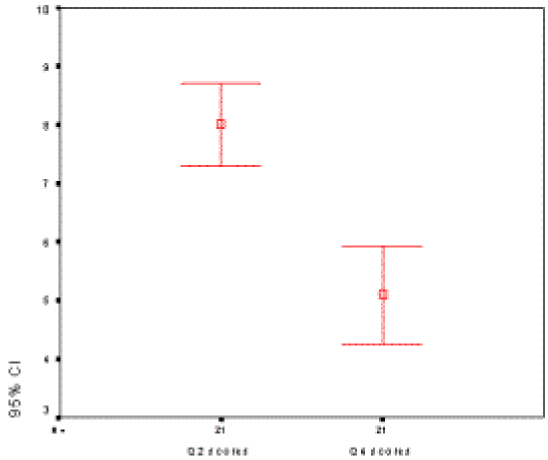


Fig. 3. Scores after the post-questionnaires with mean values for all students, with confidence interval of 95%

To determine if the differences between achievement scores were significant, the two-sample, paired, two-tailed t-test was used. The main results are presented in Table 1. The hypotheses for the test scores were that:

H_0**:** Post-test-score means for matched and mismatched sessions are the same

H_1**:** Post-test-score means for matched session are significantly higher than for mismatched session

Table 1: SPSS results for the t-test for score differences for all students

Score difference		Paired Differences					t	Significance. (2-tailed)
		Mean	SD	Std. Error Mean	95% Confidence Interval of the Difference			
					Lower	Upper		
	(Q2-Q4)	2.905	1.9469	.4249	2.019	3.791	6.837	.000

The mean values for the scores after the 1st and 2nd sessions are listed in Table 1 and it appears that the mean scores for Q_2 are much higher than for Q_4 (8.00>5.095). Analysis of student performance indicated that students achieved higher scores when studying matched session than mismatched session. Mean values and standard deviation are also listed in the Table 1. From the first post–test and second post-test achievement scores, it has been found that there was a very significant difference of correctly answered questions t(21)=6.84; p=0.000, 2-tailed. Figure 3 depicts the fact that the mean value of scores (Q_2) after matched session is much greater than the mean score value for mismatched session (Q_4).

Table2: Post test and pre test score gains differences

Scores	Total	Mean	Std. Deviation
Q2-Q1 (Post test– pre test) Matched	*72*	*3.43*	*1.660*
Q4-Q3 (Post test– pre test) Mismatched	49	2.33	1.713
Difference	23	1.1	0.053

This table shows the total score gains between matched conditions was less than ? (23) higher than in the total score for score gain in mismatched conditions. Summing the scores brought the total number of points for a matched session to 72 points, with an average score difference of 3.4 and 49 points for mismatched session, with the average score difference reduced to 2.33 points. This indicates that the higher score difference was achieved between pre and post-tests in a matched session.

Table3: Score gains between two pre and post-tests for matched and mismatched conditions

Score gains (Q2-Q1) Matched Session	**Number of students**	**Score gains (Q4-Q3) Mismatched Session**	**Number of students**
0%	0	0%	4
10%	4	10%	3
20%	2	20%	2
30%	4	30%	9
40%	5	40%	2
50%	5	50%	0
60%	0	60%	0
70%	1	70%	1

When the subscale of score increases or gains is scrutinised (Table 3), results reveal that more students (4) in mismatched session did not have any score increase (0%) than in a matched session (0). The biggest difference is in the number of students between a mismatched session and a matched session is in 30% and 50% score gains (5 students). The same number of students in both sessions achieved a score increase of 20% and 70%. On average the two groups are comparable in terms of pre-test to post-test gains.

5.1.3 Results of the pre-tests: Q1 (matched) – Q3 (mismatched)

Table4: Pre test score (Q1-Q3) differences

Scores	Q1 (pre- test –matched)	Q3 (pre- test mismatched)	Q1-Q3
Total	*96*	*58*	*38*
Mean	*4.36*	*2.64*	*1.73*
SD I	*2.20*	*1.73*	*0.48*

Table 4 indicates the results of the pre test scores for matched and mismatched conditions. It shows that pre-tests in matched conditions are more than ? (38 points) higher than in their pre-test for mismatched conditions.

5.1.4 Difference between browsing times for matched and mismatched sessions for all students

The t-test tests were also performed for session browsing times. The completion times for browsing were measured from the start of each round until the students started answering post-tests. It was expected that the students would spend less time browsing their matched session than their mismatched session.

Table 5: SPSS results for t-test for browsing times for all students

Browsing times	Mean (mins.secs)	Number of students	Std. Deviation	Significance (2-tailed)
t1 *(time to browse matched session)*	*18.87*	*21*	*13.876*	*3.028*
t2 *(time to browse mismatched session)*	*20.68*	*21*	*9.863*	*2.152*

The hypotheses for the browsing times for all students were:

H_0: Browsing times for matched and mismatched sessions are the same

H_1: Browsing time for a matched session is significantly shorter than for a mismatched session

The above table shows that the means between two browsing times do not differ hugely, where t2>t1 (20.68 >18.87mins). The results do not indicate high significance between browsing times for matched and mismatched sessions, i.e. $t(21)=-0.495$, $p<0.626$.

5.2 Relationship between the length of browsing times and the increase in scores for all students for matched and mismatched sessions

To check for any correlation between the length of browsing times between the sessions and the scores achieved in the post-tests, a Pearson-rho correlation coefficient was calculated. The times show that students on average spent different lengths of time for the two different sessions and obtained different score increases. 6 out of 21 students achieved the same scores after the matched and mismatched sessions, even though there is a high difference in the times they spent on average (~16 mins). In one third of cases, students spent a considerably longer time browsing the matched session (32.5mins on average), one half of them spent less time browsing the matched session (3.77 mins), and one sixth of them spent an almost identical amount of time on two sessions and in this instance attained the same score. On average there was an 18 % score increase for all the students between matched and mismatched sessions. The results from the analysis reveal that the Pearson's correlation coefficient is $r=0.086$, $r^2=0.00739$, or less than 1 % of browsing time difference accounts for score difference, with a p-value of 0.356, which is clearly demonstrating not a very significant correlation. It can be concluded that the post-test scores obtained in the study are a reflection of the lesson–tailoring, rather than the length of time that the students spent browsing the courseware.

5.3. Summary of results

Alternative hypothesis 1 was confirmed. All the students achieved significantly higher scores while browsing matched session for all students, $t(21)=6.837$, $p<0.000$ (two–tailed).

Alternative hypothesis 2 was not supported. Browsing times for the matched session was not significantly shorter than for mismatched session for all students, $t(21)=-0.495$, $p<0.626$.

Alternative hypothesis 3 was not supported. All students did not achieve significantly higher scores in a shorter time, i.e. there was not a strong correlation for the matched session $r = 0.086$, $p<0.356$, nor for the mismatched session $r=-0.012$, $p<0.479$.

6. DISCUSSION

The main goal of this empirical evaluation was to evaluate if there was a very significant difference in the means of scores achieved in the matched-learning-style-session versus the mismatched-learning-style-session. The relationship between score-increases achieved between the matched and mismatched hypermedia environment and browsing times were also tested. The results obtained suggest insightful data with respect to student's learning preferences. The following discussion based on the experimental findings, concerning the identified factors provides an insight into possible reasons for these findings.

Knowledge scores from post-tests: difference between means of post- scores from matched and mismatched sessions

In analysing the responses to the knowledge questions, the scores for the two session types suggested that there was a very strong relationship between matching students' learning style to the courseware as the findings suggest that all the students achieved significantly higher scores while browsing the session that matched their learning styles.

The browsing times difference between matched and mismatched sessions

With regards to the browsing times between the two sessions, the evidence suggests that there is no significant difference between the lengths of time students spent on the two sessions. On average, the time it took to browse the matched session was only slightly different than the time it took to browse the mismatched session for all students (18.87<20.68 mins.). The conclusion is that there was not a statistically significant difference between browsing times for the matched and mismatched sessions. The lack of any significant difference between browsing times for the matched and mismatched groups may be due to a number of factors. The speculation would be that since the first session was tailored to suit their preferred style of learning, it grabbed their attention for a longer period of time. The results indicate the browsing times are not affected by the session-type may seem surprising, but a closer examination shows that it is not unreasonable. It may be suspected that students who were not 'burdened' with the additional links and learning style elements (in the sequential session) would perform significantly faster on the browsing, but their scores indicate otherwise. In fact, the results do not indicate a trend towards lower times. The amount of information displayed on the screen within a global session might have an impact on the browsing times too, although that is not to say that the study tasks in sequential sessions were not less cognitively demanding.

The browsing times versus score difference increase

The data analysed indicates that there was not a correlation between student performances and browsing times. Browsing times have shown not to affect the increase in scores after each session. 24% of students achieved a higher score difference in a shorter time. Those students who spent longer (19%) browsing their matched session achieved higher mean scores, while those students who spent less time browsing their matched session (5%) achieved lower scores. The correlation coefficient calculated for all students for a matched session was $r=0.086$, $r^2=0.0074$ (or only 0.74%) variance of score differences is explained by browsing times, with $p<0.356$ and $r=-0.012$ for mismatched session, $r^2=0.014$ with $p<0.493$.

7. CONCLUSION

This was an initial study evaluating what impact the incorporation of learning-styles within educational hypermedia courseware has on learning outcomes. In this study, with its emphasis on GCSE students, the main hypotheses postulated, regarding the mean scores difference, were found to be particularly pertinent and well founded. The findings suggest that students benefit from the learning materials being adapted to suit their learning preferences. The results revealed that students have obvious different preferences for lesson presentation type. The results suggest that the learning outcomes can be improved if designers of hypermedia courseware provide a different sequence and presentation of materials to accommodate individual learning style differences. Hence possibilities for promoting more effective learning are realised. These solid results indicate that learning styles provide a good basis with which to adapt hypermedia to individual needs. Hypermedia design features, based on student's learning styles, such as structural and linking mechanisms, have significant bearing for the future development of adaptive hypermedia systems. The next stage of this experiment is to develop a more adaptive version of the system that automatically tailors itself to users' learning needs. It combines the learning styles described in this experiment with a variety of learning strategies. It provides adaptive navigational guidance and it supports cognitive learning strategies. The system prototype is currently being evaluated.

REFERENCES

1. Bloom, B.S, Englehart, M.B., Furst, E.J., Hill, W.H. and Kratwohl, O.R. (1956) *Taxonomy of educational objectives: The classification of educational goals, Handbook 1: The cognitive domain*. New York: Longman
2. Brusilovsky, P (1996) Methods and techniques of adaptive hypermedia. *User Modelling and User adapted Interaction*.**6**, pp. 87-129
3. Carver, C. A. (1996) *Enhancing student learning by incorporating learning styles into adaptive hypermedia*, [online], http://www.eecs.usma.edu/cs383/tools.htm, [Accessed November 2001]
4. Carver, C.A., Howard, R.A. and Lavelle, E. (1996) *Enhancing student learning by incorporating learning styles into adaptive hypermedia.* Proceedings of 1996 ED-MEDIA World Conf. on Educational Multimedia and Hypermedia, Boston, USA, pp.118-123
5. Clarke, J.A. (1993) *Cognitive style and computer assisted learning: problems and a possible solution*, Association for Learning Technology Journal (ALT-J), Vol.**1**, (1), pp.47-59
6. Felder R.M. and Silverman L.K. (1988) *Learning and teaching styles in engineering education*. Engineering education **78**, pp. 674-681.

7. Felder R.M., and Solomon, B. *Index of learning styles questionnaire*. [online], http://www2.ncsu.edu/unity/lockers/users/f/felder/public/ILSdir/ilsweb.html, [Accessed November 2002]
8. Ford, N. (1995) *Learning styles and strategies of postgraduate students*. British Journal of Educational psychology, **16** (1), pp. 17-26
9. Ford, N. and Chen, S.Y. (2001) *Matching/mismatching revisited: an empirical study of learning and teaching styles*. British Journal of Educational Technology, Vol. **32**. No. 1, pp. 5-22
10. Graff, M. (1999) *Cognitive style and hypertext structures*. Proceedings of the 4th European Learning styles conference, Central Lancashire
11. Grigoriadou, M., Papanikolaou, K., Kornilakis, H. Magoulas, G. (2001) *INSPIRE: An Intelligent system for personalized instruction in a remote environment*. [online] Available at http://wwwis.win.tue.nl/ah2001/papers/papanikolaou.pdf, [Accessed March 2003]
12. Kwok, M. and Jones, C. (1985) *Catering for different learning styles*. Association for Learning Technology Journal (ALT-J), **3**, 1, pp. 5-11
13. McLoughlin, C. (1999) *The implications of the research literature on learning styles for the design of instructional material*. Australian Journal of educational technology, **15**(3), pp. 222-241
14. Pallister, J., Bowen, A., Clay, R., Dilandro, C., Phillipson, O., (2001) *Longman Geography for GCSE*. London: Longman, pp. 202-206
15. Pask, G. (1976) *Styles and strategies of learning*. British Journal of Educational Psychology, **46**, pp. 128-48
16. Pask, G. (1988) *Learning strategies, teaching strategies, and conceptual or learning styles*. In R.R. Schmeck (Ed.), *Learning strategies and learning styles*. New York: Plenum Press
17. Pillay H., and Willss, L. (1996) *Computer assisted instruction and individual cognitive style preferences in learning: Does it matter?* Australian Educational Computing, Vol. **11**, No.2, pp. 28-33
18. Riding, R. and Cheema, I. (1991) *Cognitive styles: An overview and integration*. Educational Psychology, **11**(3 & 4), pp.193-215
19. Riding R.J. and Sadler-Smith, E. (1992) *Type of instructional material. Cognitive style and learning performance*. Educational studies **18**, pp. 323-340
20. Triantafillou, E., Pomportsis. A, Georgiadpu E. (2002) *AES-CS: Adaptive Educational System base on cognitive styles*. AH2002 Workshop, Adaptive Systems for Web education, 2nd International Conference, Adaptive Hypermedia and adaptive web based systems
21. Woods, P. and Warren, J. (1995) *Rapid Prototyping of an intelligent tutorial system (RAPITS)*. Proc. ASCILITE'95, Melbourne, pp. 557-563

[1] A 'm*atched*' session refers to one where the hypermedia material is adapted so that it matches a student's learning style.

[2] A '*mismatched*' session is a session where the hypermedia material presentation and navigation do not match the student's learning style.

[3] GCSE refers to General Certificate of Secondary Education

Applying Adaptive Hypermedia Techniques to Semantic Web Service Composition

Owen Conlan, David Lewis, Steffen Higel, Declan O'Sullivan, Vincent Wade
Knowledge and Data Engineering Group (KDEG),
Department of Computer Science, Trinity College, Dublin 2, Ireland.
{Owen.Conlan, Dave.Lewis, Steffen.Higel, Declan.OSullivan, Vincent.Wade}@cs.tcd.ie
http://kdeg.cs.tcd.ie

Abstract

The semantic web has the potential to revolutionize the ways in which users search and manipulate the rapidly expanding amount of information on the web. This expansion, coupled with the capability of this information to be semantically related in a multitude of different ways, leads to increased possibilities for the Web user becoming 'lost in hyperspace'. This danger stems from the fact that technologies which encourage semantic markup do not necessarily encourage appropriate vocabulary usage, resulting in inappropriate or misleading semantic relationships being formed. Users' expectations from information services on the Web are also increasing. This will lead to a proliferation of web services in this Semantic Web. Adaptive techniques, and in particular those used in Adaptive Hypermedia (AH), may be used to alleviate the semantic overload, while at the same time meeting the raised expectations of users. This approach leads to a number of research questions: How can Web Services be (successfully) used in an adaptive environment? and How do adaptive hypermedia and open hypermedia combine? This paper argues for the integration of semantic web, semantic interoperability, web services and AH technologies to meet and exceed the information requirements of users in the emerging Semantic Web.

1. Introduction

A Hypermedia system consists of Hyper-documents. Hyper-documents are nodes which have links through which the user can navigate to other nodes [3, 6]. This allows the information in the Hypermedia System to be accessed in a non-linear manner. The user can *jump* along a link from one page to another regardless of the physical location of the linked page in Hyperspace [6]. The content and link structure of a closed hypermedia system should be designed so that any possible path through the links is logically valid for the user. The World Wide Web, however, is an open hypermedia system that has not been explicitly designed and as such there may exist many linked paths that are not obviously valid for the user.

The WWW, as the primary example of an open hypermedia system, is undergoing a fundamental change as node types are developing from human readable media to web services that can be invoked directly by automated agents. Simple web services can be combined to create a more complex and feature rich composite service. Much work in service composition derives from workflow systems where a centralized server controls the flow of control and data between services. However, for service composition to scale successfully to the WWW, the service themselves must be configurable with the conditional links to the services with which they must communicate within a composition [1]. The services, therefore, become hyper-document or hyper-services. Service composition therefore involves the discovery, brokering, intercommunication and monitoring of these hyper-services.

A given service composition will have any number of attributes describing the function it performs, its input and output information, pre-requisites, required effects and other context information, and similarly so for the constituent services (which may themselves be composite services). The end result of a service composition should ensure that some form of input will be given to any service, operated upon and then given to the next service in the chain. At some point, the flow of information will reach a service which is an endpoint in the process, and the result will be passed to a person or device capable of interpreting it.

Some parallels are already visible between this concept and those in hypermedia. There is a navigation (possibly with multiple branches at different nodes) through a structure in service composition, just as there is in hypermedia. There are links between different nodes in service composition which could be statically embedded in the service or dynamically created to suit a differing set of requirements, as there are in hypermedia. This, however, is not enough to apply the more advanced and useful techniques in adaptive hypermedia to enable easier and more automated service composition.

This paper describes the challenges presented by service composition on the WWW and examines how parallels between service composition and adaptive hypermedia (AH) point to areas where advances in AH techniques could be exploited in service composition (and vice versa). The paper examines the potential benefits and challenges from exploiting the semantic mark up of web services in AH-based service composition techniques.

2. Web Service Composition

Service Composition is the orchestration of a number of existing services to provide a richer composite service assembled to meet some user requirements. The current major interest in service composition, however, stems from the emergence of web services and the possibility of composing them to provide value-added services over the WWW. Service composition techniques typically involve expressing elemental services and composite services, the latter being compositions of elemental services and other composite services. The definition of composite services requires the expression of the flow of control and information between the elemental services. Techniques for this draw heavily on business process modeling and languages for enactable work flows.

The recent emergence of web services and their expression in the Web Service Description Language (WSDL) has resulted in XML based languages being used to define composition between WSDL service definitions, e.g. BPEL4WS, allowing service composition patterns to be readily exchanged between tools and thus reused. Such service composition specifications necessarily need to be expressed at the level of abstraction used for constituent services. Though techniques for developing business process models from business requirements are well established, they involve skilled manual activities and are not amenable to the dynamic service creation needed to meet the needs of average users lacking in business process modeling skills. A 'semantic gap' therefore exists between the model the user possesses of *what* they want to do, and the service composition models that simply express *how* this may be accomplished. Some work has been performed in reconciling the behavior of a system with a high level representation of its requirements. However, even these high level requirements are complex to express and are typically elicited by skilled requirements engineers. Adaptive hypermedia techniques have been successfully applied in areas such as e-learning, to dynamically map basic user profile information to adapt hypermedia documents to user's knowledge and preferences [4].

3. Adaptive Hypermedia Techniques

Adaptivity is used to tailor a user's view of a subject to their personal requirements. Adaptive Hypermedia (AH) technologies are often used to guide a user through a body of digital material, assisting them in their comprehension of that material. There are several ways in which AH techniques may be employed to assist the user. These techniques may be classified along axes of adaptivity. The two axes from which adaptive techniques are most frequently utilized are the adaptive navigation and the adaptive presentation axes.

3.1 Adaptive Navigation

Adaptive navigation attempts to guide the user through the system by customising the link structure or format according to a user model. The form of adaptive navigation will determine the level of guidance and freedom granted to the user within the system. Hypermedia experienced users are known to be more likely to navigate in a non-linear way. Similarly users who are familiar with the subject matter are more likely to navigate non-linearly and therefore reap the benefits of Hypermedia systems [7]. The axis of adaptive navigation describes techniques that may be used to aid a user's exploration of a body of material. These techniques range from methods that restrict the user's interactions with the content, such as link hiding, to techniques that aid the user in their understanding of the information space, such as link annotation. Utilizing these techniques the user's path through an information space may be guided by the adaptive technologies. These techniques have generally been applied to closed corpus information spaces, where the content available is well defined. Adaptive Navigation techniques require some information about the nature of the resources in the information space to decide how links to that resource should be adapted for the user's requirements. They also require information about the user and their current objectives to effectively modify the navigation structure of an information space.

3.2 Adaptive Presentation

Adaptive Presentation is the customisation of content to match characteristics specified by the user model. The granularity may vary from word replacement to the substitution of pages or the application of different media. Content may be customised to contain additional information, pre-requisite information or comparative explanations. This form of adaptivity may be implemented by fragmenting the constituent content components into discrete words, phrases or paragraphs. These components or pagelets may constitute a discrete unit of information about a concept. With this approach different pagelets may be displayed for different users. An example would be a technical term or acronym with which the user is unfamiliar. The system may substitute the unfamiliar content until the user can be introduced to the technical term or acronym.

3.3 Abstraction and Candidacy

In APeLS (Adaptive Personalised eLearning Service), based on the multi-model metadata driven approach [4], the principles of abstraction and candidacy are key to facilitating effective navigation and presentation adaptivity. To facilitate flexibility in the design and implementation of new course offerings the multi-model approach was designed to include an abstraction mechanism. It was envisaged that this mechanism would enable the collaboration of many people on the development of an adaptive course. For example, the abstraction enables the course author (knowledge domain expert) to develop a narrative describing the course sequencing not in terms of the pagelets to be added, but rather using the learning concepts to be added to realisations of the course. This abstraction allows the course author and instructional designer to design the course without necessarily being concerned with the individual pieces of content that will be used to populate the final course. Similarly the content designer can develop instances of learning content without knowing how or where that content will be used.

This abstraction is facilitated through candidate groups. Candidate groups are used to group together *like* models. For example, a candidate content group concerned with a particular learning concept may contain several pagelets, each covering the learning concept in different ways. In the case of pagelets these differences may be pedagogical or technical – some pagelets may deal with the concept from different perspectives or render the material differently for different devices.

Each candidate group has associated metadata that describes the role of the group and has the identifiers of the constituent models within the group. A simple candidate content group may be visualised as in figure 1.

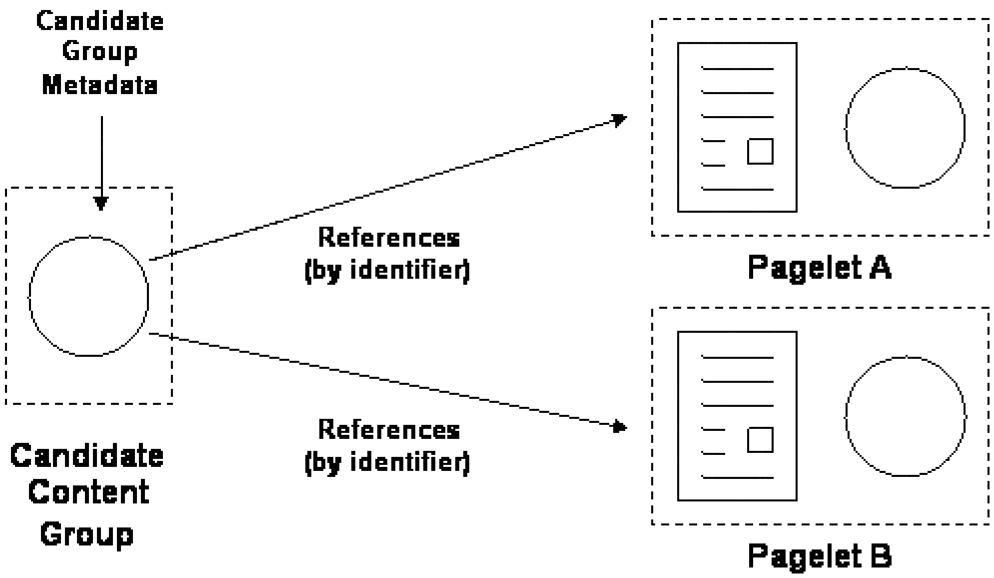


Fig. 1. Candidate Content Group

In the figure above, the candidate content group has two candidates, pagelet A and pagelet B. Pagelets in the same candidate content group are equivalent on some axis, usually the learning concept they cover. Candidate groups can be formed for any set of models in the multi-model system. For example, there can be candidate narrative groups containing narratives that produce equivalent courses according to different instructional approaches.

As candidate groups refer to their candidates by identifier it is possible for any single model in the system to be included in multiple groups. Groups do not have to be homogeneous and may contain models of different types. For example, a candidate group may have some narrative and some pagelet candidates.

In the context of an open corpus environment the capacity for creating candidate groups is limited by the potentially vast number of candidates that may exist. This process, therefore, must be based on a non-manual mechanism.

4. HyperService Composition

We propose an architecture for a hypermedia composition system which integrates the composition of services by a domain expert and the adaptive integration service in response to user needs. The whole process should begin with the Service Composition Toolkit (SCT) being fed the general task requirements from a domain expert who has some understanding of how a task can be broken down into smaller tasks which could map onto a composition of different service types likely to be available. The SCT will generate an abstract service composition that can fulfill the proposed user task and store this in a candidate repository made available to the Service Integration Engine (SIE). The SCT also uses input from a repository of User, Service and Context Classifications, which are used to impose preconditions on the use of the abstract composite service to aid the SIE in later matching it to the requirements and context of specific tasks. To improve task to composition mapping over time, the repository of previously developed services can be analyzed to see if a close match already exists.

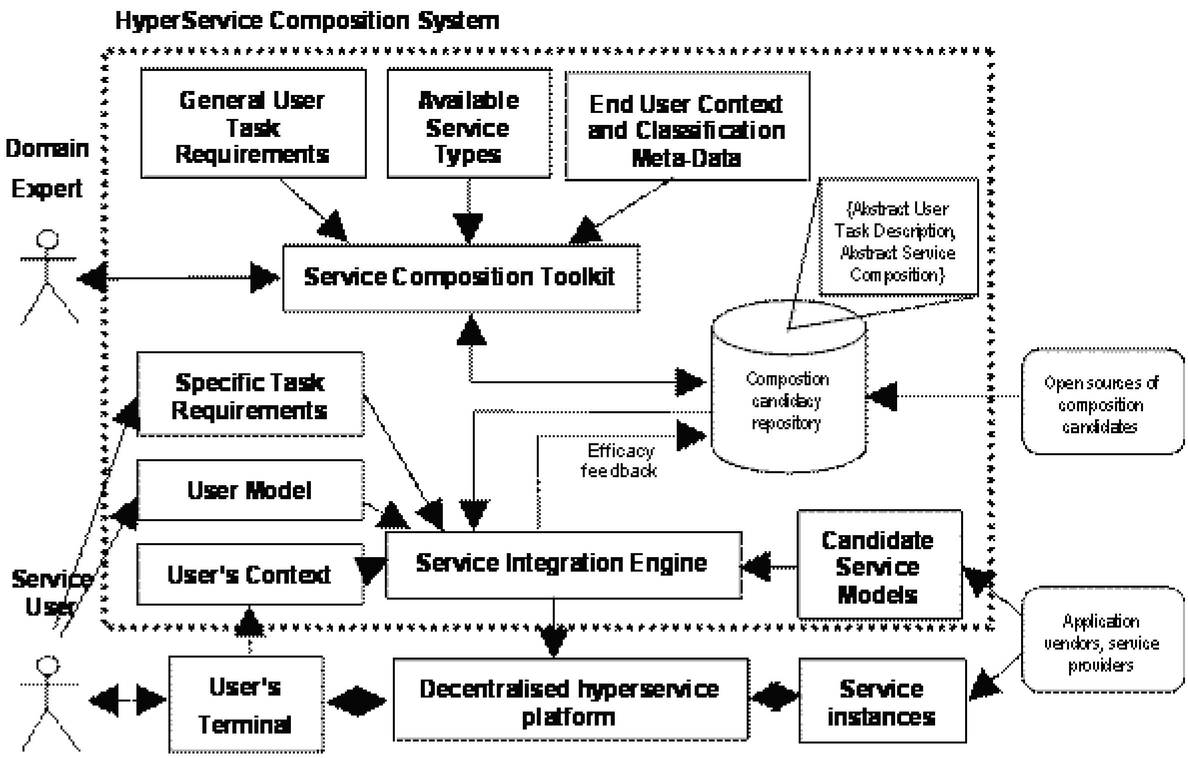


Fig. 2. HyperService Composition System

For any given task required by a user the SIE attempts to match the task requirements to general requirements and corresponding abstract service compositions contained in the candidate repository. Once a composition is selected the SIE attempts to resolve the constituent services onto available running services. The SIE also uses the User Model and User Context to personalize the composite service to the user. Failures to match the abstract composite service to concrete services or failure of user model and user context information matching the preconditions of the abstract service may result in an alternative abstract service being selected for adaptation. Successful uses of an abstract service composition are recorded in the candidate repository to aid in future matching by both the SCT and the SIE.

To illuminate this architecture further a mapping of the above concepts to existing concepts in adaptive e-learning engines is described below -

The list of sub-tasks required to complete the overall task could be mapped onto a list of learning concepts in a course (typically called a narrative). When transformed from e-learning to service composition, a sub-task like “pay for the transaction using a credit card before shipping it” could be mapped onto a learning requirement like “teach the student about regular expressions before teaching lexical analysis”.

The user model and context model could be combined to map onto the learner model of an adaptive e-learning engine. Attributes like “the user is walking down Grafton St.” and “the user would prefer to have an Irish company handle the payment for a transaction” could map onto attributes in e-learning like “the learner likes content to be explained with examples rather than theoretical descriptions” or “the learner responds well to visual explanations”.

Finally, there is the conceptual mapping of different, available services onto different, available pieces of content. In the APeLS system [conlan02], the adaptive engine is presented a number of pieces of candidate content, each of which would teach the same content but in a different manner. If the adaptive engine is presented with a candidate service, described in a vocabulary that can be mapped onto that which describes a candidate piece of content, it should be possible for the engine to select an appropriate service based on the user's preferences, as it would select a piece of content based on the same criteria.

To illustrate how this system may work in practice, an example of a course coordinator wanting to build a peer review system for her students will be used. Already developed and available services within the college might include:

- Document viewer Service - given a list of capabilities of a device and the file to be viewed, this service will convert a given document to some form which can conveniently viewed on said device. It understands that PDF is computationally intensive, so conversion to a series of JPEGs of a given size and detail can be generated for easier viewing on a low powered device. For low bandwidth devices, a text only version could be displayed.
- Access Control List (ACL) Service - Specifies that a certain entity has the right to access a certain resource from a given service. This could be built on top of existing security frameworks like Kerberos [8].
- Document Repository Service - Allows retrieval and storage of files. It should be possible to specify rules which control what can be downloaded by who. It can verify such operations either from another service or from a local set of rules.
- Candidate Selector - Very generic service; it is supplied with a group of candidates (this can be anything from users to documents) and a metric by which to select one or many of them (e.g. which document has been viewed the least times or which user has done the least amount of work).
- Grading Service - A service that accepts a user name, the name of the resource that is being graded, the scale(s) on which they should be graded and what to do with the results (typically, they should be stored in a database).
- Generic Database service - Allows simple access to a persistent relational database. The service does not perform the storage itself... it simply provides the interface to any number of databases, including MSSQL, ORACLE, MySQL, Postgresql, Access, Xindice, etc. Its input will consist of a list of operations and a set of arguments for each. Output will consist of a return code, and any data that was requested.

- Results Generator - This will be specifically developed for the task of peer review. It understands the data stored in the database, how to manipulate it and how to present results in a manner which is meaningful to both students and course administrators. The input will consist of a user name, password, operation name and output will be text displayable in a browser.

The course admin will contact the service development toolkit and she will supply it with a task description with loose keywords, along the lines of "Peer review: Project Submission, grading, limited access, document viewing" The SCT will examine a history of service compositions, seeing if there is a close match for the given criteria. Assuming there isn't, it will have to request the input of some person who (or software that) understands the required outputs.

The resulting abstract service composition will contain a “flow-chart”, outlining the different sub-tasks, their interaction with one another and various pre-conditions that must be met to allow this interaction to take place. An example could be forbidding a student to review another student's work until they themselves have submitted their project.

The list of sub-tasks for this scenario would probably include all of the afore mentioned services.

The service developer would submit this flow-chart to the SIE, which, using information based on the course administrator's circumstances and preferences, would fill in the sub-tasks with actual references to given services. A given lecturer might be very strict on plagiarism, or very concerned about security, so appropriate services could be invoked in the chain to ensure that these concerns are dealt with.

This description would be sent back to the service developer for review, to ensure that it functions correctly. If this is not the case, the flow-chart is refined and sent back to the service integrator. This process is repeated until a satisfactory result is produced.

Once a successful composition has been completed, the composition will be stored on a server somewhere, for students to access. Upon contacting the peer review compound service, the student will be asked to confirm their identity. Any running services under this instance of the composition will be marked as belonging to this user.

Firstly, the service should ask the top level results service should check to see if this user has submitted their project. If this is the case, the candidate selector should be provided with information on where to get a list of candidates (it should use the ACL service to get this information from the database service) and a metric through which it can choose which document to return.

The user may be using a PDA to view the document on the bus, so the document viewer service could convert it into an appropriate file format. Once the document has been read, the user will be given the option of submitting a review of this document. The grading service should validate that the inputs are within the correct bounds, and use the database service to store the results.

The course coordinator should be able to view submitted reviews, interpreted in a useful manner (to display various averages) with the possibility of saving the results to a spreadsheet file.

5. Further Work

Ultimately, accurate mapping of user level requirements to service composition requires modeling of the real world which forms the context in which the user naturally expresses their usage needs. Ontologies provide a way of capturing and exchanging models of the real world and making them available to automated agents. The DARPA Agent Mark-up Language initiative is defining XML-based standards for supporting the development of the Semantic Web [2], which aims to make the content of the web more amenable to processing by automated agents. As part of this initiative DAML-S [5] uses the DAML+OIL ontology language to provide semantic mark-up of web services. This includes modeling the links between a service and the outside world, by using ontologies for expressing the inputs, outputs, preconditions and effects of a service and the resources which provides it, in a way that can be mapped to semantic descriptions of the real world.

DAML-S aims to support the automation of web service discovery, invocation, composition, interoperation and execution monitoring. The principle element of the language is a Service. A Service has one or more Service Profiles which describes what the service does and includes input, output, preconditions and effects. A Service is described by a Service Model, which captures a Process Model of how the Service works. A Process can be atomic, in which case it is a single step execution, or composite, in which case it allow simple processes to be invoked through bindings to control constructs, similar to workflow control constructs, e.g. if-then-else, sequence, repeat-until, split and join etc. Both atomic and composite processes can be expressed in an abstract form called a simple process. This allows a more abstract version of a process to be defined, perhaps hiding specific preconditions or effects not relevant to a particular application or hiding the control constructs linking sub-processes of a composite service. A Service may have one or more Groundings which describe how a service may be accessed. This part of the language is integrated with WSDL, to exploit the latter's existing mapping onto communication protocols such as SOAP. WSDL is extended by using DAML+OIL for expressing the information types used in input and output parameters. Finally, the language includes the description of Resources, which provide the Service. These have types related to allocation and capacity of the resource in question, and can be composed in various forms providing guidance on the sharing of resources between services.

In terms of the proposed application of adaptive hypermedia techniques to service composition, the formal basis of DAML+OIL in description logic opens the door to the use of a range of automated reasoning techniques in the Service Development toolkit [9]. This could range from the use of rule-engines reasoning on simple conditions on service input and output to perform limited automated service composition [13] to the use of situation calculus to reason about semantic information in DAML-S specs in tools to aid developers in the simulation, verification and automated composition of web services [11]. The abstraction of existing services as DAML-S Simple Processes could be used to populate the repository of Service Types. Simple Process expressions could also be used to capture the users required inputs, output, preconditions and effect in terms that they can relate to that may be taken from the growing population of ontologies about the real world, e.g. the UNSPSC ontology of commercial product and service codes (www.unspsc.org). Candidate service composition can therefore be expressed a DAML-S composite processes made up from Simple Processes from the service type repository. By using an open language such as DAML-S for these service composition candidates, exchange or purchase of successful candidates from other organizations could be undertaken, e.g. universities could exchange successful candidate compositions combining eLearning services and specific administration services, such as coursework assessment or fee collection.

Our work on the integration of adaptive hypermedia techniques with service composition is centering on eLearning in a ubiquitous computing environment, or 'uLearning'. Ubiquitous computing is seen as a ripe application area for semantic web technologies, since similar problems in terms of the number of potential services and huge range of potential compositions exist. Therefore we are actively developing a base ontology for describing uLearning resources and basic services, including mining existing ontologies on academic learning. This will provide the semantic definitions of terms use to express user's uLearning requirements and atomic services available for composition in this context.

One potential problem raised by defining service composition candidates at an abstract level is that when that composition is mapped by the Service Integration Engine onto available concrete services there is some mismatch between semantics implemented by the concrete services. In some cases these mismatches will mean the candidate composition is not viable for the user task at hand and another alternative must be sought. However, we envisage a large number of cases where the semantics of services that must interoperate within a composition are close enough that tailored interoperability services may be automatically generated. We have conducted some research into such semantic interoperability based on the automatic generation of XSLT scripts for adapting communications between services based on a Topic Map expression of their semantics [12]. A close analogy between the expression of composition in hypermedia authoring languages and Architectural Description Languages (ADLs) is identified in [10], with hypermedia nodes mapping to components in ADLs, which implement services, and links mapping to the concept of connectors which represent communication between components. However, ADLs elevate connectors between components to first class objects, whereas the corresponding links between nodes in hypermedia models tend not to be so. Connectors are the ideal abstraction for representing the semantic adaptation functions we describe above. We therefore propose to elevate links in hypermedia model to the same status as connectors, both to provide support for representing semantic adaptors with a HyperService composition and to allow such semantic connectors to be stored as candidates for use on other service compositions. Connectors could also be defined at a syntactic level, where interlinked services in a composition have different WSDL protocol and data format bindings in their Service Groundings and thus require a suitable syntactic connector. Connectors can also represent conditional interactions and n-array interactions between services, and so could be used as reusable sub-components within composition candidates.

6. Conclusion

This paper outlines how adaptive hypermedia techniques may be applicable to the analogous field of service composition of web services in an approach we call Hyper-Service composition. An architecture is proposed that builds on an existing adaptive hypermedia engine by introducing a toolkit for developing service compositions to match user task requirements based on abstract models of existing services and known compositions. The adaptive hypermedia engine is then used to deploy concrete service composition on demand for a particular user task, based on a combination of the abstract service composition, the availability of concrete services and the user's current context. We then describe how this approach can be made more open by the use of ontologies and semantic mark-up of service descriptions, and describe research that directs us towards the use of semantic and syntactic connectors in HyperService compositions.

References

1. Benatallah, B., Dumas, M., Sheng, Q.Z., Ngu, A.H.H. (2002), 'Declarative Composition and Peer-to-Peer Provisioning of Dynamic Web Services', Proceedings of the 18th International Conference on Data Engineering (ICDE'02), ISBN: 0-7695-1531-2, March 2002, pp 297 – 308
2. Berners-Lee, T., Hendler, K., Lassila, O. (2001), 'The Semantic Web', Scientific American, pp 35-43, Issue 284 (3), 17th May 2001
3. Brusilovsky, P., "Methods and Techniques of Adaptive Hypermedia" in User Modeling and User Adapted Interaction, 1996, v6, n2-3.
4. Conlan, O., Wade, V., Bruen, C., Gargan, M. (2002), 'Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning', Proceedings of Second Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Eds. De Bra P., Brusilovsky, P., Conejo, R., Springer, LNCS 2347, May 2002, pp 100-111
5. 'DAML-S: Semantic Markup for Web Services', The DAML Service Coalition, http://www.daml.org/services/, October 2002.
6. De Bra, P., "Definition of hypertext and hypermedia", http://wwwis.win.tue.nl/ah/, 1998.
7. Eklund, J., Brusilovsky, P., "The Value of Adaptivity in Hypermedia Learning Environments: A Short Review of Empirical Evidence", http://wwwis.win.tue.nl/ah/, 1998.
8. Jochheck W., 'KERBEROS: Secure Network Authentication, Myth or Reality', Published online
9. McIlraith, S.A., Son, T.C., Honglei Zeng, H. (2001), 'Semantic Web Services', IEEE Intelligent Systems, 16(2), March/April 2001.
10. Muchaluat-Saade, D.C., Soares, L.F.G. (2001), 'Towards the Convergence between Hypermedia Authoring Languages and Architecture Description Languages', Proceeding of the ACM Symposium on Document Engineering, Atlanta, Georgia, USA, ACM Press, pp 48-57.
11. Narayanan, S., McIlraith, S.A., (2002) "Simulation, Verification and Automated Composition of Web Services", Proceedings of 11th World Wide Web Conference, May 7-11, 2002, Honolulu, Hawaii, pp 77-88.
12. Conlan, O. (2002), 'Enabling "Smart Space" Service Mobility Negotiation by Ordinary Users', Proceedings of Eurescom Summit 2002 Heidleberg, Germany.
13. Ponnekanti, S.R., Fox, A. (2002), "SWORD: A Developer Toolkit for Web Service Composition", To appear in The Eleventh World Wide Web Conference (Web Engineering Track), Honolulu, Hawaii, May 7-11, 2002 (http://swig.stanford.edu/public/publications)

Embedding Information Retrieval in Adaptive Hypermedia: IR meets AHA!

Lora Aroyo, Paul De Bra, and Geert-Jan Houben
Technische Universiteit Eindhoven - Computer Science
PO Box 513, 5600 MB Eindhoven, the Netherlands
{aroyo, debra, houben}@win.tue.nl

ABSTRACT

Traditionally, adaptive hypermedia research concentrates on "closed" applications (with fixed contents). Certain applications ask for an extension of the contents considered, with data obtained through information retrieval. This paper addresses this issue, and tries to give an insight into research questions concerning the embedding of information retrieval in adaptive hypermedia. We take a look at this issue in the context of an abstract reference model (AHAM), and in the context of a concrete implementation framework (AHA!). The goal of this paper is to identify some of the relevant questions, to give directions for solutions, and to raise the discussion on this interesting extension to adaptive hypermedia.

Keywords

Adaptive Hypermedia, Information Retrieval, Ontology, Semantic Web, Metadata.

1. INTRODUCTION

A typical assumption made in major parts of the research on hypermedia is that the content of the application is a fixed set of fragments. Moreover, it is assumed that the author of the application knows these fragments. As a consequence, the author can make decisions in the application design based on the knowledge of the data fragments and their properties. It may actually be the case that the fragment itself is not considered, but the design reasons in terms of the fragment's properties and it leaves the actual rendering of the fragment to a run-time component. A model like Dexter [17] effectively supports this separation of concerns. In the research on adaptive hypermedia (AH) this assumption is also used. Providing adaptation in a hypermedia application means that the designer decides on using different methods and techniques to make the desired adaptive application out of the available fragments. In that context, it is even more important for the designer to know the data fragments and their properties: specifying the adaptation is much easier when knowing the items used in the adaptation. For adaptive hypermedia the AHAM model [11,26], like Dexter, leaves the actual rendering of the fragments to a run-time component, therefore making it not an explicit part of the design process.

Certain areas of (adaptive) hypermedia research identify the need to consider content that is outside the application, outside meaning that the data fragments are not under direct control of the application. Often, this is interpreted in the way that the application can have access to those data fragments to include them in the presentation, but there is less knowledge available to (the designer of) the application about which fragments to consider when designing the application (and its adaptation). Applications in the area of Web information systems often take an approach like this, not knowing the actual fragments existing at the level of instances, only having information at the (schema) level of classes. Several research projects, e.g. Hera [14,15], focus on the control of adaptation in the context of data that is available only at the schema level and that is instantiated at run-time. While this approach allows for a limited notion of "outside" data, there is a need to go further. A natural next step is to include the concept of information retrieval (IR). Allowing the application to use data fragments that are obtained through IR mechanisms has an impact on the design in the sense that the designer of the application has a limited insight in the fragments and their properties. The major challenge with this step is to understand the consequence of including IR in AH applications.

This challenge leads to a number of relevant research questions.

- First, it is interesting to see how we can capture this notion of IR in terms of existing notions of AH, specifically in a reference model. We want to describe how IR actually is embedded in AH applications and try to identify how (Dexter and) AHAM can help in providing this insight.
- Second, there is an interesting issue in dealing with data that is the result of an IR request. As indicated above, the author knows part of the properties of the data but there is also a lot of uncertainty about some aspects of the data. This has an influence on the actual embedding of the "outside" data within the "inside" data. The question here is how this issue can be dealt with in concrete application frameworks like AHA! [9,10].
- Third, adaptive hypermedia is typically based on a user model. Given the limited knowledge available on data obtained through IR, the question is how this influences the user modeling in AH. We'll address this issue, by considering our ontology-based solution.

The remainder of this paper gives a short introduction to the area followed by a discussion on each of the three questions mentioned above. The nature of this paper is such that we indicate our understanding of the research issue, and we give our idea of a solution (direction). Next to this we also want to cover the issues in such a way that relevant discussions could be raised.

2. ADAPTIVE HYPERMEDIA

Adaptive hypermedia systems [2] maintain a user model in order to store users' "features", and use these to provide adaptive content and adaptive navigation support. The aim is to improve the usability of hypermedia by making them more personalized, by adapting to the user's goals. Traditionally the user modeling is based on observing the user's behavior, which is mainly browsing (following a sequence of hypermedia links). In other words, adaptive hypermedia works with a user model in order to make the hypermedia adapt to that model.

2.1 AHAM

Reference models or frameworks support the specification of adaptive hypermedia systems, e.g. by distinguishing different components of the system each modeled by a separate model. The user model (UM) is usually an overlay model of the domain model (DM) and there is a separate module (engine) applying the adaptation rules, specified in the Application Model (AM). A good example of such a reference model is AHAM [11,26], in which the UM, DM and AM cooperate to perform the adaptation.

The DM represents a semantic structure of concepts and relationships between the concepts: it identifies the concepts considered in the application and their descriptive attributes. The UM considers the same concepts and associates user-attributes to them, represented as attribute/value pairs: these attributes express for example the `knowledge-about` or `interest-in` the concept for the specific user. So, the concepts represent various topics in the subject domain and the `knowledge/interest` attribute values indicate the user's level of knowledge or interest of these concepts. The AM specifies relationships between attribute values of concepts, thus indicating the knowledge and interest values in relationship to other neighboring concepts. An important part of the AM is the set of rules to link the DM and UM in order to generate the final presentation of content. A set of requirements is given by these rules to express constraints for presenting content fragments (and the related concepts) on the basis of the current user model. The system actually reasons on the basis of the attributes' values in order to select and present content and the content format to the user. This results in the system's selection of a set of hyperlinks to be presented to the user in a specific order and in a specific color and presentation format (e.g. text, video or audio). Another set of rules defines the knowledge propagation mechanism (based on the concept attributes and the links between the concepts) for the current user model. Those rules can indicate to the system that if the user reads content fragment A, then the following links should be to content fragment B and C and the `knowledge` value of concept A and concept D (related to A) should for instance be increased by 50 (%).

2.2 AHA!

Several software platforms exist that support adaptive hypermedia applications. Here we mention Interbook [3] and AHA! [9]. Like Interbook, AHA! also offers an adaptive solution to develop on-line courses where the main browsing aspect is knowledge. Currently AHA! moves towards exploring also other aspects of browsing, such as interest-, context-, goal- or learning style-driven.

The current version of AHA! is implemented in Java as a web-based server-side (Java Servlets) adaptive application. It offers adaptation to the web pages (local or remote) requested by the user and does so on the basis of the user model, where an update of attribute values appears each time the user accesses a page. The user model, as well as the specific domain model structure of concepts and attributes is preliminary designed as an overlay of the existing generic domain model. The adaptation rules are generically constructed and designed by the author. The most recent version of AHA! contains a user-friendly authoring tool for editing all three models (DM, UM, AM). Note that in AHAM a separation between DM and AM is advocated, although in AHA! the two models are working closely together.

The main interaction between the user and an adaptive application empowered by AHA! results in the presentation and access of "desired" (by the user) links on a web page. The notion of desired is deduced from the values of the user model attributes. A link (and the content and concepts behind it) could be *interesting,not interesting or already visited.* This is usually indicated with different colors (called *good*, *bad* or *neutral*). An example of this interaction is given in the implementation of the course "2L690: Hypermedia Structures and Systems" (http://wwwis.win.tue.nl/2L690/) given to fourth year students in Eindhoven. The course uses a local collection of learning material stored in HTML pages and data in XML files for the concept structure and the user model. Currently, the research on AHA! concentrates on opening up AHA! for external sources, making authoring of DM and AM more user friendly, and a more flexible inclusion of fragments or objects. This paper is part of the attempt to include external sources that are not known beforehand.

3. RETRIEVING CONTENT FROM OUTSIDE

3.1 Content Inside

In traditional adaptive hypermedia applications, for example existing applications running on AHA!, the content is typically known to the author of the application and under his control. Actually, the content is stored, at least conceptually, inside (with) the application, and as AHAM describes the storage part of the application is closely related to the part that realizes the adaptation. Usually the author's task is to structure the content that is to be presented and to fix the storage of the content within the system. Figure 1 shows what the architecture of the application looks like in a situation where the content is available within the system: the Content layer captures the data as it is stored in the application's repository and it captures the description of the data as it is made available to the Application layer that actually provides the (adaptive) hypermedia structure to the application's users.

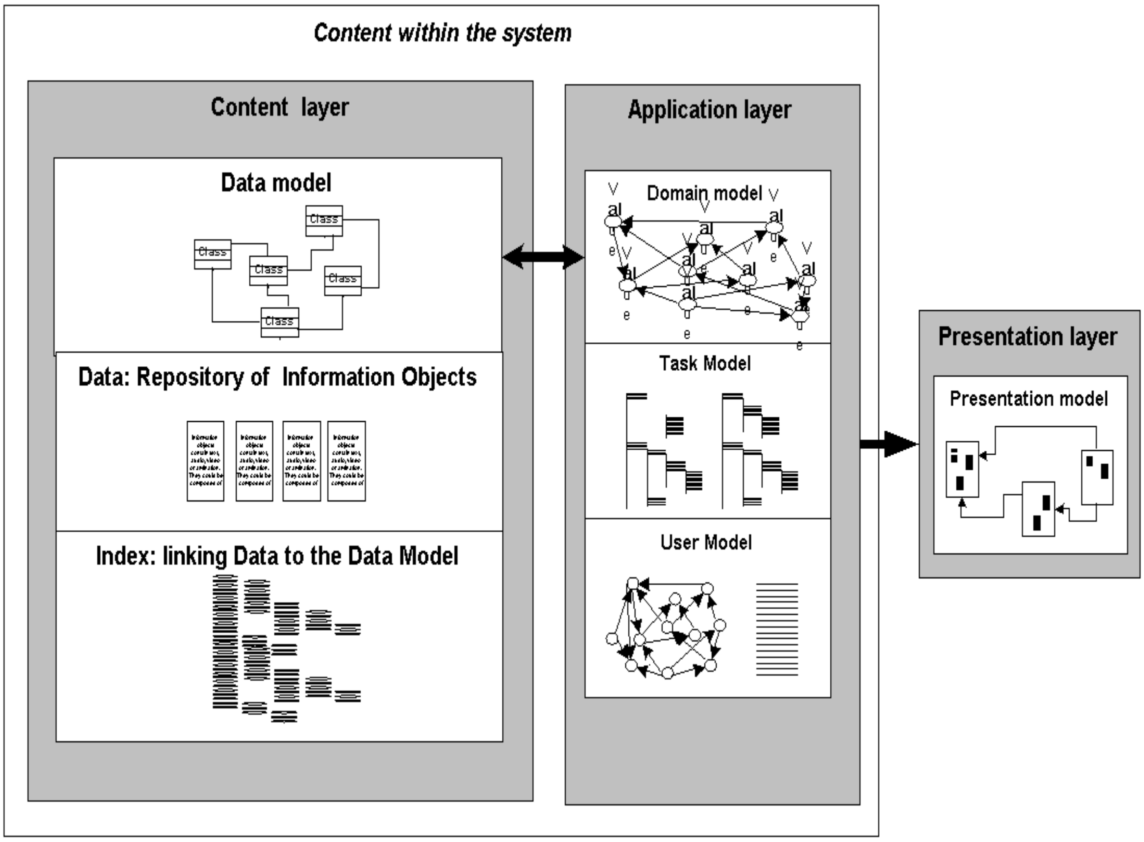


Fig. 1. Content layer inside the system.

3.2 Content Outside

Placing the content outside of the application is a move that is already visible in the context of a large class of information systems known as Web-based Information Systems (WIS) [18]. They are information systems, often database-driven, that exploit the Web paradigm and use Web technologies to retrieve data from sources reachable through the Web and deliver them to their users (see Figure 2 for Hera's perspective on WIS [14,15]). Typically, a WIS delivers the data to the users in a web or hypermedia presentation, requiring that in comparison to a handcrafted (adaptive) hypermedia application, a WIS needs to generate (automatically) a hypermedia presentation for the data to be delivered. We mention (general) methodologies or frameworks for WIS engineering, like Araneus [20], WebML [8], UWE [19], and Hera [14,15], of which only a few explicitly include the adaptation aspect. Most of these approaches consider the outside content to be known at the schema (class) level, such that the design is based on that level, leaving the actual run-time retrieving and rendering of the data at instance-level a separate and less prominent issue.

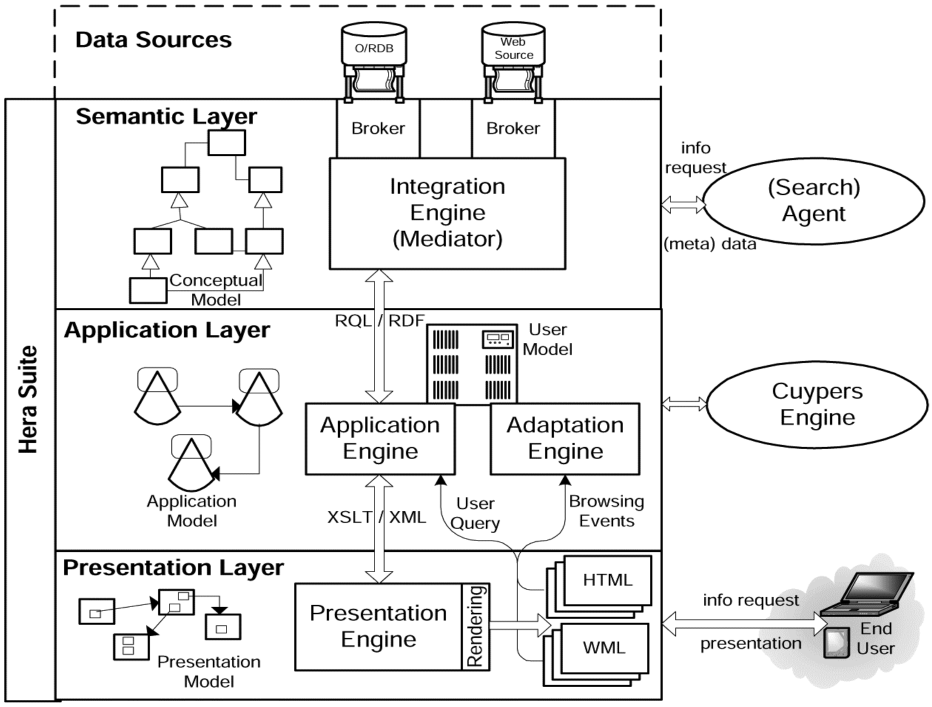


Fig. 2. WIS architecture (Hera view).

As indicated above, most of the research on hypermedia or multimedia generation [24] concentrates on the output of the application. There has been less attention to the input aspect. Some research has addressed the fact that the content that is to be retrieved on-demand can be at different, distributed places, which implies a process of integration based on a complicated alignment of the semantics associated to the different sources [25]. However, this still assumes that besides the semantic alignment the main part of the process is hard-wired. Since typically a schema-level definition is known, the application can rely on schema properties, but not on the properties of the concrete instances that exist at the present time. As an example, in [14] an approach completely based on RDF(S) is chosen in which mappings (XSLT transformations) at instance level are automatically generated from schema mappings.

As a further extension to dynamic, distributed content, a WIS may choose to access the content via information retrieval (IR). Specifically, when the data is gathered using IR techniques, not all properties of the content are available for the application. Only some metadata of the content is available for the application to perform its task. In comparison to the database-generated content, that is often implemented using fixed query dialogues, content accessed through IR implies communication on the basis of a limited and dynamic set of metadata.

3.3 IR Extension to AH Model and Application

In the case of retrieving content from outside the application, we may want to distinguish between different situations. It is possible to just have an “outside” fragment, while it is also possible to have a search (retrieval) request that results in an entire set of fragments that have to be made accessible (we address these issues further in the next section). In each of the situations, we are dealing with outside contents that have to be connected to the given AH application.

In principle both situations lead to a different implementation of what is called the resolver function in models like Dexter or AHAM. The purpose of this function is to obtain the content that “realizes” the desired conceptual fragment. In the case of outside content this function has to be related to the outside sources in order to identify the data that has to be obtained.

This identification of outside content requires that the resolver function has knowledge about the terms or concepts used in the sources. It is not sufficient anymore to just have some internal identifier for the stored data: the resolver will be more like a retrieval or search operation that mentions some terms or keywords. These search terms reflect the concept associated with the page (e.g. *“Amsterdam”*). However, this might include a mapping from the terms known as concepts in the application to terms as they are referred to in the outside space. Often there is an ontology available for that outside retrieval space, and then an ontology mapping or articulation [25] can give the term or terms to be used (e.g. *“city of Amsterdam”*). Usually, the author will want to include also additional terms in order to state the context of that retrieval (e.g. *“the Netherlands”*,*“culture”*). This would allow the author to adapt the retrieval request on the basis of his perception and knowledge of the search space (or source).

According to the AHAM reference model the author captures the relevant metadata concerning the content in the Domain Model (DM), and subsequently the User Model (UM) and Adaptation Model (AM) are based on this representation. The challenge for the author when the content is outside the application (i.e. not only inside) and IR is used to obtain the data, is to capture the relevant details of the content in terms of metadata. We assume that for the purpose of IR we have an ontology available that specifies (the structure of) the search space: see Figure 3.

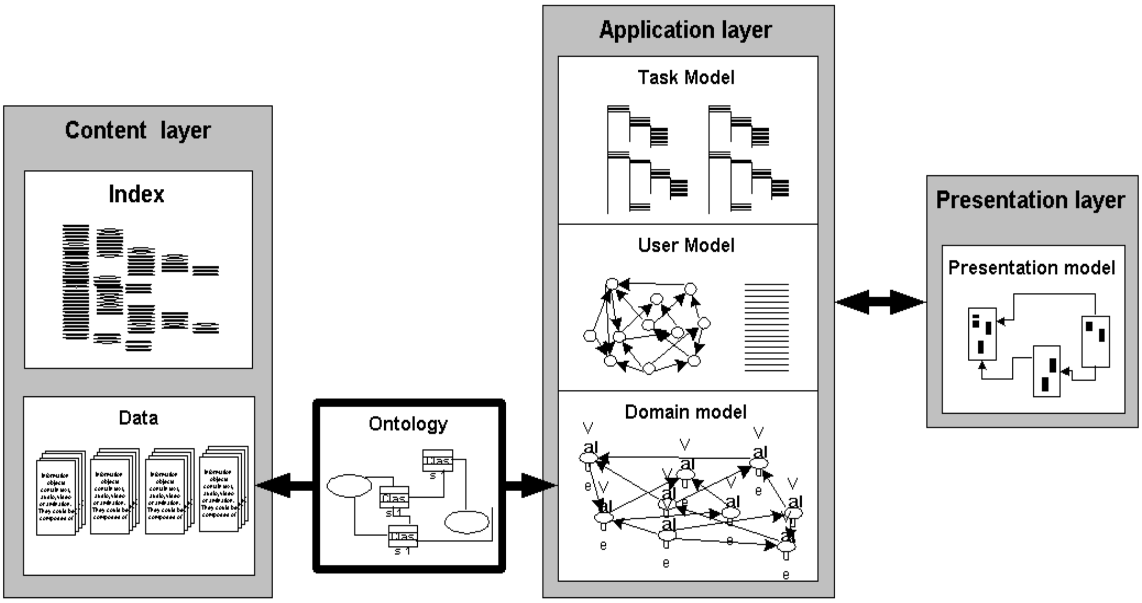


Fig. 3. Content available through ontology.

Since this ontology basically is the only means for the application to “reason” about the (outside) contents, we extend AHAM by relating this ontology (O) to the DM, UM and AM. We add a Retrieval Model (RM) that captures the relation between the retrieval space and DM. This RM will consist of a set of mappings or articulations (M) between concepts from DM (representing the application) and terms from O (representing the retrieval space). The ontology O describes the metadata of the content as it is available for the application, and hence it acts as the “contract” on which the application interacts with the content. In any adaptive hypermedia application the metadata that goes with the content is the main “tool” that the author has available to manipulate the content and is able to apply effectively in the application. If the content is outside and therefore not directly available to the author, this tool is even more important since it is the only means to reason about the content that is to be used in the application.

The retrieval model RM describes the relationship between O and DM, and is actually quite similar to the integration model from [25] (given in Appendix A). Each mapping in RM relates a concept of the conceptual model (DM in AHAM) to a collection of terms in the ontology that describes the sources. This mapping might actually include metadata on quality dimensions or weights that should be taken into account in the search or retrieval process. The mapping is called for whenever the AM decides that the resolver function has to obtain the fragment that goes with the DM concept.

It appears that with this extension to AHAM we are able to specify the necessary associations between DM and O. Implementing this in the same way as the integration process was implemented in [14], an elegant and effective RDF(S)-based solution opens up the way to combine Semantic Web and AH approaches.

4. IR EXTENSION ALTERNATIVES

As indicated in the previous section we can distinguish between two concrete situations: one in which the IR extension solely implies the inclusion of one outside fragment, and another one in which an entire navigational structure over outside content retrieved through search is added to the application. Here, we address them as illustrative examples in order to identify the relevant research questions.

4.1 Outside Fragment

We start looking at the simplest situation of embedding outside contents inside the navigation structure that has been designed for the application. In standard AH applications the content is fixed in and known to the application, and the adaptive part of the application communicates with the storage part through some internal layer that delivers data fragments when given fragment identifiers. Now let us assume for an AH application that within the given navigation structure we decide to include content (data fragments) that is stored elsewhere (outside) and that is retrieved on-the-fly. A typical result of embedding retrieval in the standard navigation is to have a link to a page that:

1. includes a fragment that is retrieved from outside the application and not from the internal Storage Layer, and
2. is like any other page in the application in the sense that it contains links for further navigation (typically this navigation is contained in fragments surrounding the embedded outside fragment).

In this way the page is included in the navigation structure as designed by the author (by supplying the outgoing links to continue browsing in the application), maintaining the main ideas reflected in that design. The nice thing is that for the user there might not even be a visible difference, except for the fact that the included fragment might contain links that are not under control of the AH application. For the author the main difference is the need to specify explicitly how to obtain that fragment.

Example A concrete example is a page on Amsterdam in our museum application, where the data on Amsterdam is included dynamically via the Web. For this purpose the fragment can be addressed through a Web address specifying the site of the Tourist Office. In this case (choosing the solution of the outside fragment) we might want to make sure that the presentation of the information from the Tourist Office is such that the reader recognizes that the information is made available through the application, but is not part of the application, and that the navigation from that information is recognized as outside navigation.

4.2 Retrieval through Search

If we take the search approach, we do not state one address to obtain a dynamically generated fragment. Then, we are confronted with the problem that the retrieval request results in a set of fragments. For this set of fragments an access structure needs to be defined, and that access structure has to be embedded in the existing application. The part of the embedding of the access structure is the same as discussed in the previous section on outside fragments. What is essentially different is the composition of this fragment offering the access structure with links to the retrieved fragments: design questions are for example how the different links are ordered and annotated.

Example Taking the same example as before offering information on Amsterdam in the museum application, the situation is different if the information on Amsterdam is retrieved through a request to a search engine. The system will obtain a set of links to interesting documents about Amsterdam, and the application has to find a way to present that list. Not only the first presentation needs to be specified, also the effect of navigation within that additional part of the navigation space.

5. PRESENTING RETRIEVAL RESULTS

One important part of using retrieval is the specification of the retrieval request: the description of *what* needs to be retrieved. However, after the retrieval request has been issued, data will be retrieved and then the application should know *how* to present the retrieval results, i.e. how to embed the retrieval results in the application. It is obvious that this part of the retrieval process, the access structure to the retrieval results, needs to be included in the author specification as well.

One of the first things the author has to decide is whether the retrieved data can be captured in one fragment or is better made accessible through a fragment that only gives an access structure (e.g. a list of links) to other fragments/pages. There are other variants possible, but these two represent the situations of single or multiple retrieval results.

Furthermore, if links are used in the access structure, it is has to be made clear how they are presented and annotated, and how this addition to the navigation space relates to the original navigation structure of the application. In this respect we refer to Web engineering methodologies (like Hera) that are designed to specify such dynamically generated access structures and dynamically composed fragments.

Let us consider the prime decision of which (part) of the retrieval results is actually presented. If the retrieved data is a (possibly large) set, then the author has to define the composition of the result fragment. This includes the definition of which elements are actually selected (e.g. top-3), and how they are ordered (e.g. based on relevance, or "first internal links, then external ones"). This design aspect asks that choices can be made by the author either at a generic level (for all cases), or at a specific level per fragment or anchor. It is clear that it depends on the nature of the application what is needed. In the case of a hand-crafted application, e.g. a course, the author can, without exactly knowing what is going to be retrieved, specify how the retrieval result is offered. In the case of a WIS with dynamically generated content, e.g. a shopping catalogue, it is much harder for the author to specify the result presentation, since in this case also the environment surrounding the embedding is only known at generic (schema) level. For the latter case methods like Hera are designed: for an effective combination of these specifications at schema and instance level RDF(S)-based approaches appear useful.

6. RETRIEVAL AND USER MODEL

In the case of adaptation, by definition the user model (UM in AHAM) plays an essential role. For the specification of the retrieval, the author does not only have to consider the actual retrieval request and the result presentation, but also has to determine the update of UM. In principle, the retrieval request contains a number of search terms that can be used to determine the UM update.

Let us consider the different origin of a couple of search terms for their effect on UM:

- Some of the search terms are taken from the source ontology (O), as they are the terms that correspond to concepts mentioned in the domain model (DM) of the application. If for example on the basis of the source ontology the search term *"Nederland"* is replaced by the term *"the Netherlands"* which happens to be a DM concept, then that concept should be updated in UM. Note that it is possible that the mapping between ontology terms and DM concepts does not have to be one to one, and therefore it is possible that the use of an ontology (search) term requires an update to other DM concepts as well.
- Often, terms (from the ontology O) are added to the request that help to specify and refine the context of the search, e.g. *"culture"* or *"Holland"* could be added to the above mentioned request for information about Amsterdam. These terms often do not require UM updates, although the author might take into account that by adding these terms general information might be retrieved that could overall add to the general knowledge.
- Terms can also be added from the UM, and in that case the corresponding UM terms could be updated. If on the basis of UM *"river Amstel"* is added to the term *"Amsterdam"*, then the concept *"river Amstel"* might be updated in UM
- An interesting second (part of) UM can be formed by the search history, remembering which search terms have been previously used. If for example a pattern in the search history can be detected, e.g. the user keeps on searching for a certain term, this can lead to a reaction by the application, e.g. a certain adaptation or even a modification to the application.

Note that in the above we limit ourselves to the situation where the UM update is only based on the (extended) retrieval request. We neglect that the retrieved data might not match those search terms, and without any additional interpretation software we cannot improve on the accuracy of the UM update.

Example Suppose content is obtained through retrieval for the DM concept *"Night Watch"*. Let us assume that the author on the basis of the ontology has defined that the content is retrieved with the request *"night watch AND nachtwacht AND Rijksmuseum"*. First, it is possible that data is retrieved that does not actually contain proper information on *"night watch"* in the sense of the DM concept: this could be the consequence of a wrong annotation in the outside search space. Second, it is possible that the content that is retrieved contains information on other DM concepts besides *"night watch"*: the retrieved data could give a survey of paintings that nicely explains what the influence of the Night Watch has been on other paintings.

The two situations described in this example ask for a decision by the author. The author should decide how the retrieved content contributes to the knowledge about this and other concepts as reflected in the UM. We assume for the moment that the mappings are fixed and known to the author, but what if they are not?

In order to take into account that the information might not be accurately annotated, the author could specify the UM update in such a way that the value associated with the concept reflects this uncertainty. If UM contains concept attributes like `knowledge` and `read`, then the author could choose to give a value like `possibly-known`, or `certainly-read` to reflect his expectation of the accuracy of the retrieval. Note that the abstract reference model AHAM only specifies attributes and attribute values. In a concrete AH system like AHA! it is possible to use numeric values between 0 and 100, and that opens up the possibility to represent nicely the *expected* value of the property. In most current AHA! applications these numeric values are used to specify the contribution of (the knowledge about) concepts to (the knowledge about) other concepts, e.g. a knowledge of 80 on `adaptation`represents that 80% of the knowledge on the concept of adaptation is obtained. For retrieval the existing AHA! mechanism can be used to represent that the author *expects* that 80% of the knowledge is obtained.

The other anomaly is that content is retrieved that (also) contributes to the knowledge about other concepts from DM. Besides the fact that the author has to specify his expectation as described above, the author can also include the extended retrieval request to look for clues how to update UM. In the example that we gave for the concept *"Night Watch"* a retrieval request was constructed with three terms: *"night watch AND nachtwacht AND Rijksmuseum"*. If the terms *"nachtwacht"* and *"Rijksmuseum"* occur in DM as well (maybe after some renaming), then the author should consider updating these concepts as well: probably not with a major increase of knowledge, but it is foreseeable that adding these terms leads to some knowledge about these concepts as well. For those terms mentioned in the retrieval request the author can update UM accordingly; for concepts that coincidentally are covered, the author cannot foresee a UM update.

As a last issue in connection with the UM update, we mention that in the situation of retrieval through search an additional navigation structure was added (on the basis of retrieved outside content). The UM can also play a role in annotating the links in that navigation structure based on the user's exploration of that structure: this would require that (dynamically) that navigation structure is made adaptive itself by adding it to the adaptive hypermedia application.

7. RELATED WORK

We shortly mention three related research fields. Also within the area of Open Hypermedia (OH) there is a major interest in the relation between the metadata descriptors of the document's content and ways to retrieve this document and apply it for reaching specific educational or other goals. In this way OH builds upon the existing WWW infrastructure, provides a powerful framework to aid navigation and authoring, and solves some of the issues of distributed information management [12]. The interesting issues in this context relate to flexibility, which OH architectures allow in providing opportunities for adding various kinds of links to the documents and creating a user-specific navigational overlay [6,7]. OH systems make it possible that the (hyper)links, which traditionally are embedded in the web documents, could be abstracted from the documents (multimedia data) and stored and managed separately, which includes also searching of links in the same paradigm as for documents. In this way OH systems could contribute in the direction of presentation of the retrieval results and the integration of IR within the adaptation process.

Another step towards more conceptualisation and semantics in the information management within hypermedia systems is the effort of Conceptual Hypermedia Systems (CHS) to provide a well-defined conceptual schema for the hypertext structure and navigation (e.g. [4,21,23]). A major research effort is performed within the context of the TourisT [5] prototype of a Tourism Public Access System. CHS give us quite a powerful mechanism to support dynamic link and document management, based on metadata descriptors of the real content. It builds upon the notion of Hypermedia and Open Hypermedia and makes the bridge with the Semantic Web. Another example in this research is the COHSE project [16], which aims at building a conceptual hypermedia system to enable documents to be linked on the basis of the metadata content descriptor. This is realised by integrating ontology reasoning service (domain modelling) and open hypermedia link service (link providing).

Due to lack of space we only mention two references with respect to adaptive information retrieval. Combining text and semantic markup directly influences IR: we mention IR performed on the basis of knowledge representation languages, such as RDF(S) and DAML+OIL (OWL), which can help us achieve more flexible and precise information retrieval [22]. Another example of adaptive IR or in other words context dependent IR, is given by systems like AIMS [1,13], where the successful IR depends on the modelling of user tasks and learning goals in relation to the domain model, on the conceptual visualisation of the IR results in a semantic network of domain concepts, and on including the user characteristics within the search query.

8. CONCLUSION

This paper has addressed our preliminary vision on the extension of adaptive hypermedia to include aspects of information retrieval. In terms of our research on the AH implementation framework AHA! (and associated, the reference model AHAM) we have identified a number of issues in connection to "opening up" AH for the retrieval of content from outside the application. We have seen how the relationships between internal (DM) concepts are made to outside contents, by means of descriptive metadata that relates the outside content to terms from a source ontology: we have introduced the Ontology and Retrieval Model to the perspective of AHAM. An important aspect of retrieving multiple fragments at a time is the design of the presentation and access structure for the collected fragments. As the user model is a central issue in AH, the desire to optimally consider that user model in dealing with outside content also has significant consequences for the design and specification of the application. Concentrating on these aspects of information retrieval, the next step is to translate the matured conceptual ideas to concrete implementations.

REFERENCES

1. Aroyo, L., Dicheva, D., and Velev, I. Conceptual visualisation in a task-based information support system, Proc. EdMedia, AACE, 2000, 125-130.
2. Brusilovsky, P. Adaptive hypermedia. User Modeling and User-Adapted Interaction 11, 87-110.
3. Brusilovsky, P., Eklund, J., and Schwarz, E. Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems, 30 (1-7), 1998, 291-300.
4. Bruza, P.D. Hyperindices: a novel aid for searching in hypermedia, ACM Hypertext, 1990, 109-122.
5. Bullock, J., and Goble, C. TourisT: The Application of a Description Logic based Semantic Hypermedia System for Tourism, ACM Hypertext, 1998.
6. Carr, L., Davis, H., De Roure, D., Hall, W., Hill, G. Open Information Services, Computer Networks and ISDN Systems, 28 (7/11), 1996, 1027-1036.
7. Carr, L., De Roure, D., Hall, W., Hill, G. The Distributed Link Service: A Tool for Publishers, Authors and Readers, World Wide Web Journal 1(1), 1995, 647-656.
8. Ceri, S., Fraternali, P., and Matera, M. Conceptual modeling of data-intensive web applications. IEEE Internet Computing, 6 (4), 2002, 20-30.
9. De Bra, P., Aerts, A., Smits, D., and Stash, N. AHA! Version 2.0, More Adaptation Flexibility for Authors, Proc. AACE ELearn, 2002, 240-246.
10. De Bra, P., and Calvi, L. AHA! An open adaptive hypermedia architecture. The New Review of Hypermedia and Multimedia, 4, 1998, 115-139.
11. De Bra, P., Houben, G.J., and Wu, H. AHAM: A Dexter-based reference model for adaptive hypermedia. Proc. ACM Hypertext, 1999, 147-156.
12. DeRoure, D., El-Beltagy, S., Gibbins, N., Carr, L. and Hall, W. Integrating Link Resolution Services using Query Routing, Proc. 5th Open Hypermedia Workshop, 1999.
13. Dicheva, D., Aroyo, L. An approach to intelligent information handling in Web-based learning environments. Proc. ICAI, 2000, 327-333.
14. Frasincar, F., Houben, G.J., and Vdovjak, R. Specification framework for engineering adaptive web applications. Proc. World Wide Web Conference, Web Engineering track, 2002.
15. Frasincar, F., and Houben, G.J. Hypermedia presentation adaptation on the Semantic Web. Proc. Adaptive Hypermedia and Adaptive Web-based Systems (AH2002), LNCS 2347, 2002, 133-142.
16. Goble, C., Bechhofer, S., Carr, L., De Roure, D., Hall, W. Conceptual Open Hypermedia = The Semantic Web?, Proc. SemWeb'01, 2001.
17. Halasz, F., and Schwartz, M. The Dexter Hypertext Reference Model. CACM, 37 (2), 1994, 30-39.
18. Isakowitz, T., Bieber, M., and Vitali, F. Web information systems. CACM, 41 (1), 1998, 78-80.
19. Koch, N., Kraus, A., and Hennicker, R. The authoring process of the UML-based web engineering approach. Proc. First Int. Workshop on Web-Oriented Software Technology, 2001.
20. Mecca, G., Atzeni, P., Masci, A., Merialdo, P., and Sindoni, G. The Araneus web-based management system. Proc. ACM SIGMOD, 1998, 544-546.

21. Nanard J. and Nanard, M. Using structured types to incorporate knowledge in hypertext, Proc. ACM Hypertext, 1991, 329-342.
22. Shah, U., Finin, T. and Joshi, A. Information retrieval on the semantic web, Proc. ICIKM, 2000, 461-468.
23. Tudhope, D., Taylor, C. Navigation via similarity: automatic linking based on semantic closeness. Information Processing & Management 33 (2), 1997, 233-242.
24. van Ossenbruggen, J., Geurts, J., Cornelissen, F., Hardman, L., and Ruthledge, L. Towards second and third generation web-based multimedia. Proc. World Wide Web Conference, 2001, 479-488.
25. Vdovjak, R., and Houben, G.J. Providing the semantic layer for WIS design. Proc. CAiSE 2002, LNCS 2348, 2002, 584-599.
26. Wu, H. A reference architecture for adaptive hypermedia applications. PhD-thesis Technische Universiteit Eindhoven, 2002.

A Flexible Layout Model for a Web-Based Adaptive Hypermedia Architecture

Peter Brusilovsky
School of Information Sciences
University of Pittsburgh
135 North Bellefield Avenue
Pittsburgh, PA 15260
peterb@mail.sis.pitt.edu

Paul De Bra and Tomislav Santic
Information Systems Department
Eindhoven University of Technology
PO Box 513
NL 5600 MB Eindhoven, the Netherlands
debra@win.tue.nl, tomi@santic.nl

Abstract

The paper presents an attempt to extend a general-purpose adaptive hypermedia (AH) architecture AHA! with flexible and rich interface by re-using some ideas explored in a more specific AH system InterBook. The most popular application (course) created with the old version of AHA! (the hypermedia course 2L670, later called 2L690, at the TU/e) used a single-window/single-frame concept representation which had advantages of its simplicity and clarity for the user. In this course concepts are always represented in a single frameless window. This mechanism is not sophisticated enough to serve as a general-purpose mechanism for concept representation. InterBook on the other hand has a much richer user interface characterized by the use of multiple windows and frames. This approach has more possibilities for representing the course domain, but has a problem of its rigid presentation structire. The author has no possibilities of adapting the user interface to the specific course which means that every course served by Interbook has the same look independent of the course characteristics. To address the lack of user interface possibilities in AHA! we developed the *Layout model* that employs the strong points of Interbook user interface. The dynamic structures of the *Layout model* are easily extendible and give author the power to adapt the user interface to the nature of the course.

1. Introduction

Numerous Web-based adaptive hypermedia systems have been developed within the last 10 years [8, 9, 10, 11]. These systems all have different “look and feel” and offer different ways of adaptation. Yet, behind this diversity an expert can find a reasonably limited set of methods and techniques [1,2]. A major motivation behind AHA! project [4,6] was developing a flexible adaptive hypermedia architecture that can be used for implementing a vide variety of adaptation methods. AHA! was created as an “assembly language” of adaptive hypermedia in the sense that any higher-level adaptation paradigm can be expressed in terms of AHA! and simulated by the AHA! engine. The most recent AHA! version [4] has shown to be very powerful in this respect. As was recently demonstrated, a reasonably advanced adaptation paradigm implemented in InterBook system [3] can be almost completely simulated by AHA! [4, 12]. Yet, with all this power, AHA! is not completely ready to serve as a universal simulator for an arbitrary AH system.

The problem here is that each AH system is characterized not only its unique model of adaptation, but also by its unique interface. While the current version of AHA! is quite ready to simulate the adaptation behavior of any AH system, simulating their interface is a lot more difficult. The AHA! engine operates in a single-window mode - in a good tradition of classic hypertext systems. In contrast, most of modern adaptive Web-based hypermedia systems use rich multi-frame and multi-window interfaces. InterBook is a good example here. It uses several multi-frame windows (textbook, glossary, and table of contents). An example of InterBook's Textbook and Glossary windows is provided on Figure 1. Other advanced AH systems also use complex multi-frame windows. Creating such multi-frame and/or multi-window interfaces in AHA! requires that the author define the frame structure and use Javascript code to synchronize the presentation in all the frames.

The goal of the project introduced in this paper was to resolve the problem by developing a flexible interface model of AHA! engine. Following the idea of AHA! that can be used to describe a variety adaptation functionalities, we wanted to develop an interface model that is used to describe a variety of Web adaptive hypermedia interfaces. The primary goal of our project was reasonably modest: we wanted to extend the AHA! engine to the extent where it can simulate the InterBook adaptation mechanism and its multi-frame interface. In doing so, we wanted to avoid narrow-minded solutions and hacks (like the Javascript hack previously used with AHA!) developing a reasonably universal approach that can be used to implement the InterBook interface along with many other interfaces. This paper presents the first results of our work. To introduce the background for our work, we start with a brief introduction of the InterBook and AHA! interfaces. After that, we present our Layout Model that extends AHA! and demonstrate how it can be used to simulate the InterBook interface in AHA!.

2. The InterBook Interface Paradigm

InterBook has two main kinds of windows - a Textbook window (left on Figure 1) and a Glossary Window (right on Figure 1). These windows correspond to two major kinds of information items supported by InterBook - a book page and a domain knowledge concept. Each window in InterBook can include multiple links to concepts and pages. A click on any page link causes the linked page to be loaded in the Textbook window. A click on any concept link causes the information about the linked concept to be loaded in the glossary Window.

Despite of its complicated interface, InterBook attempted to support a simple metaphor - one window shows one and only one information item - i.e., a textbook window shows exactly one page of a textbook at a time. While each of these two windows includes several frames, there are considered not as independent windows, but as multiple views on the same concept or page. For example: the text frame (bottom left) presents the text of the page; the navigation bar (top) presents the location of the current page among its ancestors and siblings, and the concept bar (bottom right) presents prerequisite and outcome concepts for the current page. All four frames of the textbook window are updated at the same time. Technically, a link to a textbook page is calling a whole *page frameset* to be loaded into the textbook window. This frameset, in turn, pulls several frames associated with the requested page. The frameset approach is simple to understand and also works well with most browsers' standard way of navigation using back and forward buttons and history.

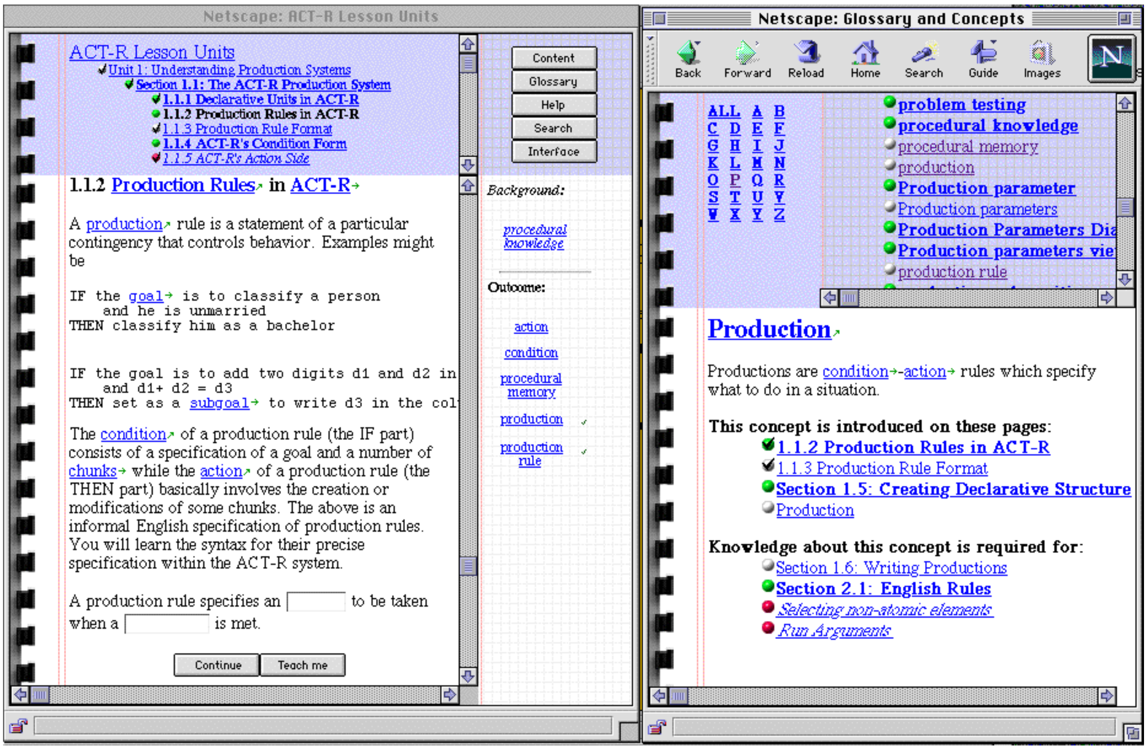


Fig. 1. InterBook interface.

3. The AHA! Interface Paradigm

AHA! was initially created to add adaptation to the course on hypermedia at the Eindhoven University of Technology (http://wwwis.win.tue.nl/2L690/). This course predates the general availability of frames in browsers. The course was therefore written using a single frame layout. The browser showed one course page at a time, with adapted links and conditionally included fragments. AHA! also added an optional header (with a progress report) and footer (with copyright statement). Header and footer were created by the author as html fragments. In Figure 3 a page with header can be seen (albeit embedded in the new Interbook-like multi-frame environment). Multi-frame applications are possible in AHA!. Figure 2 shows part of the multi-frame interface paradigm used in a course on graphical user interfaces.

In order to make this interface work in AHA!, every page must include the following piece of Javascript code:

```
<script language="JavaScript">
   parent.frames[0].location="content.xml"
</script>
```

The result is that when a link to a page is followed the leftmost frame is reloaded. It contains the “content.xml” file, which is a navigation menu in which submenus are conditionally shown, based on which page is displayed in the rightmost frame. The access to a page and the subsequent access to the menu are treated separately by AHA!. Whereas in Interbook following a link requests a complete frameset from the server in AHA! following a link requests a page, to be shown in a complete browser window or inside a frame. The AHA! engine does not “know” about a possible use of frames.

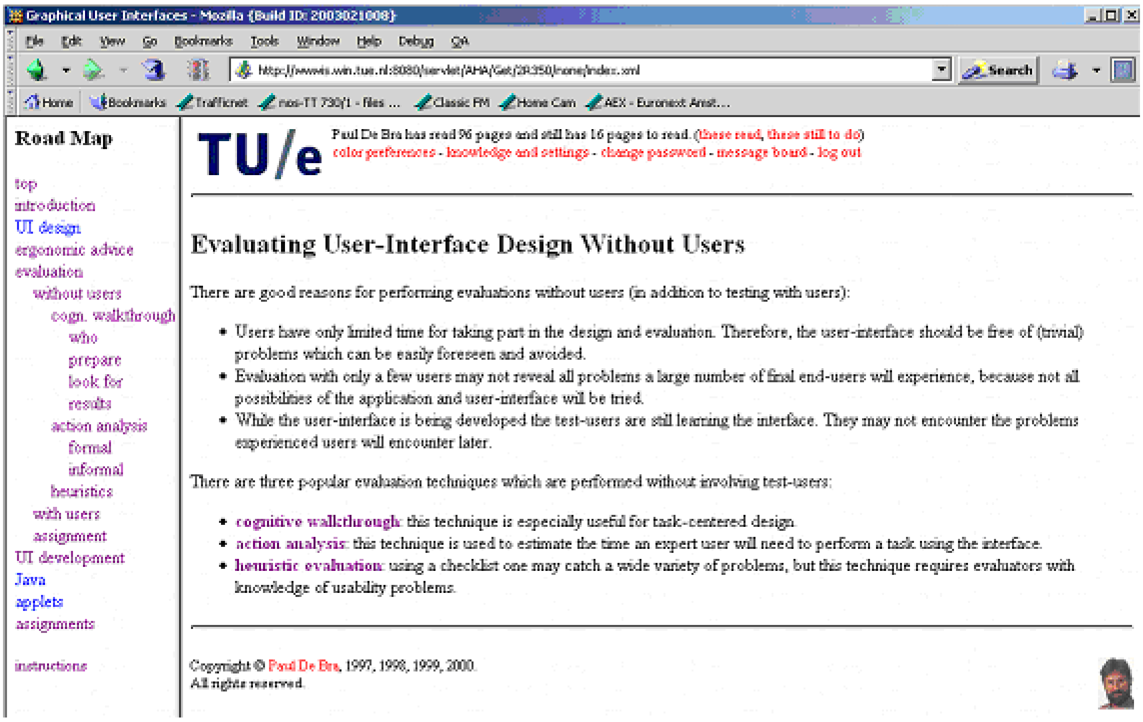


Fig. 2. AHA! multi-frame interface.

4. The View-Based Layout Model

The View-Based Layout Model is the new way in AHA! to present concepts (pages) to the user that was developed to address the lack of user interface possibilities in the earlier versions of the architecture. The *Layout model* combines the strong points of InterBook's rich user interface with the flexibility and customization style that are typical for the AHA!-architecture. This model allows every AH system developer to adapt the user interface to the course nature (without the need for the above mentioned Javascript hack).

To provide a high level of flexibility, the Layout Model was designed as a three-level interface model that is based on the concepts *views*,*viewgroups* and *layouts*. In brief, *views* are considered as atomic interface elements. Views can be grouped in *viewgroups*. One or more viewgroups form a *layout*. These concepts are presented in more detail below.

4.1 Views

Views are pieces of information about the course domain. They usually represent some relevant information about the active concept (the concept the user is viewing at the moment). A view can also represent some static information about the course domain. Views are used as pre-fabricated building blocks to construct the user interface for some specific course. Internally views are simply Java classes that generate HTML pages using underlying AHA! data structures. To present a concept to the user the system uses a set of predefined views. These predefined views can be further customized by the author of the course to develop an interface that meets the needs of the course. The author defines and customizes a view using an XML-based description language like the following:

```
<view name="v5" type="ToolboxView" title="Toolbox"
  background="IBookbluesq.bmp" params="leftspace=70">
    <secwnds>
        <secwnd link="TOC" viewgroup="TOC" img="ContentBtn.bmp"/>
        <secwnd link="Glossary" viewgroup="Glossary"
           img="GlossaryBtn.bmp"/>
    </secwnds>
</view>
```

At the moment we have already implemented a number of relatively simple basic views. The configuration of these views consists of setting the background picture, the title or changing the page margin to make the view more readable. It is possible however to implement much more complex views that will offer much more tuning possibilities to the author. We are considering parameters that will influence the content of the view page and not only the shape of it.

A view usually displays some information about the active concept including links to other relevant concepts. However a view can also contain links to other views which will offer more information to the user about the active concept. Following one of these links will result in displaying a new set of views. Views that are used directly to represent different aspects of a concept are called *primary views*. Views that present some supplementary information are called *secondary views*. These views are not visible until they are triggered by a link in one of the primary views. The author of the course will usually choose the most important views as primary views and less important views as secondary views. The connection between primary and secondary views can be specified by the author of the course in the XML view structures presented above. In the presented example *ToolboxView* can trigger two secondary viewgroups: *Table of Content* and *Glossary.*

4.2 Viewgroups and Layouts

As already said views are the building blocks for constructing concept representation. Views can be grouped in *viewgroups*. In HTML terms a *viewgroup* corresponds to an independent window and a view corresponds to a page that can be shown in a separate window or in a window frame. A set of *viewgroups* forms a *concept layout,* which is used to present a concept. Practically, it means that different aspects of a concept can be presented in several synchronized windows.

We assume that the system may have more than one *type* of concepts (pages). For example, InterBook has a *textbook page* and a *glossary concept*

that are both concepts in terms of the AHA! architecture. We also assume that an author of an adaptive course may want different types of concepts to be presented differently (this is what happens in InterBook). To support this possibility, our Layout model allows an author to define several layouts. Each concept type has to be associated with one of the layouts. Presenting concepts of the same type always in the same way (using the same layout) contributes to the user confidence in the system and avoids confusion. Links to the concepts of the same type are also annotated in the same way for obvious reasons.

The following XML structure is an example of a layout definition for two layouts that we use to simulate an InterBook style user interface:

```
<layoutlist>
    <layout name="page_c_layout" trigger="MAIN">
        <viewgroup name="MAIN" wndOpt="width=800,height=600">
                <compound framestruct="rows=20%,*" border="0">
                        <compound framestruct="cols=*,130" border="0">
                                <viewref name="v1" />
                                <viewref name="v5" />
                        </compound>
                        <compound framestruct="cols=*,130" >
                                <viewref name="v3" />
                                <viewref name="v2" />
                        </compound>
                </compound>
        </viewgroup>
        <viewgroup name="TOC"
wndOpt="resizable=1,toolbar=1,width=300,height=400">
                <viewref name="v1"/>
        </viewgroup>
        <viewgroup name="Glossary" secondary="true"
wndOpt="width=600,height=500">
                <viewref name="v4"/>
        </viewgroup>
    </layout>
    <layout name="abst_c_layout" trigger="Glossary" >
        <viewgroup name="Glossary" wndOpt="toolbar=1,width=600,height=500">
                <viewref name="v4"/>
        </viewgroup>
    </layout>
</layoutlist>
```

We have defined two layouts each associated with one of the two concept types that we use in AHA! at the moment: page concepts and abstract concepts. As can be seen in the example above each layout consists of a set of viewgroups which contain pointers to predefined views. Viewgroups use c*ompound* elements to define the place of each of the views within the window. For each viewgroup the author of the course can also define window options for the window in which the viewgroup is placed. The layout structure of layout '*page_c_layout*' above corresponds to the screen presented in Figure 3.

This layout consists of four primary views grouped into one viewgroup, which is shown in the figure, and two secondary views (Glossary and Table of Content) which can be triggered by the buttons in the ToolboxView (upper right corner).

Changing the XML configuration structures will change the layout associated with a certain concept type. The following example of an XML configuration structure uses the same views for the same concept type but grouped in a different way:

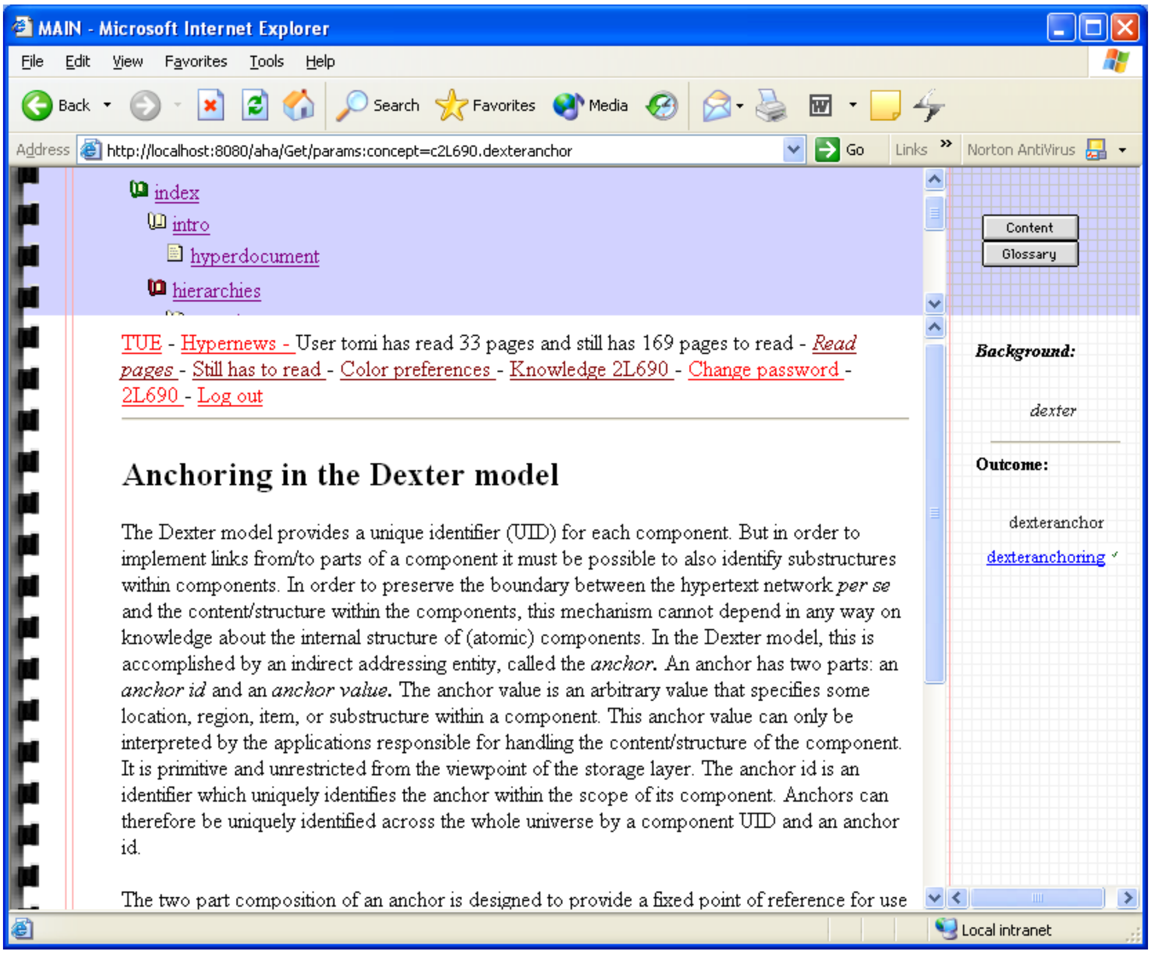


Fig. 3: InterBook style concept layout for ‘page’ concepts

```
<layout name="page_c_layout" trigger="MAIN">
        <viewgroup name="MAIN"
        wndOpt="status=1,menubar=1,resizable=1,toolbar=1,width=800,height=6
00">
                <compound framestruct="cols=200,*" >
                        <compound framestruct="rows=*,85" >
                                <viewref name="v1" />
                                <viewref name="v5" />
                        </compound>
                        <viewref name="v3" />
                </compound>
        </viewgroup>
        <viewgroup name="Conceptbar"
        wndOpt="status=1,menubar=1,resizable=1,toolbar=1,width=300,height=4
00">
                <viewref name="v2"/>
        </viewgroup>
        <viewgroup name="Glossary" secondary="true"
        wndOpt="status=1,menubar=1,resizable=1,toolbar=1,width=600,height=5
00">
                <viewref name="v4"/>
        </viewgroup>
</layout>
```

The corresponding screen layout for the XML configuration structure above is shown in figure 4.

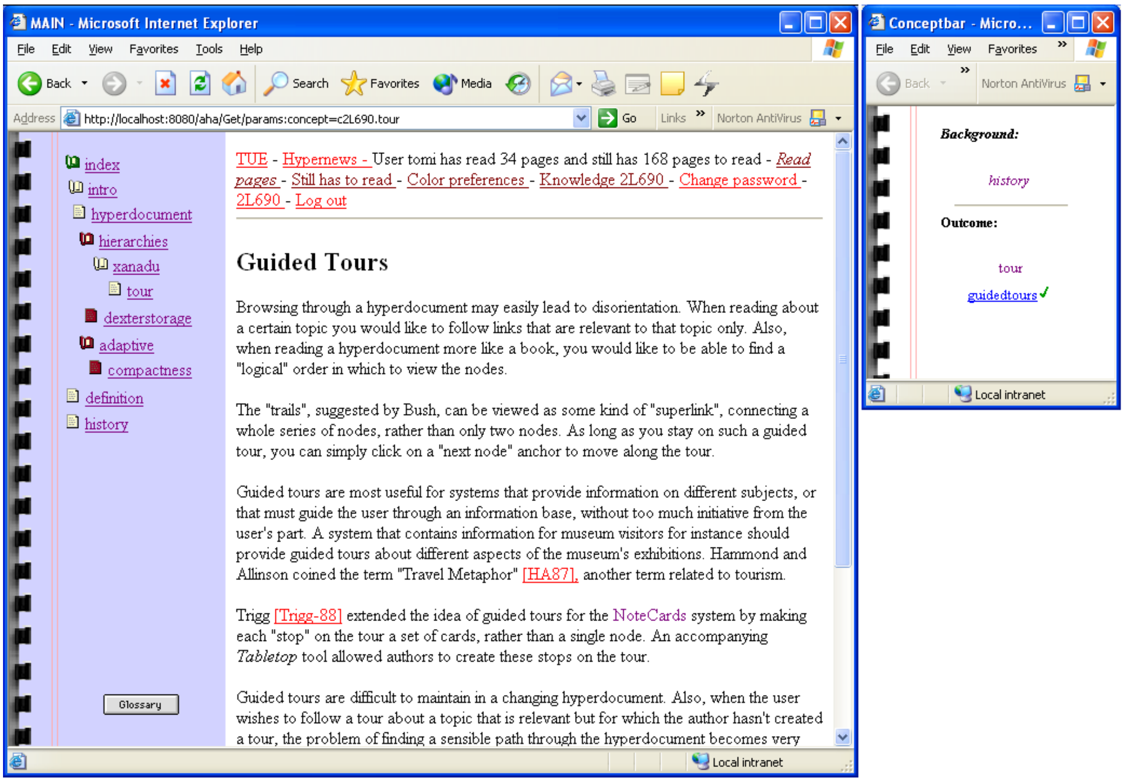


Fig. 4. second version of 'page' concepts layout

In this version of the layout associated with page concepts there are two primary viewgroups (MAIN and Conceptbar) and one secondary viewgroup (Glossary). Viewgroup MAIN consists of three views (MainView, Table of Content and Toolbox) and the Conceptbar viewgroup contains one view (ConceptbarView). Button 'Glossary' in the Toolbox view triggers the display of the secondary viewgroup *Glossary*.

5. Conclusions and Future work

The new AHA! layout model offers versatile user interface possibilities and brings AHA! one step closer to its main goal of being a generic Adaptive Hypermedia environment for all kinds of Adaptive Hypermedia applications. View based concept presentation is extremely flexible and gives a course author the power to adapt the user interface to the needs of the course. Internally views are Java objects with one task: extracting data from AHA! data structures and generating HTML pages from these data. In the future we are planning to extend the user interface adaptation possibilities by introducing the total data-presentation separation. We are thinking of giving the author the opportunity of implementing his/hers own views, in addition of using a set of predefined views. If the internal static AHA! data structures would be saved as XML files the author could use any standard XSLT editor to implement views as XSLT files which could extract data from XML formatted data structures. This model would give the author the possibility to represent the data in any desirable way without being dependent on already implemented views.

References

1. Brusilovsky, P. (1996) Methods and techniques of adaptive hypermedia. In P. Brusilovsky and J. Vassileva (eds.), User Modeling and User-Adapted Interaction 6 (2-3), Special Issue on Adaptive Hypertext and Hypermedia, 87-129.
2. Brusilovsky, P. (2001) Adaptive hypermedia. User Modeling and User Adapted Interaction 11 (1/2), 87-110, also available at http://www.wkap.nl/oasis.htm/270983.
3. Brusilovsky, P., Eklund, J., and Schwarz, E. (1998) Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems (Proceedings of Seventh International World Wide Web Conference, 14-18 April 1998) 30 (1-7), 291-300.
4. De Bra, P., Aerts, A., Smits, D., and Stash, N. (2002) AHA! Version 2.0: More Adaptation Flexibility for Authors. In: M. Driscoll and T. C. Reeves (eds.) Proceedings of World Conference on E-Learning, E-Learn 2002, Montreal, Canada, October 15-19, 2002, AACE, pp. 240-246.
5. De Bra, P., Brusilovsky, P., and Houben, G.-J. (1999a) Adaptive Hypermedia: From Systems to Framework. ACM Computing Surveys 31 (4es), also available at http://www.cs.brown.edu/memex/ACM_HypertextTestbed/papers/25.html.
6. De Bra, P. and Calvi, L. (1998) AHA! An open Adaptive Hypermedia Architecture. In P. Brusilovsky and M. Milosavljevic (eds.), The New Review of Hypermedia and Multimedia 4, Special Issue on Adaptivity and user modeling in hypermedia systems, 115-139.
7. De Bra, P., Houben, G. J., and Wu, H. (1999b) AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In: Proceedings of 10th ACM Conference on Hypertext and hypermedia (Hypertext'99), Darmstadt, Germany, February 21 - 25, 1999, ACM Press, pp. 147-156.
8. Grigoriadou, M., Papanikolaou, K., Kornilakis, H., and Magoulas, G. (2001) INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment. In: P. D. Bra, P. Brusilovsky and A. Kobsa (eds.) Proceedings of Third workshop on Adaptive Hypertext and Hypermedia, Sonthofen, Germany, July 14, 2001, Technical University Eindhoven, pp. 13-24.
9. Henze, N. and Nejdl, W. (2001) Adaptation in open corpus hypermedia. In P. Brusilovsky and C. Peylo (eds.), International Journal of Artificial Intelligence in Education 12 (4), Special Issue on Special Issue on Adaptive and Intelligent Web-based Educational Systems, 325-350, also available at http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/henze.html.
10. Melis, E., Andrès, E., Büdenbender, J., Frishauf, A., Goguadse, G., Libbrecht, P., Pollet, M., and Ullrich, C. (2001) ActiveMath: A web-based learning environment. In P. Brusilovsky and C. Peylo (eds.), International Journal of Artificial Intelligence in Education 12 (4), Special Issue on Special Issue on Adaptive and Intelligent Web-based Educational Systems, 385-407.
11. Weber, G. and Brusilovsky, P. (2001) ELM-ART: An adaptive versatile system for Web-based instruction. In P. Brusilovsky and C. Peylo (eds.), International Journal of Artificial Intelligence in Education 12 (4), Special Issue on Adaptive and Intelligent Web-based Educational Systems, 351-384, also available at http://cbl.leeds.ac.uk/ijaied/abstracts/Vol_12/weber.html.
12. Wu, H., De Kort, E., and De Bra, P. (2001) Design Issues for General Purpose Adaptive Hypermedia Systems. In: Proceedings of Twelfth ACM Conference on Hypertext and Hypermedia (Hypertext 2001), Aarhus, Denmark, August 14-18, 2001, ACM Press, pp. 141-150.

A Presentation Architecture for Individualized Content

Alberto Gonzáles Palomo, Paul Libbrecht, Carsten Ullrich
University of Saarland and German Research Center for Artificial Intelligence
Saarbrücken, Germany
{alberto,paul,cullrich}@activemath.org

Abstract

A modern approach for generating individualized web-sites is to compose a page out of individual elements, for instance XML-fragments, which is eventually transformed to . If the generated pages differ for each user, then the required transformation processes put a heavy load on the server, hence slowing down response times significantly.

The learning environment ACTIVEMATH uses this composing approach to generate learning courses that suit best the needs and goals of the individual learner. For instance, depending on their current knowledge, different users that learn the same content get presented courses that differ in length and in amount and difficulty of exercises and examples. Currently, the learning materials are transformed in one step from XML to (or other output formats).

The learning materials are encoded in a language called OMDoc which encodes semantically the fragments as well as the mathematical formulae. This article hence provides an approach to the answer ``How can Semantic Web technology be used to improve adaptation and information retrieval?". Moreover the ability to generate multiple output formats from the same content source is a central requirement of the architecture. It provides a first step towards device adaptation providing, currently, an on-screen version and a printable PDF version.

This article describes the architecture we developed to solve the performance problems that arouse out of the page generation process. In this architecture, the generation process is divided into several layers, with each layer adding/transforming well-specified data. Among other advantages, this approach allows caching of individual transformed fragments. We hope that in this way the performance problems can be reduced.

1. Motivation

A current trend in the WWW is the generation of individualized web pages. Individualization as meant as in this article covers the complete range from inserting small pieces of text (such as the user's name) to composing the complete text of a page depending on properties of the user. An area where this kind of individualization certainly makes sense is education. Although most of the (commercially and freely) available learning materials are of a static nature (pages, sometimes extended by applets, PDF , PPT, etc.) learning materials individually tailored to the student's needs definitely support their learning process in a favorable way.

One technically advanced e-learning system is ACTIVEMATH [7], a web-based learning environment that provides a throughout individualization. Learning materials, such as courses, are available in a common ``static manner", but are also constructed dynamically according to the student's learning goals and his current knowledge. The pedagogical advantages of the dynamic individualized generation are numerous: The learner has at his hands a course specifically tailored to his needs that contains only the necessary content at the adequate knowledge level. However, when we were using ACTIVEMATH for the first time in an university course, some drawbacks of technical nature arouse. Several complaints considered the sometimes slow response of the system. In the first, ad-hoc implementation, every time the student visited a page, the system (re-)builts this page from scratch. These fetching and transformation processes of course put a significant load on the server.

To overcome this and other problems detailed below, we analyzed and structured the presentation process, and, building on this analysis, developed a general architecture for the generation of individualized web pages. In short, we divided the presentation process in several separated stages, where each stage adds distinct individual information. Thereby, caching is possible in several places.

We think this architecture is of general interest. Although developed in an educational setting, the architecture we propose is general enough to be applied in all settings where throughout individualization is needed.

We will start with an overview on the generation of individualized web pages, how it is done in ACTIVEMATH, and the advantages it offers. Then, after talking about the problems that arouse out of our first, ``naive'' presentation engine, we will present the new, layered presentation architecture. The article concludes with a comparison of our architecture to existing frameworks.

2. Individualized Web Pages

From the very beginning of the web, first approaches of individualization were realized by CGI scripts which generated the complete text of the delivered page. They were followed by *script-in-page* approaches, such as Java Server Pages or Active Server Pages. There, the -code of a page contains special tags that are replaced by individual content. The advent of structured content provided by XML encodings opened the door to data processors: These engines generate browser-viewable content from a content whose structure is well defined and well known, the eXtensible Stylesheet Language Transformation (XSLT) process is a good example of such an approach: using an XML-encoded source and a few parameters, it produces the viewable content.

A good example of state-of-the-art individualized generation of web pages based on an underlying XML-representation is ACTIVEMATH. In the following section we will provide some details on ACTIVEMATH with respect to this feature.

3. ActiveMath, a Web-Based Learning Environment

ACTIVEMATH is a web-based, user adaptive environment for mathematics education. It employs content encoded in a semantic XML-representation and integrates several mathematical systems to support exploratory learning.

ACTIVEMATH generates individual courses for each learner. An example is shown in Figure 1. In the example, both users want to learn about the mathematical concept *morphism of groups*. Eva, on the left hand site, already knows a bit about the mathematics necessary to understand morphisms. Therefore, the course generator of ACTIVEMATH has composed a shorter course for Eva (see her table of content) than for Anton, on the right hand site, who misses some prerequisite knowledge. In addition, the content within a page differs. Eva know the concept *monoids* quite well, and the course generator generated a page that serves to reactivate this knowledge. Anton however, for whom this concept is unknown, gets presented a page that among others provides a detailed example on monoids.

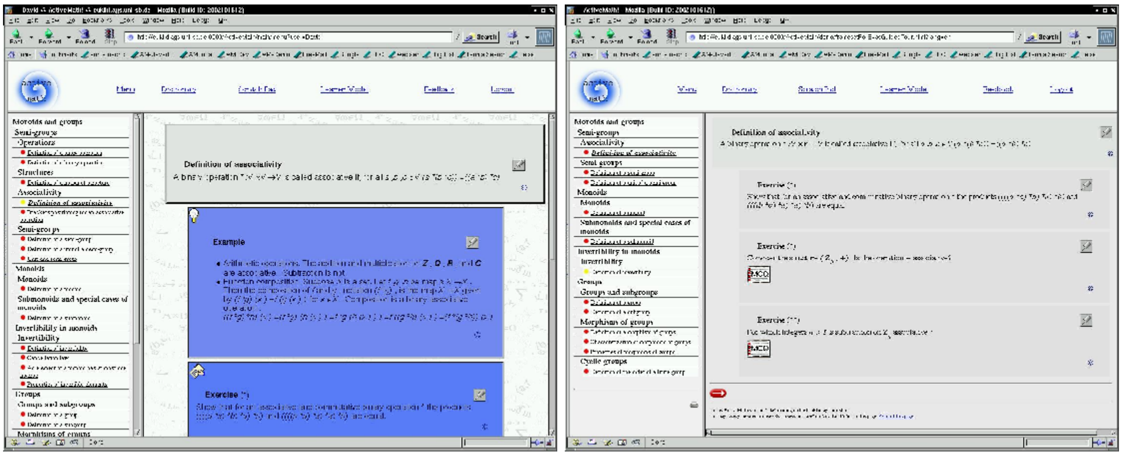


Fig. 1. Two courses for the same learning goal, but for different learners.

More generally, ActiveMath provides individualization with respect to a complete course, the elements on a page within a course, and the text within the elements. Hence, ActiveMath requires a powerful underlying knowledge representation. The following section provides some details about it (for more information, see [9]).

3.1 ActiveMath's Knowledge Representation

ActiveMath knowledge representation is a semantic XML encoding. As ActiveMath is currently targeted at teaching math, it uses `OMDoc` . `OMDoc` , documented in [6], provides means to represent the structure of a mathematical document. It allows the encoding of different categories of items at paragraph level: mathematical concepts such as definitions and theorems, and further items such as examples, exercises, and elaborative texts.

`OMDoc` allows the annotation of items and documents with metadata. The metadata covers pedagogical aspects of an item such as `difficulty` and relations between the items such as `depends-on` or `similar-to`.

The knowledge representation contains some but not all metadata of the Learning Objects Metadata standard of IEEE [5] but includes additional relationships and characterizations that are missing in this standard.

Furthermore, `OMDoc` allows to represent the semantic of a mathematical formula. Whereas in an usual text-document a+b simply represents a string and + the ASCII character 43, the `OMDoc` encoding clearly specifies which mathematical symbol is meant. For instance, besides the addition of numbers, + often denotes a group operation. To know the semantic of a mathematical object is especially important if the content is used in or interchanged between external systems such as computer algebra systems (which ActiveMath uses for interactive exercises). Another example are copy and paste operations where the user marks a formula and wants to copy the corresponding mathematical object, not its textual representation.

3.2 Current Generation and Presentation Process

Mathematical learning material, i.e., the course and its pages, is dynamically generated on the user's demand by selecting and composing the OMDoc items. The course generation process happens as follows:

Starting from the goal concepts chosen by the user or a teacher, all concepts they depend upon are retrieved recursively from a knowledge base. The result is a collection of all concepts that need to be known by the learner in order to be able to understand the goal concepts together with additional material such as examples and exercises. Then pedagogical rules are applied which take information about the user (stored in an user model) into account. These rules determine, e.g., the types of items to appear on a page, the appropriate level of difficulty and the order in which the material is presented. The result depends on the available content, the current state of the user model, and on the chosen *learning scenario*: Different documents are generated for different scenarios such as ``overview'' or ``exam preparation''. Eventually, the learning materials are ordered and put into a hierarchy.

For the presentation, the generated OMDoc pages are transformed via XSLT to , PDF , or SVG. A typical delivery of an page is triggered by an HTTP request containing the user name, page number and an identifier of the current course. The ids of the to be presented OMDoc items are looked up using the page number and the table of content of the course. Then they are retrieved from the knowledge base. The resulting XML -document is pre-processed (adding of server specific information, see Section 4.2) and is then transformed via XSLT which directly produces the page annotated with additional individual information (i.e., allowing the user to resume a course at the position he left). PDF -documents are produced in a similar way; additionally the PDFLATEX[1] program is called.

3.3 Advantages of Dynamic and Individualized Generation of Web Pages

The dynamic and individualized generation of web pages realized in ActiveMath offers several advantages:

3.3.1 Re-use of Content

Developing content, especially learning materials, is time and money consuming. By re-using and improving already created content, (hopefully) the costs are reduced and quality enhanced. ActiveMath offers a very fine-grained re-use: Instead of complete pages (or even courses), single paragraphs form the basis of the re-use. This level is not only suited for learning, for instance, parts of a *news story* can be inserted or skipped depending on the background knowledge of the reader. In this way, different parts of the content can be re-used for a variety of purposes.

3.3.2 Multiple Output Formats

Generating the content from a semantic, presentation-independent knowledge representation makes it possible to render to multiple output formats depending on the current needs of the user. Not everything that looks nice at the screen still looks nice when printed out. Guidelines for ergonomic layout differ depending on whether the content is targeted for print or online viewing. Therefore, a system should provide the possibility for elements to have different renderings depending on the output format. But this is only possible if the concrete rendering is not hard-coded in the content. For instance an `emphasize` is not equivalent to a purely presentational `bold`. The style-sheets decide which presentation is chosen for emphasize.

3.3.3 Personalization of Content

As already mentioned, ACTIVEMATH offers truly individualized content generation. The advantages of such individual generation are numerous: The user needs less time to find and to learn about the content he is interested in, he is not de-motivated by content too difficult for him to understand, neither bored by facts he already knows about.

3.3.4 Combining Learning Materials

The author, who writes the content, and the editor (in an educational setting the teacher), who selects which content to present, can but do not have to be the same person. Therefore, combining learning materials can be far more laborious than on first sight, especially in mathematics and other formal sciences: Different authors tend to use different notations. A well known example is the logical concept of *implication* which is presented differently in almost every book about logic, e.g., as -> or ? .

The concept of *composition* is a more challenging example that requires not just replacing a symbol: To denote the composition of f followed by g, in general, a German author would write $g \circ f$, whereas an English author would write fg.

Combining such materials when they are written in traditional presentation-oriented languages requires tedious manual rewriting. However, if a presentation-independent knowledge representation is used, then the is presentation specified separately from the content, for instance in a XSLT-rule. To determine their preferred presentation, authors that combine materials from different sources simply specify the presentation rules for the necessary concepts; they do not have to worry changing the content itself.

We esteemed the background information given in this section necessary specify the setting of our system and we omitted several technical details concerning semantic encoding. For a more concise overview on knowledge representation and management, please see [9].

3.4 Problems Regarding Presentation

Our first ``naive'' presentation engine that generates the desired output format directly annotated with individual data in one step gave rise to several problems:

3.4.1 Server Load

At every page request the presentation process was repeated completely starting from fetching the content to transforming it to the output format. Of course, this puts a heavy load on the server. Especially the transformation process requires a lot of resources. Caching the fully generated pages does not alleviate the problem neither, as although the students following a course would share some of the items on a page, the other items such as exercises and examples will certainly differ.

3.4.2 Performance

As a direct consequence of the high server load the response time of the system decreased, in particular if a large amount of users accessed it simultaneously. This is especially unpleasant as for web applications slow response times are critical. For instance, our students reported that the long delays were a major source of de-motivation.

3.4.3 Fixed Presentations

If several presentations for the same concept are possible, some instance has to decide which presentation to use. For students in a course this decision is normally taken by the teacher as he wants to assure that all his students work with the same notation. An independent learner however, can very well make his own choice. Our former presentation architecture used one stylesheet for all users of one ActiveMath installation, thereby disallowing a flexible choice of the presentation.

3.4.4 Missing Abstractions

The recent development of the PDF presentation process raised several missing abstractions in our current presentation architecture. As it was primarily targeted for output, the assembly process of the OMDoc content was optimized with respect to . For instance, multiple choice exercises were inserted in a special way directed at the presentation in a pop-up window (which is of course not suited for a print-version of a page).

4. Layered Presentation Architecture

4.1 Processed Data

We performed an analysis of the data that is processed/added during the presentation process to determine to which extent the process could be optimized with respect to the above problems. The analysis yielded the following kinds:

Content.
Obviously, the presented content forms the major and most important type of data. Content is mostly static in the sense that the containing text does not change (but see Section 4.4.2). However, the overall content is of course dynamic, as different users get different content.

Server-Specific Information.
This subsumes data added by the current server, such as the version of the used ActiveMath, the address of the server, links, or resource descriptions for interactive exercises.

Presentation Information.
This data specifies how specific symbols are to be rendered (represented in XSLT-stylesheets).

Personal Information.
While the individual learning materials that are presented to an user are covered by the above content type, personal information is additional individual data added on top of the content. A good example is the indication of the state of knowledge of the user with respect to an element. For instance, if Anton has only very limited knowledge of an element, it can be annotated in a special way, e.g., underlined with red color. Another example for -presentation is the preferred CSS-stylesheet of the user.

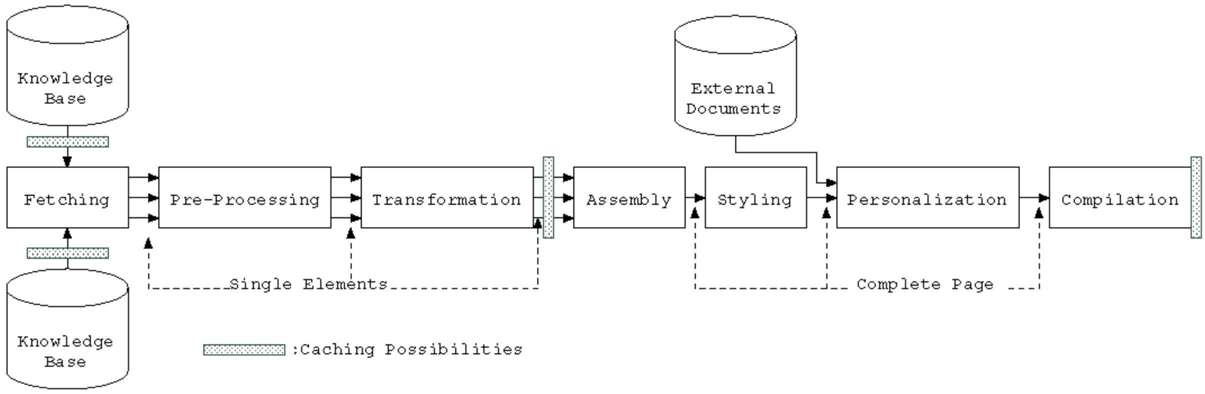


Fig. 2. A graphical representation of the layered presentation architecture

4.2 Splitting the Presentation Process

Following the analysis we split the presentation process in several layers, with each layer adding/transforming a specific kind of data. Figure 2 provides a graphical representation of the new architecture. The layers are the following:

Fetching.
: Collects requested content from the knowledge base. The output of this stage are XML-fragments.

Pre-Processing.
: Inserts server-specific information into the XML content. For instance, if the content is contained in several distributed knowledge bases, this step changes the ids of the elements to avoid duplicated ids.

Transformation.
: Performs a first transformation to the desired output format by the application of an XSLT-stylesheet to the document. The output of this stage are or LATEX-fragments.

Assembly.
: Joins the fragments together to form the requested pages. This layer uses the XSLT presentation information.

Personalization.
: Uses personal information to add individually different ``beautifications'' to the document, such as knowledge indicators, for the generation the user name in the HTTP-links and the used CSS-stylesheet, for LATEX the used macro-packages.

Compilation.
: Applies further processing to convert the content presentation into a format that can be displayed by the client. This step is not needed for as can directly be presented by web browsers. However, PDF output requires the compilation of the generated LATEX-sources.

4.2.1 An Example

This section provides a small example page of the page generation and shows how the content (simplified `OMDoc`-element) is transformed in each stage. Let's assume that the user requests a page that contains only the definition `def1`. In a first step, this content is fetched from the knowledge base:

```
<definition id="def1">
 Definition 1 with a reference to
 <ref xref="def2">definition 2</ref>
</definition>
```

The pre-processing step replaces the ids of the elements. Here, it adds the name of the knowledge base as a prefix to the id, yielding `mbX_def1`.

Then the XML-fragment is transformed to using XSLT:

```
<div class="definition" id="mbX_def1">
 Definition 1 with a reference to
 <a onClick="openInDictionary('mbX_def2')">
  definition 2
 </a>
</div>
```

In the assemble stage, the elements (in this case only one element) are assembled to form a complete page:

```
<html>
 <head/>
 <body>
  <div class="definition" id="mbX_def1">
   Definition 1 with a reference to
   <a onClick="openInDictionary('mbX_def2')">
    definition 2
   </a>
  </div>
 </body>
</html>
```

Eventually personal information is added and the page is send to the user's browser:

```
<html>
 <head>
  <link rel="stylesheet"
        type="text/css" href="colored.css"/>
 </head>
 <body onLoad="initializePageUser('Anton')">
  <div class="definition" id="mbX_def1">
   <img src="lowknowledge.png"/>
   Definition 1 with a reference to
   <a onClick="openInDictionary('mbX_def2')">
    definition 2
   </a>
  </div>
 </body>
</html>
```

4.3 Caching Possibilities

Talking about optimizations only makes sense if the assumptions under which the optimizations take effect are made explicit. ACTIVEMATH was designed with two different use cases in mind:

In the group setting a large amount of users access more or less the same content. Although the specific content of the pages can differ, the materials that need to be retrieved from the knowledge base can be specified approximately. In an educational system this would correspond to learners following a lecture, where the content will be the covered domain; in an online news server this content could be the headline articles.

The second use case is the independent self-guided user. No assumptions can be taken about what content he is interested in. For this group, caching the content of one user would not help a second one, as most probably they are interested in different topics.

We designed ActiveMath primarily for being used by a number of students learning about the same content. Hence, the caching we propose is mostly directed at the group case. The second case needs further investigation, although the proposed caches will not have any negative effect for the self-guided use case.

The diagram in Figure 2 indicates three points at which caching can take place:

At Fetching Level.
: ActiveMath can retrieve its content from different knowledge bases that can be distributed anywhere in the web. As response times will vary heavily, caching once retrieved content at the ActiveMath server can yield better performance, especially for slow connections.

After Transformation.
: Applying XSLT-transformations requires a lot of resources. Therefore we decided to cache the individual elements after their transformation. A drawback is the additional memory consumption: For every output format, a proper cache is required. In addition, personalization data has to be added after the transformation. For , this can be achieved using another XSLT-transformation and/or JavaScript, for LaTeX, macros are used.
: Problems can arise if elements are to be presented differently depending on the other elements on the page, as at the time of transformation, the transforming component does not know which elements will occur on the same page. For instance, a reference to another element can be transformed to a hyperlink that opens a new window with the referenced element if it is not present on the current page. Otherwise, if both elements occur on the same page, the focus of the page is changed to the anchor of the referenced element. However, we were able to overcome this and similar problems we encountered by the use of Javascript.

After Compilation.
: Caching a completely generated and individualized page speeds up access of often visited pages and going back/forward within a site. This cache is usually provided by the browser cache and can be controlled to some extent using the `http` meta tag `expires` that takes as an argument the date when to refetch the data from the server. However, often this decision can only be taken by the server itself. More often than not, whether a specific page changes does not depend on the time passed since the last visit but on the actions taken by the user. Adding such a cache on the server side definitely adds unrealistic memory requirements if the amount of users is unlimited. In cases that restrict either the amount of users or the number of groups that access different content, caching complete pages can be an option.

4.4 Additional Considerations

4.4.1 External Documents

Another advantage of the layered architecture is the personalization of third-party content not available in the underlying XML-representation.

Most authors don't feel comfortable changing from their preferred content format to a new one, as the change requires new tools, learning how to use the tools and the new format, and, in the case of an non presentation-oriented format, imagined loss of control over layout issues. Furthermore, for old content not to be lost, it requires extensive manual work for conversion.

In principle, performing the personalization after the assembly allows to personalize content not generated by the presentation process itself. These external documents have of course to follow certain conventions, it is certainly not possible to personalize arbitrary documents. For instance, offers manifold ways to obtain a paragraph, e.g. adding line breaks before and after a text block, or including the text within a `div` or `p` tag.

We made the experience that as a first compromise towards completely switching to `OMDoc`, authors sometimes prefer adapting their old content manually with respect to some conventions, e.g., representing a paragraph with the `div` tag, and adding a uniform `id` attribute to them. In this way, some personalization can be performed on this content, e.g., adding knowledge indicators.

Yet one has to keep in mind that this approach is only a compromise. It neither offers the full functionality of personalization nor will it satisfy the author in the long run, especially with respect to re-usability.

4.4.2 Randomized and Generated Content

Some content can not be cached at all because its text is generated on the fly. We distinguish between randomized and generated content:

A good example for randomized content are multiple choice exercises. Every time ActiveMath presents such an exercise, the order in which the possible choices are presented is randomized. As simple as it is, it hampers students to simply copy the answers from their neighbors.

Generated content is content that is generated from an abstract representation. An example are exercises involving statistics. Statistic can be applied to a number of areas, ranging from the probability whether a person falls ill to the probability that some products sells better than others. But the underlying mathematics remain the same. Therefore, an abstract representation can specify the basics of the exercise, and the concrete instantiations are generated with respect to the field of the learner.

These kinds of dynamic content should of course not be cached. Therefore, the assembly process has to be able to know whether to take an item from the cache or whether to request it again from a content-generator (not shown in Figure 2). This can be achieved by adding a non-cache attribute on the elements or by giving the assembly a list of these elements.

4.4.3 Lowering the amount of requests

We also aim at avoiding the load of a document for the sole purpose of performing small, local updates, for instance, changing the colored bullets in the table of content that indicate the mastery value of the user for the topics of the page. These updates are performed via a JavaScript connection to the server. This approach can be generalized to a message-based communication, making browser components *agents* communicating to the components of the server, thereby offering much more flexibility and reducing the amount of information traveling between the brower and the server.

5. Related Work

A huge quantity of systems in education offer individualization of content, see, for instance, the recent proceedings of the International Conference on Intelligent Tutoring Systems [3] or the proceedings of the Conference on Adaptive Hypermedia [1]. Brusilovsky [2] provides a comprehensive overview on adaptive hypermedia techniques, however the techniques he mentions focus on manipulating pre-made pages, for example hiding/showing a paragraph of text, enabling/disabling hyperlinks, or changing the sequence in which the pages are presented to the user.

Recent systems that provide the possibility of assembling pages and courses from smaller *learning objects* depending on user properties are [4] or [8]. Sadly, they do not provide details regarding the technical aspects focused in this article.

A very powerful presentation architecture is the Cocoon Publishing Framework[2]. It offers stylesheet processing added with XML -creation and caching at all levels. The Cocoon framework is very flexible, but consequently relatively complicated. Furthermore, it is hard to debug because of its purely stream-based processing (streams being either byte-streams or, most frequently, XML-parse-tree events). In comparison, our presentation architecture based on an in-memory XML -representation provides more expressive accessors and manipulations for the special case of our OMDoc

encoding. This supports authors and developers to detect and resolve presentation errors.

6. Conclusion and Further Work

We proposed a layered presentation architecture that divides the page generation in several stages, where each stage adds distinct information. Thereby, more elaborate caching strategies are possible than in an one-step generation. We think our approach is especially helpful if more elaborate individualization takes place than simply inserting a user name. In scenarios where the content presented to the user is composed of parts that are retrieved from a knowledge base, depending on individual properties, the layered architecture can very well be applied and lead to noticeable performance increases.

The implementation of the layered presentation architecture is currently underway. We will then conduct an exhaustive analysis of the performance. In particular we are interested to what extent performances increases under realistic conditions, i.e., in a course, by transforming the single elements for themselves and only later composing them to form complete pages compared to composing the complete page and then transforming it.

Furthermore, we will investigate the effects of the different cache positions on speed increase. Preliminary experimental results show that, in the case, the delivery of a page whose items are all cached after transformation is at least 100 times faster as it only involves merging byte-streams. Nevertheless, exact data can only be gained under real world conditions.

The architecture is being implemented within the ActiveMath system. ActiveMath was developed with modularity and openness in mind so that its components can be easily reused. It is open source and available free of charge in a non-commercial setting[3].

References

1. P. D. Bra, P. Brusilovsky, and R. Conejo, editors. volume 2347 of *LNCS*. Springer-Verlag, 2002.
2. P. Brusilovsky.*Adaptive and intelligent technologies for web-based education. Künstliche Intelligenz*, 4:19-25, 1999.
3. S. Cerri, G. Gouarderes, and F. Paraguacu, editors. volume 2363 of *LNCS*. Springer-Verlag, 2002.
4. O. Conlan, V. Wade, C. Bruen, and M. Gargan. *Multi-model, metadata driven approach to adaptive hypermedia services for personalized elearning.* In P. D. Bra, P. Brusilovsky, and R. Conejo, editors, *Proceedings of the second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 2347 of *LNCS*, pages 100-111. Springer-Verlag, 2002.
5. IEEE Learning Technology Standards Committee. Learning objects metadata, 1999. Available from http://www.edna.edu.au/edna/aboutedna/metadata/analysis/LOM.htm, see also http://ltsc.ieee.org/.
6. M. Kohlhase.*OMDoc : Towards an OpenMath representation of mathematical documents.* Seki Report SR-00-02, Fachbereich Informatik, Universität des Saarlandes, 2000. See also http://www.mathweb.org/omdoc.
7. E. Melis, E. Andres, G. Goguadze, P. Libbrecht, M. Pollet, and C. Ullrich. *Activemath: System description.* In J. D. Moore, C. Redfield, and W. L. Johnson, editors, *Artificial Intelligence in Education*, pages 580-582, Amsterdam, 2001. IOS Press.
8. *Adaptive learning environment for teaching and learning in WINDS.* In P. D. Bra, P. Brusilovsky, and R. Conejo, editors, *Proceedings of the second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems*, volume 2347 of *LNCS*, pages 572-575. Springer-Verlag, 2002.
9. The ActiveMath Group. *Knowledge representation and management in ActiveMath.* To appear in the Proceedings of MKM'01 in Annals of Mathematics and Artificial Intelligence, 2003.

Footnotes

...pdflatex[1]
LaTeXis a typesetting program designed for high-quality composition of content. PDFLATEX is a variant that produces PDF.

... Framework[2]
http://xml.apache.org/cocoon/

... setting[3]
http://www.activemath.org

Challenges and Benefits of the Semantic Web for User Modelling

Peter Dolog and Wolfgang Nejdl
Learning Lab Lower Saxony
University of Hannover
Expo Plaza 1, 30539 Hannover, Germany
dolog@learninglab.de, nejdl@learninglab.de
http://www.learninglab.de/~dolog, http://www.kbs.uni-hannover.de/~nejdl

Abstract

The aim of this paper is to discuss how distributed learner modelling can benefit from semantic web technologies and which challenges have to be solved in this new environment. Heterogeneity of personalization techniques and their needs raise the question whether we can agree on one common data model for user profiles, which supports these techniques. In this paper we discuss an approach where a learner model can be distributed and can reflect features taken from several standards for a learner modelling. These features can be combined according to the requirements of specific personalization techniques, which can be provided as personalization services in a P2P learning network. RDF and RDFS as key tools of the semantic web allow us to handle such situations. We also sketch an architecture for such a network, where this approach can be realized.

1. Introduction

Internet as an open environment provides us with the opportunity to share and reuse resources already available. Heterogeneity of users and resources in the web stresses the importance of customized (personalized) delivery of the resources.

A wide range of personalization techniques has been introduced based on metadata about a user or learner. The first category of techniques are based mostly on adapting user interfaces, navigation and content selection and presentation according to the user's performance in a particular domain. The performance is often evaluated in a small closed domain, e.g. an electronic course at the open university. These techniques are usually called *Adaptive hypermedia techniques* [2].

Another type of techniques is based on interests, preferences, likes, dislikes, and goals a user has. This information is mostly stored at some kind of modelling server [7]. These are the so-called *filtering and recommendation techniques*. They recommend resources according to features extracted from a resource content or according to ratings of a user or learner of similar profile.

As it was already pointed for example in [11], such single user model, which stores everything in one model on one place has its disadvantages. A user model designed for experimentation with a specific technique usually reflects the specific requirements of that technique. This means that the schemas designed for different personalization techniques do not only differ syntactically (schema elements used to encode user information) but also semantically (i.e. that elements used for representing certain user feature can have either different syntax and different meaning or the same syntax and the different meaning). They can also differ in structuring the user's or learner's feature space.

There have been attempts to standardize a learner profile (see e.g. IEEE Personal and Private Information (PAPI) [5] or IMS Learner Information Package (LIP) [6]). These standards have been developed from different points of view. The PAPI standard reflects ideas from intelligent tutoring systems where the performance information is considered as the most important information about a learner. The PAPI standard also stresses the importance of inter-personal relationships discussed also in [11]. On the other hand the LIP standard is based on the classical notion of a CV and inter-personal relationships are not considered at all.

In this paper we report on our experience in developing the first version of a learner profile to support simple personalization techniques in the ELENA project. The aim of the project is to integrate heterogeneous learning resource and service providers and enable personalized access, use and delivery of the resources and services bound to them. The network is based on the Edutella P2P infrastructure [9]. This work is the extension of our previously published work on integrating adaptive hypermedia techniques into open RDF based environments [3].

The rest of the paper is structured as follows. First, we motivate our work by discussing some characteristics of open environments (section 2). Section 3 discusses available standards. We then remark on how users or learners can be modelled by utilizing RDF and RDFS in section 4. We then discuss our approach for developing a first version of a user profile for personalization services to support personalized search in the EU/IST ELENA project in section 5 and discuss its relationship to existing user profile standards as well as its embedding into a P2P architecture.

2. Some characteristics of open learning environments

There are several characteristics of open learning environments integrating heterogeneous resource providers, which distinguish open environments from most other currently studied systems. First of all, the resource can appear and disappear in ad-hoc manner. Peers, which provide such resources can appear and disappear randomly.

Resources are authored by different people with different goals, background, domain expertise and so on. Providers of the resources can maintain the resource in proprietary databases. They can already have some personalization techniques implemented for purposes of their context. The user or learner model usually reflects the context of the techniques as well. The resources are accessed and consumed by people which differ in a wide range of characteristics.

User or learner features can already be maintained in some of integrated systems. *Human resource management systems* as one of the modules of enterprise resource planning systems incorporate a module related to maintaining information about employees. This module usually maintains information about employee identification, skills, previous jobs, previous training, current role and position within company, transcripts and so on. *Task management systems* like project management systems, Sales Force Automation, or simply Outlook with its calendar contain usually information about tasks daily performed. *User modelling servers* maintain characteristics such as likes and dislikes, interests, bought products, and so on. *Electronic course providers* already have a model of learner performance. This basically refers to learner's attended courses with ratings of his performance. The performance record also refers to the formal certificates earned from the course providers and results of work or additional material used during the course. The providers can also maintain user interests, preferences and goals.

It is obvious that in such environments the agreement on common structures and scope of information about a user can be made only within specific communities or for specific personalization techniques.

3. Standards for Learner Modelling

The two most important examples for learner modelling standards are PAPI [5] and IMS LIPS [6]. Both standards deal with several categories for information about a learner. Figure 1 depicts the conceptual view of PAPI profile.

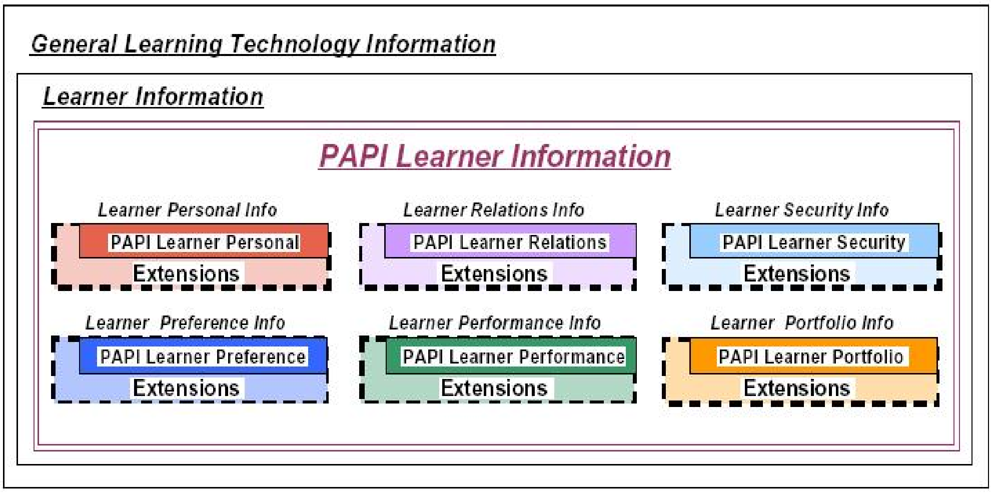


Fig. 1. The core categories in the PAPI profile [5].

PAPI distinguishes *personal*, *relations*, *security*,*preference*,*performance*, and *portfolio* information. The *personal* category contains information about names, contacts and addresses of a learner. *Relations* serve as a category for relationships of a specific learner to other persons (e.g. classmate, teacheris, teacherof, instructoris, instructorof, belongsto, belongswith). *Security* aims to provide slots for credentials and access rights. *Preference* indicates the types of devices and objects, which the learner is able to recognize. *Performance* is for storing information about measured performance of a learner through learning material (i.e. what does a learner knows). *Portfolio* is for accessing previous experience of a user. Each category can be extended.

Similarly the IMS LIP standard contains several categories for data about a user. The categories are depicted in the fig. 2.

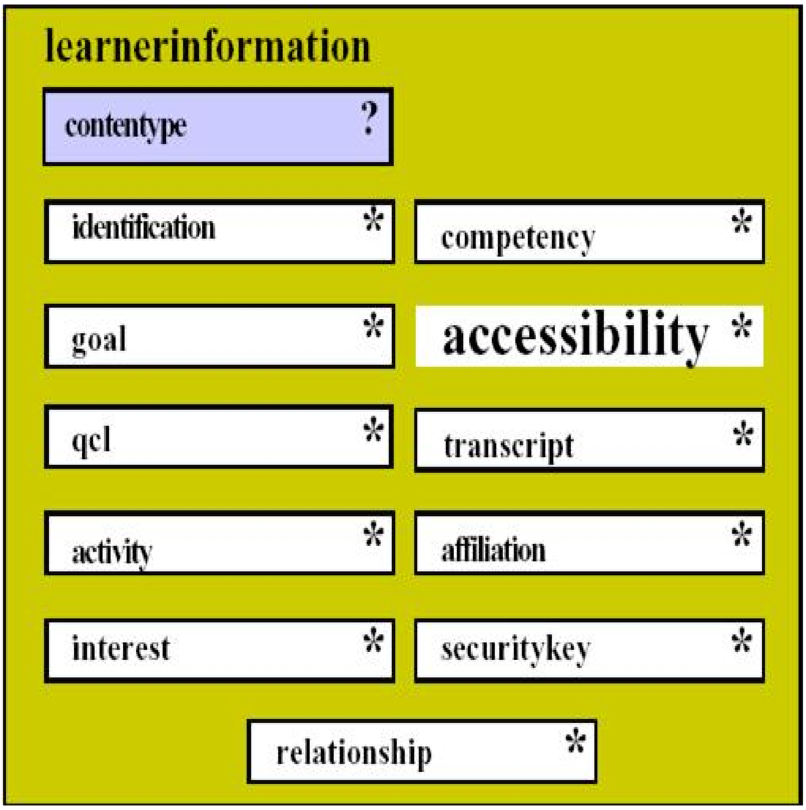


Fig. 2. The core categories for learner data in IMS LIP [6].

The *identification* category represents demographic and biographic data about a learner. The *goal* category represents learning, career and other objectives of a learner. The *QCL* category is used for identification of qualifications, certifications, and licenses from recognized authorities. The *activity* category can contain any learning related activity in any state of completion. The *interest* category can be any information describing hobbies and recreational activities. The *relationship* category aims for relationships between core data elements. The *competency* category serves as slot for skills, experience and knowledge acquired. The *accessibility* category aims for general accessibility to learner information by means of language capabilities, disabilities, eligibility, and learning preferences. The *transcript* category represents institutionally-based summary of academic achievements. The *affiliation* category represents information records about membership in professional organizations. The *security key* is for setting passwords and keys assigned to a learner.

4. Semantic web technologies

Semantic web technologies like the Resource Description Format (RDF) [8] or RDF schema (RDFS) [1] provide us with interesting possibilities. RDF models are used to describe learning resources. The RDF bindings of Learning Object Metadata (LOM) [10] can be used for these purposes. The RDF model can be used for learner description as well. RDF schemas serve to define vocabularies for medatada records in an RDF file. There is no restriction on the use of different schemas together in one RDF file or RDF model. The schema identification comes with attributes being used from that schema so backward dereferencing is again easily possible. For example the RDF model about a learner can use an attribute `performance_coding` from PAPI standard. Field from PAPI standard can be prescribed by PAPI schema. An Instance of such a record can then look as follows:

```
<rdf:Description rdf:ID="record_1">
...
    <papi:performance_coding
          xmlns:papi="http://learninglab.de/papi#">
          number
    </papi:performance_coding>
</rdf:description>
```

The abbreviated syntax can be also used where schemas are identified directly in RDF element as a root element of the RDF description. If we want to use different elements from different schemas the syntax will be as follows:

```
<rdf:Description rdf:ID="record_1">
...
    <papi:performance_coding
          xmlns:papi="http://learninglab.de/papi#">
          number
    </papi:performance_coding>
...
</rdf:description>
<rdf:Description rdf:ID="record_2">
...
    <ims:language_preference
          xmlns:ims="http://learninglab.de/ims#">
          english
    </ims:language_preference>
...
</rdf:description>
```

It says that we are using `ims` schema element `language_preference` for encoding learner language preferences. In this case the preference is English. This element was taken from the IMS schema.

The Edutella [9] P2P infrastructure allow us to connect peers which provides metadata about resources described in RDF. Edutella also provides us with a powerful Datalog based query language, RDF-QEL. The query can be formulated in RDF format as well, and it can reference several schemas. Furthermore, Edutella is able to process insert/update requests.

An example for a simple query (in Datalog syntax) over a learner model can be as follows:

```
query(student1, X, Z) :-
performance(student1, X),
performance_value(X, Z).
?- query(student1, X, Z, V)
```

The query tries to find performance values for a student. Variable X is bound to the performance record identifiers, variable Z to performance values. However, if we query the RDF model from our previous example we will not receive any result.

If we want to update the RDF description from our example with a performance value, an insert request should be used. The statement for inserting the performance value will contain `record_1` resource as a subject, `performance_value` as a predicate and the specific value for learner performance as object.

The resulting RDF description will then contain the performance value encoded similarly to the performance coding.

5. Learner profile development

We are working on personalization in the context of the EU/IST project Elena. The aim of this project is to demonstrate the feasibility of smart learning spaces. These spaces will be realized by setting up a network where heterogeneous services and resources are provided from different peers. The provision, search, booking, and delivery services should be personalized. The first goal is to support simple personalization techniques for search service. As we mentioned, personalization techniques depend on information about a user. The question for the development of such personal profile in the open heterogeneous environment was whether to use one specific standard, combine existing standards according to our needs, to develop our own profile schema, or some intermediate solution.

5.1 Methodology employed

To answer these questions, we employed a scenario based methodology for analyzing the requirements for our personal profiles. The methodology consists of the following steps:

- Collecting scenarios
- Analyzing scenarios
- Creating exemplar conceptual views of scenarios
- Abstracting and generalizing exemplar views - creating classes of user features
- Analyzing available standards
- Mapping classes from class based conceptual model to categories from supporting standards
- Refining scenarios

We could have also driven the development of a user profile by a specific subset of personalization techniques but we decided to have a broader scope taken from scenarios collected. This enabled us to analyze standards for learner profiles from different perspectives described by scenarios. Personalization techniques can be suggested according to the derived requirements from scenarios.

5.2 Scenarios analysis

We collected scenarios from several partners in the ELENA project consortium. Scenario driven approaches are quite powerful techniques for expressing high level requirements for queries user can formulate when looking for educational material or information. The scenarios helped us to better shape the diversity of personalization techniques and learner features needed for those techniques. The scenarios we analyzed are simple stories which describe usually different aspects of personalization at different levels of detail and abstraction.

One group of scenarios was oriented towards personalization based on previous experiences or performance within a domain such as finishing a certification program, enhancing the level of knowledge a learner had in some domain, and so on. A second group of scenarios was based on activities the user is involved in his job, e.g. a sale assistant should learn features of a new version of a product he should sell. He should also learn new selling techniques which are needed for this purpose. The manager wants to fill a time gap between meetings and learn something relevant to the next meeting, which can help him to succeed in negotiation, and so on.

Another group of scenarios reflected the interest to learn something from a domain which is not related to work ambitions, e.g. learn something about the history of the country the learner lives in. Interesting scenarios were also those which explicitly considered the distributed nature of the ELENA network and the availability of reusable resources, e.g. a lecturer looking for a resource to enhance his e-course or resources, to help as additional resources for newly created course. Some other scenarios reflected goals such as to move to a higher position. It means that the courses attended by people already working at the position considered as a goal can be analyzed.

An example scenario looked like the following: Bob has a meeting next month in Munich. His personal learning agent (PLA) knows Bob's activities from his calendar, and proactively tries to find out if there are any seminars that are organized in Munich the days after the meeting. A query for learning services related to computer science (Bob's preference) gives the following results: an introductory two day course in basic computer security, a refreshment course in computer security, and two seminars on advanced security technologies in networking.

Bob's PLA knows that Bob already attended an on-line course on basic security from his local university, as part of a larger seminar on computer networks, so it suggests to him only the other three seminars: one to refresh the previously obtained knowledge, and the others two to gain some new information and knowledge. One of these two advanced seminars is part of a series of seminars that lead to a certified security professional (CSP) title. Bob's PLA knows that one of the Bob's goals is to become a CSP in the future, so it emphasizes this information.

Since Bob has not forgotten much about basics in computer security yet, he decides for the advanced seminar that may help him to achieve his goal. A prerequisite for attending the seminar is knowledge about basic security. During the booking procedure Bob's PLA sends together with registration information also a certificate that confirms Bob's attendance to an on-line course on basic security. The certificate, which contains major topics that were covered in the seminar, also contains grades that are privacy sensitive. Bob's PLA decides to cover this information before sending the registration to the seminar provider. The PLA also knows that Bob is a member of IEEE, and he is thus eligible for a seminar fee discount. After the Munich seminar Bob receives a certificate which can later be used in other seminars of the series, enabling him to become a CSP. The certificate is stored in his PLA.

5.3 Resulting Model

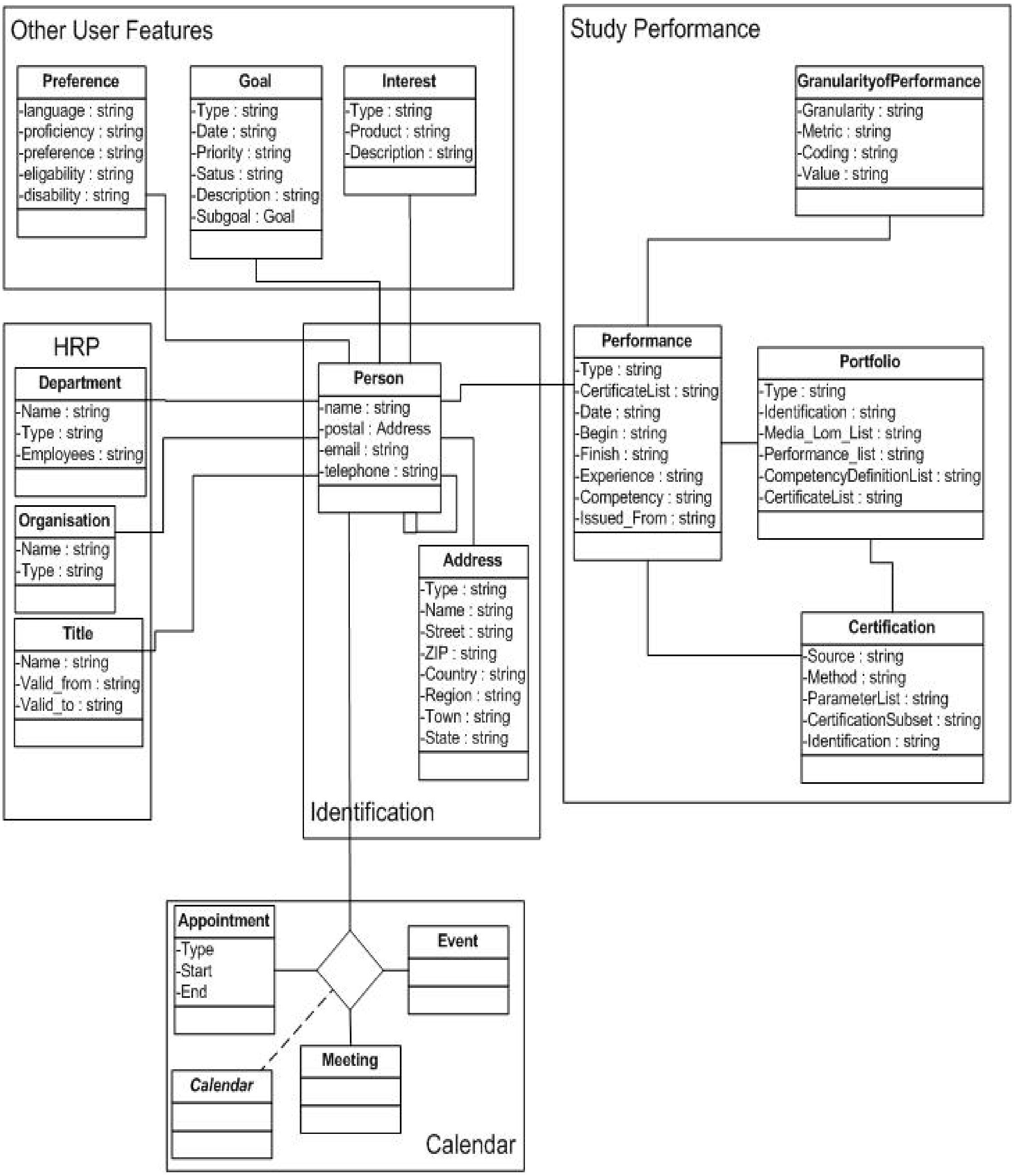


Fig. 3. First version of conceptual model of user profile for Elena. The model serves as a conceptual schema for learner features categories. The features from scenarios analyzed are then instances of the categories (classes) from the model.

Based on the standards before and our scenarios we proposed a learner model displayed in figure 3 as a base for further research. Because PAPI does not sufficiently deal with user features such as goal or interests, we employed a part of IMS profile for maintaining learner goals, interests and preferences. Human resource planning (HRP), calendar, and identification categories are the subject of existing systems and can be extracted from them. Thus we do not explicitly deal with them. The identification category is needed just for internal identification. In case of the identification category we included a subset of PAPI profile field as an example.

5.4 Example

The Bob scenario mentioned in section 5.2 can be refined as follows. Bob is a `Person`, and has a working address, which represents a company he works for. This can be an instance of `Address` class from the learner profile. Bob has proficiency preference for Computer Science (`Preference` instance). He has also the goal to earn a Certified Security Professional certificate (`Goal` instance). This goal is partially fulfilled by his study performance, namely basic security course and later after attending the course in Munich on Advanced Security Technologies in Networking II (two instances of `Performance` class). Certificates from these courses can be instantiated by `Certification` class.Appointment for Munich meeting can be instance of `Appointment` class.

For space limitation we provide here just an example of an RDF instance for the performance record for the Advanced Security Technologies for Networking II course. The example is as follows:

```
<rdf:Description rdf:ID="BOB">
 <papi:performance>
    ...
  <rdf:Description rdf:ID=
   "Advanced Security Technologies for Networking II">
      <papi:issued_from_identifier rdf:resource=
      "http://www.elena.../Test_AST345"/>
    <papi:learning_competency rdf:resource=
    "http://www.kbs.uni-hannover.de/Uli/
    ACM_CCS.rdf\# C.2.0.2"/>
    <papi:learning_experience_identifier
    rdf:resource="http://www.elena.../AST34.pdf"/>
    <papi:granularity>topic</papi:granularity>
    <papi:performance_coding>number
    </papi:performance_coding>
    <papi:performance_metric>0-1
    </papi:performance_metric>
    <papi:performance_value>0.9
    </papi:performance_value>
   </rdf:Description>
  </papi:performance>
</rdf:Description>
```

6. Discussion

Our resulting learner model is based on subsets of both mentioned standards. These standards reflect different perspectives. IMS LIP provides us with richer structures and various aspects. The categories are rather independent and the relationships between different records which instantiate different categories can be accomplished via the instances of the relationships category of the LIP standard. The structure of the IMS LIP standard was derived from best practices in writing CV's. The IMS standard does not consider explicitly relations to other people but they can be represented by relationships between different records of the identification category. However, accessibility policies to the data about different learner are not defined.

PAPI on the other hand has been developed from the perspective of a learner performance during his study. The main categories are thus performance, portfolio, certificates and relations to other people (class mate, teacher and so on). This overlaps with the IMS activity category. However, the IMS LIP defines activity category as a slot for any activity somehow related to a learner. To reflect this, IMS activity involves fields, which are related more to information required from management perspectives than from personalization based on level of knowledge. This can be solved in PAPI by introducing extensions and type of performance or by considering activity at the portfolio level, because any portfolio item is the result of some activity related to learning. IMS QCL category is similar to the certificate list enclosed in the PAPI performance and portfolio category. PAPI also has a preference category, which can be used for storing preferences about devices used for learning. This is similar to IMS accessibility category. IMS accessibility category in addition distinguishes several types of preferences, not only device preference.

PAPI does not cover the goal category at all, which can be used for recommendation and filtering techniques. PAPI does not deal with transcript category explicitly as well. IMS LIP defines transcript as a record that is used to provide an institutionally-based summary of academic achievements. In PAPI, portfolio can be used, which will refer to an external document where the transcript is stored. Although transcript is important information, it is mostly unstructured and of different format and thus hardly applicable to personalized global search in its current form. The competence category does not figure explicitly in PAPI as well. The learning experience field of the performance category can be used for encoding the competence acquired during learning.

The RDF features mentioned in section 4 have interesting implications for learner modelling. They allow us to use schema elements of both standards and also elements of other schemas. The RDF models can be accessible by different peers and even more different peers can have own representation of opposite peer. This was already discussed in [11] in the context of distributed learner modelling.

Personal learning assistants can be considered also as peers which represent a user or a learner. The metadata about a user can be provided to other peers for computing purposes (when the user allows this sharing of profiles). These peers can send and receive messages when a common language is used. Queries can be forwarded to one or more peers.

Based on this, Figure 4 depicts our current ELENA architecture. Circles represents simple providers without reasoning capabilities. Rectangles represent peers, which are able to perform programs. Multiple rectangle symbols represent metadata. Learning resources are provided through resource provider peers. Resources can be referenced by courses which represent simple learning services. Courses can be personalized by adaptation services provided by personalization peers. Courses and resources can be recommended by recommendation services or can be filtered by filtering services. Personal learning assistants support learners or user to use the network.

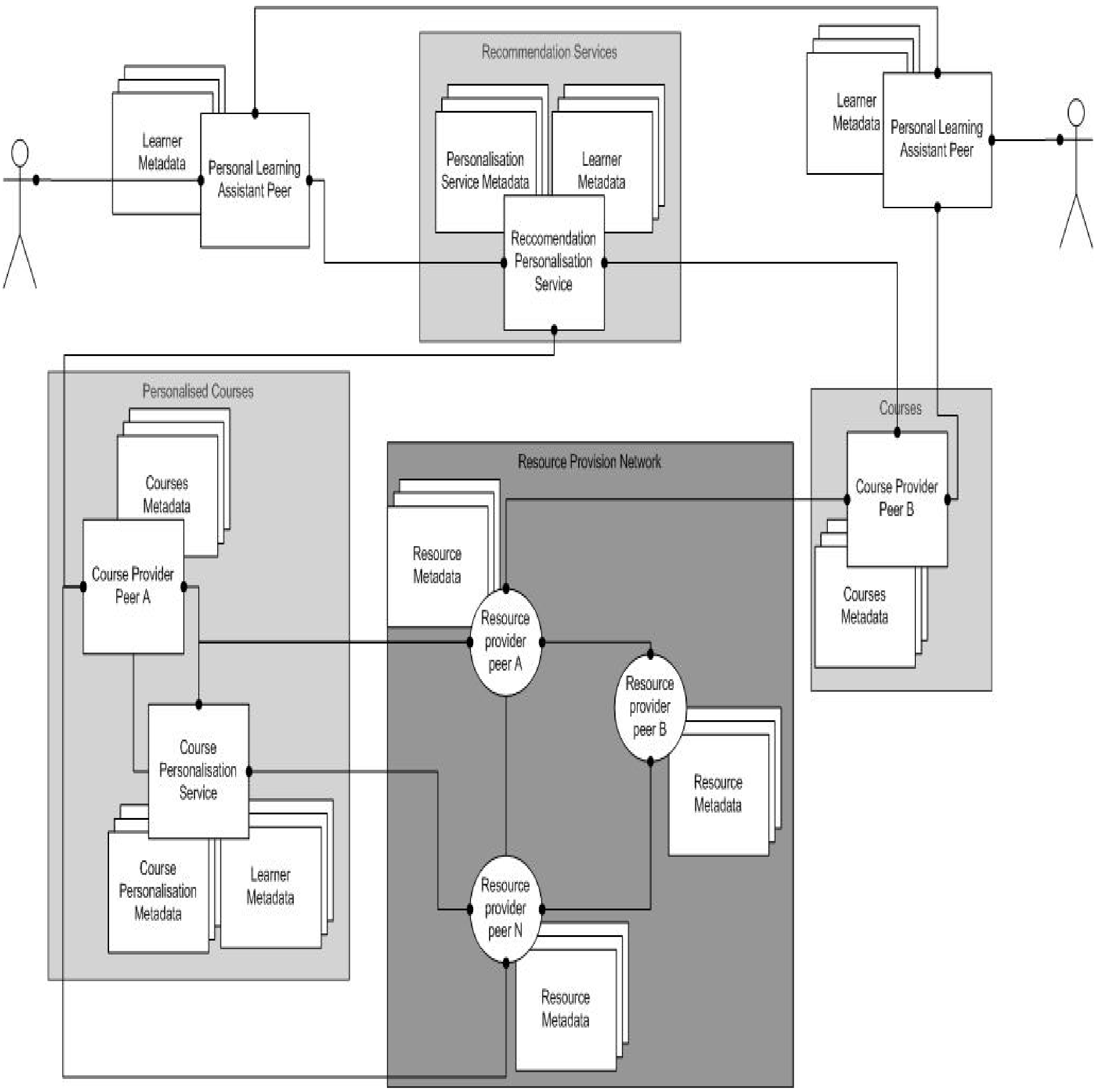


Fig. 4. ELENA architecture for personalization services. The personalization services might be generic adaptive functionalities provided and described in common language, e.g. first order logic (see [4] for details). The generic personalization services can be then reused in several courses and/or queries. The example of such generic personalization service would be recommendation of particular course fragment based on its prerequisites what can be defined independently from topics and fragments available in the course.

The Datalog-based RDF-QEL language allow us to implement reasoning capabilities by using prolog or other Datalog-based systems. These reasoning can be run on specific peers when broadcasted messages are received or after receiving results of submitted query. In [3] we already described some reasoning capabilities for some adaptive hypermedia techniques for an electronic course. The basic idea is to match learner performance descriptions to requirements (accessibility restrictions) a resource has for specific learner. It means that both (learning resource and learner model) use one schema according to the requirements for adaptation based on learner performance within the domain. The performance schema was derived from PAPI standard.

Aggregated user profile of users who visited a resource or liked the resource can be modelled as accessibility restrictions as well. However, different features and structure of this constraint will be needed. This model can be matched to other users searching for resources according to their goals, interests, etc. This mechanism is also used in some collaborative filtering techniques and can be represented by reasoning programs at different peers.

7. Conclusions and Further Work

This paper reported on recent work for the development of learner profile for the ELENA project. Work is not yet finished work but we can already identify features and characteristics important for open environments. First of all, we can definitely benefit from learner profiles standards by combining them. The scenario driven schema development helped us to better describe and analyze our problem. Diversity of learner aspects covered by scenarios allowed us to analyze standards for learner profile from an ordinary learner perspective, who is not concerned with technical issues. Our case study for personalized search services showed that even for these restricted personalization functionality we have to build on two different standards.

Many issues still have to be resolved. The technical infrastructure for this approach to personalization has to be investigated in more detail and mechanisms for provision, searching, and using such personalization services have to be introduced. Mapping or mediating between different schemas should be investigated as well when we want to provide communication between different peers. Scenarios collected helped us in the first phase but as any user input they are at different levels of detail, precision and completeness. Thus evaluation and experiments with different queries are needed to improve our scenarios and suggested personal profile. Different strategies for employing and integrating reasoning capabilities into peers will also be investigated in the future.

Acknowledgements

We would like to thank Elena project consortium members for contribution to scenarios. This work is partially supported by EU/IST ELENA project IST-2001-37264.

References

1. D. Brickley and R. V. Guha. W3c resource description framework (rdf) model and syntax specification. http://www.w3.org/TR/1998/WD-rdf-schema/. Accessed on October 25, 2002.
2. Peter Brusilovsky. Adaptive hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87-100, 2001.
3. Peter Dolog, Rita Gavriloaie, Wolfgang Nejdl, and Jan Brase. Integrating adaptive hypermedia techniques and open rdf-based environments. In *Proc. of 12th International World Wide Web Conference*, Budapest, Hungary, May 2003.
4. Nicola Henze and Wolfgang Nejdl. Logically characterizing adaptive educational hypermedia systems. In *Proc. of the AH'2003 - Workshop on Adaptive Hypermedia and Adaptive Web-Based Systems*, Budapest, Hungary, May 2003.
5. IEEE. IEEE P1484.2/D7, 2000-11-28. draft standard for learning technology. public and private information (papi) for learners (papi learner). Available at: http://ltsc.ieee.org/wg2/. Accessed on October 25, 2002.
6. IMS. IMS learner information package specification. Available at: http://www.imsproject.org/profiles/index.cfm. Accessed on October 25, 2002.

7. Alfred Kobsa. Generic user modeling systems. *User Modeling and User-Adapted Interaction*, 11(49):49-63, 2001.
8. O. Lassila and R.R. Swick. W3c resource description framework (rdf) model and syntax specification. Available at: http://www.w3.org/TR/REC-rdfsyntax/. Accessed on October 25, 2002.
9. W. Nejdl, B. Wolf, C. Qu, S. Decker, M. Sintek, A. Naeve, M. Nilsson, M. Palmér, and T. Risch. EDUTELLA: a P2P Networking Infrastructure based on RDF. In *In Proc. of 11th World Wide Web Conference*, Hawaii, USA, May 2002.
10. M. Nilsson. Ims metadata rdf binding guide. http://kmr.nada.kth.se/el/ims/metadata.html, May 2001.
11. Julita Vassileva, Gordon McCalla, and Jim Greer. Multi-agent multi-user modelling in I-Help. *User Modeling and User-Adapted Interaction*, 2002.

Using Case-Based Reasoning to Support Authors of Adaptive Hypermedia Systems

Peter Funk
Department of Computer Science and Engineering
Mälardalen University
Västerås, Sweden
peter.funk@idt.mdh.se

Owen Conlan
Knowledge and Data Engineering Group
Trinity College
Dublin, Ireland
Owen.Conlan@cs.tcd.ie

Abstract

Educational Adaptive Hypermedia Systems (AHS) are capable of producing personalized learning courses that are tailored to various learning preferences and characteristics of the learner. In the past AHS traditionally have embedded experts' knowledge in the structure of AHS content and applied appropriate design models. However, such systems have continually been criticized for believing that this is sufficient for effective learning to occur [11]. For a tutor who develops such a system there may be many permutations of narrative, concepts and content that may be combined to produce the tailored learner courses. However, the more levels of personalization the system can provide the greater likelihood exists that the system may produce an unforeseen or undesired effect. As a tutor it can be difficult to monitor the suitability of the personalized course offerings on an individual learner basis. Addressed in this paper are these questions: *How can feedback about the effectiveness of highly personalized courses offerings be gathered and returned to the author? and Can CBR techniques be utilized to identify suitable candidate content for narrative population?* By utilizing CBR techniques as a form of quality assurance for the author fine-grained stereotypes, in content or learner models, may be identified. This paper provides a high level overview of a technique for predicting/monitoring personalized course suitability and increasing the quality of delivered courses using CBR in combination with other techniques, e.g. filtering techniques.

1. Introduction

Research in knowledge management deals with methods, models and strategies to capture, reuse and maintain knowledge. Knowledge management, and in particular experience management which focuses on previous stored experience, e.g. previous learners performance and feedback in context of a specific course, is highly relevant for AHSs especially later developments where different methods and techniques from artificial intelligence are included, such as case-based reasoning, clustering and filtering techniques [4]. We will focus on two main issues:

1. How feedback is used to help the author to make improvements in the course design?
2. How feedback into the system is used to improve personalization both initially and during learner sessions (adaptation/personalization is seen as an ongoing process)?

The paper explores how AHSs may benefit from case-based reasoning and filtering techniques and some of the challenges these systems offer case-based reasoning researchers.

In adaptive education systems an experience management approach may be used to deliver personalized courses based on the both current learner performance and on the performance of learners, together with the experiences of using different narratives and content models. Using filtering techniques (filtering suggestions/adaptations through the preferences/results of other similar learners) enables the system to reuse experience from similar learners' preferences, successes and failures to improve adaptation to current users' needs and preferences. These approaches, however, traditionally suffer from a training period before the system can produce accurate recommendations.

Techniques such as category based filtering [10] may be used if learner models, content (material with the purpose to transfer specific knowledge to the learner, may have the form of text documents, animation, simulation, contain interaction, tasks, etc.), narratives and results from different individual learners are fragmented (e.g. different learners have completed different content but not a full course) and sparse. Compared with other filtering techniques category based filtering requires categorization of all items and clusters users/learners into learner stereotypes which reduce the latency problem, e.g. if previously a learner has not been able to review and give feedback on a complete course and their constituent content components, category based filtering is able to reduce this problem and give recommendations based on learner stereotypes instead of individual learners.

In rule based expert systems there are no latency problems as the personalized course is generated based on rules developed by an expert in the knowledge domain [2]. The more complexity that is built into such rules the more likely it is that the system will produce a course that does not fully cater to the learner's needs, deviating from the author's original goals. Managing large sets of rules also requires much effort from the expert (the tutors in an AH system).

The suitability of a personalized course offering can be determined by examining the learner's feedback, explicit and implicit [8]. The feedback may be given by, or requested from the user, or gathered implicitly by analysis and evaluation of the learner's progress and results. As a tutor, however, it can be difficult to spot the trends in this feedback and correlating it with the personalized courses generated.

A case-based reasoning approach is proposed for identifying and correcting potential problems with personalized courses by matching, reusing, validating and storing cases, where cases may be individual learner models, narratives or individual content models. Producing learner stereotypes using clustering techniques, and comparing the stereotypes can overcome the latency problem. This paper outlines this approach in the context of an existing research Adaptive Hypermedia Service, APeLS (Adaptive Personalized eLearning Service) system developed at Trinity College, Dublin [3].

2. Multi-model Adaptive Hypermedia Services

Multi-model Adaptive Hypermedia Services combine information models about learners, learning content and narrative structures, i.e. what knowledge to transfer to this particular learner to produce personalized course offerings to learners. This enables the Adaptive Hypermedia Service to deliver personalized eLearning courses. These three components are characterized as distinct, and separate, models within the Adaptive Hypermedia Service (see Figure 1).

The learner model contains a model of the learner with respect to learning preferences, knowledge, preferred learning style, results etc. The content model represents the learning content which may be selected to be taught to a particular learner. The learning content may be rendered in a variety of ways , e.g. text documents, slides, animation, simulation, interactive, etc. [3] proposes a mechanism that enables the personalized course structures to be described in terms of concepts rather than the pieces of learning content that teach those concepts. This abstraction enables the Adaptive Hypermedia Service to populate the concept with an appropriate piece of learning content at runtime. For example, if the learner prefers interactive content then a kinesthetic piece of content may be delivered over a non-interactive visual piece of content. The domain or narrative model is responsible for describing the possible combinations of learning concepts that may be assembled to fulfill a learner's personal learning goals.

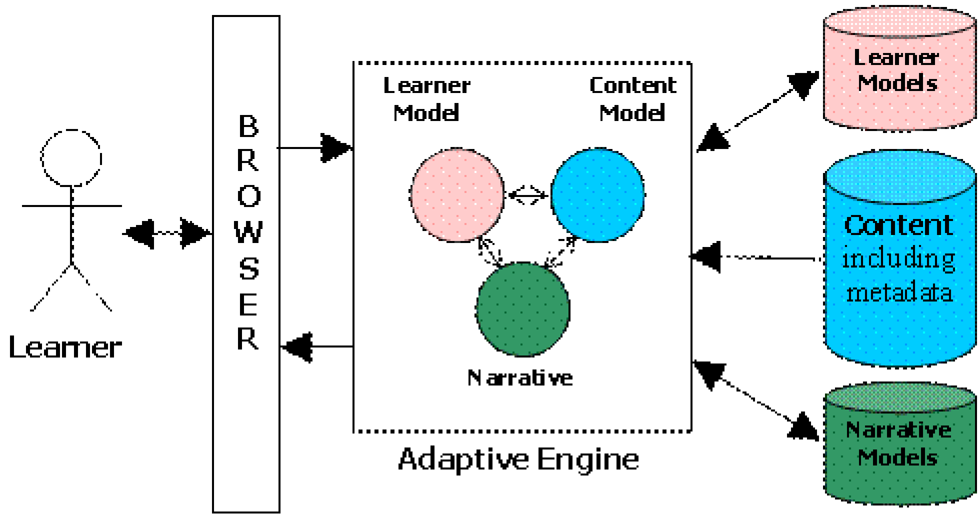


Fig. 1. Multi-model Adaptive Hypermedia Service

This approach has at least two potential problem areas that the domain expert who designs the narrative (in the form of rules/examples on how to combine, select and order material for different learners and their preferences) may not be able to foresee –

1. The sequencing of the concepts in the personalized course may not be appropriate for the learner.
2. The pieces of content selected to fulfill a concept may not be effective at doing so.

As the narrative models become more complex (or begin incorporating other narrative models) the task of foreseeing and/or diagnosing these problems in the personalized course becomes increasingly difficult for the domain expert. This task is further complicated by the ability to associate multiple base narratives with one course – each narrative produces personalized courses with the concepts sequenced in different ways catering to the learning styles of different learners.

In these situations it would be desirable to correlate learner feedback (performance on tests, explicit querying, learners' behavior during learning etc.) with the personalized course offerings to determine trends and identify potential problems. The feedback given may be classified to the following categories:

1. Feedback on concept/content reflecting the quality or suitability for this particular learner.
2. Feedback on narrative, if the selected concepts or concept sequences are relevant for this learner.
3. Feedback on learner model, does the learner have the knowledge and learning preferences reflected in their learner model.

The feedback is elicited in a number of different ways and may also need some analysis to enable conclusions. The feedback may not be deterministic, merely strengthening or weakening beliefs. Some feedback may be used to dynamically revise the personalization of the course; other feedback is used to improve the quality of future personalization for learners, in particular for similar learners. This approach fits well with the concept of case-based reasoning and collaborative filtering with minor modifications, outlined in following section 3 and 4.

3. Case-Based Reasoning and Filtering Techniques

A number of different parts in the adaptive hypermedia system can be used in a case-based reasoning approach to develop AHSs, reuse and personalization. The filtering approach is used to recommend and select items, inside our outside a case based reasoning cycle, using experience from other learners, authors and tutors.

3.1 Examples use of Case-Based Reasoning

During the process when authors design courses, a CBR approach may be used where content, concept, narrative may be reused from similar courses designed by other authors. Since the components in the AHS are standards-based [3] learning content repositories, on the web, may be used and during the authors work, the design tools may invoke a CBR system to identify and propose similar cases. Once found, the CBR system may perform some adaptation to adapt or combine the material with the authors design. Final revision is performed by the author and the case is stored in the case library, ready to be used by the personalization learning service. By combining this technique with the abstraction architecture [3] the CBR system may be used to populate concepts with appropriate content.

Learning content and concepts have a high potential for reuse. Using feedback on how successful they are in different situations their appropriateness for reuse in different circumstances may be determined. Content may also be reused in the CBR cycle for automatic personalization during learning sessions, and may even identify content not initially thought of by the tutor (the learner/tutor should be informed if this is the case). The advantage is that it is a closed cycle and able to handle both feedback, direct in the form of learners comments and indirect by comparing the learners result with the result of other cases. Feedback from tutor observation is also handled and of value when revising a course or designing a similar course.

3.2 Example use of Filtering Techniques

Filtering techniques are proposed where the main task is to identify the learner’s preferences and needs based on previous learners’ results, behavior and feedback. Category based filtering is especially suitable if all objects are categorized in advance, which is the case in the personalized learning service (the author may create additional metadata and classifications during narrative design). In the next section we will outline the proposal in more detail.

Category based filtering as described in [10] is based on category ratings. In the personalized learning service the content is categorized by concepts. A category bases filtering system is able to use the concepts as categories. To reduce the latency problem, content models and their specific narratives are clustered in clusters of similar users (a number of different approaches to perform clustering exist). Learners may be part of more than one cluster. These clusters are then merged to a stereotype learner for this cluster and will be more complete than individual learners. A new user can often be classified to belong to one of these categories, and the category can be used to guide the personalization until enough is known about the user to pass the personalization entirely on the users user model. In Figure 2 an example on how category based filtering can be used to recommend similar learner models to use in the personalization process. Clustering is used to generate a number of learner stereotypes, useful if learner models are sparse and incomplete (few learners may have completed a similar course, just covered it in part). The dotted line from learner models indicates that learner stereotypes are preferred and learner models only used if there is a good mach and no good mach amongst stereotypes (stereotypes capture experience from a larger group of similar learners).

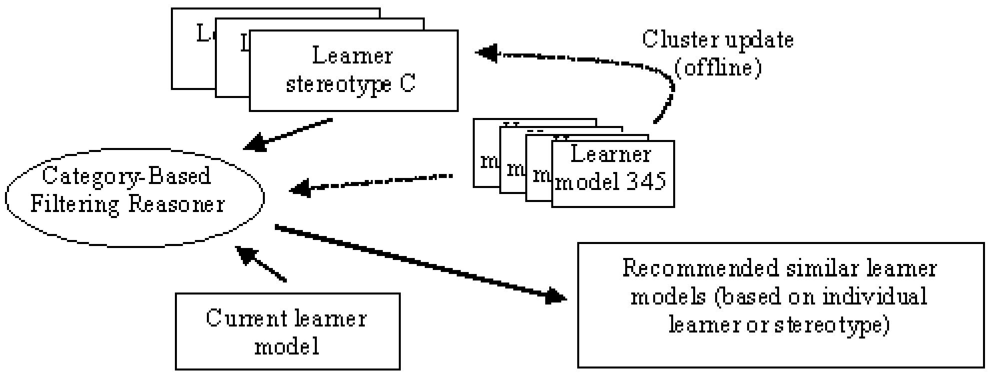


Fig. 2. Outline of an adaptation of category based filtering, as described in [10].

A learner model may also include references to previously learned courses, their content models, content, narratives, feedback and results (used in the personalization process).

4. Case-Based Approach to AHS

As discussed in the previous section there are a number of tasks and problems that can be addressed with case-based reasoning and filtering techniques. In this section we will describe the APeLS system in terms of cases and explore how case-based reasoning may be applied in AHS systems, both aiding the experts to improve and refine eLearning courses and also improve online personalization by using feedback on learning objects and concepts to make modifications in content model and selected learning objects (content). In section 2 we discussed the problems for a domain expert to be able to see all implications on how a course can be personalized from the set of narrative rules and how well a selected learning object for a concept meets all learners' personalization requirements.

Each concept, added by the execution of a narrative, requires some prior competences from the learner and results in additional learned competences after successful completion. Each concept is, after personalization, populated by a candidate selector (CS) and the learning style of the learner should be considered in selecting the candidate. The socket on the lower left side of the CS is an illustration of the learning style, i.e. a learning concept is not completely seen as a black box. The learning style or pedagogical approach of the successful candidate should engage the learner. Candidates may also be used to cater for technical limitations of the delivery device (learner working from different locations may have bandwidth limitations on some occasions, e.g. not able to view video examples).

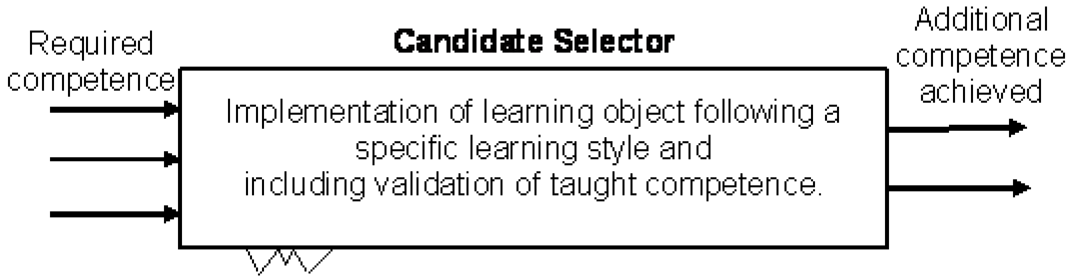


Fig. 3. Structure of a candidate selector

Candidate selection may also be used at the course level. In order to meet all on the learning goals of the course, the course has do be comprised from a number of concepts and corresponding Candidate Selectors. The CSs should match the learner's preferred learning style and sequence, combination, granularity and order should follow the narrative rules. Constructing a course could be viewed as a planning task where requirements and restrictions are enforced by the narrative and competencies required and learned.

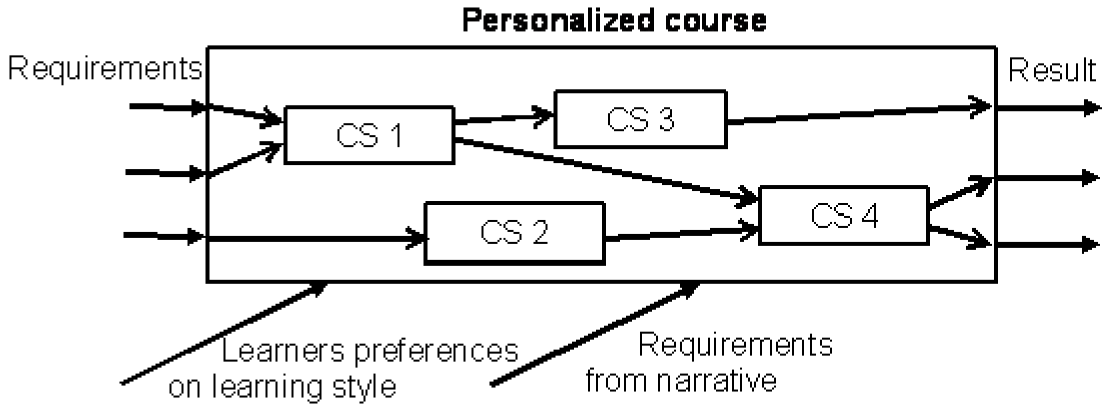


Fig. 4. Example of a personalized course with candidate selectors

Ordering is determined by the narrative, where experts have created rules and each learning object may have prerequisite requirements (knowledge or competencies the learner is supposed to have) and outcomes (knowledge the learner has acquired after successful completion of the learning object). The full search space may be very large since different parts of the multiple models may be used as input information along with feedback from the learner. CBR systems are able to handle models and processes and are both able to correct problems and reuse parts of models and models in full, see e.g. [6].

4.1 Improving the Quality of Personalization using CBR

Describing the APeLS system in terms of cases, both case selectors, concepts and courses are seen as cases, giving a number of interesting opportunities for case-based reasoning researchers and valuable benefits to the AH system. Cases can be reused in part and in full.

Looking at the system in this CBR view also reveals the usefulness of separating the narrative from the learning content [3]. The narrative can be used both in validation of content, case selectors and courses, and also be used as adaptation rules during the reuse phase, see Figure 5.

Concepts, candidate selectors and content can now be used in a CBR system and aid the tutor/author in adapting an existing course or in designing a new course. There are a number of exciting challenges for case-based reasoning researchers as many of the tasks in AHSs are complex. How to populate concepts with content is, for a tutor, a difficult task, and using content containing learner feedback in a case-based approach is interesting to explore (candidate selector using CBR). An interesting issue is the interactive nature and relationship between the different parts of the AHS, and an interesting approach is to work on more than one case-library at the same time. Other avenues include combining CBR with category based filtering systems to recommend appropriate candidates/examples that are used in the adaptation and validation process.

Also ensuring the quality for a new course or after adapting a course is a relevant issue and the challenge in how to use the feedback in cases; it may even be relevant to look for similar cases which previously have shown to be unsuitable. These cases can be used in the adaptation and validation process to ensure the author/tutor does not produce a course that repeats mistakes previously made (e.g. if experience has shown that a particular configuration leads to more failures amongst learners). This may also be valuable when performing on line adaptation - if the learner changes his/her preferences and is proposed a personalized course with corresponding features, the system may recognize from other learners with similar preferences and courses that the learning result takes longer time or learners score lower and even suggest alternatives (e.g. some engineering students selecting personalization with examples may score higher than engineering students selecting abstract theoretical learning content).

5. Summary and Future Work

In this paper we outlined how an existing research Adaptive Hypermedia Service [3] may benefit from CBR and filtering techniques in a number of different ways, in particular how author/tutor may be aided in designing and adapting courses. An important issue is feedback collected in the cases and used both in the adaptation process and the validation process to insure quality and efficiency. Using category based filtering and clustering learner models to find similar learner models used in the personalization process is one example such benefits. Another example discussed is how CBR may be used to populate concept groups with learning content, potentially using the case library. Using less successful examples in the CBR adaptation and validation process to avoid repeating solutions shown to be less successful is proposed. Also the fact that the adaptation may be improved by comparing the current user model with other similar users involved in a similar course is interesting, and a challenge would be to construct a CBR system uses multiple case libraries.

The plans for the future are to select a number of the ideas and challenges outlined in this paper for improving the effectiveness of adaptive features of AHS, and implement them in a collaborative project between Trinity College, Dublin and Mälardalen University, Västerås. It is hoped that this approach will yield greater and more focused feedback to the author/tutors of adaptive courses enabling them to improve the learning experience for the learner. Also the reuse of components in AHS systems will be further aided by this approach and it is hoped that such an approach will result in shared repositories of components between different universities worldwide enabling tutors to design efficient and high quality personalized courses for their students.

The proposed implementation will combine the adaptive AHS system, APeLS, developed by Trinity College, Dublin [3] with the category-based filtering approach and CBR system developed at Mälardalen University, Västerås [6], [10] to produce a system that uses the information gathered about learners partaking in personalized courses to produce recommendations to the course author as to how the adaptive features may be better tuned and aid reuse.

References

1. Aamodt A. and Plaza E.: Case-Based Reasoning: Foundational Issues, Methodological Variations and System Approaches. AI Communications, vol 7 (1994), pp 39-59.
2. Brusilovsky, P.: Adaptive educational systems on the world-wide-web: A review of available technologies. In Proceedings of Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (AHS'98), San Antonio, TX, 1998.
3. Conlan, O., Wade, V., Bruen, C., Gargan, M.: Multi-Model, Metadata Driven Approach to Adaptive Hypermedia Services for Personalized eLearning. In the Proceedings of Second International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, AH 2002 (2002) 100-111.
4. Ferrario M. A. and Smyth B.: Collaborative Knowledge Management & Maintenance. In Proceedings of Workshop "Knowledge Management by Case-Based Reasoning (GWCBR 2001), Baden-Baden, Germany.
5. Funakoshi, K., Ohguro, T.: A content-based collaborative recommender system with detailed use of evaluations. In Proceedings of 4th International Conference on Knowledge-Based Intelligent Engineering Systems and Allied Technologies, Volume 1 (2000) 253-256.
6. Peter Funk. Processes and Structured Transition Rules for Case-Based Reasoning. In Workshop on CBR Approaches for Process-Oriented Knowledge, ICCBR'2001, Fenstermacher K., Tautz C. (eds.) (2001).Vancouver, Canada.
7. Hayes, C., Cunningham, P., Smyth, B.: A Case-Based Reasoning View of Automated Collaborative Filtering. In Proceedings of 4th International Conference on Case-Based Reasoning, ICCBR2001 (2001) 243-248.
8. Kobsa, A.: User modeling: Recent Work, Prospects and Hazards. In Adaptive User Interfaces: Principles and Practice, M. Schneider-Hufschmidt, T. Kühme, and U. Malinowski, (eds.). 1993, North-Holland: Amsterdam.
9. Rich, E.: User Modeling via Stereotypes. Cognitive Science 3, 329-354, 1979.
10. Sollenborn, M., Funk P. Category-Based Filtering and User Stereotype Cases to Reduce the Latency Problem in Recommender Systems. 6th European Conference on Case Based Reasoning, ECCBR2002, 10 pages, in press, Springer Verlag Lecture Notes Series, 2002, Aberdeen, Scotland.
11. Stauffer, K.: "Student Modelling & Web-Based Learning Systems", 1996.
12. Ungar, L.H., Foster, D.P.: Clustering Methods for Collaborative Filtering. Proceedings of the Workshop on Recommendation Systems (1998). AAAI Press, Menlo Park California.

How to give the user a sense of control over the personalization of AH?

Marek CZARKOWSKI, Judy KAY
School of Information Technologies
University of Sydney
Sydney, NSW 2006, Australia
{marek,judy}@it.usyd.edu.au

Abstract

This paper describes work towards a scrutably adaptive teaching system. We outline our motivations for scrutability. Then we give an overview of Tutor3 and a report on a qualitative evaluation to assess whether users could understand the results of the adaptation and how to change their user model to control the adaptation. This report indicates that participants in the evaluation could, generally, understand how the material was adapted and how to control that adaptation. It also highlights some serious challenges for supporting scrutability.

Keywords

Scrutable, user model, adaptive hypertext, adaptation control, adaptation visualization, adaptive web environment.

1. Introduction

A scrutable adaptive system enables the user to delve into the way that the system has been adapted. We have several major motivations for this work: the first relates to issues of control over private information; the second is our belief that electronic teachers should be scrutable since they should be able to account for their actions in terms of the real drivers for adaptation and this maintains the position of the learner as the human who controls the machine; finally, there is the goal of improving learning. Key concerns to achieving these goals are how to best present the results of adaptation to the user in a way that they can understand what caused the adaptation and how they can control it.

While an accessible student model provides an excellent basis for reflection and for discussion, we would like to explore ways for the learner to see the *implications* of their student model in terms of the adaptations that it effectively controls. Essentially, we want to enable a learner of an adaptive system to be able ask `How was this material adapted to *me*?', `How might this system teach this to someone else?' and to be able to see just how their student model affected what they saw in the hypertext document. At this stage, the user model is quite simple. However, this makes a logical starting point for exploring ways to effectively provide support for users to scrutinise the adaptation of a system.

The design of Tutor3 was informed by our evaluation of its predecessors [4,5]. Tutor, the first version, was evaluated in a field trial. This appeared to be quite successful, with a total of 113 students registering with the system, 29% exploring the adaptivity and, of these, 27% checking what had been included as part of the adaptation and where content had been excluded. This evaluation identified limitations, too. While it did show material that had been adaptively *included*, it showed only the location of material *excluded*. Also, some users failed to appreciate that they could, affect the adaptation, at any time, by altering their answers to the profile questions. In essence, the Tutor evaluation showed promise but also identified ways that the scrutability support was incomplete.

These concerns were addressed in Tutor 2. We performed a qualitative evaluation to assess whether users could scrutinise its adaptation effectively. The student should be able to:

- Determine what content was *included*/*excluded* on a page;
- What *caused* the adaptation;
- Understand how to change their profile to *control* what content is *included*/*excluded*.

The evaluation of Tutor2 indicated that users could not do these things. So, we completely redesigned the adaptation explanation to the form it has in Tutor3. We note that the whole purpose of a *scrutable* adaptive hypertext is to ensure that the student can, in practice, scrutinise it to understand and control the adaptation.

Section 2 gives an overview of the learner's view of Tutor3. Section 3 describes the design of the evaluation study and Section 4 reports the results. The final section discusses these and conclusions.

2. Overview of Tutor3

Tutor3 is a web-based tutoring environment. The system employs the adaptive presentation technique of inserting and removing page fragments. For other adaptive presentation techniques the user is referred to [1.2]. The author creates lesson content in ATML, an XML syntax that adds simple mark-up to HTML. Mark-up is added to adaptive page fragments to describe the user model attribute/value pairs a user must hold to be shown that fragment within the lesson page. This approach is similar to the adaptive presentation technique used in AHA! (De Bra, 2002), although Tutor3 allows only simple boolean rules in the page fragment criteria. In this paper, we focus on aspects of Tutor3 related to scrutability. For other aspects, see [4,5].

The starting page gives an overview of Tutor3 and links to register and login. Once logged in, first time users see a welcome message explaining the need to answer some questions and that:

> `The answers you give will be stored by the system, in what Tutor calls your *profile*, and will help Tutor customise each lesson to you. You will be able to review and change your profile at any time and, by doing so,alter how the material is customised to you. At the bottom of every lesson page, there is an explanation of how the lesson has been customised to you.'

Each unit starts with a set of questions, as in Figure 1. These set the values in the student model. For example, the first question asks the student if their goal is to *Learn the material* or *Revise the material*.

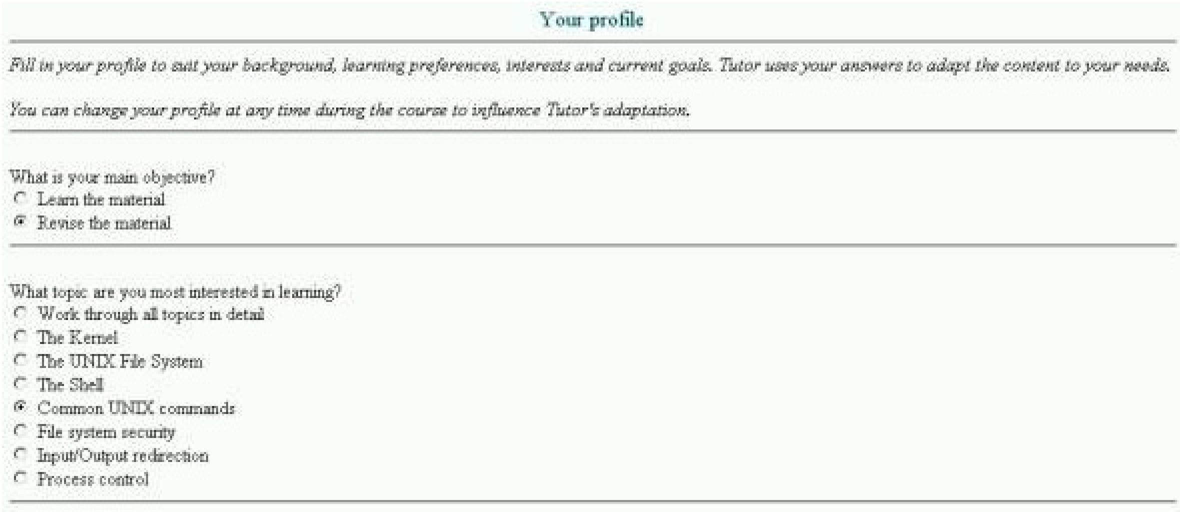


Fig. 1. Example of a profile display. Tutor3 presents students with a set of questions and uses their answers to establish their student model (profile). This screen shot and later ones are all from Netscape 4.7.

Next, Tutor3 presents lesson content as in Figure 2. The icons, near the top, link to various Tutor facilities. These include the course map, a page for the student to make notes, the glossary and on-line help. Importantly for this paper, the head accesses the profile. This enables the student to alter this at any time and so, to influence the adaptation.

As the Figure 2 has adaptation, it has the link: *How was this page adapted to you?* Clicking this, a hidden section appears as in Figure 3 to explain the adaptation. On pages without adaptation, this link is replaced by the text: *There was no adaptation on this page*.

The adaptation explanation, as in Figure 3, shows each profile question that influenced adaptation of the current page. Beside the question is a short form of the student's current answer, as in the student model, and a *show me* hyperlink. For example, the first row in Figure 3 indicates some adaptation was caused by the user answering *revise* to profile question *What is your main objective?* Clicking *show me* opens a separate browser page, the Adaptation Explanation, as in Figure 4.

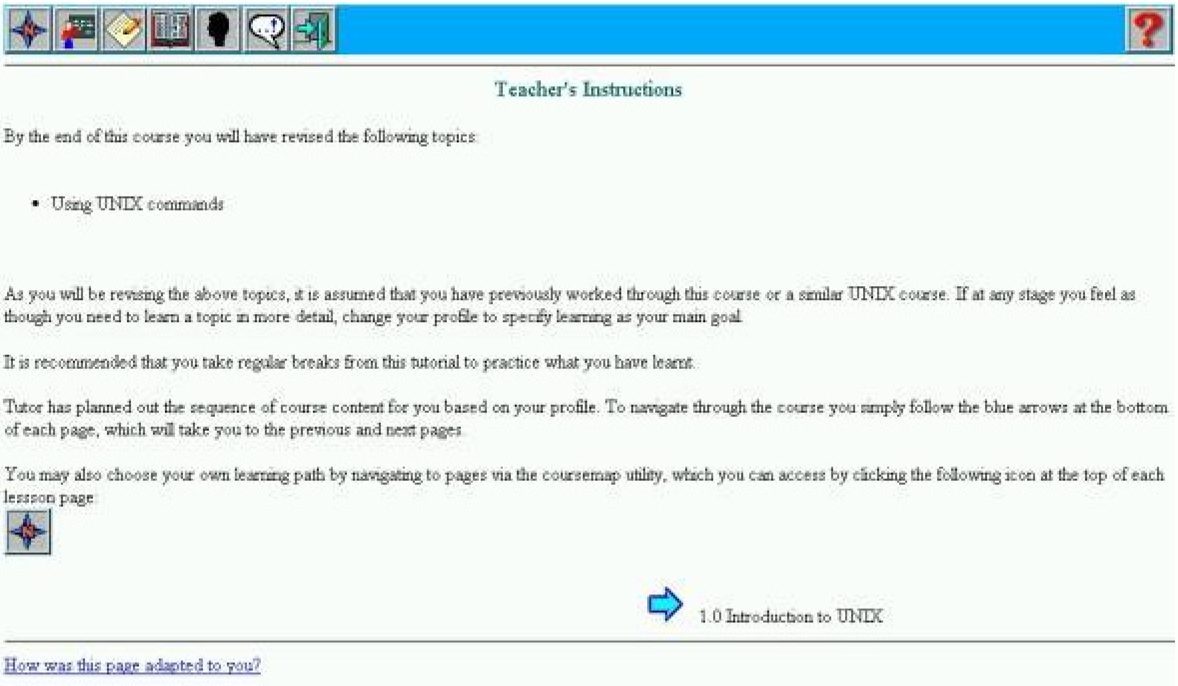


Fig. 2. Example of an adapted lesson page. Clicking the hyperlink *How was this page adapted to you?* at the bottom of the page provides an explanation of adaptation that occurred on the page (see Figure 3).

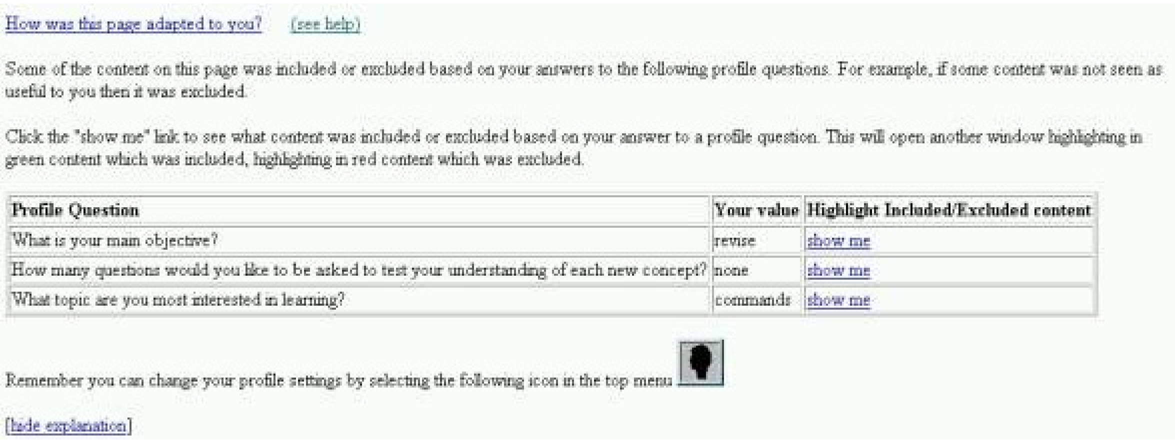


Fig. 3. Example of an explanation section at the bottom of an adapted lesson page. This explanation is displayed by clicking the hyperlink *How was this page adapted to you?* from a lesson page (see Figure 2). Clicking the *show me* hyperlink opens a separate browser page showing content that was included or excluded (see Figure 4).

Content that was *included* has a green background, while *excluded* content has a red background. This is explained in the Key at the upper right. In Figure 4 the word *revised* in the first paragraph and the second paragraph beginning with *As you will* is highlighted in green, the word *learnt* in the first paragraph is highlighted in red. At this point, the student can see both the page in Figure 3 and that in Figure 4 in separate browser windows so they can compare the Adaptation Explanation against the original lesson page. In fact, since the Adaptation Explanation for each profile question is displayed in a separate browser window, the user can compare multiple adapted versions of the page.

An info icon is displayed within each highlighted section of content which was included or excluded. Moving the mouse over this icon pops up a further explanation. For example, in Figure 4, moving the mouse over the info icon in the first highlighted section, before the word *revised*, pops up the text *This content is included if your value for the above profile question is revise*.

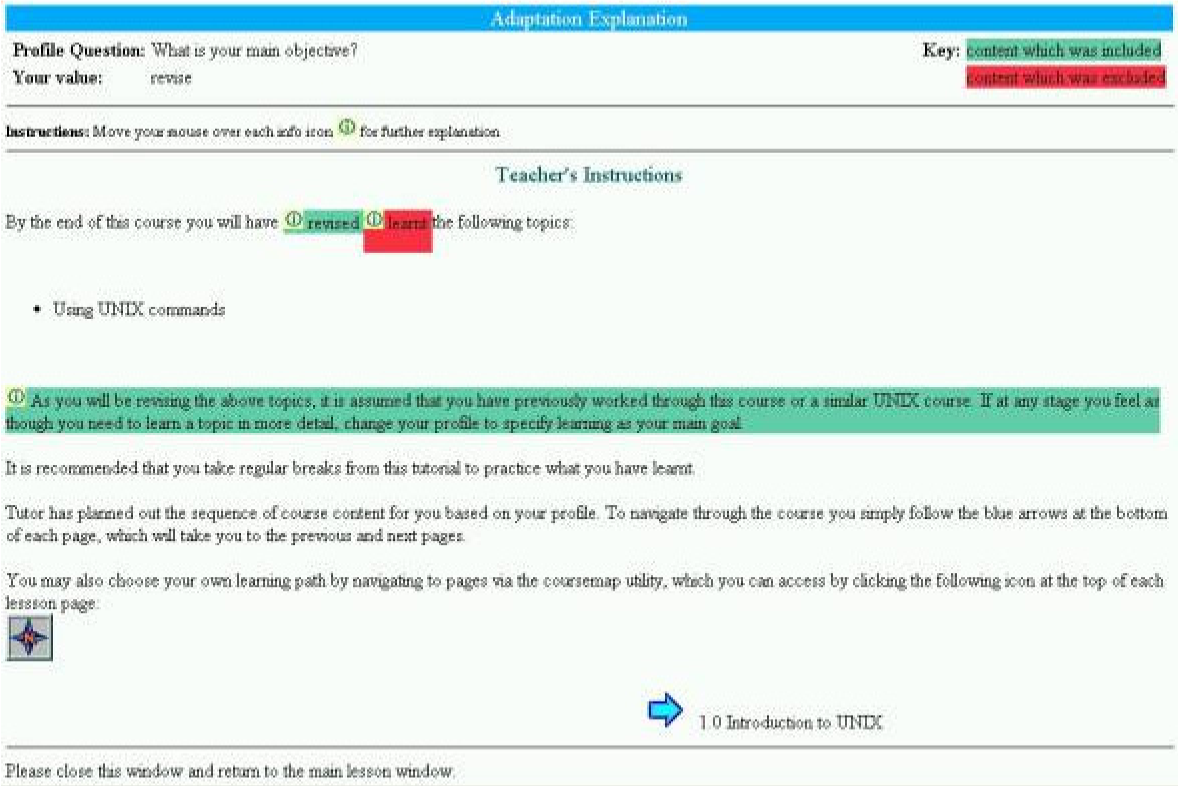


Fig. 4. Example of an adaptation explanation page showing content that was included and excluded based on the student's answer (*revise*) to the profile question *What is your main objective?*. Content that was included is highlighted in a green background colour (in the figure the word *revised* in the first paragraph and the second paragraph beginning with *As you will* is highlighted in green). Content that was excluded is highlighted in a red background colour (in the figure the word *learnt* in the first paragraph is highlighted in red). Moving the mouse over the info icon in a highlighted section pops up a further explanation. For example, moving the mouse over the info icon in the first highlighted section, before the word revised, pops up the text *This content is included if your value for the above profile question is revise*.

3. Evaluation

Our evaluation was qualitative, based on a think-aloud [9]. This has the merit of being relatively low cost and giving insights into the causes of difficulties. Following Neilsen, we selected five participants for the evaluation. There have different backgrounds and varying degrees of computer literacy: one was a secondary school student, two were third year computer science degree students and there were two adult participants with basic computer literacy skills.

We designed the evaluation around a fictitious person, Fred. This meant that all participants were dealing with the same student profile and we could predict the exact adaptation each should see. This, in turn, meant that all participants did exactly the same task and saw exactly the same screens.

Each participant was provided with a worksheet. This described Fred's learning goals, interests and background. Participants were asked to assume the role and background of Fred and use Tutor3 to start working through the beginning of the Introductory UNIX course. Participants were presented with one page of the worksheet at a time so they could not jump ahead.

The tasks of the worksheet were designed so that each basic issue was explored in three subtasks. This provided internal consistency checks, an important concern since there are degrees of understanding and we wanted insight into just how well each participant was able to scrutinise the adaptation. Extracted questions from the worksheet are shown in Table 1. The worksheet asked the participants to:

- Explain the use and effect of a Fred's answers to profile questions;
- Explain how Tutor3 decides what content is included/excluded on a page;
- Demonstrate control over adaptation by changing answers to profile questions.

Table 1. A sample of questions in a worksheet completed by participants in the evaluation of Tutor3.

Concept	**Questions presented in worksheet**
Explain the use and effect of Fred's answers to profile questions	• Will Tutor use Fred's profile settings? If so, what will it use Fred's profile settings for? • Where does Tutor get information about Fred to use to perform adaptation of the content to suit him? • Will Fred be able to influence or control the way Tutor adapts content to him? If so how?
Explaining how Tutor3 decides what content is included/exclude d on a page	• What would you do in the system to find out whether Tutor adapted any material on the Teacher's Instructions page to you? • Did your answer to the profile question "What is your main objective?" have any effect on the contents of the Teacher's Instructions page? How do you know this? • Was any content specifically excluded because of your answer to the profile question "What is your main objective"? If so, what was the content?

Demonstrating control over adaptation by changing answers to profile questions	• Now consider what the first sentence on the page would read had you answered the profile question "What is your main objective" with "Revise the material". Without changing your answer just yet, write out how you think the sentence would read. "By the end of this course you will have …"
	• Explain what actions you would have to perform in the system to change your answer to the profile question "What is your main objective"?
	• Change your answer for the profile question "What is your main objective" to "Revise the material".

The first part of the worksheet instructed the user to register with the system, log on and select the Introductory UNIX course. The first task was to complete the profile of Fred (as in Figure 1). The worksheet did not specify the answers for each question but did describe Fred's learning goals, interests and background. Each participant was asked questions from the first block in Table 1.

The second part of the worksheet asked each participant to navigate to a specific page, that shown in Figure 2. Then, participants were asked to determine whether any of the content on that page was adapted to Fred, based on the answer to the profile question *What is your main objective?* The participant had to indicate any adapted content and whether it was included or excluded. To perform this task, the user had to notice and click the hyperlink *How was this page adapted to you?*

The third part of the worksheet involved what-if experiments. The participant was asked to guess the content of a specific paragraph, assuming Fred were to change his answer for the profile question *What is your main objective?* to *Revise the material*. Answering this question should not have required any guesswork. The answer is on Adaptation Explanation as in Figure 4.

4. Results

Table 2 summarises the results of the evaluation. It shows the range of age, academic background and computer literacy. Overall, they represent a quite computer literature group.

The fourth row of Table 2 summarises the participants' understanding of the core concept that answers given in the profile were used to adapt the content. They first had to answer the 8 profile questions, as Fred would have. These appeared as in Figure 1, with additional questions about preferences for abstract explanations, numbers of examples and self-test questions, interest in historical background and understanding of other operating systems. For this experiment, the critical aspects about Fred's background were that he wanted to learn (not revise) and he was not interested in historical background information. All participants answered these profiles questions consistently with the information we provided about Fred.

All participants understood that profile answers controlled the adaptation. However, Participant 4 believed they would not be able to influence or control the way Tutor adapts content. Participant 3 believed they might have the ability to do so but did not understand how. Participant 5 initially did not appreciate that profile answers could be changed later. Overall, at this stage, having just completed the profile, participants understood that profile answers affected adaptation, but did not know whether they would be able to control the adaptation. Note that this is inspite of the message at the top of the screen (see Figure 1):

> 'Fill in your profile to suit your background, learning preferences, interests and current goals. Tutor uses your answers to adapt the content to your need.
> You can change your profile at any time during the course to influence Tutor's adaptation.'

Table 2. Summary of qualitative evaluation of Tutor3.

	Participant 1	**Participant 2**	**Participant 3**	**Participant 4**	**Participant 5**
Age group	21-25	21-25	18-21	18-21	14-18
Background	Post study adult	Post study adult	Year 3 Comp Sci degree student	Year 3 Comp Sci degree student	Secondary school student
Computer literacy	Basic	Basic	High	High	Moderate
Understood role of the Profile	Yes	Yes	Yes	Yes	Yes
Understood green background shows adaptively included content	Partly – and needed help to access explanation	Yes but needed help to access explanation	Learnt this concept through the evaluation	Yes	Yes
Understood red background shows adaptively excluded content	Partly – and needed help to access explanation	Yes but needed help to access explanation	Learnt this concept through the evaluation	Yes	Yes
Understood profile answers may be changed to alter adaptation of content	No – did not know how to access profile	Yes	Yes	Yes	Yes
Time to complete worksheet	50 min	31 min	20 min	22 min	28 min

The fifth row shows how well the participants understood the Adaptation Explanation of the text *included*, based on their written responses and actions performed to answer the middle three questions in Table 1. This required that they appreciate that the green background indicated adapted content had been *included,* because of the answers to the questions on the Profile Page. Participants 1 and 2 needed help in accessing the explanation in order to continue the worksheet. Allowing for this, all participants other than Participant 3 did understand that the green background indicated content that had been included. Participant 1 stated that they did not understand why the other (non-adapted) content was not also given a green background. Participant 3 did not notice the highlighting feature when completing this stage of the worksheet. This participants' method for determining whether content had been adapted was to go back to the profile page, change their answer, navigate back to the lesson page and study the content for changes. This participant correctly identified the included content. Later, Participant 3 discovered the highlighting feature and showed understanding of the concept.

The next row shows the participants' understanding of the Adaptation Explanation for material that was *excluded*. It shows exactly the same pattern as for *included* material. One exception occurred with Participant 3, who could not identify excluded content through their un-aided visual approach to determining whether content was excluded.

The second last row shows the participant's understanding of the central notion that they could alter the adaptation by altering their answers to the profile. The worksheet asked participants to alter the adaptation on a page (see last three questions in Table 1). The first question asked the participant to determine the content of a page assuming a specific profile setting, without actually changing the profile. To answer this question, we expected the participant would examine the adaptation explanation which shows how the content would be affected based on a change to the profile. Participant 3, who had not yet discovered the adaptation explanation, was not able to determine the expected page content. Participant 4 had previously demonstrated an understanding of the adaptation explanation but did not realise that the facility could be used to predict the content of the page in this case. The final question in this section asked participants to change their profile to see if their expectation of how the page would appear was correct. This would demonstrate the ability to change the profile to achieve a desired effect on adaptation. All participants were able to change their profile as required, except Participant 1, who could not work out how to access the profile.

The final row shows the time taken for the full experiment. Participant 1 took about twice as long as most of the other participants, who took 20 to 31 minutes. Since this included the time to complete the worksheet and to think-aloud, it is not indicative of the time needed to explore the way material is adapted.

Overall, Participants 2, 4 and 5 understood how to interpret the adaptation explanations and were able to use the explanations to control the inclusion/exclusion of content to a desired effect. Participant 1 understood the adaptation explanation but could not work out how to change the adaptation. Participant 3 did not discover the adaptation explanation until late in the evaluation but then demonstrated understanding of it.

A key concern, highlighted by the experiment, was that users had great difficulty finding the adaptation explanation. Participants 1, 2 and 3 required help to find the link *How was this page adapted to you?* Participant 4 found this link without help. Due to a browser anomaly, Participant 5 was presented with the adaptation explanation without the user having to click the link.

5. Discussion and Conclusion

The purpose of a *scrutable* adaptive hypertext is to ensure that the user can scrutinise it to visualize, understand and control the adaptation. This paper has described the Tutor3 system. It complements previous work on open user models, by taking the next step, and supporting openness of the process of adaptation to present the results of adaptation and a way of altering the adaptation. From the user's point of view, Tutor3 enables them to thoroughly explore the *implications* of the user model.

At this stage, Tutor3 has an extremely simple student model. It is entirely defined by the student's answers to the questions on a profile page. That page is presented at the beginning of each login to Tutor. It is also available from every Tutor page. On the one hand, this is a weakness, since the processes for constructing the student model are so restricted. However, at this stage, this simple model makes a logical starting point for exploring ways to effective support for users to scrutinise the adaptation of a system. It seems unwise to proceed to more complex underlying student models until we can demonstrate that we can support scrutable adaptivity in relation to an intuitive and simple underlying student modelling process.

We have reported a small, qualitative evaluation of the scrutability support. Our most striking finding is that it is quite difficult to support scrutability. This may be, in part, because the notion is new. Tutor3's scrutability support appears to be quite good, with some reservations. It seems that users need help in getting started. A simple tutorial session should help. Of course, this will only help if students do it! At the same time, our evaluation indicates that once students make their way to the Adaptation Evaluation page, they can understand how to scrutinise the core adaptation elements: the role of the profile, the material that has been included/excluded, what caused that adaptation and how to alter it.

References

1. Brusilovsky, P. (1996). Methods and Techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction, Vol. 6, n 2-3, pp. 87-129, Kluwer academic publishers.
2. Brusilovsky, P. (2001). Adaptive Hypermedia. User Modeling and User-Adapted Interaction, Vol. 11, n 1-2, pp. 87-110, Kluwer academic publishers.
3. Bull, S. P Brna, H Pain (1995). Extending the scope of the student model, User Modeling and User-Adapted Interaction, 5(1), 44-65.
4. Czarkowski, M and J Kay, (2000) Bringing scrutability to adaptive hypertext teaching, ITS'2000, Intelligent Tutoring Systems, Gauthier, G, C Frasson, K VanLehn (eds), Intelligent Tutoring Systems, Springer, 423--432.
5. Czarkowski, M and Kay, (2002) A scrutable adaptive hypertext, De Bra, P, P Brusilovsky, R Conejo (eds), Proceedings of AH'2002, Adaptive Hypertext 2002, Springer, 384 - 387.
6. De Bra, P., Aerts, A., Smits, D., Stash, N. (2002). AHA! Version 2.0, More Adaptation Flexibility for Authors. Proceedings of the AACE ELearn'2002 conference, October 2002, pp. 240-246.
7. Dimitrova, V., Self, J.A. and Brna, P. (1999). The interactive maintenance of open learner models, International Conference on AIED, Le Mans, France, 1999, 405-412.
8. Kay, J. (1995). The um toolkit for cooperative user modelling, User Modeling and User-Adapted Interaction, 4(3), 149-196.
9. Nielsen, J. (1994). Estimating the number of subjects needed for a thinking aloud test, International Journal of Human-Computer Studies, 41(1-6), 385-397.

10. Self, J.A. (1990). Bypassing the intractable problem of student modelling, in C. Frasson and G. Gauthier (eds.), Intelligent Tutoring Systems: at the Crossroads of Artificial Intelligence and Education, Ablex, Norwood, N.J., 107-123.
11. Self, J.A. (1999). Open sesame?: fifteen variations on the theme of openness in learning environments, International Journal of Artificial Intelligence in Education, 10, 1020-1029.

An Empirical Study of the Effect of Coherent and Tailored Document Delivery as an Interface to Organizational Websites

Cécile Paris, Mingfang Wu, Anne-Marie Vercoustre, Stephen Wan, Peter Wilkins, Ross Wilkinson
CSIRO, Australia

Abstract:

Increasingly, people find information about corporations through their web sites, as opposed to reading printed brochures. Typically, the only tools they have to find that information is searching and browsing. But can we do better with a modest user model and coherent delivery? Can we provide, through the web, more relevant and coherent information about a corporation, information specially tailored to specific clients and presented in a useful and understandable manner? Our conjecture was that a tailored hypermedia system offering tailored and coherent information would indeed provide a good alternative to the current search-and-browse facilities, also allowing corporations to be more responsive to their clients' needs.

In this paper, we present a preliminary study aimed at testing this hypothesis. We designed an experiment aimed at investigating the effect of delivering coherent and tailored information in response to user's query, in comparison to delivering the results of conventional search engines. We employed our experimental system, PERCY, which exploits information retrieval, user modeling and discourse-oriented technologies to retrieve and select parts of existing documents from CSIRO website and assemble them back into a coherent and tailored webpage. Our experiment consisted of asking users to evaluate and compare the results produced by PERCY and those produced by a typical search engine, given a query about the some project or research topic at CSIRO. Our experimental results indicate that users preferred the output of Percy, as it attracted more their attention and they understood the results better.

Keywords:

information retrieval, discourse model, evaluation, tailored delivery, coherent delivery

1. Introduction

Increasingly, people find information about corporations through their web sites, as opposed to reading printed brochures. Typically, the only tools they have to find that information is searching and browsing. It is quite often hard for visitors with very different backgrounds and goals to readily find the information they need, especially when they are not familiar with the corporate organization or its website structure. This is because search results are usually neither tailored to the visitors' needs, nor presented in a way that is easily understandable and recognizable to the visitor.

Can we do better with a modest user model and coherent delivery? Can we provide, through the web, more relevant and coherent information about a corporation, information specially tailored to specific clients and presented in a useful and understandable manner? While tailoring of search results has been exploited in the past [5,6,7], the effect of coherent delivery has not. Yet, the content *and* the presentation of the information to be delivered are equally important. On the one hand, content needs to be relevant and targeted to the advocated community. On the other hand, the organization and appearance of the presentation should help its understanding by the reader and enhance its usefulness for the reader. From the point of view of the author (or the corporation), it should also facilitate the delivery of the intended message to the audience in a clear and succinct way.

Within this context, our conjecture was that there might be a more effective way to provide the information that users needed. In particular, we believed that producing a coherent document about the relevant information would be more easily understood and more useful to the users. Our conjecture was thus that a tailored hypermedia system offering tailored and coherent information would indeed provide a good alternative to the current search-and-browse facilities, allow corporations to be more responsive to their clients' needs, and also provide the right information to the right people.

We observed that most corporations (including ours) typically have paper-based brochures intended to inform the readers of the work performed within the organizations. (CSIRO, for example, has a set of brochures about the different research areas and projects in the organizations.) These brochures form coherent documents that not only provide the relevant information but also introduce the corporation to the readers, provide additional information when appropriate (e.g., contact details). There are two additional important aspects of typical corporate brochures: first, the information is not simply listed as bullet points but rather unfolds in a natural progression of topics. Second, a corporation usually prepares a set of different brochures to send different messages to the different targeted communities. Taking our own organization -- CSIRO (Commonwealth Scientific and Industrial Research Organization) -- as an example: to the general public, we would like to send the message that CSIRO is doing good science and is beneficial to the community; to the university students, we would like to say that CSIRO is an excellent organization with which to study and work; finally, to the business/market developers, we would like to help them to identify which CSIRO research and development outcomes may have potential business interests, and how CSIRO could transfer these outcomes to their business applications. CSIRO thus has different sets of brochures, each presenting the appropriate message while also addressing the needs of the targeted audience.

We hypothesized that such a way of presenting the information would also be useful as an interface to a website instead of the traditional search-and-browse facility. Our system, PERCY, thus aims at delivering such brochure-like documents as a result of a query. Because PERCY generates these documents automatically on the fly, it can actually produce brochures which are much more tailored to the individual needs than paper-based brochures can be. The approach proposed in PERCY is to capture not only a visitor's potential information needs but also the information about the visitor’s personal background, so that the retrieved documents (or parts of documents) can be filtered and tailored. The information can then be organized and presented according to *discourse rules*. With this delivery method, we aim to provide the visitors with more appropriate and readable information than the output of a conventional search engine.

In this paper, we present an experiment we performed to validate this hypothesis. The remainder of this paper is structured as follows. First, we introduce PERCY, our tailored hypermedia system that produces tailored and coherent information about a specific topic in the context of a company web site. We then present our experiment, which aimed at evaluating the value of such delivery system. We conclude with a short discussion of the results.

2. PERCY

Information retrieval is a proven technology for searching large document collections. However, the search result is usually a list of discrete units that are not easy for users to assimilate. In contrast, the discourse-oriented technologies for adaptive hypermedia systems aim at ensuring that the result is coherent and appropriate through explicit text planning or manual authoring. In our work, we have been designing a new approach to tailored information delivery by integrating both technologies [10, 11]. As shown in Figure 1, this approach exploits a user profile, a set of discourse rules (which represent how one presents information in a particular domain) and a set of web data sources. Given a user query, the system uses the discourse rules and the user model to decide which information to present and how to present it in a coherent way.[1] PERCY is a specialized application embodying this approach. As previously mentioned, it produces coherent brochure-like documents about a corporation, personalized to the user's queries and user type. Our current prototype uses CSIRO as the test corporation.

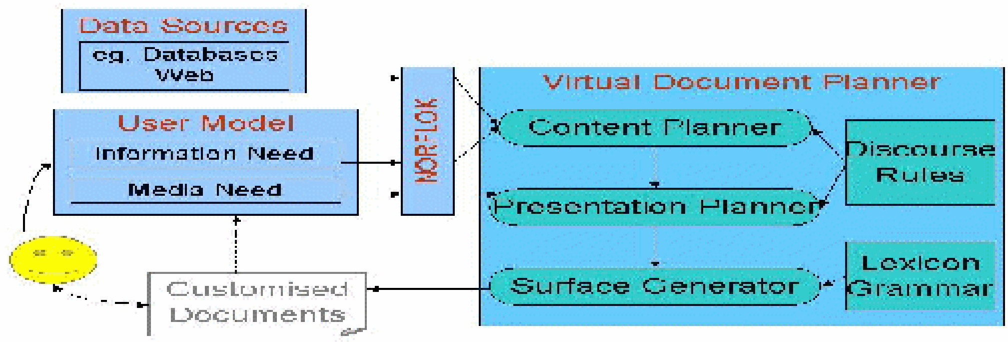


Fig. 1. The System Architecture of PERCY

The first task for PERCY is to obtain a visitor's profile through a web form. The profile is then stored in the user model for information retrieval, filter and presentation planning. Based on a requirements analysis performed in conjunction with professional science communicators, business development personnel and our website managers, we have identified for CSIRO the following characteristics as being important:

Stereotypical information:

- Current job position (e.g., a scientist or a research student or a CEO);
- Industry sector (e.g., science, engineering, finance);

Individual information:

- Interests (e.g., data mining, collaborative technology), which forms the initial query sent to the system; and
- Preferred delivery channel (e.g., web page, hand-held device).

1Our approach also allows for the delivery of the information on a variety of delivery channels, including paper, hand-held devices and the web [2].

The stereotypical information is used at the high level to decide what kinds of information to deliver. For example, a scientist and a CEO may use the same set of keywords to describe their interests when they visit the CSIRO website. However the research scientist may want to know more about the research activities and technical publications, while the CEO may want to know the potential business opportunities. The information about "industry sector" is important to guide PERCY to focus on the most relevant activities in the appropriate sector whenever possible. The individual information is then exploited to decide what information to deliver and how to deliver it. The information about "delivery channel" helps PERCY ensure the document produced is appropriate in its layout to the delivery platform (e.g., a web screen is different from a printout). We will discuss only web-based delivery in this paper.

Given a query about a topic, the virtual document planner (VDP) first creates a discourse tree by calling the content planner (or discourse planner). The discourse tree represents which information is to be included in the text and how it should be organized. It uses metadata to retrieve relevant portions from existing webpages. The discourse planner, modeled on the one described by Moore and Paris [9], uses a library of discourse rules (or plans). These indicate how a discourse goal can be achieved, and how various segments are related to each other to form a coherent whole[2]. The VDP then calls a presentation planner, which decides how best to express the discourse tree. The presentation planner decides on the exact amount of the information to be included in the document and, if necessary, the navigation needed. It also decides upon a presentation suitable for the medium chosen by the user, and augments and annotates the discourse tree appropriately. Finally, the surface generator traverses the final discourse tree to generate the text, delivering information retrieved from web sources, generating new sentences when appropriate, and, finally, producing device specific layout tags.

The discourse rules used by PERCY were produced from a study of corporate brochures. The document which results from PERCY thus typically starts with a brief introduction of the corporation. Then a summary of the search results is presented. This summary explicitly states which query terms were matched or not matched. To improve the search results, PERCY exploits a thesaurus. The final document specifies the synonyms that were added as keywords to the search, to ensure the user is not confused by the results. Finally, summaries of specific projects relevant to the query are given, together with contact information. When the document is a hypertext presentation, links to details are included. Overall, the structure of the document produced reflects that of a paper brochure of the corporation.

Figures 2 and 3 show part of PERCY's output for two queries[3]. Example A is generated for a *scientist* who is interested in *language technology* and in the *science* sector. Example B is generated for a CEO who is also interested in *language technology*, but in the *tourism* sector. Comparing the two examples, we can see that Example A contains more descriptive information about research topics, and that it includes selected publications. On the other hand, Example B gives information about how CSIRO relates to industry. Both examples present the selected information in a coherent manner, reflecting the structure of a conventional brochure, which is a natural (and expected) way to present information about a corporation.

2The Rhetorical Structure Theory (RST) [8] proposed by Mann and Thompson is used here to represent coherency amongst text segments.

3Because of space limitation, Figure 2 and Figure 3 show only the top two matched projects.

3. The Evaluation

3.1 The hypothesis

A visitor gets information from a corporate website either through browsing or searching. PERCY is intended to complement these two mechanisms. With the PERCY system, a visitor can quickly get a bird's eye view of an interesting topic, then can either search the website with a more focused topic, or browse the website through the links suggested by PERCY.

We want to evaluate how useful the PERCY is as a personalized corporate information delivery mechanism, compared with the normal browsing and searching mechanisms. As the effectiveness and speed of obtaining information by browsing largely depends on the usability of that website, comparing PERCY with the website would make the result inclusive. In this experiment, we chose to investigate whether users would *prefer* a tailored and coherent delivery mechanism over the conventional search engine results, and, in that case, we also wanted to try to identify why this may be so.

3.2 Evaluation criteria

There are three important indicators that reflect the effect and effectiveness of an information delivery mechanism for a corporation. The first indicator is a visitor's interactive experience with the delivered information. The second indicator is the visitor's attitudes towards the received message. And the third indicator is the further interactions with the corporation caused or influenced by the visitor's attitude and belief. The third indicator is indirect, and hard to test in a laboratory setting. In our experiment, we concentrated on the first two indicators.

The first two indicators are measured through two sets of questionnaires to assess the visitors' satisfaction with the delivered content (i.e. the quality of the tailored content) and the presentation format [3] (i.e. the quality of the presentation), the visitor's intention (whether subjects have intention to get further information), and preference. Given two systems being evaluated and compared, one questionnaire (the post-system questionnaire) gets the subjects' attitude independently on each test system, while the other questionnaire (the exit questionnaire) gets the subjects' attitude of the second system compared with the first system.

3.3 Experimental setting

Test systems The two test systems are the CMIS (CSIRO, Mathematics and Information Science) website search engine (P@NOPTIC) [1, 4] and PERCY used on the web. To a user, PERCY is very much like a search engine. The only difference is that the user sends only a query to a search engine, while PERCY requires users to provide minimal amount of personal information in addition to the query.

Subjects Only one type of users (university students) was considered in this experiment. So the stereotypical information for all subjects was set as research students from science sector. During the experiment, these students were asked to play the role of research students, interested in the four given research topics. For each topic, subjects were required to find out what CMIS is doing in that area, who are the key players, and other research related issues. They were allowed a maximum of 5 minutes for each topic.

Twenty subjects were recruited. Nineteen of them were university students (from the second year and above) and the other one was a fresh graduate working in a research environment. All subjects were from the computer science department of an Australia university. The average age of the subjects was 21.65. On average, they had 5.33 years online searching experience; their familiarity with CSIRO and CMIS was very low (1.95 and 1.6 on average respectively, on a five-point Likert scale). Fourteen of our subjects had never searched any CSIRO's website before; two of them had searched the CMISO's website for summer studentship position and staff profile, and they claimed that they had obtained what they had searched for.

Experimental design Four search topics (or queries) were selected from a wide range of CMIS research topics, namely: human computer interaction, mathematic modeling, image analysis and language technology; on the assumption that our subjects would have general knowledge of these topics. As we focused on testing the effect of the delivery, the query for each topic was fixed. No mechanism for providing relevance feedback or for supplying a new query was provided; subjects were restricted to explore the two different types of deliveries. While subjects were interacting with a delivery, they were allowed to browse away from the presented interface to any degree.

To reduce the effect of appearing order of systems and topics, the Latin-square experimental design was adopted in this experiment, as shown in Table 1. In this design, four test topics are divided into two blocks (B1 and B2). Topic blocks and systems are then rotated. Each subject uses each system on one block of topics. Subjects are required to fill in an entry questionnaire, a post-system questionnaire (refer to Appendix I) after each system, and an exit questionnaire (refer to Appendix II) at the end.

Table 1. Experimental design

Subjects	**System, Topic**	
No. 1-5<	Percy, B1	Search, B2
No. 6-10	Percy, B2	Search, B1
No. 11-15	Search, B1	Percy, B2
No. 16-20	Search, B2	Percy, B1

3.4 Experimental results

We used two questionnaires: a post-system questionnaire and an exit-questionnaire. The post-system questionnaire was to test whether the users obtained the information they desired, in a form that was useful, and whether the output of the system served as a good introduction to what the organization was doing on a specific topic. The post-system questionnaire included 9 questions, which meant to evaluate the systems with respect to the content of the information provided, its format, its coherence in terms of presentation, and, finally, whether based on what they obtained, they would like to have further interactions. The exit questionnaire was to elicit users' preferences between the two systems. We first present the results from the post-system questionnaire, followed by those of the exit questionnaire.

3.4.1 Results from the post-system questionnaire

The results from the subjects' responses to post-system questionnaire are shown in Table 2. Each question in this questionnaire measured certain aspects of the delivered information. (See the Appendix for the questionnaires themselves.)

Table 2. The average of the responses to the post-system questionnaire across all subjects

	Q1	Q2	Q3	Q4	Q5	Q6	Q7	Q8	Q9
Search	3.50	3.60	3.35	3.70	3.65	3.50	2.75	3.95	3.15
Percy	3.65	3.30	3.65	3.55	4.00	3.80	3.50	3.90	2.90
p <	NS	NS	NS	NS	NS	NS	0.06	NS	NS

In general, there were no significant differences[4] between two systems on all questions in the post-system questionnaire. However, PERCY tended to get higher scores for the coherence of the presentation of the information, especially a strong tendency appears for question 7 ("I think the presented information serves well as a useful online brochure"), which meant to measure coherence and readiness of the information delivered. This is encouraging and may point to the fact that users liked obtaining information in a coherent and well presented manner. We examine the results in more detail below.

Content Questions 1-4 measured the content of the delivered information. For questions 1 ("The system provides sufficient information") and 3 ("The information provided by the system meets my need"), PERCY got higher scores than the search system. For questions 2 ("The system provides sufficient information") and 4 ("The system provides me comprehensive information"), the search engine got higher scores. It is hard to interpret these 2 results, especially as there are no significant differences. The higher scores for the search engines for Questions 2 and 4 may be related to the amount of information presented: the search engine presents every piece of information related to a project, while PERCY selects only the most important information given the query and the user.

Format Questions 5 ("The structure of the presented information is clear to me") and 6 ("I think the presented information is organized in a useful format") measured the format of the delivered information. For these two questions, PERCY got higher scores than the search system. While we can notice a tendency,, we cannot derive strong conclusions yet as the differences are not significant.

Question 7 ("I think the presented information serves well as a useful online brochure") provides a measure for both content and format, and was meant to validate whether the delivery was an appropriate introduction or interface to a corporation. Again, the PERCY system has the higher score than the search system. The difference is significant at 0.06. While this may be because subjects were likely to compare the delivery with their own model (or expectation) of a brochure, it provides validation for the use of personalized and coherent delivery as an interface to a corporation's website.

Intention Questions 8 ("I would like to get more information on some specific projects presented") and 9 ("I would like to get more information because I have not got the needed information") were aimed at understanding the subjects" desired further actions. Both systems rated the same on question 8. For question 9, however, more subjects from the search system thought they want to get more information because they had not obtained the needed information. This is curious and interesting since, at the same time, subjects gave higher scores to the search system for questions 2 and 4, as we saw earlier. These seemingly contradictory results may indicate that the *amount* of information presented is not related to the *quality* of the information.

4All significant tests are based on the two tailed t-test.

3.4.2 Results from the exit questionnaire

Besides the post-system questionnaire, subjects were also required to fill in an exit-questionnaire based on their experience of the two systems. This exit-questionnaire was to evaluate their preferences, since it was based on their experiences with both systems. In contrast, the post-system questionnaire was purely based on their experience of that particular system. (Again see the appendix for the questionnaire.)

Preference The subjects' responses to the exit-questionnaire are listed in Table 3. They show that subjects ranked PERCY higher than the search engine for all questions. Notice that for question 6, a lower score indicates more focused delivery. Significant differences are found for questions 4 ("The information delivered by the second system attracts my attention better"), 5 ("The second system provides a better explanation on why a piece of information is presented") and 7 ("Overall, I prefer to use the second system as an online brochure of the searched topic").

Table 3. The average of the responses to the exit-questionnaire across all subjects

	Q1	**Q2**	**Q3**	**Q4**	**Q5**	**Q6**	**Q7**
Search	2.70	3.00	3.10	2.80	3.30	3.50	2.70
Percy	3.70	3.60	3.50	4.20	4.20	3.40	4.00
p <	NS	NS	NS	0.01	0.03	NS	0.04

4. Discussion

It is interesting to note that the two systems were evaluated as very similar in terms of the quality and quantity of information delivered (as shown for the results of the post-system questionnaires). Yet, subjects did prefer the system that tailored information to their needs *and* presented content in a coherent fashion. Such a kind of information delivery mechanism allowed the subjects to focus their attention better on their interested topics, to understand better why a piece of information is presented, and finally, to have a better overview of the organization's projects in the focused area. The study thus suggests that producing coherent and tailored hypermedia based on a user stereotype is an effective way to deliver information about an organization, probably more effective than the traditional search and browse mechanisms. We believe that, as the amount of information available for users to look at increases, such a means of delivery could in fact become increasingly valuable, as the need for focused attention (and fewer distractions) increases.

The tailored hypermedia we produced necessitated only a small user model and a retrieval engine coupled to a discourse planner. Yet this enables our system delivers tailored information in a coherent manner (in this case, resembling a corporate brochure), which was found useful by the users in comparison to the output of a conventional search engine. This approach thus provides an attractive interface to an organization's website.

The experiment is of course still limited, in terms of the number of subjects, their types, and the scope. For example, while subjects judged that PERCY attracted their attention better, whether this increased focus can generate further interaction with the corporation or the corporate website needs to be explored further.

References

1. CSIRO, http://www.panopticsearch.com
2. Colineau, N. and Wan, S. (2001) Mobile delivery of customized information using Natural Language Generation. In *Monitor* (Special Issue on Wireless Communication Special), 26 (3), September-November 2001. pp 27 - 31.
3. Davis, F. D. (1989). Perceived Usefulness, Perceived Ease of Use, and User Acceptance of Information Technology. *MIS Quarterly (13),* 1989, pp. 319-340.
4. Hawking, D., Bailey, P. and Craswell, N. (2000). Efficient and Flexible Search Using Text and Metadata. CSIRO Mathematical and Information Sciences, TR2000-83.
5. Korfhage, R. R. (1984) Query Enhancement by User Profiles. In *Proceedings of the 7th annual international ACM SIGIR conference on Research and development in information retrieval.* Cambridge, England. pp. 111-121.
6. Laine-Cruzel, S., Lafouge, T., Lardy, J.P. and Abdallah, B. N. (1996). Improving Information Retrieval By Combining User Profile and Document Segmentation. *Information Processing & Management* Vol. 32(3), pp. 305-315.
7. MacKinnon, L. & Wilson, M. (1996) User Modelling for Information Retrieval from Multidatabases. In C Stephanidis (Ed.) *Proc. 2nd ERCIM Workshop on "User Interfaces for All"*. Prague, Czech Republic, November 1996.
8. Mann, W. C. and Thompson, S. A. (1988). Rhetorical Structure Theory: Toward a functional theory of text organization. In *Text* 8(3): 243-281.
9. Moore, J. D. and Paris, C. L. (1993). Planning Text for Advisory Dialogues: Capturing Intentional and Rhetorical Information. In *Computational Linguistics*, Cambridge, MA. Vol 19(4), 651-694.
10. Paris, C., Wan S., Wilkinson R. and Wu M. Generating Personal Travel Guides – and who wants them? (2001). In *Proc. of the International Conference on User Modeling (UM2001)*; Sonthofen, Germany, July13-18, 2001.
11. Wilkinson R, Lu S, Paradis F, Paris C, Wan S and Wu M. (2000). Generating Personal Travel Guides from Discourse Plans. In *Proc. of International Conference on Adaptive Hypermedia and Adaptive Web-based Systems*. Trento, Italy, August 2000.

Appendix

Appendix I – Post–system questionnaire

Q1: The system provides me accurate information. 1-5

1	2	3	4	5
Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree

Q2: The system provides sufficient information
Q3: The information provided by the system meets my need.
Q4: The system provides me comprehensive information.
Q5. The structure of the presented information is clear to me.
Q6: I think the presented information is organized in a useful format.
Q7: I think the presented information serves well as a useful online brochure.
Q8: I would like to get more information on some specific projects presented.
Q9: I would like to get more information because I have not got the needed information.

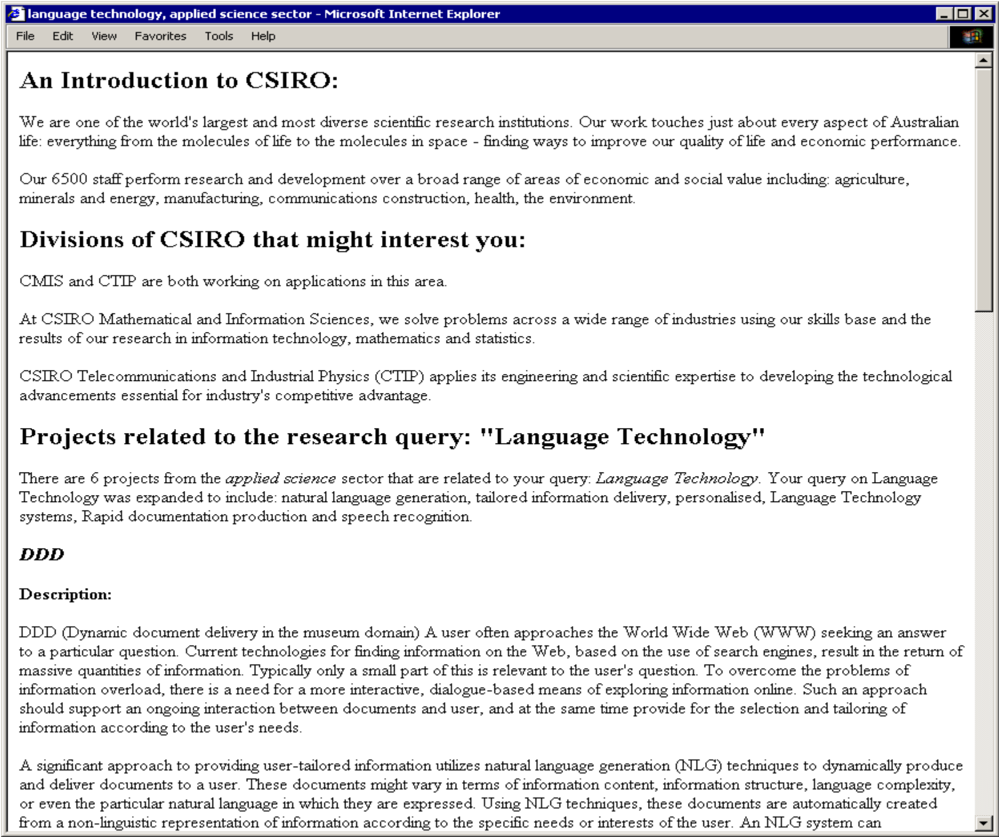
Appendix II – Exit questionnaire

You have just used two different systems that deliver the searched information in two different forms. In answering each the following questions, please use the **first system** as your basis of comparison, i.e. how good is the **second system**, as compared to the first one, in the following aspects.

Q1: The second system enables me to focus on the search topic better.

1	2	3	4	5
Strongly disagree	Disagree	Neither agree nor disagree	Agree	Strongly agree

Q2: The second system makes my search task easier.
Q3: The second system delivers the higher quality information.
Q4: The information delivered by the second system attracts my attention better.
Q5: The second system provides a better explanation on why a piece of information is presented.
Q6: The second system provides more comprehensive information.
Q7: Overall, I prefer to use the second system as an online brochure of the searched topic.



language technology, applied science sector - Microsoft Internet Explorer

An Introduction to CSIRO:

We are one of the world's largest and most diverse scientific research institutions. Our work touches just about every aspect of Australian life: everything from the molecules of life to the molecules in space - finding ways to improve our quality of life and economic performance.

Our 6500 staff perform research and development over a broad range of areas of economic and social value including: agriculture, minerals and energy, manufacturing, communications construction, health, the environment.

Divisions of CSIRO that might interest you:

CMIS and CTIP are both working on applications in this area.

At CSIRO Mathematical and Information Sciences, we solve problems across a wide range of industries using our skills base and the results of our research in information technology, mathematics and statistics.

CSIRO Telecommunications and Industrial Physics (CTIP) applies its engineering and scientific expertise to developing the technological advancements essential for industry's competitive advantage.

Projects related to the research query: "Language Technology"

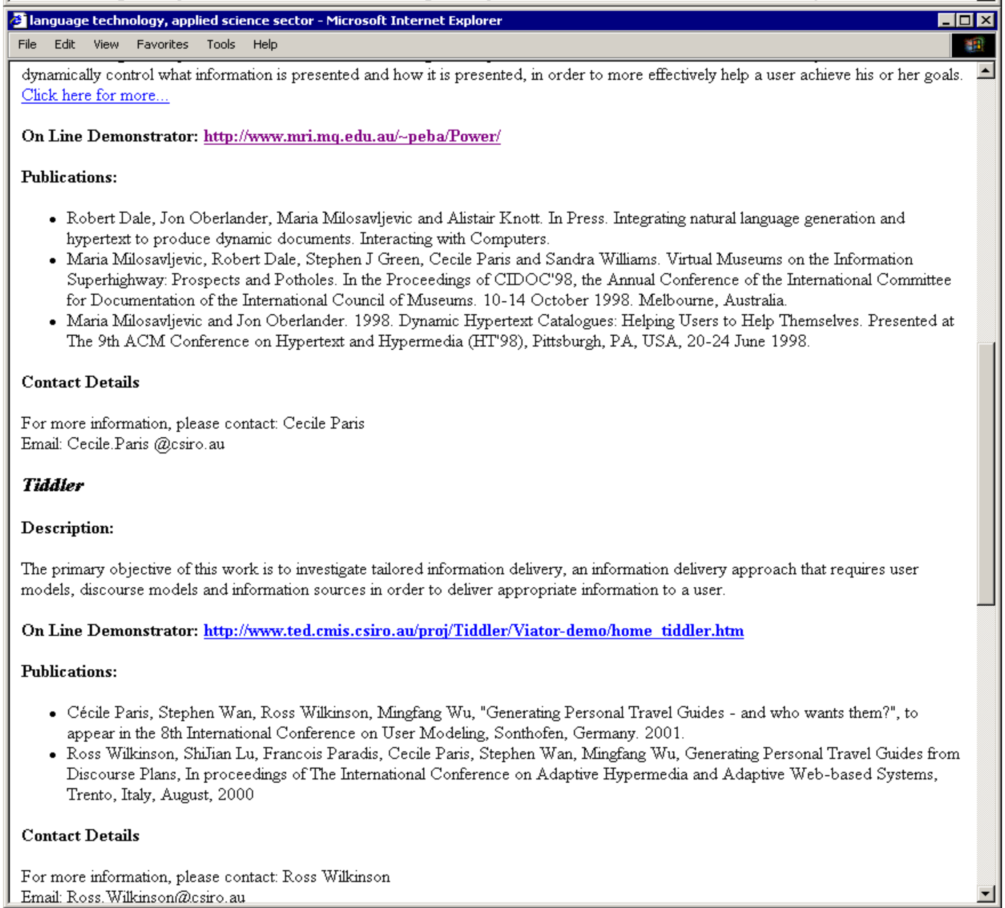
There are 6 projects from the *applied science* sector that are related to your query: *Language Technology*. Your query on Language Technology was expanded to include: natural language generation, tailored information delivery, personalised, Language Technology systems, Rapid documentation production and speech recognition.

DDD

Description:

DDD (Dynamic document delivery in the museum domain) A user often approaches the World Wide Web (WWW) seeking an answer to a particular question. Current technologies for finding information on the Web, based on the use of search engines, result in the return of massive quantities of information. Typically only a small part of this is relevant to the user's question. To overcome the problems of information overload, there is a need for a more interactive, dialogue-based means of exploring information online. Such an approach should support an ongoing interaction between documents and user, and at the same time provide for the selection and tailoring of information according to the user's needs.

A significant approach to providing user-tailored information utilizes natural language generation (NLG) techniques to dynamically produce and deliver documents to a user. These documents might vary in terms of information content, information structure, language complexity, or even the particular natural language in which they are expressed. Using NLG techniques, these documents are automatically created from a non-linguistic representation of information according to the specific needs or interests of the user. An NLG system can



language technology, applied science sector - Microsoft Internet Explorer

dynamically control what information is presented and how it is presented, in order to more effectively help a user achieve his or her goals.
Click here for more...

On Line Demonstrator: http://www.mri.mq.edu.au/~peba/Power/

Publications:

- Robert Dale, Jon Oberlander, Maria Milosavljevic and Alistair Knott. In Press. Integrating natural language generation and hypertext to produce dynamic documents. Interacting with Computers.
- Maria Milosavljevic, Robert Dale, Stephen J Green, Cecile Paris and Sandra Williams. Virtual Museums on the Information Superhighway: Prospects and Potholes. In the Proceedings of CIDOC'98, the Annual Conference of the International Committee for Documentation of the International Council of Museums. 10-14 October 1998. Melbourne, Australia.
- Maria Milosavljevic and Jon Oberlander. 1998. Dynamic Hypertext Catalogues: Helping Users to Help Themselves. Presented at The 9th ACM Conference on Hypertext and Hypermedia (HT'98), Pittsburgh, PA, USA, 20-24 June 1998.

Contact Details

For more information, please contact: Cecile Paris
Email: Cecile.Paris @csiro.au

Tiddler

Description:

The primary objective of this work is to investigate tailored information delivery, an information delivery approach that requires user models, discourse models and information sources in order to deliver appropriate information to a user.

On Line Demonstrator: http://www.ted.cmis.csiro.au/proj/Tiddler/Viator-demo/home_tiddler.htm

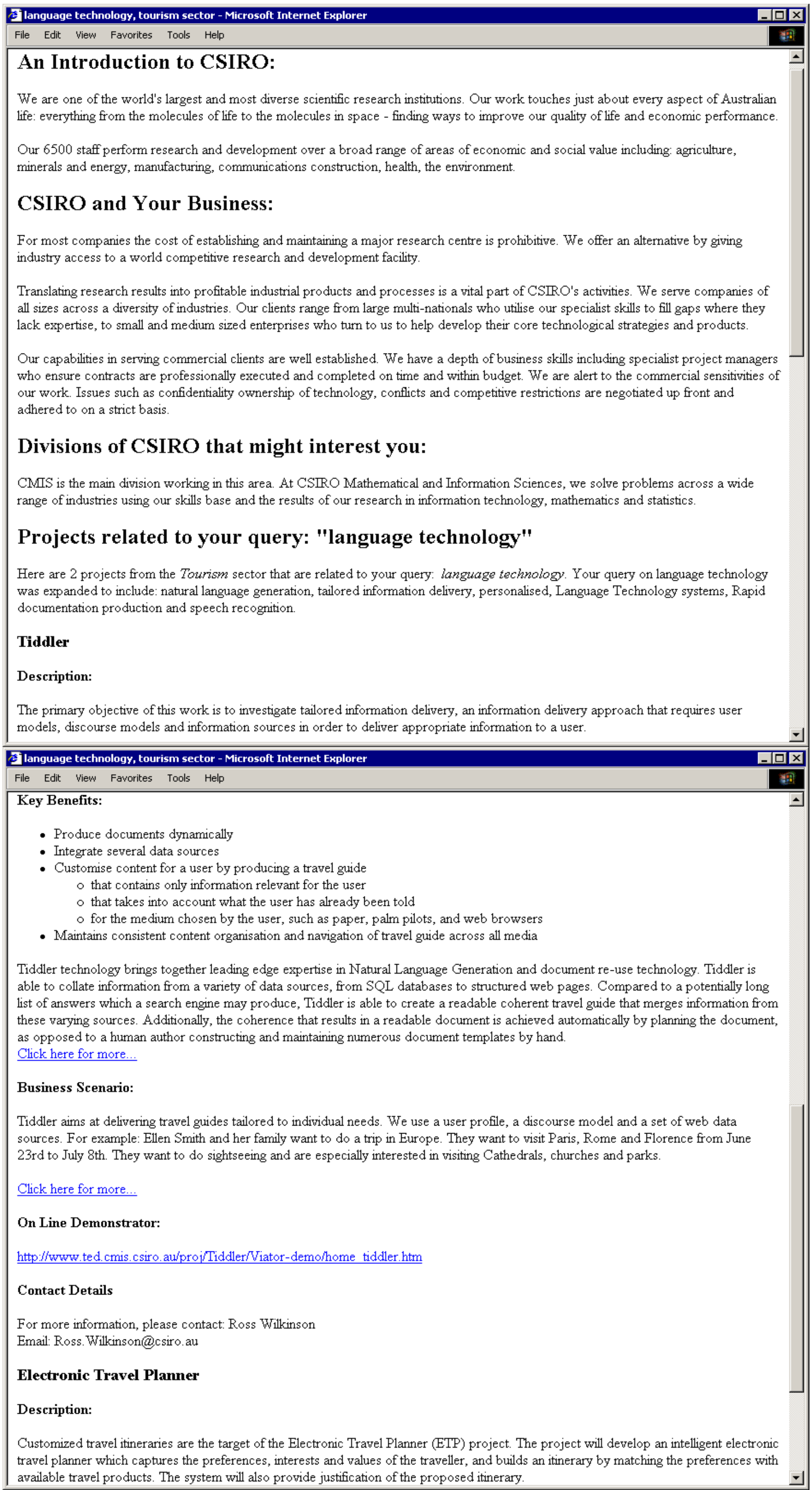
Publications:

- Cécile Paris, Stephen Wan, Ross Wilkinson, Mingfang Wu, "Generating Personal Travel Guides - and who wants them?", to appear in the 8th International Conference on User Modeling, Sonthofen, Germany. 2001.
- Ross Wilkinson, ShiJian Lu, Francois Paradis, Cecile Paris, Stephen Wan, Mingfang Wu, Generating Personal Travel Guides from Discourse Plans, In proceedings of The International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, Trento, Italy, August, 2000

Contact Details

For more information, please contact: Ross Wilkinson
Email: Ross.Wilkinson@csiro.au

Fig. 2. Example A: the delivery for a scientist, with query in Science sector



language technology, tourism sector - Microsoft Internet Explorer

File Edit View Favorites Tools Help

An Introduction to CSIRO:

We are one of the world's largest and most diverse scientific research institutions. Our work touches just about every aspect of Australian life: everything from the molecules of life to the molecules in space - finding ways to improve our quality of life and economic performance.

Our 6500 staff perform research and development over a broad range of areas of economic and social value including: agriculture, minerals and energy, manufacturing, communications construction, health, the environment.

CSIRO and Your Business:

For most companies the cost of establishing and maintaining a major research centre is prohibitive. We offer an alternative by giving industry access to a world competitive research and development facility.

Translating research results into profitable industrial products and processes is a vital part of CSIRO's activities. We serve companies of all sizes across a diversity of industries. Our clients range from large multi-nationals who utilise our specialist skills to fill gaps where they lack expertise, to small and medium sized enterprises who turn to us to help develop their core technological strategies and products.

Our capabilities in serving commercial clients are well established. We have a depth of business skills including specialist project managers who ensure contracts are professionally executed and completed on time and within budget. We are alert to the commercial sensitivities of our work. Issues such as confidentiality ownership of technology, conflicts and competitive restrictions are negotiated up front and adhered to on a strict basis.

Divisions of CSIRO that might interest you:

CMIS is the main division working in this area. At CSIRO Mathematical and Information Sciences, we solve problems across a wide range of industries using our skills base and the results of our research in information technology, mathematics and statistics.

Projects related to your query: "language technology"

Here are 2 projects from the *Tourism* sector that are related to your query: *language technology*. Your query on language technology was expanded to include: natural language generation, tailored information delivery, personalised, Language Technology systems, Rapid documentation production and speech recognition.

Tiddler

Description:

The primary objective of this work is to investigate tailored information delivery, an information delivery approach that requires user models, discourse models and information sources in order to deliver appropriate information to a user.

language technology, tourism sector - Microsoft Internet Explorer

File Edit View Favorites Tools Help

Key Benefits:

- Produce documents dynamically
- Integrate several data sources
- Customise content for a user by producing a travel guide
    - that contains only information relevant for the user
    - that takes into account what the user has already been told
    - for the medium chosen by the user, such as paper, palm pilots, and web browsers
- Maintains consistent content organisation and navigation of travel guide across all media

Tiddler technology brings together leading edge expertise in Natural Language Generation and document re-use technology. Tiddler is able to collate information from a variety of data sources, from SQL databases to structured web pages. Compared to a potentially long list of answers which a search engine may produce, Tiddler is able to create a readable coherent travel guide that merges information from these varying sources. Additionally, the coherence that results in a readable document is achieved automatically by planning the document, as opposed to a human author constructing and maintaining numerous document templates by hand.
Click here for more...

Business Scenario:

Tiddler aims at delivering travel guides tailored to individual needs. We use a user profile, a discourse model and a set of web data sources. For example: Ellen Smith and her family want to do a trip in Europe. They want to visit Paris, Rome and Florence from June 23rd to July 8th. They want to do sightseeing and are especially interested in visiting Cathedrals, churches and parks.

Click here for more...

On Line Demonstrator:

http://www.ted.cmis.csiro.au/proj/Tiddler/Viator-demo/home_tiddler.htm

Contact Details

For more information, please contact: Ross Wilkinson
Email: Ross.Wilkinson@csiro.au

Electronic Travel Planner

Description:

Customized travel itineraries are the target of the Electronic Travel Planner (ETP) project. The project will develop an intelligent electronic travel planner which captures the preferences, interests and values of the traveller, and builds an itinerary by matching the preferences with available travel products. The system will also provide justification of the proposed itinerary.

Fig. 3. Example B: the delivery for a CEO, with query in the Tourism sector

ILASH: Incorporating learning strategies in hypermedia

Namira Bajraktarevic, Wendy Hall
Intelligence, Agents, Multimedia Group
Dept of Electronics and Computer Science
University of Southampton
{nb99r, wh}@ecs.soton.ac.uk

Patrick Fullick
Research and Graduate School of Education
Southampton
plf@soton.ac.uk

Abstract

This paper describes the main ideas behind the adaptation and architecture of an educational hypermedia system called ILASH that incorporates learning strategies into hypermedia. The paper describes a computational framework designed to provide adaptive support for learning by using ILASH. The framework includes different adaptive techniques embedded in the text and link structure to scaffold and encourage the use of strategies and a dynamic student model. The pedagogical rationale underlying the study and adaptive features employed in the system are presented.

1. Introduction

Adaptive hypermedia systems make an attempt at scaffolding learners, by adapting and matching the content presentation and interaction to their goals. Many such systems tend to be designed in such a way that supports the learner interaction. They facilitate the activities of the learner by adapting to some learner preferences. However, there is a need to acknowledge the importance of supporting the learners in applying their effective learning strategies. Incorporating the most effective strategies into hypermedia systems takes the instructional designers one step closer to developing learning resources that match the learner's profile more closely.

Individuals tend to develop learning strategies in order to deal with learning materials and therefore these strategies can be regarded as "cognitive tools, which enable students to complete tasks and solve problems" (McLoughlin, 1999). Weinstein and Mayer (1986) defined learning strategies as "behaviours and thoughts that a learner engages in during learning". The most effective learners will use multiple strategies to ensure that they monitor their comprehension. Frequent comprehension checks are an important part of an effective learning process. This is particularly important in a hypermedia environment, where students can get easily distracted and "lose coherence of what they are reading" (Foltz, 1996). Applying an inappropriate learning strategy or not knowing how to apply a learning strategy, may prove a big stumbling block for some students and is likely to hamper their comprehension. Foltz (1996) points out that "a model of hypertext comprehension must consider both the information the reader gains from the text and how that information can affect the readers choice of strategies for proceeding through the text". Jonassen (1988) suggests extending the learner's cognitive approaches to learning through adaptive, intelligent use of computer courseware and learning materials.

The consequences of learning strategy differences have not been pursued in the field of adaptive educational hypermedia. According to Hammond (1993) basic hypermedia systems may fail to provide students with the support, direction and engagement that learning requires. This failure suggests implications for the design of hypemedia-based learning and introduction of adaptivity. With an adaptive hypermedia-based system we can individualize the learning process and allow students to apply a learning strategy that is proving to be the most effective for a given task. The success of adaptive systems can be measured if they can cognitively adapt to the student. The adaptive system can attempt to emulate actual processes employed by the students for effective learning processes.

The focus of adaptation in ILASH is to provide a representation of an appropriate strategy for students while learning, whether it is summarising or questioning.

2. Learning strategies

The reviewed literature shows that a large number of learning strategies have now been identified and teachers have been encouraged to enable students to use a variety of learning strategies in secondary education. A number of definitions of learning strategies have been used in the field of educational psychology. The research has acknowledged the importance of learning strategies and many studies examined their efficacy. There is a need to take into account the fact that students differ and that these differences and preferences in learning need to be taken into account. One method or learning strategy may be ineffective for some students who could learn more effectively using a different strategy (McKeachie, 1995). Jonassen (1998) suggested that learning strategies could be embedded inside a hypermedia program. Effective learning involves knowing when to use a specific strategy, how to access that particular strategy, as well as when to abandon an ineffective strategy (Jones, Palinscar *et al.* 1987). Having said that, many students are not aware of what strategies work for them. Some students may experience difficulty in selecting the main idea or the concept and supporting details. They treat each sentence with the same importance. "Learning will be easiest when there is a strong correlation between the way in which new material is presented to us and our learning preferences. Conversely, we find learning more difficult when there is a large disparity between our learning strategy preference and the supplied learning presentation. Styles are fixed but strategies are adaptable processes we can use to respond to the demands of a learning situation" (Laing 2001). According to Nist and Simpson (2000), research validated strategies are small in number, however, extensive research for the past two decades indicates that some of the strategies for constructing meaning are more significant than others (Dole et al., 1991). A previous study has been conducted to examine which learning strategies students use (McLeod, Heiko and Lockwood 1998). It was found that many students made good use of 'higher order' learning strategies such as "questioning" and "summarisation" strategies.

One of the definitions of learning strategies used by the students was described by Garner (1987), who suggested that 'text summarization is a tool for making a cognitive process and for monitoring it. As a cognitive strategy, it allows learners to synthesize information from multiple sources and diverse perspectives". Research indicated that embedding learning strategies in software is effective. The students using the software with the learning strategies embedded, performed better (Thornburg and Pea, 1991). Hsiao (1997) conducted a combined study in which learning strategies were embedded into hypermedia based system (such as note-taking, summarisation, reflective questions). They also embedded prompts to encourage strategy-use, combined with the application of field–dependent and field-independent cognitive styles. The "**summarizing**" strategy (S) provides the opportunity to identify, paraphrase, and integrate important information. The "**questioning**" (Q) strategy enables students to generate questions and identify the kind of information that is significant enough that it could provide the substance for a question. Students pose this information in a question form and self-test themselves to ascertain that they can indeed answer their own question. While studies in effective learning strategies continue to emerge, the relevance of applying these strategies in the field of adaptive hypermedia learning has not been determined. Bull (2000) created an adaptive system that recommended individual language learning strategies in order to help students become more effective learners. The model combined representations of learning styles and current strategy use. A new strategy was recommended based on the information obtained from the student model. The recommended strategies were from cognitive, metacognitive and memory domain.

3. System description

The system contains courseware targeted at GCSE[1]-level students. The Physics courseware has been adapted from Fullick (2001). The chapters chosen for the study contain scientific concepts, principles, and theories that are used to explain observations of the natural world. The first (adaptive) session contains the courseware on "The behaviour of waves" and the second (non-adaptive) session contains the courseware on "The Solar System". Each session contains the same number of pages, and the student's knowledge is accessed at the end of each lesson. As part of the system usage, the students browse the adaptive session first and then complete the post test, followed by a non adaptive session and second post-test. The post tests contain the same number of questions, and they are tied to lesson objectives and three levels of Bloom's taxonomy (1956)[2]. The results of the post tests between adaptive and non adaptive sessions are compared.

Page layout

The basic structure of the page layout is that the pages are divided into two formats: S_type (corresponding to the "summarising strategy" and Q_type (corresponding to the "questioning" strategy). Three factors were viewed as essential and sufficient to design the layout of an environment conducive to studying: the learning strategy, the text and link presentation and the structural signals. The S_type pages have a top-down approach where the material is presented with key-points summarised at the end of each page. The S_type page presentation provides contextual clues to help students with getting the gist of information (by using headings, diving text in small chunks etc). The aim is to provide the students with some elements of a summarising strategy.

The Q-type pages have a question asked after each paragraph (which contains an explanation of a concept). Arburn and Bethel (1998) suggest that directing the attention to deliberate questioning activities may encourage the students to confront misconceptions which they have grown comfortable with, so that in resolving their discrepancies more meaningful learning may occur.

Adaptive features of the system

The system allows the user interface, linking and content structure to change according to student's knowledge state. Student's recall and understanding of content is continuously checked and an appropriate strategy is selected. The adaptive techniques used in ILASH are adaptive presentation (adaptive layout presentation) and adaptive navigation support[3] (adaptive annotation and hiding of links) in the table of contents and the adaptive side bar (See Figure 1 and 2). For adaptive presentation a set of pedagogical rules of knowledge prerequisites is created, that determines which layout and which pages should be presented.

These rules also determine which 'additional information' should be presented along with a concept and which 'examples' should be shown. Students are prevented from jumping to pages for which they lack prerequisite knowledge. (The pages that describe concepts are divided into prerequisite concepts by the author). The case is similar for the additional material related to the concepts (such as 'examples', 'science people' sections, 'interesting facts', 'ideas' etc.). Some pages have examples of concepts associated with them and some do not. The links to the pages that the student is 'not ready to learn' become hidden and a 'cross' icon is placed next to them. The links to lessons that the student is ready to cover are displayed in the table of contents with a 'green tick' icon next to them. The percentage of completed material is also displayed in the table of contents. Previously viewed chapters, currently available pages and newly available links to chapters are presented in the adaptive side bar (See Figure 2).

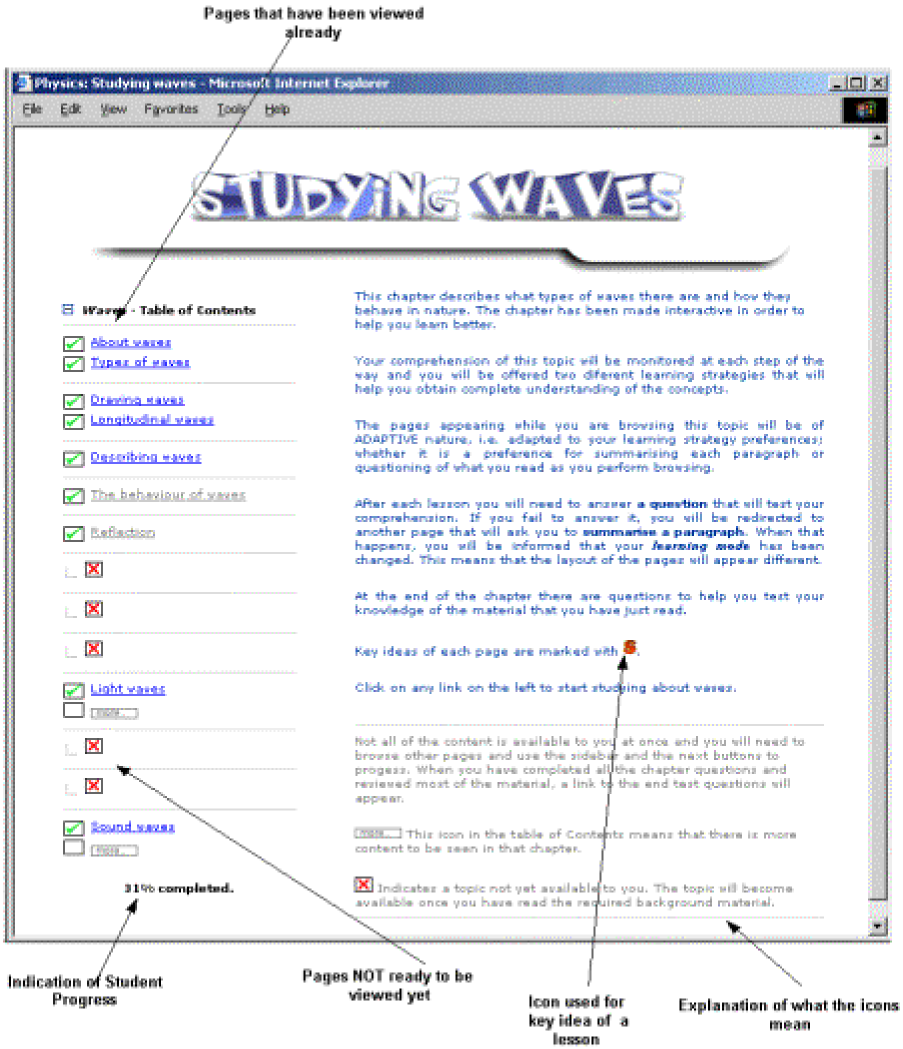


Fig. 1. User interface with adaptive table of contents

Student model (SM)

The student model is used to adapt the display characteristics of the interface and the appropriate learning strategy, to the needs of the student. Student's interaction is reflected immediately in the system and in the learning strategy selection. The knowledge that the student has attained is collected through direct questioning methods. The student model is dynamically updated and triggers the system to select the most appropriate learning strategy for each lesson. The student model contains the following information: *Student _ID, History of visited links, time spent on each page, learning strategy preference and the number of switches between the two strategies.*

Adaptation algorithm

The following algorithm was used to determine the student model in relation to the student's knowledge:

For each of the lessons

If score after each lesson is correct

THEN

The Learning strategy is preferred by the student, keep on using pages with that preference

ELSE

The Learning strategy is NOT preferred, a different strategy needed

ENDIF

The system monitors the history of visited pages and supplies strategies at different points. The students are free to 'jump' between paragraphs. The technique follows a pattern of supplying different strategies adaptively, depending on student's progress. In the adaptive version of the system, each page had the following links available in the adaptive sidebar:

- *Index*, takes the student to the table of contents page, providing an overview of the lessons
- *Search*, allows the student to find clarification for keywords
- *Examples*, takes students to an example of a concept
- *Science people*, presenting more detailed information on the scientist mentioned in a lesson
- *Ideas and evidence,* presents the key points summarised for the concept
- *Back,* taking students to the previous page
- *Next,* with a description on what the next page is about
- *Glossary* links, providing students with definitions of terminology used in the content
- *Adaptive chapter links*: these links change in the adaptive part of the system to describe previously visited links, currently viewed links and next links available for viewing.

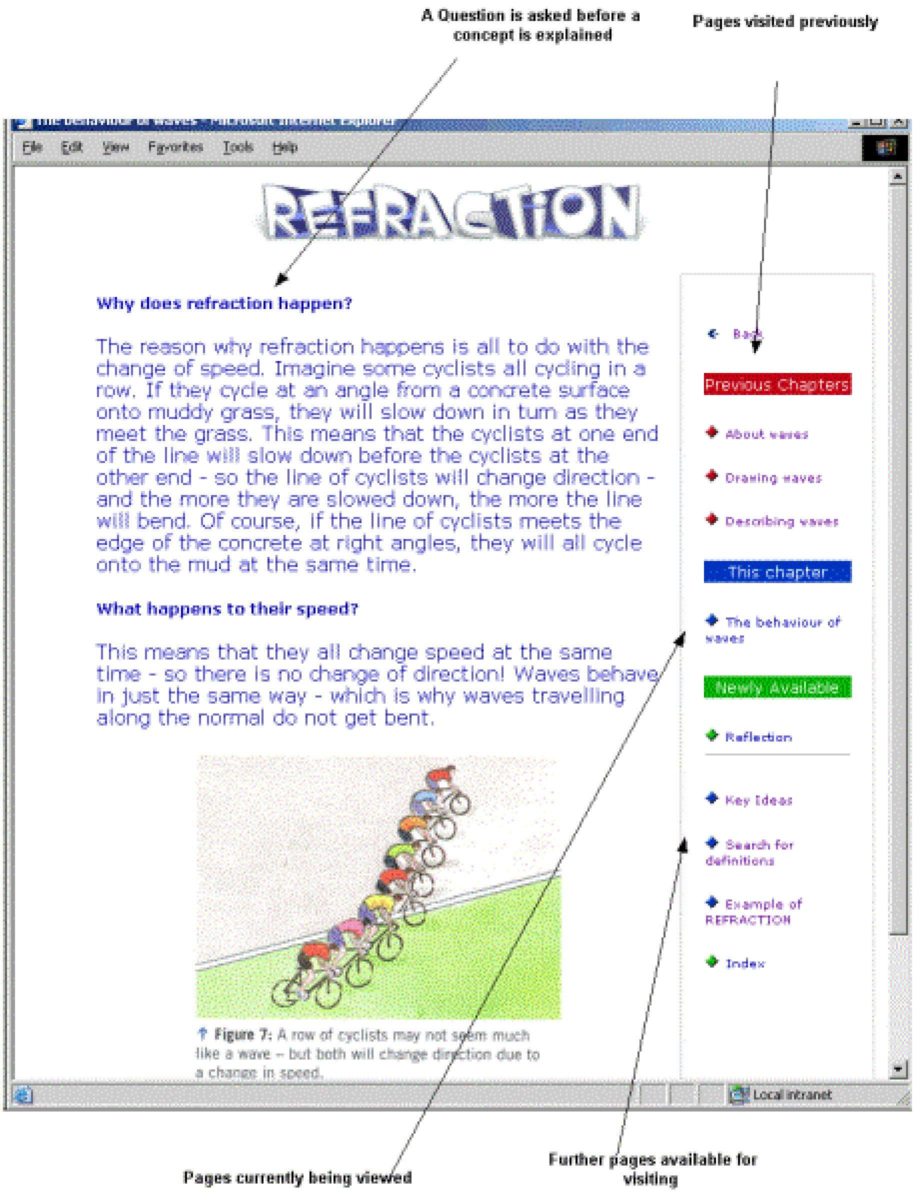


Fig. 2. Adaptive sidebar layout

(1) Adaptive session

In this first, adaptive part of the system the students log in and start browsing. The students are not able to see all the pages at first (See Figure 1). The links that the system provides become available as the student learns more. The students start browsing pages that embed "summarising" strategy elements (S_type page layout). At the end of a lesson the student is asked to summarise it ("summarising" strategy check) and the student fails to provide a correct answer, then a different learning strategy ("questioning" strategy) is provided (the Q_type page layout). At the end of that lesson, when a strategy check point is reached, and if the student fails to answer, then the students can continue to browse the lesson, but the links to the pages they can browse are restricted until a concept is mastered. The post-test is presented after the students have completed 75% of studying.

(2) Non-adaptive session

This is the second part of the system where students re-log in. This version of hypermedia courseware offers students unrestricted navigation throughout the lessons. The students can apply whatever learning strategy they wish. Summaries of key points are provided in the non–adaptive side bar (See Figure 3). The student's comprehension is tested after each lesson, but no clues are given if they provided an incorrect answer. At the end of the chapter they take another post-test. The student's behaviour is monitored and the history of links is logged. Questions asked at the end of adaptive and non-adaptive parts of the system follow Bloom's taxonomy of educational objectives on knowledge, comprehension and synthesis of information. The questions were created so that the students were asked to demonstrate that they fulfil these conditions by being asked to demonstrate these cognitive levels by having to define, match and classify information, as well as to describe and explain concepts in their own words.

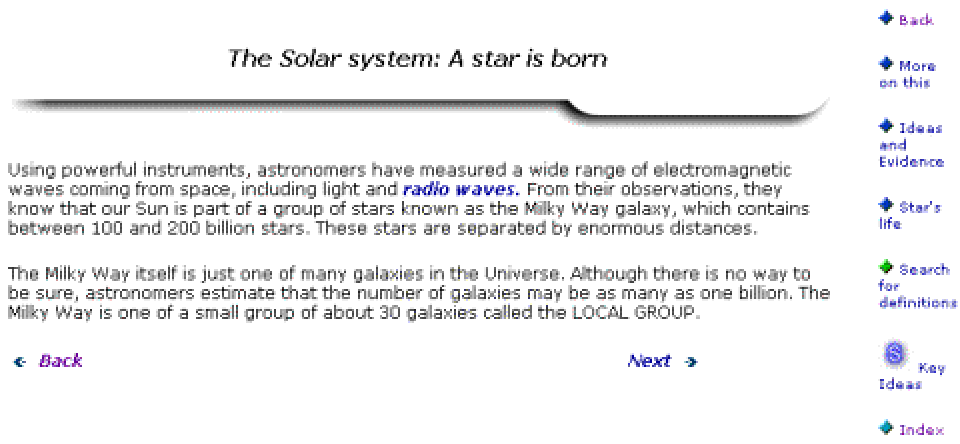


Fig. 3. Non adaptive layout

4. Technologies used in the experiment

For this study the following technologies were used: The PHP scripting language, MySQL, a relational database engine, Apache web server, HTML, XML (Extensible Markup language) and XSL (Extensible stylesheet language) technologies. PHP is a server side scripting language that can be embedded in html documents. This provides an easy way to incorporate dynamic content within what was previously a static document. Also, PHP is well suited for reading information from web forms and maintaining sessions between pages. This is important in order to keep the name of the currently authenticated student, their browsing actions and history in the student model. Sessions are used to maintain student-specific information. XML and XSL allow content to be separated from the presentation, where XML is used to store the content and XSL is used to present pages with different layouts.

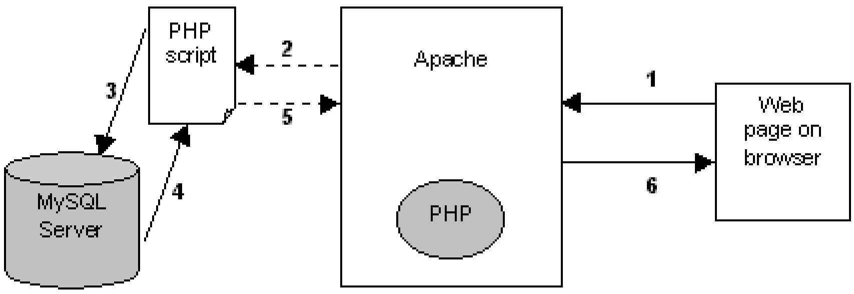


Fig. 4. Architecture of the system

Figure 4 shows the system architecture. A student requests a particular page from the Web server by means of a Web browser (1). In response to this request the Web server calls a PHP script (2), which is executed by the PHP preprocessor (3), pulling data from the database (4). The results are processed by the rest of the PHP script (5) and turned into HTML, which is returned to the student's browser (6). The current implementation of the system is as shown above, using Apache server. The student model is stored in a MySQL database. The courseware is stored in XML files. The students need to use a browser capable of parsing XML and XSL. No specific client side software needs to be installed on the client's PC.

5. Conclusion and future work

A review of the literature indicated that there is a paucity of investigations concerning the application of learning strategies in the field of adaptive educational hypermedia. In this study an adaptive hypermedia system has been created that provides the adaptation of learning strategies. This is achieved by applying adaptive presentation and navigational support. The aim of the study is to prove that by using adaptive features in hypermedia based educational systems, students' learning outcomes and comprehension can be enhanced. The system adaptively scaffolds students and allows them to apply effective learning strategies. In terms of specific learning strategies, the study hopes to find significant improvement in a student's achievement following an adaptive lesson. The use of the aforementioned strategies may provide students with the tools to enhance their success in hypermedia based studying. The system is currently being evaluated.

References

1. Arburn T. and Bethel L.M (1998) Teaching strategies designed to assist community college science student's critical thinking, The University of Texas, Austin.
2. Bloom, B.S, Englehart, M.B., Furst, E.J., Hill, W.H. and Kratwohl, O.R. (1956) *Taxonomy of educational objectives: The classification of educational goals, Handbook 1: The cognitive domain*. New York: Longman.
3. Brusilovsky, P. (1996) Methods and techniques of adaptive hypermedia. *Journal of User modelling and User adaptive interaction*, **6**, 2-3, pp. 87-129.
4. Bull, S. (2000) Individualised recommendations for learning strategy use, [online], available from http://www.eee.bham.ac.uk/bull/papers/ITS00_LS.htm, [Accessed June 2002]
5. Duke, N.K and Pearson, D. (1991) Effective individual techniques, [online], available from http://ed-web3.educ.msu.edu/pearson/pdppaper/Duke/ndpdp.html, [Accesses October 2002]
6. Dole, J. A., Duffy, G. G., Roehler, L. R., and Pearson, P. D. (1991). Moving from the old to the new: Research on reading comprehension instruction. Review of

Educational Research, **61**, pp. 239-264.).
7. Foltz, P.W. (1996) Comprehension, Coherence and Strategies in Hypertext and Linear text. In Rouet, J.F., Levonen, J.J., Dillon, A.P. and Spiro, R.J. (Eds.) *Hypertext and Cognition*. Hillsdale, Lawrence Erlbaum Associates.
8. Fullick, P. (2001) *AQA double award coordinated science–physics*, Oxford, Heinemann, pp. 40-77.
9. Garner, R. (1987) Metacognition and reading comprehension, Norwood, New Jersey, Ablex Publishing, pp.105-126.
10. Hammond, N. (1993) Learning with hypertext: problems, principles and prospects. On C. McKnight, A.Dillon, J..Richardson, Hypertext – A Psychological perspective, NY,: Ellis Horwood.
11. Hsiao, Yu-Ping (1997) The effects of cognitive styles and learning strategies in a hypermedia environment. a review of literature, [online], available from www.edb.utexas.edu/mmresearch/Students97/Hsiao, [Accessed June 2002]
12. Jonassen, D. H. (1988) Integrating learning strategies into courseware to facilitate deeper processing, In D. Jonassen (Eds.) Instructional designs for microcomputer courseware, Hillsdale, Lawrence Erlbaum Associates. pp.151-181.
13. Jones, B.F, Palinscar, S.A., Sederburg O.D., Carr, G.E. (Eds.) (1987) Strategic teaching and learning: Cognitive instruction in the content areas. Alexandria, V.A, Association for supervision and curriculum development.
14. Laing, M. (2001) Teaching learning and learning teaching: an introduction to learning styles, New Frontiers in Education XXXI/**4**, pp. 463-475.
15. Liu, M. and Reed, W. M. (1994) The relationship between the learning strategies and learning styles in a hypermedia environment, *Computers in Human Behaviour*, **10** (4), pp. 419-434.
16. McLoughlin, C. (1999) The implications of the research literature on learning styles for the design of instructional material, Australian Journal of Educational Technology, 1999, **15** (3), pp.222-241.
17. McKeachie, W.J. (1995) Learning styles can become learning strategies, NTFL Vol.4, **6**, [online], available from www.mtfl.com/html/pi/9511/article1.html, [Accessed October 2002].
18. McLeod, R. J., Daniel, H. and Lockwood, P. V. (1998). A study of learning strategies used by students with the Oz Soils Interactive Multimedia Program. In C. McBeath and R. Atkinson (Eds.), *Planning for Progress, Partnership and Profit*. Proceedings EdTech'98. Perth: Australian Society for Educational Technology. [online], available from http://cleo.murdoch.edu.au/gen/aset/confs/edtech98/pubs/articles/m/mcleod.html, [Accessed July 2002].
19. Nist, S.L. and Simpson M. (2000) College studying, in Handbook of reading research, Vol. **III**, [online], available from http://www.readingonline.org/articles/handbook/nist, [Accessed July 2002].
20. Palincsar, A.S. and Brown, A.L. (1984) Reciprocal teaching of comprehension-fostering and comprehension-monitoring activities. *Cognition and Instruction*, **1**(5), pp.117-175.
21. Thornburg, D.G. and Pea, R.D. (1991) Synthesizing instructional technologies and educational culture: Exploring cognition and metacognition in the social studies. Journal of Educational Computing Research, **7** (2), pp.121-164.
22. Thüring, M., Mannemann, J. and Haake, J. (1995) Hypermedia and cognition: Designing for comprehension, *Communications of the ACM*, **38**(8), pp. 57-66.
23. Welling L. and Thompson L., (2001) PHP and mySQL web development, SAMS publishing, USA.
24. Weinstein, C.E. and Mayer, R.E. (1986) The teaching of learning strategies. In M.

Wittrock (Ed.), *Handbook of research on teaching*, New York, Macmillan, pp. 315-327.

[1] GCSE denotes General Certificate of Secondary Education in the UK

[2] According to Bloom's cognitive domain taxonomy 'knowledge' is defined as an ability to acquire, to identify, to recognize knowledge of facts, specifics and abstractions and to recall previously learned information. 'Comprehension' implies the ability for translation, interpretation, extrapolation of meaning of information and understanding of the facts. 'Synthesis' is defined as an ability to discriminate, distinguish, reintegrate and organise the information and the relationships into a meaningful whole.

[3] See Brusilovsky (1996) for the definition of the techniques.

Automatic Authoring in the LAOS AHS Authoring Model

Alexandra I. Cristea
Faculty of Computer Science and Mathematics
Eindhoven University of Technology
Postbus 513, 5600 MB Eindhoven, The Netherlands
tel: +31-40-247 4350
a.i.cristea@tue.nl

Abstract

In this paper, we extend the automatic authoring techniques that can be built based on the LAOS model, a five-layer AHS authoring model. As the LAOS model itself is fairly complex, although information-rich, an adaptive hypermedia author needs a lot of system support to be able to populate all its levels with the corresponding information. Therefore, such automatic authoring techniques, which are actually automatic transformation (and interpretation) rules between the different layers of the model, have been designed. These automatic rules represent, in the area of adaptive systems, designer-goal oriented adaptation techniques. They should represent the goal of the designer that is authoring the hypermedia (such as the pedagogical goal in educational adaptive hypermedia). Therefore, this paper represents yet another step towards an adaptive hypermedia (or adaptive course) that 'writes itself'. The focus here is on automatic transformation between the *domain* and a newly introduced *goal and constraints model*, to show that the effort of introducing this new layer can be minimal.

Keywords

Adaptive authoring, adaptive hypermedia, AHS, AHAM, ontologies, MOT, Dexter model

1. Introduction

Adaptive hypermedia system (AHS) are becoming nowadays popular, due to their connection with the W3C and IEEE LTTF [18] movements towards (*ontology*-based) customization and the *semantic Web*[24]. The success of commercial adaptive systems as Firefly, or research AHS as AHA! [15], Interbook [4], TANGOW [6] and others has pushed AHS forward. Their edge over classical ITS systems relies on their simplicity: they contain a simple domain -, user model (usually an overlay - of the domain model), aimed at a quick response, which is extremely beneficial in the speed-concerned WWW environment. However, for quite a long while there has been a lack of powerful authoring tools for adaptive hypermedia [2, 7]. One of the main reasons was the great (but fruitful) diversity in AHS implementations, many with implicit models[27].

Here we build on AHAM [27] and on the LAOS model [12] that allows a more flexible model for adaptive hypermedia authoring. As authoring of information rich adaptive hypermedia is difficult and time-consuming, we have added, next to the LAOS model that allows high flexibility, some methods to bypass the workload for the adaptive hypermedia author. Here we show, for instance. (adaptive, adaptable) automatic authoring techniques that can lead to more powerful AHS authoring tools. Instead of having the author populate the layers of an adaptive hypermedia model such as LAOS, the system can take many of the tasks over and perform them with no or little authoring intervention. Here we are going to highlight some examples of such *automatic authoring*, as we call it.

The paper is organized as follows. In section 2 we briefly recall the LAOS model as well as the definitions we need for the automatic transformations. Section 3 introduces automatic transformations that can lead to designer adaptation: automatic adaptation of the designed hypermedia (e.g., courseware) itself to the designer goal. In section 4 we exemplify some automatic transformations between two concrete layers, the *domain* and the *goal and constraints layer*, that are allowed by the LAOS model, and compute some *flexibility degrees* to show the expressiveness of the possible transformations and give also some examples and implementation instances from MOT [13]. In section we present a short discussions about the benefits and implications of such automatic, designer goal oriented transformations. Finally, section 6 draws conclusions.

2. LAOS Layered Model

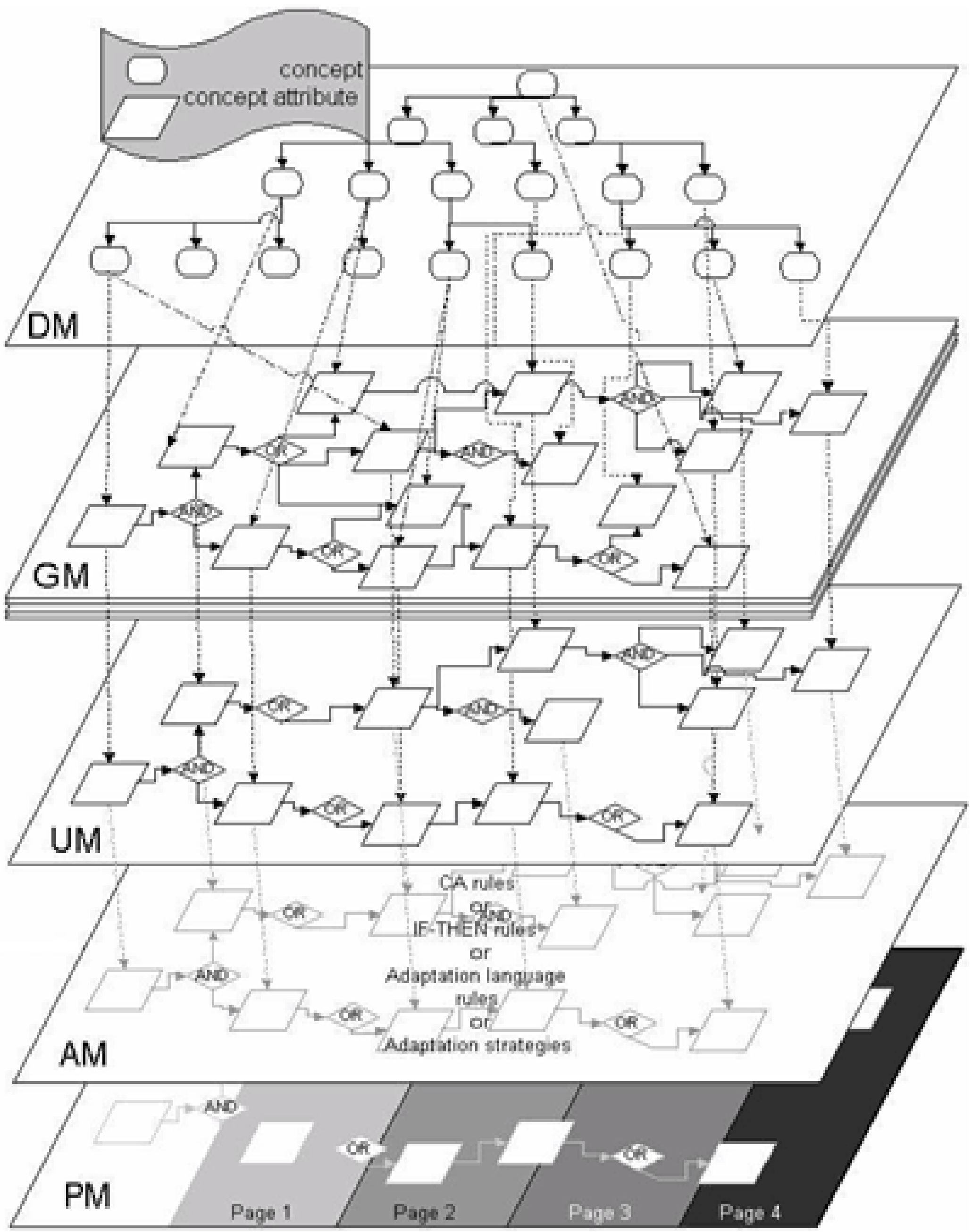


Fig. 1. The five level authoring model

The LAOS model (figure 1) is a generalized model for generic adaptive hypermedia authoring and was introduced in [12], therefore we are not going into many details about it here. It is based on the AHAM model [27] which in turn extends the Dexter reference model [17] for the specific field of adaptive hypermedia. The LAOS model is composed of five components: the *domain model* (DM), *goal and constraints model* (GM), *user model* (UM), *adaptation model* (AM) and *presentation model* (PM), as can be seen in Figure 1.

The idea is based on the book–course or book–presentation metaphor: generally speaking, when making a presentation (GM), be it for the Web or not, we base it on one or more references (DM). Simplifying, a presentation (GM) is based on one or more books (DM)[1]. This is why we need an intermediate presentation (GM) layer. The rest of the layers are shared with the AHAM model, so the motivation of using them is similar to the motivation of the previous model.

The basic idea is that such a model is easier to maintain than a small but compact model with all needed information in the same place. A change in user information will go, for instance, directly into the *user model* (and might influence the adaptation model) but has nothing to do with the domain – so will not influence the *domain* - or *goal and constraints model*. A presentation style change or update, on the other hand, will influence only the *goal and constraints model* (if it is a content related presentation style change) or only the *presentation model* (if it is a interface related change). So, each type of information is kept separately from information of other type, thus allowing maintainability.

Moreover, with the LAOS structure, dynamic (adaptive) presentation generation becomes possible. The actual presentation seen by the user can contain both elements of the *goal and constraints* - as well as elements of the *domain model* (e.g., for clarification of an explanation based on only the GM, the other elements/ objects of the respective concept, or the other concepts related to the current concept, can be referred, via a jump over one layer). This increases the flexibility and expressivity of the created adaptive presentations, as we shall see by computing the flexibility indexes of automatic transformations, for instance[2].

In the following we are going to list the definitions regarding the different layers that are going to be used in the automatic transformations.

2.1 Conceptual Layer

Definition 1. We consider a concept map *CM* of the AHS to be determined by the tuple <C,L>, where C represents the set of concepts and L the set of links ($CM \subseteq$CM, the set of all concept maps of the AHS).

Definition 2. A concept $c \in C$ is defined by the tuple $< A_c, C_c >$ where A_c ($A_c \neq \emptyset$) is a set of attributes and C_c a set of sub-concepts.

Definition 3. A_{min} is the minimal set of (standard) attributes[3] required for each concept to have ($A_c \supseteq A_{min}$).

Definition 4. A concept $c \in C$ is a composite concept if $C_c \neq \emptyset$.

Definition 5. A concept $c \in C$ is an atomic concept if $C_c = \emptyset$.

Definition 6. A link $l \in L$ is a tuple $<c1, c2, n_l, w_l>$ with $c1 \in Ci$, $c2 \in Cj$ *start* and *end* concepts, respectively, n_l a name or label of the link and w_l a weight of the link.

Definition 7. An attribute $a \in A_c$ is a tuple $<var, val>$, where *var* is the name of the attribute (variable or type) and *val* is the value (contents) of the attribute[4].

Definition 8. Each concept *c* must be involved at least in one link *l*. This special relation is called *hierarchical link* (or link to father concept). Exception: root concept.

Concept *c* is determined by its identification $i \in \{1,\ldots,C\}$ (where C=card(C)) and the attributes of concept *i* are $a_i[h]$, with $h \in \{1,\ldots,A_i\}$ and $A_i \geq A_{min}$(where A_i=card(A_c) and A_{min}=card(A_{min})).

2.2 Goal and Constrains Layer

The goal map *GM* of the AHS is a special *CM*, as follows.

Definition 9. A concept $c \in \mathcal{C}$ in *GM*is defined by the tuple $< A_c, C_c >$ where A_c (card(A_{min})=2)[5] is a set of attributes and C_c a set of sub-concepts.

Definition 10. A link $l \in L$ in *GM* is a tuple $<c1, c2, n_l, w_l>$ with $c1 \in C$, $c2 \in$ CM.C[6] *start* and *end* concepts, respectively, n_l a name representing the type (i.e., hierarchical or AND/OR connections) of the link and w_l a weight of the link.

2.3 User Model

UM and AM have been described relatively well by AHAM [26].

However, another way of representing the UM is given in [10], where we view the UM as a concept map (CM). In such a way, relations between the variables within the UM can be explicitly expressed as relations in the UM, and do not have to be “hidden” among adaptation rules (figure 2). The components of the concept-based user model are:

1. a concept map of user variables and their values (UVM)
2. a history concept map (HM): an overlay model of visited attributes and concepts from the GM and DM (a copy or a pointer to them) with extra historical variables and their respective values attached (e.g., a set of [date of visit, duration] for each visit)
3. a future concept map (FM): an overlay model of attributes and concepts from the GM and DM to be visited (copy or pointers to them); this map should be in the general case dynamic, i.e., its components (concepts, attributes, links, variables, values) can vary according to the AM(adaptation model) application by the AE (adaptation engine) and the user’s decisions.

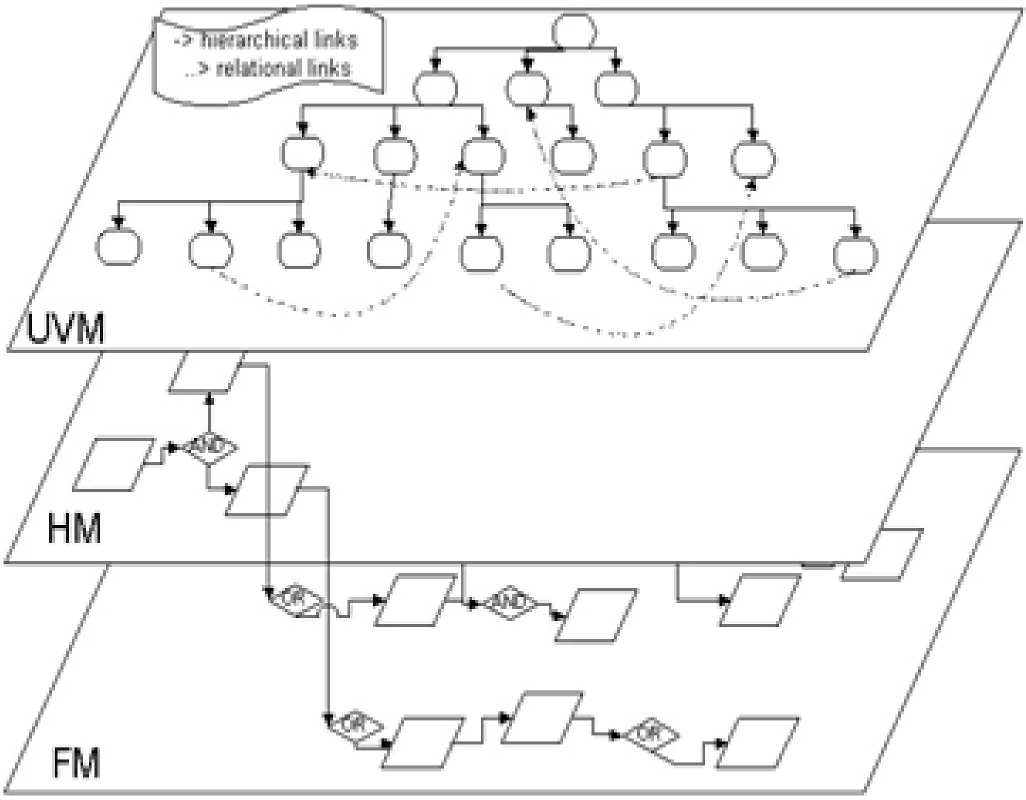


Fig. 2. The 3-leyered user model in LAOS

2.4 Adaptation Model

We have introduced in [9] a new three-layer adaptation model (defining *low level assembly-like adaptation language*, *medium level programming adaptation language* and *adaptation strategies language*) that we are in the process of refining and populating, but this is beyond the scope of the present paper.

2.5 Presentation Model

The PM has to take into consideration the physical properties and the environment of the presentation and provide the bridge to the actual code generation for the different platforms (e.g., HTML, SMIL [25]).

3. Adaptive, Adaptable Generation of Automatic Transformations

In [11] we have defined the notion of designer adaptation (and adaptability), as *adaptation (*and *adaptability) to* the *design (authoring) goals*. In this paper we elaborate on the different possibilities of implementing such adaptation, based on the LAOS adaptive hypermedia authoring model introduced in [12].

In the following, we will present some automatic transformation possibilities from one layer to the other of the LAOS model, which can be performed in exclusivity by the authoring system, triggered or not by the designer's (author's) specific request.

These processes are normally done by hand during adaptive hypermedia design and authoring, and many of them are considered to be domain dependent (and therefore embedded in the domain functionality).

Here we find some patterns that allow us to generalize and automatically perform these transformations. In the first case, we talk about *adaptable design generation*, whereas in the second, we can talk about *adaptive design generation*. A simple rule system, for instance, can be implemented to make the choice between adaptability and adaptivity, and then within adaptivity, among the different adaptive options[7] presented.

Adaptivity implies that the system makes the inferences about the possible choices, and then takes the decision that is conforming to its adaptation model. The system then executes the choice.

In *adaptability*, the inferences about possible choices, as well as the selections are made by the user, and the system then executes the choice.

However, a combined version of adaptivity and adaptability is possible. The system can make the inferences about possible choices, and then allow the user to make the selection (or decision). This we call *adaptive adaptability*.

Moreover, we look at the *flexibility index* of many of the automatic transformations presented in the following sections, defined as the combinatorial index giving all the functionalities that can be covered by such transformations.

4. From Domain Model to Goal and Constraints Model

This section discusses the automatic (adaptive, adaptable) *goal and constraints model* generation from the *domain model*, according to some presentation constraints and goals (e.g., pedagogical strategy or pedagogical technique). This transformation can be viewed as the first step from *information* to *knowledge*. This is due to the fact that, as said in [13] for instance, the *lesson[8] level* repeats the information contained in the concept level, now modeled and grouped based on pedagogical goals.

4.1 According to Concept Attribute Type

A very simple way of using the concept attributes can be for the selection of the specific types that are the only ones to go in the goal and constraints model. This transformation has been used for demonstration purposes by the MOT adaptive hypermedia authoring system [13].

I.e., for A_{min} *={title, keywords, introduction, text, explanation, pattern, conclusion}* (A_{min} *=7) as defined in section 2.1, we define* $A_{transf} \subseteq A_{min}$ *as* A_{transf} *={title, introduction}* (A_{transf} *=2), as the transfer set from DM to GM, for implementing a goal-constraints model representing the elements for the pedagogical goal "introductory presentation" (e.g., for a beginners course, or for an overview on the whole material).*

If A_{transf} *={title, explanation}* (A_{transf} *=2) we can implement a goal-constraints model representing the elements for the pedagogical goal "motivational presentation" (e.g., for a motivational overview on the whole material, that is to attract students towards it).*

As a third example, we mention the selection of A_{transf} *={title, text}* (A_{transf} *=2), with which we can implement a goal-constraints model for a "motivational presentation" (e.g., for a motivational overview on the whole material, to attract students towards it).*

As a fourth and last example, we mention the selection of A_{transf} ={title, keywords, pattern, conclusion} (A_{transf}=4) that implements a goal-constraints model representing the elements of the pedagogical goal "rehearsal" (e.g., for a summary or resuming presentation of the whole material, that is to remind students what they have learned in that lesson, and what the main important points – patterns - are).

Obviously, many more such pre-selections can be done. It is easier to imagine these types of transformations, if we go back to the book metaphor, and we are trying to construct a presentation based on a book: first, we are going to select the material that is going to be presented, as the whole book might be too long (hence, *constraints*) and its focus might be elsewhere (hence, *goal*). In this way, we build our *goal and constraints model*. Next, we are going to focus on the order, style, etc. of the presentation, which will appear in the *adaptation model* (and partially in the *presentation model*). Finally, we will interact with our audience and decide on skipping some parts or going into more details into others, depending on their reaction to our story (so, *user model* building and processing).

If we look at the combinatorics of these transformations, the *flexibility degree*, computing in this case the number of different lesson materials (so, the number of different sub-layers in the *goal and constraints model*) that can be generated automatically just with this simple procedure, is as follows. The different ways of selecting attributes from a concept C1 are:

$$flex(1) = \sum_{i=1}^{card(\mathcal{A}_{c1})} C(card(\mathcal{A}_{c1}), i) \geq \sum_{i=1}^{A_{max}} C(A_{min}, i) =$$
$$= \sum_{i=1}^{A_{max}} \frac{A_{min}!}{(A_{min} - i)!\,i!}$$

where *C(a,b)* are combinations of *a* elements taken *b* at a time.

This number is flex(1)≥87 for A_{min} =7 and represents the number of possible selections from C1 attributes if we don't care about the order. However, because in the *goal and constraints layer* the order starts being important, as opposed to the *domain layer*, the actual formula is:

$$flex(1) = \sum_{i=1}^{card(\mathcal{A}_{c1})} P(card(\mathcal{A}_{c1}), i) \geq \sum_{i=1}^{A_{max}} P(A_{min}, i) =$$
$$= \sum_{i=1}^{A_{max}} \frac{A_{min}!}{(A_{min} - i)!}$$

where *P(a,b)* are permutations of *a* elements taken *b* at a time.

So flex(1)≥ 13699, which is a much greater number.

Again, this is just the flexibility degree for one single concept and its extracted attributes. In a hypermedia concept map, there are many concepts. If we consider very simple automatic transformations, such as implemented in MOT [13], where all concepts in a concept map *C* are transformed in the same way, then these numbers don't change. However, if we allow concepts to be transformed independently, the flexibility degree will drastically grow.

4.2 According to Link Type

As said in the link definition for the conceptual layer (section 2.1), beside the obligatory hierarchical links, concepts can be involved in several other relations (links), which are defined by their *start point*, *end point*, *name* (type) and *weight*. In [11] we have shown that simply by using the attribute structure of the concepts, and labeling links between concepts with the name of the attribute that presents some relatedness, a great number of links can be automatically generated. However, in the LAOS structure we allow other types of links between concepts, which may not be automatically generated or related to attribute types.

These link types can be used to generate new, specific links at the level of the GM model.

A very simple example is the selection of some selected type of links only, that are to be taken over by the GM model (e.g., only *name* links).

In MOT, the automatic transformation functions described in section 4.1 go hand in hand with an automatic transformation into a standard, hierarchical, ordered link structure.

In other words, the selected attribute subset will keep (almost) the same *hierarchical structure* as its DM source: if a concept *C1* was a sub-concept of concept *C2*, and, let's say, we use the transformation of choosing only the {*title*,*introduction*} attributes; then, *L11=C1.title* and *L12=C1.introduction* will be sub-concepts of *L21=C2.title*, and the former attribute *C2.introduction* becomes concept *L22*, which is also a sub-concept of *L21*. In this way, the hierarchical link structure in *domain model* is translated into a hierarchical link structure of the *goal and constraints model*[9]:

$$LL21 \supseteq LL22, LL11, LL12$$

Moreover, concepts in the GM have an *order relation*, as opposed to concepts in the DM, which are represented as concept *sets* (so without order within a hierarchical depth). The solution implemented in MOT is to first list the (selected) attributes of a DM concept, and then the sub-concepts of the same concept. In our example case, this means that the order relationship is:

$$LL21 > LL22 > LL11 > LL12.$$

Finally, relations in the GM have a type, which can be hierarchical, as describe before, or {AND/OR}. The latter are relations between elements from the same hierarchical depth. Automatically, all elements at a certain hierarchical depth are transformed in the MOT GM into concepts connected via an 'AND' relation:

AND(LL21 , AND(LL22 , LL11 , LL12)).

These can then be manually altered (e.g., into 'OR' relations), and added weights, but we are not going into details about this here.

The link-based transformation above is very simple, taking into account just the hierarchical link relations in the DM, but it is useful to illustrate the many different types of links that can be generated in the GM from only such a simple link sub-set. Here, one hierarchical relationship (together with the implicit attribute relationship) at DM level generates 3 relationships at the GM level. Please note that the above transformations don't take into account the relatedness links. By using these relations we could design an extended version of the three GM links above, as follows.

If, for instance, in the above setting, concept *C1* is related (e.g., via a 'text' attribute relatedness relation) with *C3* (*link(C1,C3,'text','70%')*), and we write the new GM concepts resulting as *LL31=C3.title* and *LL32=C3.introduction*, then we could write an automatic transformation from *domain* - to *goal and constraints model* that would generate:

LL21⊒ LL22 , LL11, LL12,LL31,LL32

LL21 > LL22 > LL11,LL31 > LL12,LL32

AND(LL21 , AND(LL22 , LL11 , LL12,LL31,LL32)).

It is easy to see that this transformation would integrate in the introductory presentation also all related concepts.

4.3 Combination of Concept Attribute Type and Links

As previously said, MOT is already combining (a primitive version of) the above. However, much more complex and interesting combinations are possible.

The total number of possible combinations is obviously huge, as for each concept attribute type transformation there will be different possible link type transformations, making the total *flexibility degree* a *product of the independent flexibility degrees of the two transformation types*.

5. Discussion

In sections 3, 4 we have shown a small, illustrative number of different types of automatic (adaptive, adaptable) transformation possibilities that can be directly performed by the adaptive hypermedia authoring system, in order to make the task of the author easier. These transformations are based on the data design defined by the LAOS model, which allows a concept-oriented approach for data design, analysis and usage.

It is interesting to note the great number of different design possibilities these automatic functions permit, computed in the form of a *flexibility degree*, which shows also the range of the adaptivity of the final system.

Moreover, although only some example transformations from the *domain* – to the *goal and constraints model* have been discussed and analyzed here, more types are possible (such as *domain* - to *adaptation model*, *goal and constraints* - to *adaptation model*, etc.). In practice it is reasonable to expect that these transformations will be parallel. This combination of all transformations may be leading to a situation where one transformation may be setting some restrictions on another one, but most of the time, these multiple transformations together will generate an increased number of possible functionalities.

We have not extended all the examples or computed the flexibility degree from all the cases, as the space in the paper did not permit it. Instead, we have tried to give some flavor of the different possible automatic transformations, their applicability and their diversity.

6. Conclusion

In this paper we have introduced different possible automatic authoring techniques between two specific layers of LAOS, a five level AHS authoring model with a clear-cut separation of the representation levels: the *domain model* (DM), the *goal and constrain model* (GM), the *user model* (UM), the *adaptation model* (AM) and finally the *presentation model* (PM).

We have previously shown [1] that authoring of adaptive hypermedia is a difficult task, which might be the main impediment that keeps AHS from being wider spread. Therefore we have implemented some goal-oriented automatic authoring techniques in MOT [13] that have the role to help the AHS author and ease the authoring burden. The implementation in MOT is mainly for demonstration purposes at this stage, and has therefore to be further developed.

In the current paper we have worked at the design for such a development from a more general, partially theoretical point of view. We have given a few examples for the automatic transformations, we have introduced and computed the flexibility degree offered by such transformations, and we have discussed the significance and extension possibilities of such transformations.

In this way, we are gradually advancing towards adaptive hypermedia that 'writes itself', being therefore adaptive not only to the final AHS user, but also to its designer (or author).

Acknowledgements

This research is linked to the European Community Socrates Minerva project "Adaptivity and adaptability in ODL based on ICT" (project reference number 101144-CP-1-2002-NL-MINERVA-MPP).

References

1. Aroyo, L., Cristea, A.I., and Dicheva, D. A Layered Approach towards Domain Authoring Support. In Proceedings of ICAI 2002 (Las Vegas, US) CSREA Press.
2. Brusilovsky, P. Adaptive hypermedia, User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 2002, 87-110.
3. Brusilovsky, P., Schwarz, E., Weber, G. ELM-ART: An intelligent tutoring system on world wide web. In Proceedings of International Conference on Intelligent Tutoring Systems (ITS'96) (Montreal, Canada, June 1996), 261-269.
4. Brusilovsky, P., Eklund, J., and Schwarz, E. Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems, In Proceedings of Seventh International World Wide Web Conference (14-18 April 1998) 30 (1-7), 291-300.
5. Calvi, L., and Cristea, A.I. Towards Generic Adaptive Systems Analysis of a Case Study. In Proceedings of AH'02 (Malaga, Spain, May 2002) Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 2347, Springer, 79-89.
6. Carro, R. M., Pulido, E. Rodríguez, P. Designing Adaptive Web-based Courses with TANGOW .In proceedings of the 7th International Conference on Computers in Education, ICCE'99 (Chiba, Japan, November 4 - 7, 1999) V. 2, 697-704.
7. Cristea, A.I., and Aroyo, L. Adaptive Authoring of Adaptive Educational Hypermedia, In Proceedings of AH 2002, Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 2347, Springer, 122-132.

8. Cristea, A.I., and Calvi, L. The three Layers of Adaptation Granularity. UM'03. Springer (in press).
9. Cristea, A.I., and De Bra, P. Towards Adaptable and Adaptive ODL Environments. In Proceedings of AACE E-Learn'02 (Montreal, Canada, October 2002), 232-239.
10. Cristea, A.I., and Kinshuk. Considerations on LAOS, LAG and their Integration in MOT. ED-MEDIA'03 (in press).
11. Cristea, A., De Mooij, A. Designer Adaptation in Adaptive Hypermedia. In Proceedings of ITCC'03 (Las Vegas, US 28-30 April 2003) IEEE Computer Society (in press).
12. Cristea, A., De Mooij, A. LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators. In Proceedings of WWW'03, Alternate Education track. (Budapest, Hungary 20-24 May 2003).ACM (in press).
13. Cristea, A., De Mooij, A. Adaptive Course Authoring: MOT, My Online Teacher. In Proceedings of ICT-2003, IEEE LTTF International Conference on Telecommunications, "Telecommunications + Education" Workshop (Feb 23 - March 1, 2003 Tahiti Island in Papetee - French Polynesia) (in press).
14. Cristea, A.I., Okamoto, T., and Kayama, M. Considerations for Building a Common Platform for Cooperative & Collaborative Authoring Environments. In Proceedings of AACE E-Learn'02 (Montreal, Canada, October 2002), 224-231.
15. De Bra, P. and Calvi, L. AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, vol. 4, Taylor Graham Publishers, 1998,115-139.
16. European Community Socrates-Minerva project (project reference number 101144-CP-1-2002-NL-MINERVA-MPP). http://wwwis.win.tue.nl/~alex/HTML/Minerva/index.html
17. Halasz, F., Schwartz, M. The Dexter Hypertext Reference Model. Communications of the ACM, Vol. 37, nr. 2, pp. 30-39, 1994.
18. IEEE LTTF, Learning Technology Task Force. http://lttf.ieee.org/
19. IMS (Instructional Management System) http://www.imsproject.org
20. Kayama, M., Okamoto, T. and Cristea, A.I. Exploratory Activity Support Based on a Semantic Feature Map. In Proceedings of AH'00, LNCS 1892, Springer, 347-350.
21. Mizoguchi, R., Bourdeau, J. Using Ontological Engineering to Overcome Common AI-ED Problems, International Journal of AI in Education, 11 (2), 107-121.
22. My Online Teacher. http://wwwis.win.tue.nl/~alex/MOT01/TeachersSite-html/index.html
23. Wu, H., De Bra, P. Sufficient Conditions for Well-Behaved Adaptive Hypermedia Systems. In Proceedings of the First Asia-Pacific Conference on Web Intelligence: Research and Development (Maebashi, October 2001). Lecture Notes in Artificial Intelligence, Vol. 2198, Springer, 148-152.
24. WC3, Semantic Web. http://www.w3.org/2001/sw/
25. W3C, SMIL, Synchronized Multimedia Language. http://www.w3.org/AudioVideo/
26. Wu, H., De Kort, E., De Bra, P. Design Issues for General-Purpose Adaptive Hypermedia Systems. In Proceedings of the ACM Conference on Hypertext and Hypermedia (Aarhus, Denmark, August 2001) 141-150.
27. Wu, H. A Reference Architecture for Adaptive Hypermedia Applications, doctoral thesis, Eindhoven University of Technology, The Netherlands, ISBN 90-386-0572-2.

[1] This is why the GM layer is so dense: from one DM multiple GM versions can be generated.
[2] Please note however, that automatic transformations represent in themselves a restriction on the total flexibility of the system, because they do not add new data, but are based on re-usage of inherent information. The actual flexibility allowed by the LAOS model (given by the combination of all possible elements allowed by LAOS) is therefore much greater.

[3] This is to 'force' the authors to give at least some minimal information about the concept they are defining, in order to be make the semantics of the concept machine-readable (minimal ontology-based meta-data tagging).

[4] With values being volatile or not according to AHAM [26].

[5] Each *GM* concept has only 2 attributes: '*name*' and '*contents*'.

[6] Links can be added between any concept of the owned *GM* to any concept of the whole CM space of concepts, within *GM* or jumping a level, to the DM.

[7] here, transformations.

[8] an educational-oriented version of the LAOS goal and constraints level.

[9] which can be regarded also as a hierarchical inclusion relation.

Is it possible to devise a unifying model of adaptive hypermedia and is one necessary?

James Ohene-Djan, Matt Gorle
Department of Computing
Goldsmiths College
University of London
SE14 6NW, UK
j.djan;g.gorle@gold.ac.uk

Christopher P. Bailey, Gary B Wills, Hugh C Davis
Department of Electronics and Computer Science
University of Southampton
SO17 1BJ, UK
{cpb,gbw,hcd}@ecs.soton.ac.uk

Abstract:

In this paper we address the question of whether it is possible to devise a unifying model of adaptive hypermedia (AH) in the form of a standardised ``plug `n' play" architecture. Such an architecture, we argue, could the form the basis of a measurement indicator for analysing the amount of adaptivity in a hypertext structure, or to compare and contrast different AH component technologies. We also address the question of how to develop AH systems that are interoperable and interchangeable. The predominantly application-orientated approach to AH development that has prevailed to date has, in part, resulted in relatively low levels of interchangeability and interoperability between AH systems. To address this situation recent AH research has begun to focus on the mechanisms and structures underpinning AH systems. Specifically, several formal models have been proposed which address these issues, the prominent examples being Adaptive Hypermedia Application Model (AHAM), the Fundamental Open Hypermedia Model (FOHM) and the Goldsmiths Adaptive Hypermedia Model (GAHM). In this paper we propose a unifying component-based architecture for AH within which these approaches to modelling can be compared and understood. Using this architecture we show that although each model uses different representations there are important underlying commonalities. It is envisaged that through an understanding of these commonalities, it may be possible for the AH community to devise a standardised ``plug `n' play" architecture for the development of future AH systems.

Keywords:

adaptation, adaptive, modelling, component technology, construct, hypertext, hypermedia, user model, architecture

1. Introduction

Adaptive Hypermedia (AH) systems use knowledge provided by (or captured about) specific users to tailor the information and links presented to individual users. By applying the knowledge they accumulate, adaptive features are made available that support users in navigation and information acquisition. Such support may include limiting options for traversal to information units, tailoring content and presentation of information units, suggesting relevant links to follow and providing additional information on links and information units. Early AH systems were devised to address a particular application area or information domain and were implemented with whatever technologies were available, guided by the intuition of their developers. With the advent of engineering-oriented approaches [20], researchers began to formalise the process of building AH systems. Brusilovsky has provided an account of early AH research [5]. Surveys of more recent AH concepts and the research that has given rise to them can be found in [6]. Current research has started to focus on the mechanisms and structures present in AH systems. In particular, several formal models, emphasising the importance of hypermedia structures and their manipulation, have been proposed. Qualifying representative examples are the Adaptive Hypermedia Application Model (AHAM) [10], the Fundamental Open Hypermedia Model (FOHM) [1] and the Goldsmiths Adaptive Hypermedia Model (GAHM) [25]. In this paper we propose a unifying model as a general, component based architecture for AH within which the AHAM, FOHM and GAHM approaches to modelling AH are compared. Each model takes a structure-oriented approach, in which emphasis is placed on the structures used to specify hyperdocuments and their meta-data rather than the content they contain. We illustrate that although each model uses different representations of hypermedia structures and has differing approaches to manipulating these structures, there are important underlying commonalities in terms of the components found within AH systems and the interactions between them. For the purposes of this paper, we make a distinction between *Personalisation* and *Adaptation*. Any action that alters the structure of a hyperpage, or one of its' component parts, is referred to as a *tailoring action*. *Personalisation* actions are user-initiated tailoring actions. *Adaptive* actions are system-initiated personalisation actions. The remainder of this paper is structured as follows. Subsection 1.1 outlines the motivating factors behind the research reported. Section 2 presents an architecture for general AH systems within which the key components found within AH systems are delineated and their roles described. Using this architecture as a template, AHAM, FOHM and GAHM are described in Sections 3, 4 and 5. Section 6 presents a discussion of the opportunities afforded by the proposed architecture. Section 7 compares our results with those of others and draws conclusions.

1 Motivation

Although AH research has delivered a variety of systems, [9,7,12,13,21,24,28] there is still no consensus as to the components that should comprise such systems. The research reported in this paper is motivated by the view that such a consensus could be achieved through a generalisation of the components within AH systems and the interactions between them. With such a consensus we believe that it is possible to facilitate a systematic investigation of the space of possibilities for personalisation and adaptation (P&A) actions. The technology-driven approach towards the development of AH systems that has prevailed makes a principled testing of the benefits of P&A actions difficult. In many cases it is uncertain whether what is being tailored is a unique, distinctive, property of hypermedia-based interaction or instead simply arises as a consequence of coupling user-interface, database and link management components in a particular manner. Therefore a further motivating factor of our research is how to devise an architecture, at a suitable level of abstraction, that can represent the functionality that is unique to AH systems. Such an architecture should clearly delineate the discrete groupings of functionality found within AH systems, so that they may be viewed as interchangeable and, possibly, interoperable components of AH systems. In the context of this paper, we view *personalisation* as the process of handing over to the user the ability to take actions to tailor hyperpages, thereby overriding, in principle, each aspect of a hyperpages' content and presentation. *Adaptivity* is viewed as the process of allowing the system to take the initiative in tailoring actions in the light of the systems' inference of a users' information goals and history. We argue that system-initiated tailoring (adaptivity) is therefore, in principle, as expressive as user-initiated tailoring and requires no technologies other than those involved in user-modelling and in decision-making from a user-model. With respect to the view above, a final motivating factor is to devise an architecture which accommodates personalisation yet treating the components required to perform adaptation, (i.e., a user-model and decision-making algorithm) as "plug 'n' play" black box components. We view this as beneficial, as it would enable appropriate user-models and decision-making algorithms for particular application areas to be included as and when required. To address these motivating factors, in Section 2, we detail a general, open architecture for personalisation and adaptation. This architecture reflects the predominant approaches to the design of AH systems and aims to illustrate the interactions that take place between their components.

2. The Architecture

A general, open architecture for hypermedia systems, of the kind depicted in Figure 1 is assumed. This architecture reflects the predominant approach to the design of hypermedia systems, whether web-based or otherwise. It is assumed that a core of AH functionality is a client technology loosely coupled to one or more user interface servers (UISs) and database servers (DBSs). An example UIS is a web-browser, and an example DBS is any Database Management System that supports client/server architectures (e.g. Oracle, PostGreSQL).

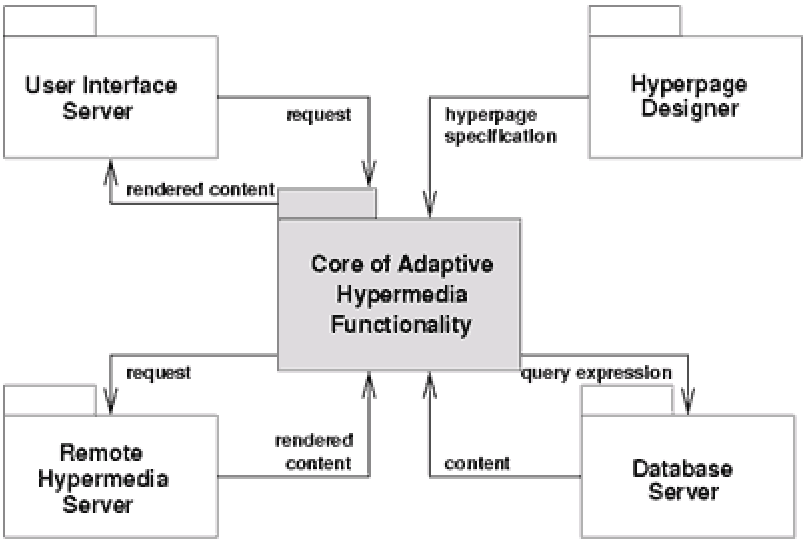


Fig. 1. A General, Open Architecture for Hypermedia Systems.

UISs capture requests for hyperpages. These are channelled into the core of AH functionality. If a request is for a local hyperpage (i.e., one known to the core), the core responds by composing a rendering expression that can be rendered by UISs, possibly after querying one or more DBSs to fetch some or all of the content required by the requested page. It is assumed that the AH system may rely on other hypermedia systems, in the role of a client/server, in a similar manner to the server-browser model in the WWW. In this case the client AH system plays no active role, other than to forward requests received from the UIS to the remote server and return responses from the latter to the former. Implicit in Figure 1 is the assumption that P&A actions in AH systems are independent of any adaptive actions that might be provided by UIS and DBS components of a hypermedia system. This core of AH functionality, the shaded area in Figure 1, is the component that performs the tailoring and is responsible for what users experience as hypermedia-based information retrieval. In the following subsection, we outline our view of these structures. Following this, we describe the components within the core of AH functionality, namely the *user-model* and *composer & tailoring engine*. Figure 2 depicts the architecture of the core of AH functionality.

2.1 Hypermedia Structures

The simplest form of structure is a single, flat formal text, or hyperpage. However, to tailor the information contained within such a page, it is necessary to subdivide it into information units. A hyperpage can be viewed as a sequence of information units (containing, for example, text, numbers, graphics or video), or nodes. These unit have associated with them, a set of references to other units. Each information unit is comprised of a specification of its' content and a specification of how to present (or render) its' content. A specification of content may be the content itself or a reference to the content in the form of query expressions that can be evaluated into content. The specification of presentation defines how the content in the content specification is to be presented by a UIS (e.g., a WWW browser). It takes the form of a formal text in a language that the intended UIS can render (e.g., WML, XHTML, etc.). There is assumed to be a binding between the two specifications. Generally, such a binding may take the form of the renderable text being interspersed with variables referencing the content. We define a hyperdocument to be a collection of hyperpages whose formal properties enable certain navigational operations to be performed over it. For example, with reference to an unordered collection and one of its' members, one can only request another member. If, however, the collection is known to be totally ordered, then requests for the next and previous member are meaningful. Clearly, the issue of designer-imposed structure on collections of hyperpages is a very important one, but it is also orthogonal to the architecture and therefore one that is not further addressed in this paper.

2.2 Meta Data

The purpose of meta-data is to describe the content and/or behaviour of a hyperpage or its' component parts. One can induce from a definition of a hyperpage (and recursively, its' component parts) a set of meta-data possibilities. Such meta-data may take the form of notes, generated by authors, users or the system. Generally, such notes are user-generic attributes of interest (e.g., the level of difficulty of a hyperpage is high) or instructions on how to tailor a renderable text. Meta-data can also be used to represent more abstract concept of links, tours and other structures. The level of granularity that a hyperpage is subdivided by will determine the set of meta-data possibilities. For example, a flat hyperpage may only be annotated at the hyperpage level, while a hyperpage comprising of a sequence of information units enables annotation at the level of an unit or one of its component parts (e.g., a link). Within this architecture, it is assumed that both hyperpages and their associated meta-data are stored in a *hyperlibrary*. It is assumed that the implementation of such a library would provide the functionality of a modern database system (e.g., associative querying, scalable retrieval and versioning).

2.3 The User Model Component

A *user model* may be viewed as a store of an individual users' information goals and history. Such a model is the basis upon which users may provide input to the adaptive process. User models can take many forms [4,14,15] and many different techniques have been used to acquire information about users [19,16]. A user model may provide the means for a user to feed preferences into the adaptive process when it is system-initiated. The model reflects a users' goals and history. User modelling generally involves eliciting details from the user overtly or covertly. These details are then used by the adaptive mechanism to perform system-initiated tailoring. There are various types of user modelling, such as preference analysis [13], stereotyping [5] and activity analysis [2].

2.4 The Engine Component

The architecture within the core, depicted in Figure 2, shows two distinct groups of components. The *personalisation mechanism*, is responsible for providing the functionality required for user-initiated tailoring (personalisation) of hypermedia structures. The *inference mechanism*, is responsible for providing the additional functionality required for system-initiated tailoring (adaptivity). The dynamics of the personalisation mechanism may be understood as follows. The personalisation process starts with a personalisation request, conveyed by a user, to the personalisation mechanism via a UIS. A personalisation request specifies a scope across hyperpages, or meta-data and the actions that the user wishes to effect. On receiving this request the *composer & tailoring engine* parses it to determine which hyperpages should be personalised and how, thereby generating a *personalisation program*. This program is a formal set of instructions, in a language known to the tailoring engine, that, when interpreted, retrieves the identified hyperpages, carries out the actions specified in the personalisation request and generates their user-specific versions, which are then stored.

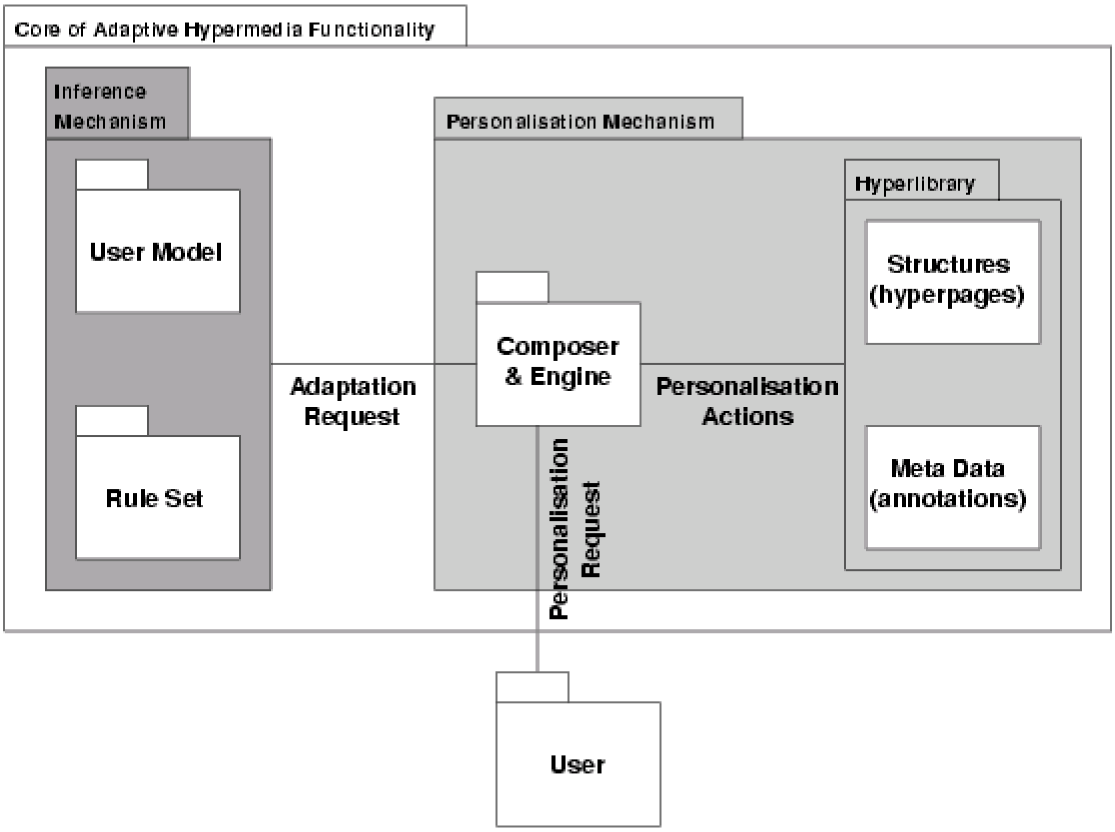


Fig. 2. An Architecture for Adaptive Hypermedia

Adaptivity is viewed as the process of allowing the system to take the initiative in personalisation actions, in the light of the systems' inference of a users' information goals and history (i.e., user-model). For adaptivity, a user-model and decision-making algorithm (containing a set of rules) are required. The role of these components is to initiate personalisation actions, possibly after consulting the user. When the hypermedia system identifies an opportunity to adapt (e.g., a request for a hyperpage) it does so by generating an adaptation request, which is a system-initiated personalisation request that carries out an action which the user may have issued, were he or she motivated to do so. Using the architecture described, system-initiated personalisation is, in principle, as expressive as user-initiated personalisation and requires no functionality other than that provided by the inference mechanism. We argue that, using this architectural approach the components of this inference mechanism may be treated as ``plug 'n' play'' black boxes that can be non-disruptively added to an AH system once an appropriate API has been developed. The additional dynamics provided by the inference mechanism are as follows. The adaptive process starts when an event (e.g., a request for a hyperpage) is detected by some monitoring system. As a result of the detection of an event the inference mechanism consults the user-model (i.e., a store of an individual users' information goals and history) to determine its current state. Using this knowledge the decision-making algorithm component applies a set of rules to each goal in the user-model. If a rule is satisfied then a set of rules and the user-model are used by the decision-making algorithm to construct and then suggest to the user a well formed personalisation request. This request is then submitted to the personalisation mechanism. It is assumed that the core of AH functionality can also handle requests for hyperpages. Upon receiving a request, the composer component parses the hyperpage specification into a series of instructions that, when executed, convert the hyperpage specification into a renderable text (hyperpage), that is then returned to the UIS that initiated the request. Note that since we are drawing an architecture within which the operation of AH systems in general may be understood, we do not specify the formal structure or semantics of personalisation requests and hyperpage specifications. To do so would limit the architecture to specific forms of hypermedia structures and specific languages for their manipulation. Furthermore, no constraints are imposed, regarding the choice of rules, user-model and decision-making algorithms, other than that they must exhibit the dynamic behaviour described in this section. Using the architecture described in this section we now show how the AHAM, FOHM and GAHM approaches to modelling hypermedia may be represented within this architecture.

3. AHAM

One of the first formal models for adaptive hypermedia, the Adaptive Hypermedia Application Model (AHAM) [33,10] builds on the earlier models formulated for more traditional hypertexts. AHAM is designed around an extended version of the DEXTER model [11]. DEXTER separates the components of a hypertext system into three major layers; the within component layer which stores the contents of the domain, the storage layer which contains the structure (nodes and links) between objects in the component layer, and the runtime layer which presents the hypertext information to the user. The DEXTER model also includes an anchoring layer to allow addressing of individual chunks of data within the component layer, and a presentation specification layer which provides the runtime layer with information on how to present specific hypertext components. The AHAM extension to DEXTER allows it to support adaptive hypermedia applications by separating the storage layer into a domain model, a user model and an adaptation model. This allows AHAM to provide a formal model for expressing adaptive hypermedia applications at the abstract level. AHAM is not the only model based on Dexter. The Munich Reference Model [18,17] follows the same approach taken in AHAM by extending Dexter's storage model and adding user and adaptive models. Architecturally, the two reference models are almost identical. The major difference between the two approaches is that AHAM specifies an adaptation rule language, while the Munich model contains an object-oriented specification written in the Unified Modelling Language (UML).

3.1 Hypermedia Structures

Much like our notion of hypermedia structures, AHAM's domain model uses concept components to represent the abstract representation of an information item in an adaptive hypermedia domain. The structure of a concept is broken down into a set of attribute-value pairs, a sequence of anchors and a presentation specification. To form a hypermedia, concepts are arranged in a directed acyclic graph. Atomic concept components represent a single fragment of information and their anchors reference the physical information, while composite components use a 'children' attribute to specify a sequence of smaller composite components or atomic concepts. As in the Dexter model, the raw data is stored in the within-component layer and all concept anchors reference the data in this layer. Presentation specifications determine how the particular data is to be displayed/rendered, although their application-specific nature means they are not modelled by Dexter or AHAM.

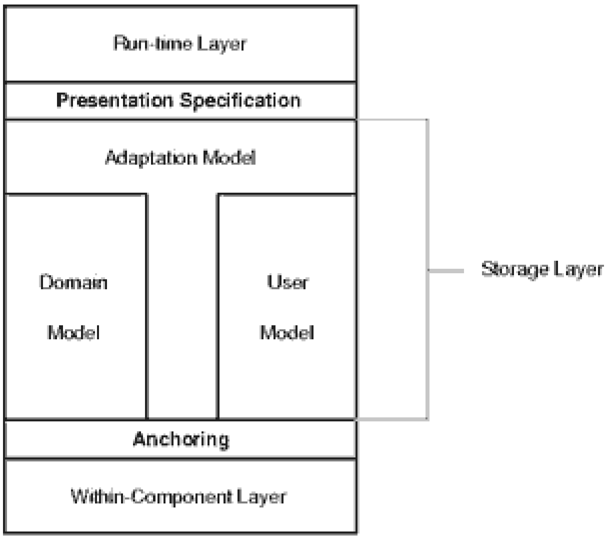


Fig. 3. The Architecture of AHAM

3.2 Meta Data

AHAM's Meta-data, in the form of attribute-value pairs can be associated with both atomic concepts and higher-level composite components. At the hypermedia structure level, these storage units provide a means for describing the relationship types between concepts. AHAM also specifies a user model, overlaid on top of the domain model, to determine factors and actions that affect the user. The user model is also a set of attribute-value pairs that can be used to represent user-centric meta-data such as the required knowledge level of a user, or the status of a concept (read, ready to be leaned,). AHAM does not specify the complete set of possible meta-data values but allows authors to produce their own. AHAM also does not allow the ability to dynamically add new attributes-values, such as annotations, at runtime by the users of the system.

3.3 Engine Component

To combine the hypermedia structure and meta-data (or in AHAM terminology, domain and user models) AHAM uses an adaptation model which contains a set of adaptation rules, and an interpreter (or engine) to process these rules. Adaptation rules, written by a system designer, are stated in the form of event-condition-action clauses which provide the required mechanism to initialise the user model, update the user model and generate instances of adapted information.

4. FOHM

Work at the University of Southampton, has concentrated on analysing the fundamental components and structures of hypermedia systems. This work is part of the larger open hypermedia community [32,29,30,31] which have developed formal models for representing the structure and associations that exist within the underlying data components of hypermedia systems. To this end, a new open hypermedia model was developed; the Fundamental Open Hypermedia Model (FOHM) [23]. FOHM is largely based on the prior work with the Open Hypermedia Protocol (OHP) [8] which was designed to provide a reference model and architecture for Open Hypermedia systems. OHP placed an emphasis on the different structures belonging to hypermedia domains and raised the issue of how context might affect such structures. FOHM extends these ideas by developing a generalised model to represent the structure of these domains, and then provides the facility to attach context and behaviour objects to the model at various locations. While FOHM provides a flexible theoretical model of structure which can be used to build hypertexts, an engine, Auld Linky [22], is required to instantiate and process the model. Auld Linky stores a database of FOHM objects (in XML format) and responds to queries from client applications for FOHM structures. Although FOHM was originally designed from an Open Hypermedia perspective, its application within the adaptive hypermedia field has been presented in [3].

4.1 Hypermedia Structures

The primary structures in FOHM are the data item and the association. Following earlier hypertext models, data items are attached to associations using a process of reference.

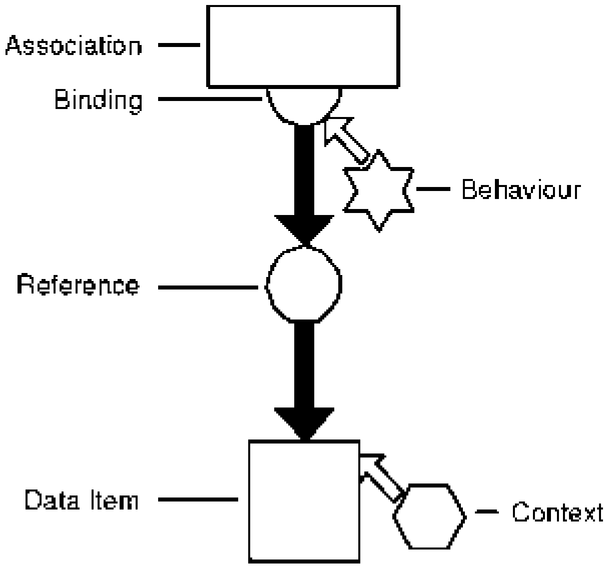


Fig. 4. The Structure of a FOHM Object

Data objects are components that encapsulate a piece of information. Associations are links that relate together data objects and other associations. By combining these structures together, FOHM can support highly complex hypertext domains. During FOHM's development, several common arrangements of FOHM objects have been identified. *Tours* provide a sequential path across a set of objects, *Level's of Detail* are tours linking together increasingly detailed information and *Concepts* are associations that relate the same conceptual information using different presentation styles (i.e. handling different media representation of the same data). These structures can then be combined and arranged to suit a variety of hypertext domains.

4.2 Meta Data

To enhance the power of FOHM, two additional objects, context and behaviour can be used as meta-data/annotation components. They are implemented using attribute-value pairs (in a similar manner to the attribute-value meta-data in AHAM). Context objects provide a means of limiting, or scoping, the current 'view' of the FOHM model. With this technique, a context object is attached to a FOHM query and it acts as a modifier, restricting the set of available FOHM objects that can be provided to the subset which have valid matching contexts. In an adaptive hypermedia domain, context objects can be used to represent restrictions on user views of a domain. This is analogous to the technique of user modelling, since a context object can be used to represent the current level of user understanding in a given subject. Behaviour objects provide an event driven mechanism for specifying a set of actions. For example, a behaviour object can be attached to the 'on traversal' event of an association (such as a standard hyperlink) to specify the changes to the state of the system after the user has activated the link. For an adaptive hypermedia system, behaviour objects can be used to as a means of updating user models with new information based on the actions taken by the user.

4.3 Engine Component

The engine component of FOHM is realised by Auld Linky. Auld Linky manages a hypertext domain model marked up in XML as FOHM objects. When a client sends a personalisation request to Auld Linky (in the form of a FOHM association query) they can chose to optionally provide a Context Object. Auld Linky analyses the domain model to find parts that match the query pattern. It then collapses parts of the structure that fail to match the supplied Context. In this way, clients receive a contextual (personalised) view of the FOHM domain.

5. GAHM

The Goldsmiths Adaptive Hypermedia Model(GAHM) [25,26,27], developed at Goldsmiths College, University of London, takes a formal approach to the modelling of personalisable, adaptive hyperlink-based systems. The main contributions of which are an abstract model of core hypermedia-based functionality and, secondly, the definition of an abstract model of personalisability extending the core. Furthermore, an architecture is drawn for adaptivity using active rules as the decision-making component. The model is comprised of three groups of functions: H-Region functions that model non-personalisable hypermedia-based interaction; P-Region functions that model user-initiated tailoring of hypermedia content; A-Region functions that model system-initiated tailoring of hypermedia content. Within the H-Region, hyperpages are modelled as formal specifications. The semantics of hyperpage specifications are given with reference to an abstract machine whose operation and instruction set is formalised. The abstract machine is used to illustrate the execution of hyperpage specifications, thereby yielding renderable hyperpages. Induced from the formal definition of hyperpage specifications is a set of annotation possibilities and personalisation & adaptation (P&A) actions. These enable all design decisions realised as hyperpage specifications to be revised. Within the P-Region, personalisation is modelled as the user-initiated process of annotating and rewriting a hyperpage specification into a version that is associated with the user who took that action. It follows that the hyperpages users see may reflect their user-model if they wish. When personalisation functionality is non-disruptively added to the H-Region, a designer can annotate a hyperpage in preparation for differences in users' goals and histories. A user can request to personalise both hyperpages and their annotations. Personalisation requests allow users to specify which hyperpages are to be personalised and how they should be transformed. Hyperpage annotations and personalisation requests are modelled as formal specifications and a formal language has been defined for this purpose. Set-theoretic and relational algebraic expressions are used to represent the semantics of personalisation requests. Within the A-Region, adaptivity is modelled as the process of allowing the system to take the initiative in personalisation actions. When adaptivity functionality is layered over Hand P-Regions, both users and designers can define strategies as to when the system should take the initiative and actively tailor interaction.

```
page{
 chunk{
  entry{
    [<a name="Intel 80386"></a>]  }
  content{
    A := 'Intel 80386',
    B := '(Commonly abbreviated to "386", trademark
 "Intel386") The successor to the',
    C := 'Intel 80286',
    D := 'microprocessor',
    E := '.  It was the first Intel processor with
 32-bit data and address',
    F := 'bus',
    G := 'es.  It can address four ',
    H := 'gigabytes',
    I := ' (2^32 bytes) of memory; however, 16
 megabytes is a typical maximum in'
    J := 'IBM PC'
    K := 's.'
  rendering{
    [<h2>                          A </h2><p>] B
    [<b><a href="80286.html">      C </a></b>]
    [<b><a href="microprocessor.html">D </a></b>] E
    [<b><a href="bus.html">        F </a></b>] G
    [   <a href="gigabyte.html">    H </a>    ] I
    [   <a href="ibmpc.html">       J </a>    ] K }}
 chunk{
  entry{
    [<a name="Intel 80386"></a>]  }
  content{
    A := 'select definition
          from   glossary
          where  term = 'i386'  }
  rendering{
    A  } } }
```

Fig. 5. An Example Hyperpage Specification

In summary, the model is an abstract model, as many steps removed from concrete implementations as necessary to allow a systematic, exhaustive investigation of P&A issues in hypermedia systems. The model is an open model, insofar as hypermedia systems are viewed as clients of a variety of servers, in particular data and user-interface servers. Personalisation involves a transfer of ownership of the process of interaction with a hyperdocument from designers to users. To ensure that the set of personalisation actions is consistent, its elements are induced from the formal definition of the hyperdocuments they act upon.

5.1 Hypermedia Structures

Within the GAHM, hyperpages are defined to be a sequence of chunks, each of which is comprised of a *content specficiation* (C-Spec) and a *rendering specfication* (R-Spec). A C-Spec may take the form of data values or requests to DBSs for data values and may be associated with a set of *template variables*. Conceptually, a template variable is a placeholder for the content denoted by the C-Spec. An R-Spec defines how content is to be rendered. It takes the form of a formal language that the intended UIS can render (e.g, HTML). This renderable text is interspersed with template variables, acting as placeholders for the content defined by the C-Spec. A chunk may be associated with a set of *entry points*, enabling it to be referenced, and a set of *exit points*, allowing for the establishment of a navigable link to another hyperpage or chunk. A hyperdocument is defined to be a collection of hyperpages whose topology enables navigation between them.

5.2 Meta Data

The kinds of personalisation actions modelled by GAHM are based on the annotating and rewriting of hyperpage specifications. Annotation pairs a hyperpage specification with notes of interest to the user. These notes allow the assignment of values to attributes of the hyperpage and also allow the specification of rewriting actions over renderable texts (composed hyperpages). The existence of annotations facilitates the personalisation of a specified hyperpage and the recording of information about a hyperpage. Figure 6 shows an annotation in GAHM.

```
annotation{
 page:
  description := 'a brief description of the
                  Intel 80386 microprocessor';
  level       := 1;
  see-as-well := [http://www.intel.com/];
  keyword     := '386', '80386',
                  'Intel', 'CPU';
 [1, chunk]:
  description := 'Intel 80386';
  keyword     := '386', '80386',
                  'Intel', 'CPU';
  see-as-well := '[microprocessor.html]'
 [2, chunk]:
  description := 'a brief description of the
                  Intel 80386 microprocessor,
                  fetched dynamically';'
 [2, chunk (R-spec)]:
  keyword     := 'Rendered using HTML'; }
```

Fig. 6. Annotating Fig. 5

5.3 Engine Component

In GAHM, personalisation is viewed as the process of handing over to the user the ability to annotate, and/or rewrite, hyperpage specifications. Users can personalise both hyperpage specifications and annotations. A *personalisation request* is an editing command over hyperpage specifications that causes a modified version of a hyperpage to be versioned by the user who issued the request. A personalisation request specifies which hyperpages to personalise and what form to change them into. It is, therefore, a request to override the original decisions of the designers of a hyperpage (and of course, past expressions of preference by the user). A few examples of personalisation requests that a user might issue to the hyperdocument, of which Figure 5 is a page, are given in Figure 7.

```
select-page-if            % Example 1
  true
hp-then-do {
  insert [1, chunk (R-spec)]
     "These pages belong to student X"  }
                          % ---------
select-page-if            % Example 2
  [5, chunk] contains "Electronic Mail"
hp-then-do {
   insert [1, chunk]
     chunk {
       content  { X := 'Many of the latest
                        electronic mail systems
                        now provide support for
                        sound and video files.'  }
       rendering { [<I>] X [</I>]} } }
                          % ---------
select-page-if            % Example 3
  true
ann-then-do {
  insert page :
       "http://www.gold.ac.uk.com"
    -> "http://www.goldsmiths.ac.uk";  }
                          % ---------}
```

Fig. 7. Personalising Fig. 5

Example 1 in Figure 7 is a personalisation request to tailor content. Its' effect is to insert the string ``These pages must be revised" into the rendering expression of the first chunk of all hyperpages. **Example 2** is a request to add content. This request is applied to all hyperpages that contain the string ``Electronic Mail" in the 5th chunk. **Example 3** is a request to rewrite all occurrances of the string ``www.gold.ac.uk" to ``www.goldsmiths.ac.uk". Its' scope is all hyperpages. Within the GAHM, adaptivity is modelled as system-initiated personalisation. The GAHM approach to adaptivity centres on adaptive function. This function implements an inference engine over a decision theory, specified as a set of active rules, that describes which actions are more likely to yield the most benefits given a user-model. The function is responsible for suggesting personalisation actions, as described above.

6. Discussion

Due to increased interest in the use of personalisation and adaptation technologies, in the development of hypermedia applications, we believe that the hypermedia community is likely to benefit from a clearer consensus of how strong foundations for such applications may be laid. The architecture proposed is purposefully cast at a level of abstraction above that of concrete systems, which has allowed for the representation of alternative models of AH. Using the architecture proposed, we have been able to clearly model the functionality of the components of AH systems in terms of its hypermedia structures, meta-data and engine component. By doing this, we have shown that additional AH components, such as user-models and decision-making algorithms, can be treated as interchangeable components of an AH system. Our architecture also demonstrates how the expressiveness of P&A actions is ultimately determined by the structural approach chosen to define the hyperpages. The level of granularity that a hyperpage is subdivided by will determine the set of meta-data possibilities. In turn these semantic references to the content are the constraining factor when devising an engine for the manipulation of hyperpages. Through the comparison of three qualifying models of AH, using the proposed architecture, we have shown that although taking different perspectives, all three address the same structural concerns:

1. the formulation of tailorable hypermedia structures,
2. the use of meta-data to provide semantics, or context, for these structures,
3. the development of mechanisms of hypermedia structures and their associated meta-data.

However, there are several noticeable differences between the three models that would have an impact on developers when choosing an architecture to model an AH system. Firstly, FOHM, while flexible and expressive, models only the personalisation mechanism aspect of an AH system. FOHM provides all the required hyperlibrary functionality while Auld Linky acts as the composer and engine. However, to be used within an adaptive environment, it needs to be coupled with these "plug 'n' play" components previously discussed. Unlike FOHM, GAHM and AHAM both support the personalisation mechanism and the inference mechanisms needed to model the core AH components. Another difference that will be important to system designers, is that AHAM has been designed to operate on pre-defined data. In other words, AHAM does not handle dynamically generated data or meta-data at run time. This restriction was imposed to secure full knowledge of the adaptation rules at design time and therefore guarantee that all rules terminate, or at least identify those that do not. However it then limits the ability of any AHAM-based systems to create annotations on hypertext objects (such as pages or chunks) by users and then offer personalisation actions based on this meta-data. In contrast to FOHM and AHAM, from a developers perspective, the GAHM is a functional model. Its' expressiveness is in its ability to clearly represent the functions required to compose hyperpages together with those required to tailor them. GAHM illustrates the process of adaptation through the specification of an adaptive engine utilising a rule base and a formally defined user model. The adaptive function is formalised as an interpreter of active rules. Its' operation is based upon the dynamic updating of a users' history and information-seeking goals. However the GHAM's language for personalisation, although rich, is tightly coupled to the structure of its' hyperpages. As such, it may not be applicable in differing settings. A final point to note is that, due to the functional nature of the GAHM, a degree of respecification may be required by a developer wishing to implement the GAHM using a modern object-oriented methodology. It can be argued that, even though the architecture proposed in this paper could be presented in equivalent ways, the methodological procedure of isolating the various components and describing their high-level functionality and interactions is a contribution that may be used in other settings, under differing assumptions and using alternative conceptualisations of hypermedia systems.

7. Conclusions

In this paper we present a proposal for a component-based architecture that can be used to illustrate the components required for adaptivity. It is shown that, through an understanding of the structures that underlie these components, it is possible to devise an architecture in which personalisation and adaptation are clearly defined. We have also shown that structure-oriented adaptive hypermedia models, AHAM, FOHM and GAHM, can be mapped onto this architecture. Although each model uses different representations we believe that there are important underlying commonalities. This architecture has, therefore, provided a mechanism by which we can express these commonalities for P&A, i.e., hypermedia structures, meta-data and the engine component. Through an understanding of these common components and their relationships, we believe that it is possible for the AH community to devise a standardised "plug 'n' play" architecture. Such an architecture will enable the rapid development of future AH systems, by avoiding the reengineering of common components. In addition, the identification of the commonalities between models of AH provides the first step towards the development of a set of services for communication between various AH components. Such a set of services could form the basis for a consensus as regards standards for interchangeability and interoperability between components of AH systems.

References

1. Bailey, C. (2002). *An Agent-Based Framework to Support Adaptive Hypermedia*. PhD thesis, University of Southampton.
2. Bailey, C., El-Beltagy, Samhaa, R., and Hall, W. (2001). Link augmentation: A context-based approach to support adaptive hypermedia. In *Hypermedia: Openness, Structural Awareness, and Adaptivity*.
3. Bailey, C., Hall, W., Millard, D. E., and Weal, M. J. (2002). Towards open adaptive hypermedia. In *Proceedings of the Second International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, pages 36-46.
4. Brajinik, G., Guida, G., and Tasso, G. (1987). User modelling in intelligent information retrieval. *Information Processing and Management*, 23(4):305-320.
5. Brusilovsky, P. (1996). Methods and Techniques of Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 6(2-3):87-129.
6. Brusilovsky, P. (2001). Adaptive Hypermedia. *User Modeling and User-Adapted Interaction*, 11(1-2):87-110.
7. Brusilovsky, P., Schwarz, E., and Weber, G. (1996). ELM-ART: An Intelligent Tutoring System on World Wide Web. *Lecture Notes in Computer Science*, 1086:261-269.
8. Davis, H. and Lewis, A. (1996). A draft proposal for a standard open hypermedia protocol. In Wiil, U. and Demeyer, S., editors, *Proceedings of the 2nd Workshop on Open Hypermedia Systems Hypertext '96*, Washington D.C.
9. De Bra, P., Aerts, A., Smits, D., and Stash, N. (2002). AHA! meets AHAM. In Bra, P. D., Brusilovsky, P., and Conejo, R., editors, *Adaptive Hypermedia and Web-based systems: Second International Conference, AH 2002*, LNCS, pages 213-222, Malaga, Spain. Springer.
10. De Bra, P., Houben, G.-J., and Wu, H. (1999). AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In *Proceedings of the 10th ACM conference on Hypertext and Hypermedia*, Darmstadt, Germany. ACM.
11. Halasz, F. and Schwartz, M. (1990). The Dexter hypertext reference model. In *Proceedings of the Hypertext Standardization Workshop*, volume 500-178 of *NIST Special Publications*, pages 95-133, Gaithersburg, MD 20899. National Institute of Standards and Technology.
12. Hohl, H., Boecker, H.-D., and Gunzenhaeuser, R. (1996). Hypadapter: An Adaptive Hypertext System for Exploratory Learning and Programming. *User Modelling and User-Adapted Interaction*, 6(2-3):131-156.
13. Höök, K., Karlgren, J., Waern, A., Dahlback, N., Jansson, C., Karlgren, K., and Lemaire, B. (1996). A glass box approach to adaptive hypermedia. *User Modeling and User Adapted Interaction: Special Issue Adaptive Hypertext and Hypermedia*, 6(2-3):225-261.
14. Karlgren, J., Höök, K., Lantz, A., Palme, J., and Pargman, D. (1994). The glass box user model for filtering. In *Proceedings of the 4th International Conference on User Modeling*, pages 221-222, Bedford, MA, USA.
15. Kay, J. and Kummerfeld, B., editors (1994). *Proceedings of the Workshop on Adaptive Hypertext and Hypermedia User Modelling '94*, Cape Cod.
16. Kobsa, A. (1993). User modelling: Recent work, prospects and hazards. In *Adaptive user interfaces: Principles and pratice*, pages 111-128. Amsterdam: North-Holland Elsevier.

17.Koch, N. and Wirsing, M. (2001). Software engineering for adaptive hypermedia systems. In *Proceedings of the Third Workshop on Adaptive Hypertext and Hypermedia*, pages 205-210.
18.Koch, N. and Wirsing, M. (2002). The Munich reference model for adaptive hypermedia applications. In Bra, P. D., Brusilovsky, P., and Conejo, R., editors, *Aptive hypermedia and Web based systems: Second International Conference, AH 2002*, LNCS, pages 213-222, Malaga, Spain. Springer.
19.Kok, A. (1991). A review and synthesis of user modelling in intelligent systems. *The knowledge Engineering Review*, 6(1):21-47.
20.Lowe, D. and Hall, W. (1999). *Hypermedia and the Web An Engineering Approach*. Wiley. ISBN 0-471-98312-8.
21.Mathe, N. and Chen, J. (1996). User-centered Indexing for Adaptive Information Access. *User Modeling and User-Adapted Interaction*, 6(2-3):225-261.
22.Michaelides, D. T., Millard, D. E., Weal, M. J., and Roure, D. C. D. (2001). Auld linky: A contextual open hypermedia link server. In *Proceedings of the Seventh Workshop on Open Hypermedia Systems*, pages 52-64.
23.Millard, D. E., Moreau, L., Davis, H. C., and Reich, S. (2000). Fohm: A fundamental open hypertextmodel for investigating interoperability between hypertext domains. In *Proceedings of the ACM Conference on Hypertext, May 30 - June 3*, pages 93-102, San Antonio, TX.
24.Ng, M. H., Hall, W., Maier, P., and Armstrong, R. (2002). The application and evaluation of adaptive hypermedia techniques in web-based medical education. *Association for Learning Technology Journal*.
25.Ohene-Djan, J. (2000). *A Formal Approach to Personalisable, Adaptive Hyperlink-Based Interaction*. PhD thesis, Department of Computing, Goldsmiths College, University of London, University of London. Available from http://homepages.gold.ac.uk/djan/jodthesis.pdf.
26.Ohene-Djan, J. (2002). Ownership transfer via personalisation as a value-adding strategy for web-based education. In Brusilovsky, P., Henze, N., and Millán, E., editors, *Proceedings of the Workshop on Adaptive Systems for Web-based Education, held in conjunction with 2nd International Conference on Adaptive Hypermedia and Adaptive Web Based Systems*, Málaga, Spain.
27.Ohene-Djan, J. and Fernandes, A. (1998). Modelling personalizable hyperlink-based interaction. In Brusilovsky, P. and De Bra, P., editors, *Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia (Ninth ACM Conference on Hypertext and Hypermedia HYPERTEXT'98)*, pages 87-91, Pittsburgh, USA. ISSN 0926-4515.
28.Vassileva, J. (1996). A task-centered approach for user modeling in a hypermedia office documentation system. *User Modeling and User-Adapted Interaction*, 6(2-3):185-224.
29.Wiil, U., editor (1997). *Proceedings of the 3rd Workshop on Open Hypermedia Systems Hypertext '97*, Southampton, England.
30.Wiil, U., editor (1998). *"Proceedings of the 4th Workshop on Open Hypermedia Systems Hypertext '98*, Pittsburgh, PA.
31.Wiil, U., editor (1999). *"Proceedings of the 5th Workshop on Open Hypermedia Systems Hypertext '99*, Darmstadt, Germany.
32.Wiil, U. and Østerbye, K., editors (1994). *Proceedings of ECTH 94 Workshop on Open Hypermedia Systems*, volume R-94-2038.
33.Wu, H. (2002). *A Reference Architecture for Adaptive Hypermedia Applications*. PhD thesis, Techniche Universiteit Eindhoven.

Formalising the Design Process of Web-based Adaptive Educational Applications using an Object Oriented Design Model

Simos Retalis
Computer Science Department
University of Cyprus
P.O. Box 20537 Nicosia, Cyprus
Tel: +357 22 892246, Fax: +357 22 339062
retal@cs.ucy.ac.cy

Andreas Papasalouros, Manolis Skordalakis
Software Engineering Laboratory
National Technical University of Athens
15780 Zografou, Athens, Greece
Tel. +302 10 7722486, Fax. +302 10 7722519
{andpapas,skordala}@softlab.ntua.gr

Abstract

The work presented in this paper is comprised of a visual design model for adaptive hypermedia educational systems that follows the principles of the object-oriented paradigm. The model is primarily originated on the Object Oriented Hypermedia Design Method, it has influenced by the AHAM. The Unified Modeling Language (UML) serves the purpose of notation syntax and semantics for this model. The theoretical analysis of the model is accompanied by a case study application for the design of an adaptive web-based testing system.

Keywords

Adaptive Hypermedia, Educational systems, visual design modeling

1. Introduction

It is generally agreed that there is a need for modeling of hypertext and hypermedia applications [17].In contrast to generic software engineering, where significant progress has been made the past twenty years, there is still a great deal of work to be done on formalizing process models, and defining methodologies or design methods for hypermedia applications. These models can be reference models, facilitating the common understanding of the structure and behavior of particular hypertext applications, or design models that facilitate the process of design and development of such applications.

In this paper we propose a design model for *Web-based Adaptive Educational Applications*. A *Web-based Adaptive Educational Application (WAEA)* is defined as a dynamic web-based hypertext application, i.e. a set of dynamically generated web content, which provides a learning environment to its users. This environment comprises electronic content for study as well as a set of tools that facilitate the study of a learner such as web-based questionnaires, glossaries, communication tools, etc. This model focuses on content, which is considered as hierarchically structured, usually dynamically created, personalized assembly of predefined learning resources, either created from scratch or reused. These resources can be available in any form such as files, database entries, etc.

Our model is based on the Unified Modeling Language [18, 19] that is a standard, extensible formalism for visual object-oriented modeling. A design model like this can be used as a framework [1, 10] for authors of hypertext applications to develop and apply methodologies in order to create adaptive educational applications (and not general purpose) in a disciplined and controlled fashion. It incorporates the principle of separation of concerns in the design of hypermedia applications, dividing the design of the application in three stages: conceptual, navigational and presentational. We also claim that this separation of concerns aligns with the three types of adaptation, navigation and presentation. Beyond a design model, if the development of open, portable, maintainable WAEA is to be facilitated, there is a need for a formally specified description of the WAEA. This description must be automatically generated from the aforementioned design model, at least to an extent, and must be easily ported to specific run-time environments that will deliver the specific WAEA.

This model has been built with the following requirements that in mind [16]:

1. *Formalisation*: its notation system must describe a WbEA and its constituents in a formal manner
2. *Completeness*: its notation system must be able to fully describe a WbEA, including all types of its constituents, the relationships among them and their behavior
3. *Reproducibility*: its notation system must describe a WbEA and its constituents in an abstract level so that repeated execution/adoption is possible for specific subject domains
4. *Compatibility*: its notation system must fit in with the available standards and specifications (IMS, IEEE LTSC, SCORM, etc.)
5. *Reusability:* its notation system must make it possible to identify, isolate, decontextualize, exchange and re-use constituents of a WbEA.

The rest of the paper is structured as following: In section 2, an overview of the three steps in the design process as well as the main components of the model are described. In section 3 follows an analysis of the components of the model giving more emphasis on conceptual design. In section 4, this model is compared to other models and some concluding remarks are given.

2. CADMOS-D: A Hypermedia Design Method for WEHA

CADMOS-D (design) is a method for the creation of the detailed design of a web-based educational application, which includes structural details of the learning resources, the navigational schema and templates for describing abstractly the graphical user interfaces. This method follows the principles of the object oriented hypermedia design method (OOHDM) [17]), which has provided systematic ways to design generic hypermedia applications and not especially educational ones.

CADMOS-D method proposes a stepwise design process, as shown in Figure 1: Conceptual Design, Navigational Design and Interface Design. The intermediate products of each step are validated according to guidelines for formative evaluation of the instructional design (checking structural, navigational, aesthetics and functional issues). The whole design process is considered to be iterative, where in each iteration loop the design artefacts are evaluated and the feedback from the evaluation is used for their improvement, until they reach the desirable level.

As facilitator and framework in the design process, a UML based model has been used. This model consists of the following components: Domain Model, User Model, Teaching Model. Each one of these is a logical group of model elements and is represented as a package, the standard UML grouping mechanism, as shown in Figure 2.

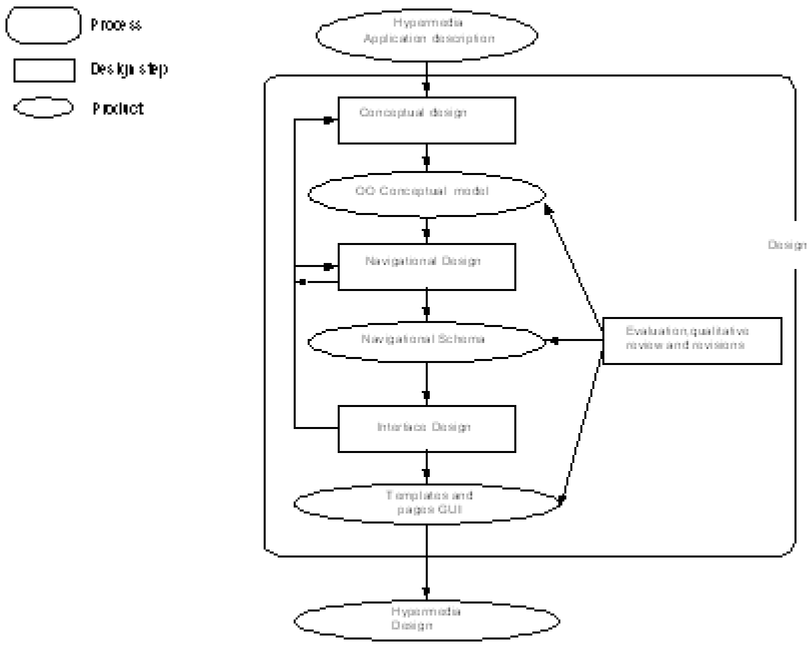


Fig. 1. The three design steps.

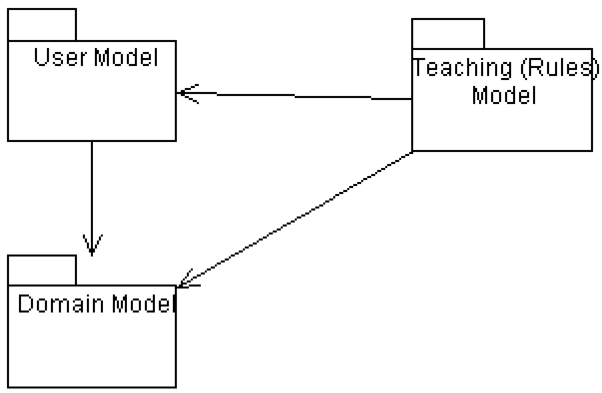


Fig. 2. The models and their dependencies

The *Domain model* defines the concepts of the subject that is going to be taught with their semantic interrelationships. It can be considered as the Ontology [22] of the subject to be learned by the students. The Conceptual Model provides an objective definition of the knowledge subject. This definition is provided by the author of the educational application who is considered as a subject matter expert.

The *User Model* consists of two different parts, each one containing two types of elements: The *Overlay Model*, which is the domain specific part of the user model and defines the status of the learner's knowledge of the specific concepts covered by the learning material. The state of this model is frequently updated as a result of the learner's interaction with the learning content, for example the reading of learning material, the taking of an on-line test, the interaction with simulations, etc. The knowledge is defined as a structure of concepts (schema) and this structure is built during the user's learning activities. The Overlay Model depicts the system's awareness of the current status of the user's knowledge about the domain of the specific application as it is stated in the Conceptual Model. The elements of this sub-model are called *UserScheme* [2],and there can be one *UserScheme* element for each class of the Conceptual Model.

The second part of the User Model defines elements that are used to represent the usually predefined user knowledge profile either concerning the knowledge of the particular domain (novice, intermediate, expert, etc) or corresponding to the user's preferences or learning style. According to [3, 4] this constitutes the Stereotyped User Model. The elements of this submodel are called *User.*

The *Teaching Model* contains rules as Object Constraint Language [23] expressions applied to the appropriate UML elements, mainly classes. Constraints are conditions that must hold for the specific model they are applied. OCL is a formal language for applying constraints to UML models. OCL is a language for the specification and not the implementation of particular systems. The rules defined in our Teaching model are applied as two types of constraints:

- *Invariants*, that is conditions that must *always* be true in the context they are applied (concept components, concept relationships).
- *Postconditions*, that are conditions that must be met *after* the execution of a method or operation of a specific class.

The constraints are applied to specific model elements, defined by the keyword **context,** as will be shown in the following example.

Based on these three models, the designer is called upon to create the conceptual, navigational and presentation designs of the WEAH.

2.1 The Conceptual sub-Model

The *Conceptual sub-Model* defines the concepts of the subject that is going to be taught with their semantic interrelationships. The main entity of the conceptual model is the C*oncept,* which depicts a main idea or topic of interest into the educational application. Concepts are abstract entities that do not carry actual content by themselves. They can contain meta-data or other descriptions, but the actual content is defined in the associated Resources. The *Resources* are the actual fragments of content that compose the WAEA, text, images, sounds, videos, etc, which are static, reusable components or dynamic components. Two (or more) concepts can be associated with *Relationships,* which capture the semantic links between these concepts. Both concepts and relationships in the Conceptual Model are described as attribute-value pairs.

2.2 The Navigation sub-model

The Navigation sub-model captures the decisions about how Concepts, Relationships and Resources of the Conceptual Model are mapped to actual hypertext elements Pages and Links, and how the conceptual relationships defined in the Conceptual Model are driving the structuring of the learning content. The *Navigation sub-model* is composed by two other sub-models:

2.2.1 The Navigation Structure sub-Model

This model defines the structure of the WAEA and defines the actual web pages and the resources contained in these pages.

This structure is composed of the following elements:

- *Content,* which is the top-level container in the hierarchy of an electronic content organization.

- C*omposite* entities that are used as containers, thus composing the hierarchical structure of learning content. The chapters and subtopics in which an electronic tutorial or book are organized are examples of composite entities.
- *Access structures* elements, namely *indexes* and *guided tours,* which are related to Content or Composite components
- *ContentNodes*, which are the actual pages of the learning content. Content, Composite and ContentNodes are associated with Concept elements, or directly with Resources, in the Conceptual Model.
- *Fragments* that are contained into the ContentNodes. Fragments correspond to Resource elements in the Conceptual Model.
- *Links* between ContentNodes as well as between Fragments. Note that these links are associative links [9, 20] implementing domain specific relationships of the conceptual model. They are not structural links denoting, for example, the transition from a page in the learning content to the next one.
- Composite, ContentNodes, Fragments and Links have a predefined attribute of Boolean type named *included*. This denotes whether or not a specific element (and all its descendants in the hierarchy) is included in the created hypertext or not, as a result of adaptation.

2.2.2 The Navigation Behavior sub-model

The *Navigation Behavior sub-model* defines the run-time behavior of the WAEA. Earlier research [7, 14, 24] has proposed the use of statecharts for the modeling of hypertext and web based applications. The Navigation Behavior model uses statecharts, as they are incorporated in the UML in order to specify the dynamic transitions of the hypertext structures as the user interacts with the WAEA. Every containing element of the Navigation Structure Model (Content, or Composite) is associated to a composite state in the Navigation Behavior Model, while every ContentNode corresponds to a simple state. Thus, the hierarchy of the navigational elements defined in the Navigation Structure Model corresponds to the hierarchy of nested states in the Navigation Behavior Model. The events that fire the transitions in the Navigation Behavior Model correspond to structure links into the ContentNodes: next, previous, up level, etc. In addition, guard conditions in these transitions can define alternative navigational transitions, which correspond to conditional behavior of the WAEA, thus implementing content sequencing and adaptive navigation.

2.3 The Presentation sub-model

The *Presentation sub-model* deals with the presentation aspects of the elements defined in the Navigation Model.

The presentation model is by itself separated in two additional sub-models: *Presentation Structure sub-model*, which defines the allocation of the navigational elements to actual user interface web elements: Web pages, frames, framesets, etc. Elements of this model, which is a variation of the synonymous model proposed in [14], are the following: *frameset,frame, window.* The aforementioned elements are associated with one or more elements of the Navigation Model.

User Interface sub-model, that captures the layout, colors, styles, etc of the entire web pages as well of atomic elements of the pages. This model consists of *Presentation* elements, which define the layout and style of associated elements of the navigation model.

3. A case study: An adaptive web-based testing system

We exemplify the design model by applying its concepts to an adaptive web-based testing system that conforms to the IMS QTI standard for question and test interoperability [http://www.imsproject.org]. The system enables either the editors to create/edit questions and tests (multiple choice, fill-in the blanks, etc.) or the simple users to be assessed answering to a series of questions of a test. The system supports adaptive exercise sequencing [7], customizing the succession according to which the the questions are launched to the user. The answer to a particular question (right or wrong) changes the sequence of the test questions according to specific simple rules. The testing system was developed using the Java technology [http://java.sun.com], as applets, and utilizes the extensible Markup Language [http://www.w3c.org/xml] for data storage. An example of the graphic user interface for a question is depicted in Figure 3.

In the example of Figure 3, a section of the test on the History of the World War II. contains seven questions. For the three first: question1, question2 and question3 we have applied the following simple rule: If a user answers correctly to the first question, the system skips question2 and immediately presents question3, else it continues with question2. As shown in Figure 3, the user can confirm his answer and then the button next is active, presenting a link to the next question.

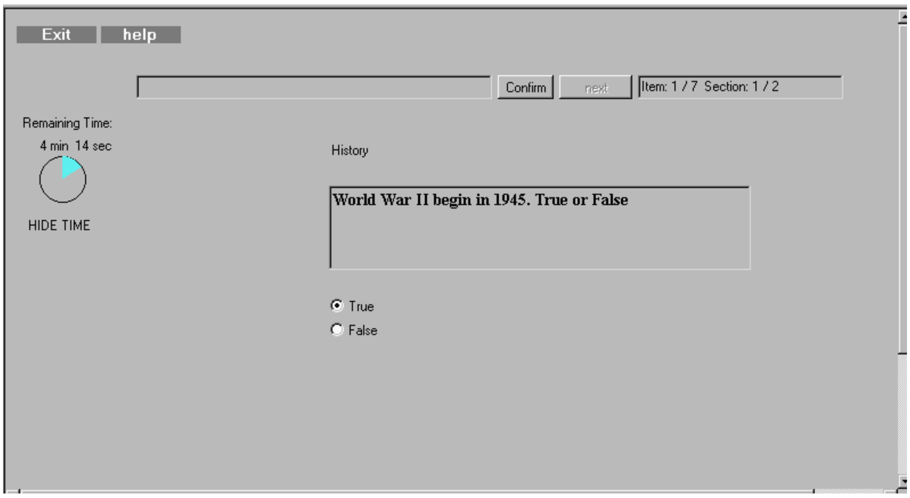


Fig. 3. An example of the user interface

As shown in Figure 4, the conceptual design model of this part of the system is presented. In this figure, we show some conventions that hold in the naming of the roles in the associations between the model elements. For example, when a concept component (e.g. the page entitled "question1") is associated with a UserModelScheme element, the role name of the UserModelScheme is, by convention, "user". An anchor is connected with its containing component with an association role named *uid*, which is the component's unique identifier. The role names of the specifiers related to a specific relationship are ss1, ss2, etc.

We will show how the previous rule can be defined in the model with OCL expressions (constraints) applied to the appropriate contexts. For the first constraint we have:

```
context question1::confirmed(): void
 post: if self.answer=self.correct_answer then
   self.user.answered=TRUE
 else self.user.answered=FALSE
```

This is a postcondition applied to the operation *confirmed* of the resource “question1”. The operation named *confirmed* is the default operation that is called whenever the user presses the Comfirm button while *accessed* is the default operation that is called whenever a component is accesed during the user’s navigation. This kind of relationship between a domain model component and its corresponding user interface element is typically defined with state transition diagrams. It defines that if the user gives the right answer, then the corresponding user model attribute *answered* is set to TRUE, else it is set to FALSE.

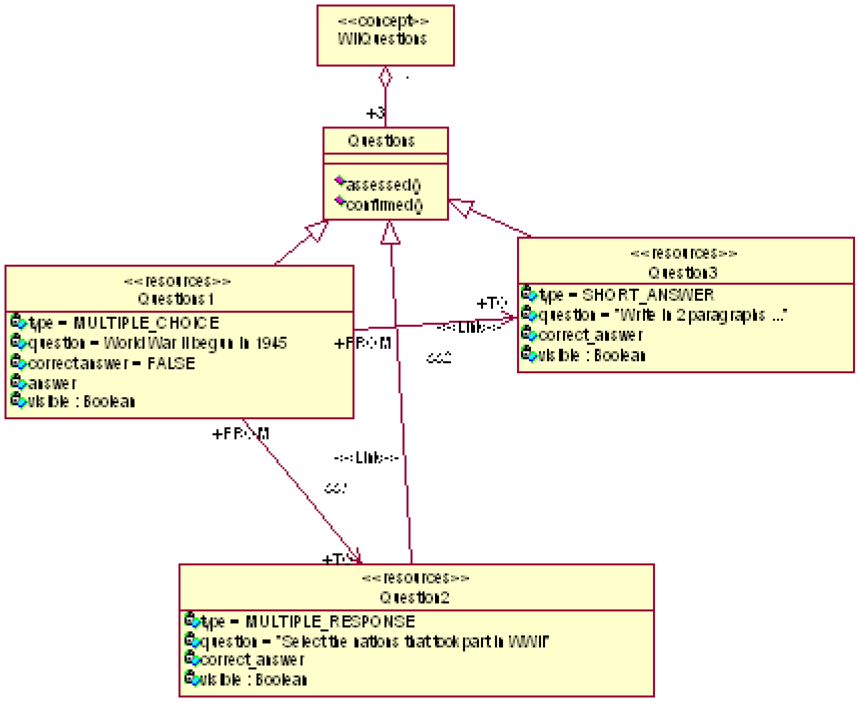


Fig. 4. The adaptive question system example

The second constraint is applied to the relationship named “relationship1” that concerns the links named “ss1” and “ss2”. If question1 is correctly answered then question2’s presentation specification attribute *visible* would set to SHOW and this question would be presented to the user. Else if the answer to question1 is wrong for the specific user then the opposite would happen, that is the question3 would be presented. This is defined with the following OCL expressions that are applied to relation1 as invariant conditions:

```
context relationship1 inv:
self.user.answered=true implies
 (self.ss1.TO.visible=SHOW and
 self.ss2.TO.pres.visible=HIDE)
context relationship1 inv:
self.user.answered=false implies
 (self.ss1.TO.visible=HIDE and
 self.ss2.TO.visible=SHOW)
```

Note that the keyword **implies** means that when the expression is evaluated then if the condition left to the “implies” keyword is true then the condition to the right must also be true, in order for the whole expression to be true. In all the previous OCL examples the keyword **self** represents the UML classes that are the context in the specific OCL expression. It is evident that the OCL rules have been extracted-prescribed by the teaching model.

4. Discussion

The definition of a Design Model can facilitate the process of developing software projects, regardless of the domain of the application [9]. This facilitation is even more important in fields where the people involved in the development process come from different backgrounds so there is an increased need for a means of communicating design decisions. However, this improvement is often *confuted* by the lack of formalism in the definition of such models. This lack of formalism has certain negative aspects:

- Poorly defined models, which are based on the intuition of the designers rather than in predefined 'rules'.
- It is impossible to automate the authoring of models by means of specific Case Tools.
- It is impossible to automate the process of automatic code generation based on the models created (forward engineering).

Up to our knowledge only one similar attempt has been made for WAEA. In [7] a layer approach for the modeling of Adaptive Educational Applications is provided, together with a method for the design of such applications. This approach is similar to ours in the distinction of three views of Adaptive Educational Application depicted as layers: A conceptual Layer, a lesson layer and a student adaptation and presentation layers, which resemble our separation in three sub-models, i.e. conceptual, navigational and presentational. A second main similarity is that both approaches recognize that the authoring of WAEA is driven by an initial mapping of the available resources in a high level conceptual model. The main differences from this approach are in the way of mapping of the initially defined concepts into specific navigation and presentation elements, as well as the specific formalism used in our approach, namely UML.

The conceptual model we propose is based on or influenced by previous established models for web and hypermedia engineering. HDM [10] provides a model for high-level hypertext design. Like Dexter model [11] preserves the hierarchical structure of hypertext nodes but in addition supplies domain specific concepts, namely *Entities* and*Components,* so facilitating the definition of a conceptual *schema* of a hypertext application. The application itself is considered as an implementation of the previous schema and thus the model provides separation between conceptual design and implementation. In the same manner our model provides a conceptual organization of the educational through structuring of learning resources. The implementation of the previous organization in terms of navigation structure, user interface layout and actual content creation is taking place in later steps than the conceptual design, described in this paper.

OOHDM, described in [17] clearly proposes the separation between Conceptual, Navigational and User Interface Design steps in the development of hypermedia applications, suggesting certain types of models for each step. We follow this separation in the process of designing WbEA, though in the present paper we only deal with conceptual design.

Conallen's approach [5]) introduces a UML extension for web application architecture modeling. While web-site modeling is more implementation oriented, introducing the web page as the core modeling element, a purely conceptual approach is adopted for the modeling of the business logic implemented in server side components. Unlike our model, it refers to generic web based applications and not particularly educational ones.

Hennicker and Koch [15] also extend UML for generic web application modeling. Unlike Conallen [5], they follow the distinction between conceptual, navigational and static presentation modeling.

Süß et al. [21] provide a UML extension by means of a meta-model for teachware management. Their meta-model can be separated into two sections, one concerning a conceptual model and one concerning a navigational model, implementing the previous one. They have also developed an XML based language, LM^2L that implements their meta-model. User Interface design as well as personalization of teachware is achieved by defining XSL stylesheets and applying them on LM^2L files.

EML [16] is not a strict educational hypermedia concept, but it defines a formal, XML based, language for modeling different aspects of the educational process e.g. activities, peoples' roles, content, etc, concerning specific lessons or courses called "units of study". It is a high level model aiming at facilitating instructional design in the context of e-learning.

This model has been tried out in small scale WAEA development projects. It still evolves but it seems that the notation used is valid and can be applied to every subject domain. The fact that this model uses a very specific and formal notation enables the development of a CASE tool that will support the design process. We do not expect the user to know OCL in order to describe the relationships and constraints applied. On the contrary, we are building a tool that will translate rules that the user will describe in "if then else" format (like the rules in the IMS sequencing [13]) into OCL that could be into understandable by an adaptive engine. Finally, we are in the process of modeling via reverse engineering a big WAEA in order to check the validity of this model with a "real" system.

References

1. P. De Bra and L. Calvi. AHA! An open Adaptive Hypermedia Architecture, *The New Review of Hypermedia and Multimedia*, vol. 4, pp 106-112 (2000).
2. P. De Bra, G.J. Houben and H. Wu. AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. In *Proceedings of the tenth ACM Conference on Hypertext and Hypermedia,* Darmstadt, Germany, February 1999. ACM Press.
3. P. Brusilovsky. Adaptive educational systems on the world-wide-web: A review of available technologies. In *Proceedings of Workshop WWW-Based Tutoring,* 4th International Conference on Intelligent Tutoring Systems (ITS'98), San Antonio, TX, 1998.
4. P. Brusilovsky. Methods and Techniqes of adaptive hypermedia. *User Modeling and User Adapted Interaction*, vol. 6, pp 87-129, 1996.
5. J. Conallen: Modeling Web Application Architectures with UML, *Communications of the ACM*, 42:10, Oct. 1999, pp. 63-70.
6. A. Cristea and L. Aroyo. Adaptive Authoring of Adaptive Educational Hypermedia. In *Proceedings of the 2nd International Conference on Adaptive Hypermedia and Adaptive Web System,* Malaga, Spain, May 2002.
7. M. C. Ferreira de Oliveira , M. A. Santos Turine and P. C. Masiero. A Statechart-Based Model for Hypermedia Applications, *ACM Transactions on Information Systems*, Vol. 19, Issue 1, 2001.
8. P. Fröhlich, N. Henze, and W. Nejdl. Meta Modeling for Hypermedia Design. In *Proceedings of IEEE Metadata Conference,*
Maryland, USA, Sept. 1997.
9. D. Garlan and M. Shaw. Software Architecture: Perspectives of an Emerging Discipline. Prentice Hall, New Jersey, 1996.

10. F. Garzotto, D. Schwabe and P. Paolini. HDM- A Model Based Approach to Hypermedia Application Design. *ACM Transactions on Information Systems*, vol. 11, n. 1, pp. 1-26, 1993.
11. F. Halasz and M. Schwartz. The Dexter Reference Model. *Communications of the ACM*, vol. 37, n. 2, pp. 30-39, 1994.
12. T. Isakowitz, E. Stohr and P. Balasubramanian. RMM: A Methodology for Structured Hypermedia Design. *Communications of the ACM*, vol. 38, n. 4, 1995.
13. IMS Global Learning Consortium Inc. *IMS Simple Sequencing Information and Behavior Model*. Available at http://www.imsproject.org/simplesequencing/
14. N. Koch and A. Kraus. The Expressive Power of UML-based Web Engineering. In *Proceedings of the 2nd International Conference Workshop on Web Oriented Software Technologies,* Malaga, Spain, June 2002.
15. N. P. de Koch. Software Engineering for Adaptive Hypermedia Systems, PhD Thesis, Verlag Uni-Druck, Munich 2001.
16. Rod Koper. Modeling units of study from a Pedagogical Perspective: the pedagogical meta-model behind EML. Available at http://eml.ou.nl.
17. D. Lowe and W. Hall. *Hypermedia & the Web, an Engineering Approach*, John Wiley & Sons, 1999.
18. OMG. UML Profile for CORBA Specification. Available at http://www.omg.org/uml.
19. OMG. Unified Modeling Language Specification. Available at http://www.omg.org/uml. September 2001.
20. A. Papasalouros and S. Retalis. Ob-AHEM: A UML-enabled model for Adaptive Educational Hypermedia Applications. *Interactive educational Multimedia*, vol. 4, 2002.
21. C. Süß, B. Freitag and P. Brössler. Meta-Modeling for Web-Based Teachware Management. In *Advances in Conceptual Modeling*, ER'99 Workshop on the World-Wide Web and Conceptual Modeling , Paris, France 1999. Springer.
22. M. Uschold and M. Grunninger. Ontologies: Principles, Methods and Applications. *Knowledge Engineering Review,* vol.1, n. 2, pp. 93-155, 1996.
23. J. Warmer and A. Kleppe. *The Object Constraint Language: Precise Modeling with UML.* Addison Wesley, Reading Mass., 1999.
24. Y. Zheng and M. Pong. Using statecharts to model hypertext. In *Proceedings of the ACM conference on Hypertext,* Milan, Italy, 1992, ACM Press.

Adaptive Presentation Supporting Focus and Context

Theophanis Tsandilas
Department of Computer Science
University of Toronto
Toronto, Canada
fanis@cs.toronto.edu

m.c. schraefel
Department of Electronics and Computer Science
University of Southampton
Southampton, UK
mc at ecs.soton.ac.uk

Abstract

This paper focuses on how content adaptation is provided in adaptive and adaptable hypermedia systems. Questions that we investigate are: How focus and context can be supported by content-adaptation techniques? Are there any techniques that can be easily generalized to adapt the content of generic Web pages without requiring much effort from the author of the pages? How different adaptation techniques should be compared? We propose a new technique of adaptive presentation of Web content, which derives from fisheye views. This technique applies adaptation by modifying the scale of the visual elements in Web pages. We present an adaptable Web application that applies the technique to a set of real-world pages. We also identify existing adaptation techniques that relate to the proposed technique and examine their strengths and weaknesses. Finally, we present and discuss the results of a pilot study which compared our fisheye technique against stretchtext adaptation. The results indicate that our technique is promising while they give valuable feedback about future work.

1. Introduction

Decreasing the cognitive overload caused by the presence of information which is irrelevant to the goals of Web users has been a main goal of Adaptive hypermedia (AH) systems. Following Brusilovsky's taxonomy [4], we can distinguish between two types of adaptation techniques employed by AH systems to support this goal: adaptive-presentation techniques and link-adaptation techniques. While link adaptation aims at providing navigational support to hypertext users, the goal of adaptive presentation is to adapt the content of the pages according to the users' goals, knowledge, language or other user characteristics.

In this paper, we present a new technique of adaptive presentation which is based on the use of multiple levels of zooming to adapt the content of a typical Web page. This technique is influenced by existing *focus+context* approaches for information visualization, in particular, fisheye views. We view content adaptation as a process of moving the focus within a page rather than hiding or changing parts of the page. Context is always visible and can be easily brought into focus by the user. This approach balances the trade-off between overloading the users with less relevant information and preventing them from having the control of the content in a page. Adapting the level of zooming of visual elements in a page can be considered as a new technique of *canned text adaptation* [4]. We demonstrate a Web application which integrates the technique into an adaptable, user-determined [14, 15] interaction model.

We acknowledge that other techniques of content adaptation [3, 8, 9] also allow users to access information that is out of focus. We discuss strengths and weakness of each of these techniques. We also present the results of a pilot study which compared *stretchtext*, which is a well-studied adaptation technique, against our technique. Finally, we discuss the implications of the results for future work.

2. Focus, Context and Fisheye Views

Supporting context and focus has been the goal of several techniques in the community of Human Computer Interaction. Most techniques are based on fisheye views [5], which provide both local detail and global context in a single display. Fisheye views have been applied to visualize information in several domains. Furnas [5] applied fisheye views to program code, tree structures and calendars. Fisheye techniques were used by Sarkar and Brown [12] to support viewing and browsing graphs. Bederson [1] applied fisheye zooming to pull-down menus with the goal to reduce the cognitive load caused by long lists of choices. Greenberg et al. [6] introduced fisheye views to support group awareness when multiple people work within a single window. The technique that we propose in this paper is highly inspired by their groupware fisheye text viewer.

Techniques based on fisheye views have also been applied to hypertext applications [7, 10]. These techniques provide fisheye views of collections of Web pages or hypertext networks rather than fisheye views of the content within pages. On the other hand, Bederson et al. [2] developed the *Multi-Scale Markup Language (MSML)*, a markup language implemented using the HTML `<Meta>` tag to enable multiple levels of zooming within a single Web page. Their goal, however, was to produce interactive Web pages which can be zoomed-in and zoomed-out rather than adapt the content of the pages according to user goals or interests. Finally, Tsandilas and schraefel [14] applied zooming to visualize hyperlinks. According to this work, hyperlinks that relate to user goals are presented with large fonts, whereas irrelevant hyperlinks are presented with small fonts. Font sizes are continuously changed as the user specifies interests by means of interactive manipulators.

Fisheye-view techniques define a *Degree of Interest (DOI)* function which specifies how the elements of the visualization are presented. The actual definition of the DOI function is application depended. Different approaches use different techniques to visualize information with respect to the DOI function. Noik [10] classifies fisheye-view approaches into two main categories: *filtering* and *distorting* fisheye views. Approaches that belong to the first category use thresholds to constraint the display of information to relevant or interesting elements. Approaches that belong to the second category, on the other hand, apply geometrical distortion to the visualization. This is usually performed by altering the positions and the sizes of the visualized elements, for example, elements of interest are zoomed in, whereas irrelevant elements are zoomed out. Fisheye-view techniques usually assume that there is a single focal point, and the value of the DOI function decreases with distance to this point. However, several fisheye approaches [6, 12] support multiple focal points at the same time.

3. Applying Zooming to Adapt the Presentation of Web Pages

In this section, we present how content adaptation can be achieved by varying the level of zooming of the visual elements in individual Web pages.

3.1 Expressing the Degree of Interest (DOI)

Adaptation provided by AH systems is based on a user model that captures information about the user. We mainly focus on information finding tasks that Web browsing involves, so we assume that the user model captures the user's current interests. More specifically, we consider a finite set of information topics $T = \{t_1,\ldots,t_n\}$ and represent the user model as a vector $I(i_1,\ldots,i_n)$, where i_i is a value that represents relevance between t_i and the current interests of the user. Each Web page is considered as a collection of individual segments s_j, which can be paragraphs, sections or other page parts. The content of each segment is represented by a vector $V_j(w_{j1},\ldots,w_{jn})$, where w_{ji} is a value that represents relevance between t_i and s_j. These values can be either assigned by a human, for example, the author of the page or automatically derived by using information retrieval techniques. For example, they can be calculated as the cosine [11] between the feature vector that represents the segment and the feature vector that represents the information topic [14]. In the rest of the paper, we assume that V_j is known for all the segments s_j in a page.

Based on the above discussion, we define the degree of interest DOI as the function:

$$DOI(s_j) = \sum_{i=1}^{n} i_i \cdot w_{ji}$$

According to this definition, $DOI(s_j)$ grows as the user's interests become relevant to the content of s_j. This definition of the DOI function differentiates from the original conception of fisheye views. Proximity is not measured in terms of geometrical distance, but it refers to the semantic distance between the content of the different segments in the page. Furthermore, the focal point is determined by the focus of the user's interests rather than the user's current focus of attention. Finally, it is clear that multiple focal points are supported as multiple segments in a page may be relevant to the user's interests.

3.2 Visualization

Page adaptation is based on the DOI function that was presented in the previous paragraph. More specifically, assuming that l_{max} is the maximum size of a visual element within s_j, adaptation is performed by adjusting its size to the value $l = l_{max} \cdot DOI(s_j)$, where the DOI value has been normalized between 0 and 1. In order to prevent a page element from being totally hidden when the associated DOI value is very small, we consider a value l_{min} which determines the minimum size that the element can have. The size of any element within a page can be adapted. For instance, text is adapted by modifying its font size and images are adapted by modifying their height and width. Figure 1a demonstrates the application of the technique to the paragraphs of a page.

Fig. 1. Adapting the size of the text and images in a Web page

Although the technique changes the size of the various visual elements, other features of the page's layout are preserved. The reader of the page gets direct feedback about the quantity and the structure of the material within the minimized paragraphs. In other words, while only the most relevant parts of the page are on focus, contextual information about the content, quantity, and layout of less relevant or irrelevant parts of the page is provided. Another advantage of the technique is that multiple degrees of relevance can be represented, as the sizes of the elements can be assigned a wide range of values. It can represent multiple variations of relevance between the content of the segments of the page and the user's interests and also capture the uncertainty of the adaptation algorithm about the interests of the user. However, dual representations can also apply by distinguishing between two only sizes for each element type. Such an approach could be considered as a filtering fisheye technique, since a threshold value of DOI is used to determine the size of the page elements. Its main advantage is that it provides clarity requiring the user to distinguish between only two different states of adaptation.

3.3 Prototype implementation

In order to test the adaptation technique and explore the interaction issues that it involves, we applied it to a small set of Web pages. These pages were taken from a Web site about cultural events in Toronto. We first decided on a small set of topics that related to these pages, such as music, dance, theatre, visual arts, and cinema. Each page element, and more precisely each paragraph and image in the pages was associated with a vector of values, where each such value specified the relevance between the element and a particular topic. This was achieved by assigning a unique identifier to the each element in the page and adding JavaScript statements to define the associated vector of relevance values. Additional JavaScript code was implemented in separate files to provide functionality for dynamic adaptation of the different elements.

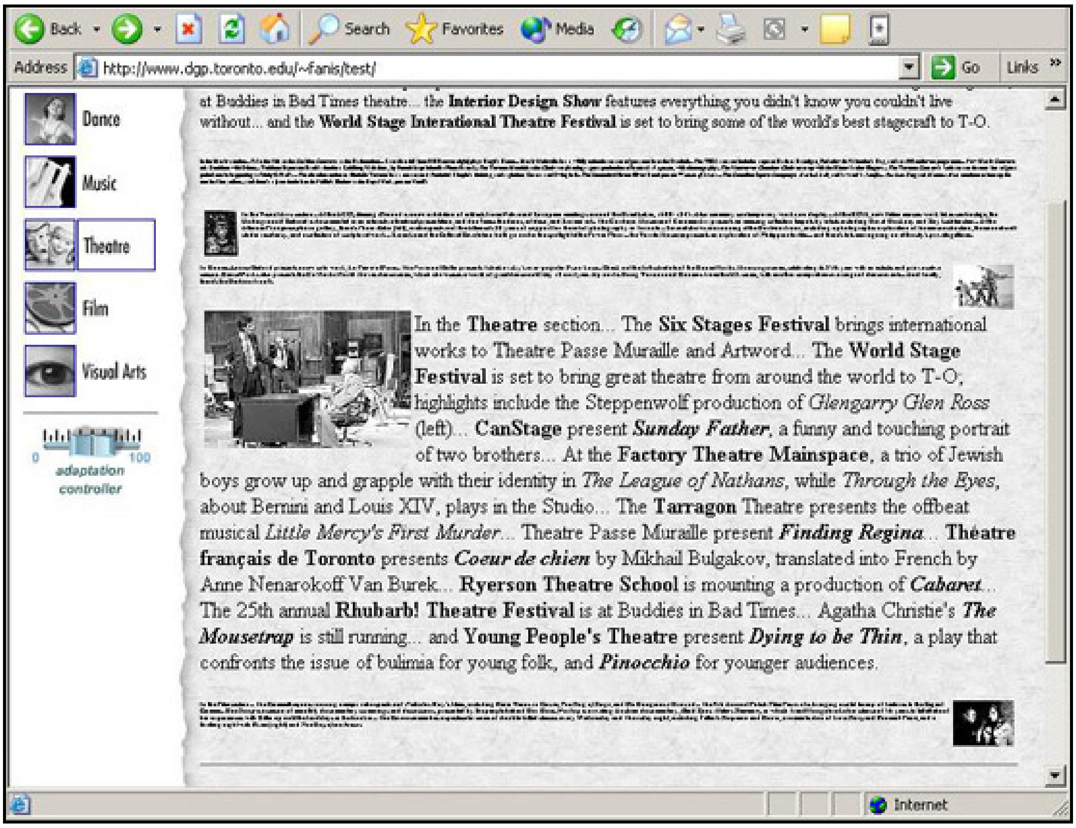


Fig. 2. Paragraphs related to theatre are on focus

Figure 2 presents a view of a page when the topic theatre is on focus. As shown in the figure, pages are presented in the right frame of the browser’s window, while the left frame is used to control the adaptation. Five icons at the left frame correspond to different topics. The user can change the focus of the browsing session by clicking on a different icon. This change is reflected to the presentation of the page in the right frame, where the size of the page's elements is adapted based on their relevance with the selected topic. In order to achieve smooth transitions between the consequent views of a page, we employ animation which is performed by gradually changing the sizes of the elements in the page. As the user navigates between different pages the information about the current focus is preserved and new pages are adapted accordingly.

A first version of the prototype helped up to identify some problems and think about their solutions. These problems concerned the legibility of text that appeared out of focus and the lack of a simple and quick mechanism that would enable users to easily bring this text into focus. Arguments supporting the value of presenting paragraphs with illegible fonts derive from previous research [2, 7] and commercial products such as Acrobat Reader which have successfully employed thumbnails of pages to provide context. The layout of a page and the presence of elements like pictures are preserved by a page’s thumbnail and can provide valuable hints about its actual content. However, individual paragraphs cannot provide as rich layout information as whole pages do. Furthermore, the above approaches provide mechanisms that allow users to easily bring pages into focus. The easiest solution to the above problem is to constraint the minimum value of the font sizes so that the text is always readable. As different people have different visual abilities and in order to balance between illegibility of text and cognitive overload caused by extra information brought to the user's attention, we decided to add a slider in the left frame of the pages as Figure 2 demonstrates. The slider was implemented in Flash MX. By moving the slider, the user can adjust the sensitivity of the adaptation and modify the minimum size of the text fonts. When the slider moves to zero, no adaptation is performed. The slider allows the user to control the adaptation process and adjust it according to his/her current goals.

Another technique that we use to handle the illegibility problem is the use of glosses that are activated when the user moves the mouse over paragraphs with small font sizes. A gloss provides hints about the content of the paragraph, such as a list of the most relevant topics. Glosses have been used by other systems to provide information about hyperlinks [16] or hidden text within pages [13].

Finally, we provide a mechanism that allows fluid transitions of individual paragraphs form context to focus. More precisely, by double-clicking on a paragraph that is out of focus, the user can zoom in the text of the paragraph together with its containing images. Animation is used to smoothly change the zooming level. If the user double-clicks again, the paragraph is zoomed-out to its initial size. This mechanism can be considered as a local rather than a global change of focus. The global adaptation of the page is not affected when a paragraph is double-clicked. In other words, temporary changes in the user's attention are not translated into switches of the user's current interests.

In summary, users specify the focus of their browsing tasks by selecting a topic of interest, which determines the adaptation of the viewed pages. By hovering the mouse over minimized paragraphs, they can get fast feedback about the content of the paragraphs. By double-clicking on them, they can maximize them and read the content in detail. If they decide to switch the focus of the adaptation, they can click on a different topic and change the view of the page. Finally, they can control the degree of the adaptation by manipulating the slider. This interaction model gives powerful control to the user and at the same time, provides smooth transitions between the different views of the page. Since adaptation is not automatic but is directly controlled by the user, the prototype can be considered as an adaptable rather than an adaptive system. However, the generalization of the approach to automatic adaptation is straightforward.

4. Related Techniques of Content Adaptation

Brusilovsky [4] identifies five different techniques for adapting canned text: (1) inserting or removing fragments, (2) altering fragments, (3) stretchtext, (4) sorting fragments, and (5) dimming fragments. Scaling fragments that our approach suggests can be considered as a sixth technique. Each of the above techniques has advantages and disadvantages. The two first techniques provide only focus but not context. In addition to that, the second technique requires additional effort by the author of the adaptive pages who has to provide different versions of a fragment's content for each user type. The fourth technique provides both focus and context but their boundaries are not clearly shown. It may also not be clear to the user that order of presentation signifies order of importance or relevance. Its main disadvantage, though, is that reordering the fragments within a page can disturb the natural flow of information and the text may become incomprehensible.

The techniques that best support both focus and context and highly relate to our technique are stretchtext and dimming. Stretchtext enables users to expand and collapse additional text within a page. MetaDoc [3] was the first system that employed stretchtext as an adaptation technique. It provided different views of hypertext documents for users with different expertise. PUSH [8] also used stretchtext to adapt the content of hypertext documents to different information tasks. The advantage of the above approaches is that although text that is judged as irrelevant or redundant is hidden, the user can open it by clicking on a link or a representative icon. The amount of context that is provided by this approach depends on the ability of the link anchor or the icon to inform the user about the content of the hidden fragment. Compared to our technique, the main disadvantages of stretchtext are: (1) it does not provide any feedback about the quantity and layout of the hidden information; (2) support of context depends on the selection of a representative text or icon for the adaptable fragment, which is a procedure that needs special design considerations from the author of the page; and (3) it can visualize only two states of adaptation, i.e., fragments are either visible or hidden.

Dimming was introduced by Hothi et al. [9]. According to this approach, parts of the document containing information that is out of the user's focus are shaded instead of being hidden or zoomed-out. Information in context, in this case, is rich and directly accessible. Multiple states of relevance could also be represented by applying multiple levels of shading. The main drawback of dimming is that it does not reduce the size of the adapted page. Also, although shaded, irrelevant information can easily gain the attention of the users disrupting them from their main task.

5. Pilot Study

The most common approach of evaluating an adaptive system is to compare it against its non-adaptive version. This approach was adopted by the evaluations of both MetaDoc [3] and PUSH [8]. Although these evaluations showed that the adaptive versions of the systems improved the users' performance in several information tasks, they did not explain whether the employed adaptation technique, i.e., stretchtext, was better than other adaptation techniques. The question that arises is whether other adaptation techniques would have similar or better results if used by the same systems. It is not clear whether it was the particular adaptation technique or it was the efficiency of the adaptation mechanism that resulted in the observed improvements in the users' performance.

Comparing two adaptation techniques is not an easy task. Different adaptation techniques have been designed for different domains and different tasks. They cannot easily be separated by the system in which they have been used. However, as our goal is to identify techniques the use of which can be easily extended to generic Web pages and common browsing tasks, we conducted a preliminary study which compared two different adaptation techniques: the zooming technique that we presented in Section 3, and a stretchtext-like technique.

5.1 The techniques

In order to simplify the evaluation procedure and avoid biased conclusions in favour of one technique, we tried to eliminate the differences between the implementations of the two techniques. Thus, we focused on a single variation of the interface, which is the way that out-of-focus paragraphs are visualized. In the case of the zooming technique, we used a single level of zooming to present paragraphs in context. The fonts were selected to be legible and glosses were disabled. The topic icons and the slider in the left frame were removed as well.

The stretchtext version was based on the same implementation. The font size of paragraphs in context was set to zero. However, each paragraph had a representative title or introductory sentence whose font size was not adapted. The interaction model was exactly the same for both cases. Depending on the technique, the user could double-click on the body of the minimized paragraph or the paragraph's title to zoom in or expand, respectively, the paragraph. By following a similar procedure, the user could minimize or collapse the paragraph. Animation was used in both cases to smooth these transitions. Figure 3 shows two versions of the same page corresponding to the two different techniques that we tested.

(a) Applying the zooming technique (b) Applying the stretchtext technique

Fig. 3. Zooming and stretchtext adaptation applied to the same page

5.2 Design of the study

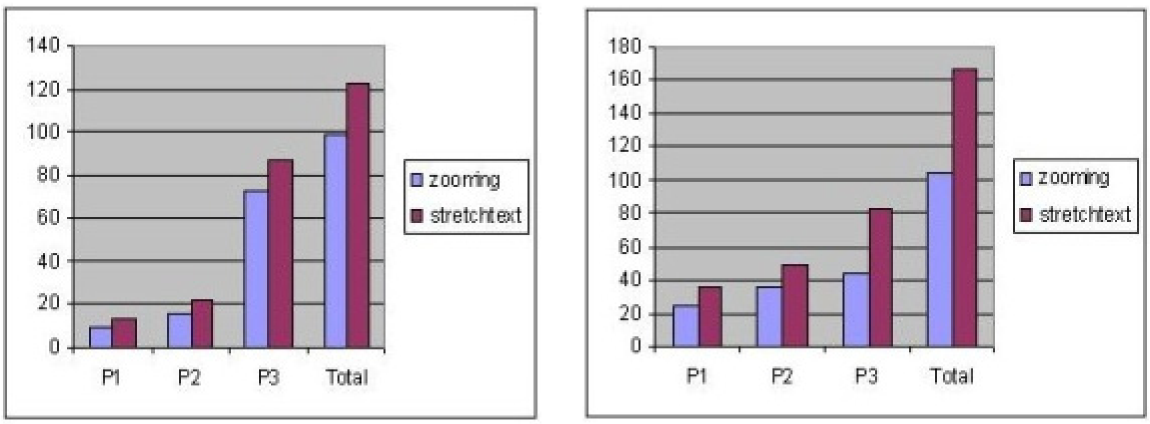
The pilot study was conducted in the Digital Graphics Project (DGP) laboratory in the University of Toronto. All the sessions were performed on the same machine with a 18-inches flat monitor. We used Internet Explorer v6.0 and 1280x1024 pixels as screen resolution. Times New Roman was used as text font, with size set to 18px for in-focus text, and 10px for out-of-focus text in the case of the zooming technique. This resulted in page layouts like the ones presented in Figure 3. User actions were captured using JavaScript. JavaScript code posted the captured user actions together with time stamps to a servlet running locally, which saved the data into a log file.

Two female and four male subjects participated in the study. They were all graduate student in Computer Science. Subjects had to complete 12 different tasks on three different pages which involved information about cultural events in Toronto. The first page (P1) contained 6 paragraphs, the second page (P2) contained 8 paragraphs, and the third page (P3) contained about 75 paragraphs. Only P1 and P2 contained pictures. Links were removed from the pages, i.e., only navigation within the pages was allowed. Tasks were divided into two main categories. The goal of the first 6 tasks was to compare the ability of each technique to help users to locate information within the three pages. Each of these 6 tasks involved two subsequent questions. The first question asked the subjects to locate one piece of information contained in a paragraph that was in focus. The second question asked the subjects to locate one piece of information contained in a paragraph that was either in focus or in context. The goal of the other 6 tasks was to test the ability of each technique to support information gathering. The subjects were asked to find either 3 pieces of information within P1 or P2 or 5 pieces of information within P3 that satisfied a particular condition. In the first case, 2 out of the 3 answers were in paragraphs displayed in focus, while one answer was in a paragraph that was in context. In the second case, 3 out of the 5 answers were in paragraphs displayed in focus, while 2 answers were in paragraphs that were in context. The page adaptation was fixed for each task and was based on a small set of topics such as music, dance, theatre and cinema. The subjects were not given any information about the adaptation mechanism.

The subjects could answer a question by selecting with the mouse the relevant piece of information and clicking on the "select" button in the left frame of the browser's window. They were told that they should give answers as precisely as possible. They were not allowed to use any search facility of the browser. The study was designed considering only one independent variable, the adaptation technique. Dependent variables that we considered were the number of correct answers, the number of double-clicks on paragraphs, and the mean time that the subjects spent for each answer. The subjects were split into two different groups. The tasks to which the two techniques were applied were switched between the two groups. All the subjects tried both techniques in similar tasks. In order to eliminate the learning factor, the sequence of the tasks was different for each subject in a group. Before the beginning of the main session, the subjects were trained to locate and gather information using the two techniques on a fourth page taken from the same Web site. The time spent for training was about 10 minutes, while the time spent for the main session varied from 30 to 40 minutes. At the end of the experiment, the subjects were asked to fill in a questionnaire which allowed them to rate the two techniques and give us additional feedback.

5.3 Quantitative results

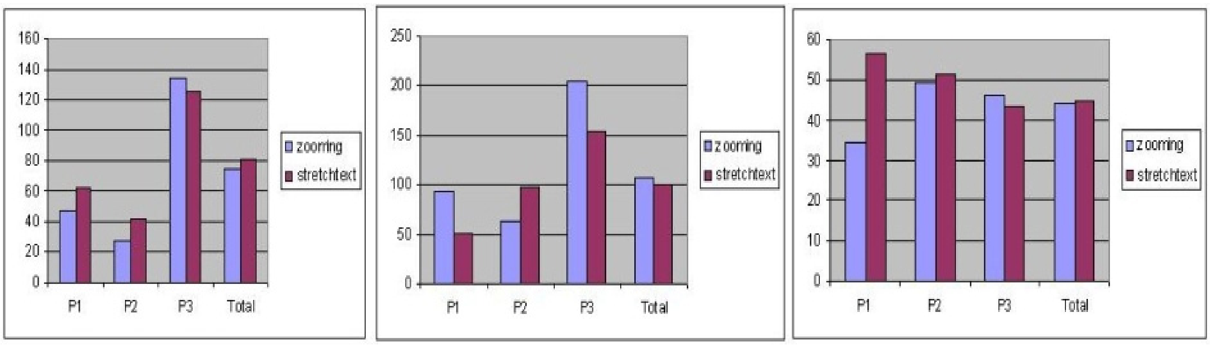
Figure 4 presents the distribution of double-clicks for the two approaches. As shown in the figure, the number of double-clicks was greater for both types of tasks and for all the three pages when stretchtext was used. This result was expectable since when the zooming technique was applied, the subjects could read the text without having to maximize it. However, the results show that although the text in the out-of-focus paragraphs was readable, the subjects used the zooming mechanism rather frequently. The logged user actions also showed that almost always and for both techniques, a double-click maximizing a paragraph was followed by a double-click minimizing the paragraph. Exception to this rule was the behaviour of one subject who preferred to maximize a small number of paragraphs without minimizing them after.



(a) Tasks for locating information (b) Tasks for gathering information

Figure 4. The distribution of the number of double-clicks for the two techniques

The analysis of the answers that the users gave did not lead to any conclusion in favour of one technique or the other. All but four answers, which were equally distributed between the two techniques, were correct. Also, one gathering task was misunderstood by a user, so the corresponding data was not included in the analysis. Figure 5 presents the mean times that the subjects spent before giving an answer. We show again how the times were distributed among the three pages. Figure 5a shows the mean times spent for answers that involved locating information that was in focus, Figure 5b shows the mean times for answers that involved locating information that was in context, and Figure 5c shows the mean times for answers that were part of information gathering tasks. The results show that there were no differences in the total mean times for the two techniques. In contrast to our expectations, the zooming approach did not perform better in tasks that involved locating information in context. We think that the reason for this result is the cost that is associated with reading small font sizes, which was not quantified by our experiment. It seems that expanding or zooming in a paragraph can accelerate the reading process. Another observation that we can make is that stretchtext performed better on the large page, whereas the zooming technique performed generally better on the two smaller pages. One possible explanation for this is that zoomed-out paragraphs occupy significantly larger space than simple paragraph titles. This results in greater scrolling times within pages. This phenomenon became intense in the case of the large page reducing the performance of the zooming technique.



(a) Locating information in focus (b) Locating information in context (c) Gathering information

Figure 5. Mean times per answer

5.4 Preferences

The analysis of the subjects' ratings and comments about the two techniques shows a slight advantage of the zooming technique over the stretching technique. 3 out of the 6 subjects liked the zooming technique more than the stretching technique for information locating tasks, 2 subjects were not sure, while one subject preferred the stretching technique. 4 subjects rated higher the stretchtext technique for locating information that appears in focus, while one subject rated higher the zooming technique. One the other hand, 4 subjects preferred the zooming technique for locating information that appears in context over one subject who preferred the stretchtext technique. We can observe that the preferences of the subjects do not match the mean times of the performed tasks as presented in Figure 5a-b. This shows that the task completion time should not be the only measure for evaluating adaptive techniques. Concerning information gathering tasks, 2 subjects rated the zooming technique higher; one subject rated the stretchtext technique higher, while equal scores were given by the three other subjects. The last question of the questionnaire asked the subjects to evaluate the zooming and the stretchtext technique in overall and compare them against common browsers which do not provide any kind of adaptation. 3 out of the 6 subjects rated the non-adaptation approach higher than the two adaptation techniques. This seems to be reasonable since during the tests, page adaptation did not always facilitate the tasks that the subjects had to perform. Finally, 4 subjects rated higher the zooming technique over the stretchtext technique.

We should note that one subject rated the zooming technique low in all the questions. He commented that in contrast to the zooming technique, the stretching technique provided summarization and abstraction which facilitated his browsing tasks. He also disliked the way zooming was implemented. He said that as text became larger, the number of lines and the position of the words in the resized paragraphs changed and this made him loose the flow of information. Another comment that we received was that the small fonts were not easily readable. Sometimes the subjects had to move closer to the screen to read the text. Careful selection of the fonts could reduce this problem, but variations in the visual abilities of different people should also be considered.

6. Conclusions and Future Work

In this paper, we introduced a new technique based on fisheye views for adapting the presentation of content in Web pages. We stressed the need of supporting both focus and context when adapting the presentation of a page's content. We argued that the proposed technique can balance the trade-off between information overload and lack of context. We conducted a pilot study to compare the technique against stretchtext-based adaptation. The small number of subjects that participated in the study does not allow us to make general claims about the efficiency or usefulness of our technique against stretchtext. The study, however, indicated that the fisheye technique seems promising. We should mention that although both techniques that we compared performed almost equally in terms of the time that the subjects spent to complete their tasks, the authoring of the stretchtext pages required additional effort. We had to manually select appropriate sentences in the paragraphs as representative titles. The structure and content of the Web site on which we based our experiments facilitated this task, but this might not be the case if the technique was applied to other pages. Our study did not evaluate how the expressiveness of the paragraph titles could affect the performance of stretchtext adaptation. It also revealed issues that were not taken into consideration from the beginning. In a future evaluation, we have to consider variables such as the size of the pages, and the size and legibility of the fonts. We also plan to evaluate navigation between pages in addition to navigation within pages. Finally, it would be interesting to examine how the two compared techniques could be integrated. Representative titles could be combined with zoomed-out page fragments. This approach could provide both summarization and feedback about the layout and the size of the adapted fragments.

Our evaluation differentiated from other evaluations of AH systems which usually measure the performance of the adaptive system against its non-adaptive version. We believe that comparing an AH system against its non-adaptive version does not clearly evaluate the performance of the adaptation technique, since the results depend on the efficiency of the underlying adaptation algorithm. A future goal, however, is to study how different adaptation techniques affect the threshold over which an adaptive system starts performing worse than its non-adaptive version.

Acknowledgments

The pages used in the prototype and the pilot study were taken from http://www.whatsuptoronto.com. We thank Luke Murphy for his permission to use the content of the pages. We also thank the students from the Department of Computer Science in the University of Toronto who participated in our study.

References

1. Bederson, B. B. Fisheye menus. 13th Annual ACM Symposium on User Interface Software and Technology, UIST, San Diego, California, United States, ACM Press, 2000.
2. Bederson, B. B., J.D. Hollan, J. Stewart, D. Rogers, A. Druin, and D. Vick. A zooming web browser. In SPIE Multimedia Computing and Networking (Proceedings of the SPIE, San Jose, CA, January, 1996), volume 2667, pages 260--271, 1996.
3. Boyle, C. and Encarnacion, A. O. MetaDoc: An Adaptive Hypertext Reading System, User Modeling and User-Adapted Interaction, Vol.4, No.1, pp.1-19, 1994.
4. Brusilovsky, P. Adaptive hypermedia. User Modeling and User Adapted Interaction, Ten Year Anniversary Issue (Alfred Kobsa, ed.) 11 (1/2), 87-110, 2001.
5. Furnas, G. W. Generalized Fisheye Views. CHI '86, Boston, Massachusetts, USA, ACM Press, 1986.
6. Greenberg, S., C. Gutwin and A. Cockburn. Using Distortion-Oriented Displays to Support Workspace Awareness. People and Computers XI (Proceedings of HCI '96), Springer-Verlag, 1996.
7. Holmquist, L.E. The Zoom Browser: Showing Simultaneous Detail and Overview in Large Documents. In Human IT vol. 2 no.3, ITH, 1998.
8. Hook, K. Evaluating the utility and usability of an adaptive hypermedia system. In: J. Moore, E. Edmonds and A. Puerta (eds.) Proceedings of 1997 International Conference on Intelligent User Interfaces, Orlando, Florida, January 7-9, 1997, ACM, pp. 179-186, 1997.
9. Hothi, J. and Hall, W. and Sly, T. A Study Comparing the Use of Shaded Text and Adaptive Navigational Support in Adaptive Hypermedia. Lecture Notes in Computer Science 1892:335-342, 2000.
10. Noik, E. G. Exploring large hyperdocuments: fisheye views of nested networks. Proceedings of the 5th ACM Conference on Hypertext, Seattle, Washington, United States, ACM Press, 1993.
11. Salton, G. Developments in automatic text retrieval. Science 253: 974-979. 1991.
12. Sarkar, M. and M. H. Brown. Graphical Fisheye Views of Graphs. CHI '92 Conference, Monterey, CA United States, ACM Press, 1992.
13. schraefel, m.c. ConTexts: Adaptable Hypermedia. Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, Trento, Italy, 2000.
14. Tsandilas T. and m.c. schraefel. User-Controlled Link Adaptation. To appear in the 14th ACM Conference on Hypertext and Hypermedia, Nottingham, UK, 2003.
15. Wadge, Wm. and m.c. schraefel. A Complementary Approach for Adaptive and Adaptable Hypermedia: Intensional Hypertext. Third Workshop on Adaptive Hypermedia, August 15, 2001. Aarhus, Denmark.
16. Zellweger, P. T., B.-W. Chang and J. D. Mackinlay. Fluid links for informed and incremental link transitions. Proceedings of the 9th ACM Conference on Hypertext and Hypermedia: links, objects, time and space-structure in hypermedia systems, Pittsburgh, Pennsylvania, United States, ACM Press, 1998.

Towards a Decentralized Search Architecture for the Web and P2P Systems

Jie Wu
Swiss Federal Institute of Technology, Lausanne
School of Computer and Communication Sciences
1015 Lausanne, Switzerland
jie.wu@epfl.ch

Abstract

Search engines are among the most important applications or services on the web. Most existing successful search engines use a centralized architecture and global ranking algorithms to generate the ranking of documents crawled in their databases, for example, Google's PageRank. However, global ranking of documents has two potential problems: high computation cost, and potentially poor rankings. Both of the problems are related to the centralized computation paradigm. We propose a decentralized architecture to solve the problem in a P2P fashion. We identify three sub-problems in the big picture: a logical framework for ranking computation, an efficient way of computing dynamic local ranking, and a cooperative approach that bridges distributed local rankings and collective global ranking. In the paper we summarize the current knowledge and existing solutions for distributed IR systems, and present our new ideas. We also provide initial results, demonstrating that the use of such an architecture can ameliorate the above-mentioned problems for Web and P2P search engines.[1]

Keywords:

search engines, information retrieval, P2P systems, link analysis, swarm intelligence, decentralized algorithms

1. Introduction

Search engines for large scale distributed systems, e.g. the Web, the emerging P2P systems, face two radical challenges: a huge collection of documents and the processing of them in preparation for information retrieval, and the generation of a proper ranking of the huge number of documents. The state-of-the-art technologies of dealing with these two problems have big limitations such as high computation cost, potentially poor rankings, etc.. The focus of my PhD thesis work is to develop a decentralized architecture for efficiently searching and ranking documents with returned results of high quality. The work covers three main issues in the big picture of my new decentralized search architecture: firstly, a mechanism inspired by Swarm Intelligence of obtaining more dynamic and more semantically meaningful rankings of documents local to Web sites; secondly, a ranking algebra which provides the algebraic ground of computing document rankings; and finally, the idea of global Web site ranking which is the key to establish the global Web document ranking in a decentralized way, and a decentralized algorithm of computing the global Web site ranking. Substantial results have been achieved and further work is going on smoothly.

2. The Research Question

We brief the established IR models [13] here at first. Then we see why these models do not fit well the Web IR systems.

2.1 Centralized Search Systems

The classical model for a centralized IR system is: $S = (T, D, Q, \delta)$ where D is a document collection, Q is the set of queries, and $\delta : Q \to 2^D$ is the set of mappings which assign every query to a set of relevant documents. T is a set of distinct terms where two relations are defined: *synonymous*: $\rho \subset T \times T$ where $\rho(t_1, t_2)$ implies that t_1 is a synonym of t_2; *general* : $\gamma \subset T \times T$ where $\gamma(t_1, t_2)$ implies that t_1 is a more general term than t_2. Many IR systems use a thesaurus T to expand a user query by including synonyms of the keywords in the query. An example of a valid generalization is $\gamma(animal, fish)$. A partial ordering of documents can be defined based on the concept of generalization. Let $t(d_i)$ indicate the list of unique, non-mutual synonymous keywords[2] of document d_i. Partial ordering $\preceq$ is defined as:

$$t(d_1) \preceq t(d_2) \Leftrightarrow (\forall t' \in t(d_1))(\exists t'' \in t(d_2))(\gamma(t', t''))$$

This is a partial ordering because two documents with terms that have no relationship between any pairs of terms will be unordered. What is mainly used in query processing of IR systems is the so-called $inclusiveness$ property of this model. An IR system is $inclusive$ only when the documents corresponding to a general query q_1 must be a superset of all documents corresponding to a more specific query q_2 where $q_1 \preceq q_2$:

$$\begin{array}{c} (q_1, q_2 \in Q) \\ (q_1 \preceq q_2) \to (\delta(q_1) \supset \delta(q_2)) \end{array}$$

The advantage of being $inclusive$ is that, if two queries q_1 and q_2 are presented such that $\gamma(q_1, q_2)$, it is not necessary to retrieve from the entire document collection D for each query. Rather the system can obtain the answer set $\delta(q_1)$ for q_1, and then simply search $\delta(q_1)$ to obtain the $\delta(q_2)$.

2.2 Model of Distributed Information Retrieval

A model of decentralized IR can be built by partitioning the centralized IR system $S = (T, D, Q, \delta)$ into n local IR systems $S_i = (T_i, D_i, Q_i, \delta_i), i = 1, \cdots, n$, where T_i, D_i, Q_i, δ_i are the individual thesaurus, document collection, set of queries, and mapping from queries to document sets of each local IR system. The whole distributed IR system can be redefined as $S = (T, D, Q, \delta)$ where $T = \bigcup_{i=1}^{n} T_i$, $D = \bigcup_{i=1}^{n} D_i$, and

$$Q \supset \bigcup_{i=1}^{n} Q_i, \preceq_j = \preceq \bigcap (Q_j \times Q_j)$$

which means the queries can be obtained by combining the queries at each local site. Moreover, the partial ordering at each site j only pertains to the queries at site j. As for each query in the grand system, the document collection for a query contains the documents whose descriptors are at least as specific as the query.

$$(\forall q \in Q)(\forall d \in \delta(q) \in 2^D, q \preceq t(d))$$

Based on this model, the hierarchy represented by γ is established and partitioned among the different sites. A local site at a lower hierarchy is called a $\boldsymbol{subsystem}$ of a higher one if it satisfies several specific criteria. [13] A query sent to the distributed IR system is then forwarded to the local $\boldsymbol{subsystem}$s where a local query is performed. The local responses are afterwards sent back to the originating site where the final result set is combined from the local ones. For example, if S_1 is a $\boldsymbol{subsystem}$ of S_0, then the query results at site S_0 contain those found in S_1:

$$\forall q \in Q, \delta_1(q) = \delta_0(q) \bigcap D_1$$

2.3 Problems of Existing Web Search Systems

Web IR systems, usually referred as Web search engines, are special IR systems, which can be built in a centralized or decentralized fashion. They are quite different from traditional IR systems mainly in the size of the document set, the organization of the set (ad hoc but linked by Web links vs. mostly independent), and the way the document set is built (by crawlers vs. according to specific criteria chosen by the people preparing and collecting the documents). For Web search engines, ranking computation of documents is a key component to return results highly satisfying users' information needs as searchers are usually only interested in the top few retrieved documents. There are decentralized search systems studied and built according the distributed IR model, including meta-engines and research-oriented prototypes, which however never reached the level comparable with non-meta engines. In general, large-scale experiments have not been seen using these approaches so their effectiveness and efficiency for Web search engines remain unknown. The classical model of distributed search systems briefed above is suitable for traditional information systems such as library, static collection of medical documents, etc.. When dealing with information seeking in Web search engines for the following reasons, it has some original sin because proper partition of a hierarchy is extremely important in this model, otherwise the resulting hierarchical subsystems may not be valid in the sense of returning correct search results, namely the result-containing property may be broken. [13] The reason behind is that the partitioning of the Web is not controllable by people. We can not re-organize the whole Web according to *Document partitioning* or *term partitioning* as we wish and do in a traditional distributed IR model. We believe these are the reasons why main Web search engines take a centralized architecture and mainly rely on global ranking algorithms. Global ranking algorithms, e.g, Google's PageRank, for centralized search engines, have been extremely succussful as people have known. However, global ranking of documents has two potential problems: high computation cost and potentially poor rankings. Both of the problems are related to the centralized computation paradigm. [5] There are also more specific problems because of the unique properties of the Web:

1. Coverage studies show that a small percentage of Web pages are in all search engines. Moreover, fewer than 1% of the Web pages indexed by AltaVista, HotBot, Excite, and Inforseek are in all of those search engines. [8] This fact also justifies the use of meta search engines, which however never reached the level of success comparable with non-meta engines.
2. It is likely that the larger the indexed subset of the web, the higher the recall and the lower the precision, for a given query. Query-based search engines still return too much hay together with the needle. One possible reason accounting for this is the current ranking algorithm is not really capable of differentiating the Web documents in the huge Web collection pertaining to the queries.
3. On the other hand, Web directories do not have enough depth to find the needle. The reason is that they are usually compiled manually or semi-automatically thus the timeliness and availability are largely limited. A reasonable decentralized architecture will enhance the situation greatly.

Thus we propose to decentralize the task of searching and ranking. In our work, first of all we introduce a ranking algebra providing such a formal framework. [5] Through partitioning and combining rankings, we manage to compute document rankings of large-scale web data sets in a localized fashion. Secondly we propose innovative ways of computing Web document rankings based on ideas inspired by Swarm Intelligence. [6] Thirdly we put dynamic interactions among the Web servers in our architecture that enables the decentralized Web search system to compute timely and accurate global rankings in a Peer-2-Peer fashion. We achieve initial results, demonstrating that the use of such an approach can ameliorate the above-mentioned problems. The approach presents a step towards a decentralized search architecture for Web and P2P systems.

3. Existing Works and Their Limitations

Research on distributed IR systems has not been limited to the abstract model. Running systems were also built to realized the previously proposed ideas. In these systems both engineering issues common to distributed systems and algorithmic issues specific to IR need to be taken care of.

3.1 Harvest

Harvest [7] is a distributed crawler-indexer architecture which addresses the main problems in crawling and indexing the Web: Web servers get requests from different crawlers of search engines which increase the servers' load; most of the entire objects retrieved by the crawlers are useless and discarded; no coordination exists among the crawlers. But it seems most of further Harvest applications are in the field of caching Web objects instead of providing advanced internet search services. State of the art indexing techniques can reduce the size of an inverted file to about 30% of the size of the text (less if stopwords are used). For 100 million pages, this implies about 150GB of disk space. Assuming that 500 bytes are required to store the URL and the description of each Web page, we need 50GB to store the description for 100 million pages. The use of meta search engines is justified by coverage studies that show that a small percentage of Web pages are in all search engines. Moreover, fewer than 1% of the Web pages indexed by AltaVista, HotBot, Excite, and Inforseek are in all of those search engines. [8]

3.2 WAIS

Wide Area Information Service (WAIS) [9] is a very early piece of work in the area of web-based distributed query processing. It was popular at the beginning of the 1990s before the boom of the Web. A WAIS system only forwards queries to certain servers based on a preliminary search of the content of those specific servers. The servers use some special fields in the documents such as *headline* of a news article or *subject* of an email to describe the content. This approach serves as a compromised solution between forwarding the request to all servers, and forwarding the request to only those servers that match the very detailed full-text index.

3.3 GLOSS

The work of Glossary-of-Servers Server (GLOSS) [10] builds a server to estimate the best server for a given query based on the vector-space model. Each individual server is characterized by its particular vector. The top n servers are then searched and the results are combined. The work explored several means of characterizing a server. It is estimated that the index on the GLOSS server is deemed to be only 2 percent of the size of a full-text index.

3.4 STARTS

Stanford Proposal for Internet Meta-Searching (STARTS) [11] is a protocol for distributed, heterogeneous search. It was designed from scratch to support distributed information retrieval and includes features intended to solve the algorithmic issues related to distributed IR, such as merging results from heterogeneous sources.

3.5 Z39.50

Z39.50 [12] is a standard for client/server information retrieval which defines a widely used protocol with enough functionality to support most search applications. It was firstly approved as a standard in 1995 but is under revision recently. The protocol is intended to query bibliographical information using a standard interface between the client and the host database manager which is independent of the client user interface and of the query database language at the host. The database is assumed to be a text collection with some fixed fields. The protocol is used broadly and is even part of WAIS. Not only the query language and its semantics, but also the way of establishing a session, communication, and exchange of information between client and server are specified in the protocol. It was originally conceived only to operate on bibliographical information, but has been extended to query other types of information as well.

3.6 Modern P2P Systems

Modern P2P systems developed very quickly in recent several years. Search functionalities in these systems however are really preliminary and limited. Most use the naive way of broadcasting requests such that the whole P2P network is flushed. And no systematic and mature public search engine like the Web counterpart Google appears yet. This leaves much space for us to study and integrate the requirements into our architecture.

4. My Approach: The Architecture

In our architecture, we introduce the logical abstract *aggregator* of the processing units of a decentralized search system. A picture of aggregator graph is illustrated here.

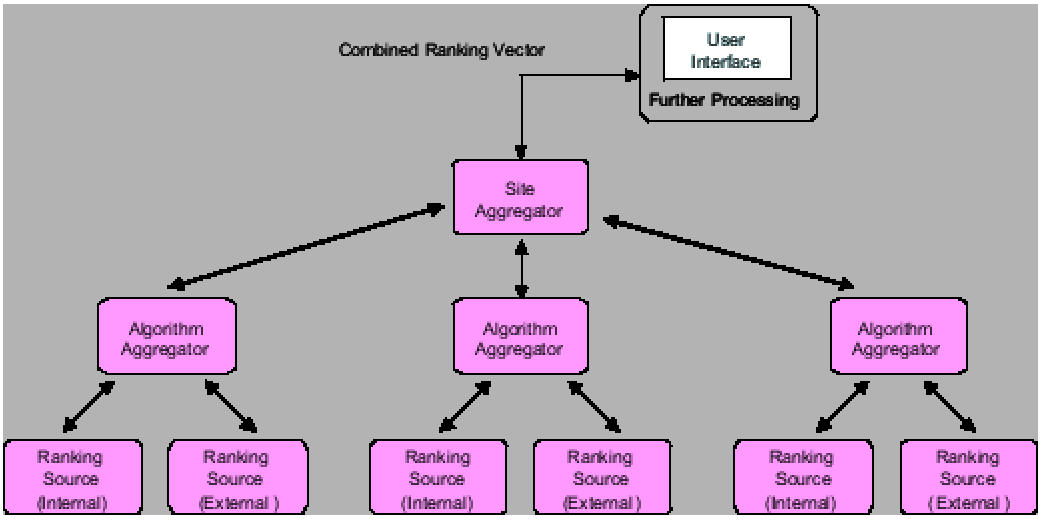


Figure 1: Aggregator Graph

We have 3 types of different roles in our architecture. *User* is the first role who sits on top of all and submits queries to member(s) of a decentralized search system. Original ranking *sourcer* is the role locating at the bottom that provides original ranking vectors of source documents. In the middle is the role *aggregator* which takes input from original ranking sources or intermediate aggregators and compute an aggregated ranking vector according to a ranking algebra. This newly computed ranking vector can be in turn exported to higher level aggregators as their input. There are two types of aggregators: a *site* aggregator or an *algorithm* aggregator. A site aggregator combines the results of different sites while an algorithm aggregator combines the results of different algorithms. Details about the ranking algebra and interactions among sites in the architecture are elaborated in the following subsections.

4.1 Rank Composition: The Algebra

In our experiments we found [5] that different rankings established in different contexts (in particular local vs. global contexts) can be of great interest. Thus we deviate from the view of usual Web search systems that all documents are ranked within a single, absolute ranking. Rather we see rankings as first-class objects, that can be produced, exchanged and manipulated as any other data object. To make this precise, we introduce now a framework for ranking computation that defines what the type of rankings is, and how rankings are manipulated. We will use an algebraic framework for rankings, a ranking algebra, similarly as it is done for other types of data objects (such as using relational algebra for relations). The ranking algebra will allow to formally specify different methods of combining rankings, in particular, for aggregating global rankings from local rankings originating from different semantic contexts. We define the domain of objects that are to be ranked. Since rankings can occur at different levels of granularity there will not be rankings of documents only, but more generally, rankings over subsets of documents (partitioned zones). In order to be able to compare and relate rankings at different levels of granularity we introduce now a partial order on partitions. We also introduce an operator to make it possible to directly relate the elements of two partitions to each other (and not only the whole partitions as with cover). Link matrices defined over partitions are the basis for computing rankings. A number of operations are required to manipulate link matrices before they are used for ranking computations. We introduce only those mappings that we have identified as being relevant for our purposes. The list of operations can be clearly extended by other graph manipulation operators. We also need the ability to change the granularity at which a link matrix is specified. This is supported by the contraction operator. In certain cases it is necessary to directly manipulate the link graph in order to change the ranking context. This is supported by a link projection. Normally rankings will be normalized. As for link matrices we also need to be able to project rankings to selected subsets of the Web. In many cases different rankings will be combined in an ad-hoc manner driven by application requirements. We introduce weighted addition for that purpose. After having these necessary definitions, we can apply the ranking algebra to produce different types of rankings by using different ranking contexts such as:

- Global site ranking: The global site ranking is used to rank the selected Web sites using the complete Web graph.
- Local site ranking: In contrast to the global site ranking we use here as context only the subgraph of the Web graph that concerns the selected Web sites.
- Global ranking of documents of a Web site: This ranking is the projection of the global PageRank to the documents from a selected site.
- Local internal ranking for documents: This corresponds to a ranking of the documents by the document owners, taking into account their local link structure only. The algorithm used is PageRank applied to the local link graph.
- Local external ranking for documents: This corresponds to a ranking of the documents by others. Here for each document we count the number of incoming links from one of the other Web sites. The local links are ignored.

Now that we have seen different ways to derive rankings using the ranking algebra, we illustrate of how these rankings can be combined in order to produce further aggregate rankings. This will be again specified by using ranking algebra expressions. Thus we address several issues that have been discussed in previous sections and demonstrate two points:

1. We show that global document rankings can be determined in a distributed fashion, and thus better scalability can be achieved. Hence ranking documents based on global information not necessarily implies a centralized architecture.
2. We show how local rankings from different sources can be integrated, such that rankings can be made precise and can take advantage of globally unavailable information (e.g. the hidden web) or different ranking contents. Thus a richer set of possible rankings can be made available.

The application of the ranking algebra to compute the rankings occurs at both the site aggregators and the algorithm aggregators.

4.2 Local Ranking in the Dynamic Web Society

Traditional ranking models used in Web search engines rely on a static snapshot of the Web graph, basically the link structure of the Web documents. However, visitors' browsing activities indicate the importance of a document. In the traditional static models, the information on document importance conveyed by interactive browsing is neglected. The nowadays Web server/surfer model lacks the ability to take advantage of user interaction for document ranking. We enhance the ordinary Web server/surfer model with a mechanism inspired by swarm intelligence to make it possible for the Web servers to interact with Web surfers and thus obtain a proper local ranking of Web documents. The proof-of-concept implementation of our idea demonstrates the potential of our model. The mechanism can be used directly in deployed Web servers which enable on-the-fly creation of rankings for Web documents local to a Web site. The local rankings can also be used as input for the generation of global Web rankings in a decentralized way. This innovative way of computing local Web document rankings is used at the level of *sourcer* in the aggregator graph. We use Web *pheromone* to record users' visiting information which reflect how interesting or how important a Web document is from the viewpoints of the surfers. Whenever a surfer accesses a page, some Web pheromone is left on the page. The Web server assumes the role of the Nature and maintains the Web pheromone information of all the local documents. Pheromone accumulation, evaporation, and spreading strategies are defined and applied to all documents. The higher pheromone density a document has, the more important in a general sense it has and thus the higher it would be ranked. In our model, the Web surfers here are the natural agents in a self-organizing system just like the ants in their social intelligent system. [14] Surely surfers are not non-intelligent, but as human have only really limited insight on the Web and most of the time can only follow the hyper links created by someone else without any knowledge of the structure of the Web graph, so here we the surfers as the primitive agents that abide by simple operation rules. Furthermore, the Web server here is not only a passive listener to the requests for Web documents, but also an active participant of the self-organizing system by assuming the role of an arbiter who assures the rules are carried on during the interactive interactions between the requesting visitors and the requested Web documents which form together the ecological environment for the self-organizing system.

4.3 Gluing All Together: Global Site Ranking

Here we go a further step to introduce the ranking of Web sites in the global Web. We have already the way that every local Web site can use to generate absolutely timely local Web document ranking with potentially high quality. We also have a ranking algebra that an aggregator can use to combine inputting rankings to get the intermediate and final ranking result. But we still lack one thing: how the local rankings can be compared with each other? How are the computed float values from different local Web sites interpreted when putting together? Global site ranking is our criteria as the answer. Just like Web documents, Web sites are also considered to have different degrees of general importance. Nobody will deny the higher importance of Yahoo! or Google or W3C Web sites. By computing global site ranking we get the base where the construction of aggregators can be built on. What is required for the computation of global site ranking is the knowledge of link structure among the sites on the Web. We will study on two sub problems:

1. How the knowledge is exchanged and shared by all participating Web sites?
2. How big a part of the Web graph of which the link structure information is needed for a Web site to compute an approximation of global site ranking that is good enough?

As a beginning, we will let the Web sites use the naive broadcasting way to exchange and share the knowledge of link structure among sites. As the number of Web sites is much much smaller than the number of Web documents, the computation of ranking is definitely tractable at the scale of sites. In the future, we may study more efficient ways of synchronizing the knowledge of the global Web among the Web sites. The second sub problem is more complex. We are thinking about identifying a subset of more critical sites in the computation of the approximate global site ranking for every particular Web site. Then this critical subset is used to hopefully have a good enough approximation of the real global site ranking based on the whole Web graph.

5. Results Achieved So Far

5.1 Ranking Algebra

We apply the ranking algebra in a concrete problem setting. We performed an evaluation of the aggregation approach described above within the EPFL domain which contains about 600 independent Web sites identified by their hostnames or IP addresses. We crawled about 270.000 documents found in this domain. Using this document collection we performed the evaluations using the following approach: we chose two selected Web sites with substantially different characteristics, in particular of substantially different sizes. For those domains we computed the local internal and external rankings. Then we applied the algebraic aggregation of the rankings obtained in that way, in order to generate a global ranking for the joint domains. For local aggregation we chose a higher weight (0.8) for external links than internal links. One motivation for this choice is the relatively low number of links across subdomains as compared to the number of links within the same subdomain. The resulting aggregate ranking for the joint domains is then compared to the ranking obtained by extracting from the global ranking computed for the complete EPFL domain (all 270.000 documents) for the joint domains. The comparison is performed both qualitatively and quantitatively. We have better qualitative results. In the top 25 list of the aggregate ranking result, the top 4 are obviously more important than the top listed results from the global PageRank. We can assume that this is an effect due to the agglomorate structure of these document collections. These play obviously a much less important role in the composite ranking due to the way of how the ranking is composed from local rankings. It shows that the global page ranking is not necessarily the best possible ranking method. We obtained similar qualitative improvements in the ranking results of other domains. As for quantitative results, one can observe that basically the aggregate ranking approximates the rankings computed on the selected subsets. This is an interesting result, since the aggregate ranking is performed in a distributed manner, computing separate rankings for each of the subdomains involved. This shows that by aggregation one can obtain at least as good results in a distributed manner as with global ranking using the same information. Details are included in the paper [5].

5.2 Swarm Intelligent Web Server Module

We developed a swarm intelligent module for the popular Apache Web server software. Although it still has bugs which make the server instable and crash from time to time, we did manage to make some preliminary but very interesting experiments with it. Firstly we made a game of Quest for Treasure. The idea was to start a quest for a treasure in order to see, if in a self-organised system, changes to the environment will result in a collective optimisation of navigation. We had 12 rooms where the visitors could navigate through. In two of these rooms were treasure chests, which had to be explored. Above each button was the actual pheromone density (on the right side of the vertical bar) of the underlying link. Visitors had to use for the navigation the density which was computed and shown by the server module. Red numbers remind people that the pheromone there has very high density. After a certain time we could just follow the red links and we found the treasure. The next morning we removed one of the treasure chests, and we could observe that during the next few hours the colors had changed. On the way to the room where we removed the chest the density of pheromone was decreased and the red links again led now to the one and only chest. So we could observe a very simple form of self-organization by collectively using the Web pheromone information from the server module. Hence, we demonstrate that the swarm of internet surfers is indeed more intelligent than a single surfer. Then we did a small-scale experiment with a lab Web site which has about 200 Web pages in order to explore this possibility of generating document ranking from our swam intelligent module, which we call ***intelligent*** ranking. We installed the module and requested volunteers to surf the site. The experiment lasted for 2 days (because of the instability of the module). We pre-computed the static PageRank ranking. We found that the PageRank ranking and the intelligent ranking is quite different. Firstly, the top ranked document is different; Secondly, in the top 17 documents of both rankings, only 6 (35%) are the same. More details can be found in [6].

5.3 Simulation of Cooperative Web Servers

We are investigating proper simulation environments for Web and P2P systems. Factors taken into consideration include scalability, which is probably the most important for a Web scale problem; ease of development, for example, the language used, the modularity, the interface definitions, etc.; administrative capabilities in order to monitor the simulation, customize parameters and settings, observe the progress of execution, gather statistical information and results; visualization; etc.. After that, we will develop the prototype system of our decentralized architecture for Web and P2P search engines and implement the algorithms that we have briefly discussed. We will focus on the cooperative interactions among the Web servers.

6. Conclusion

6.1 Summary

As the global site ranking is more or less stable because at a grand scale the Web is more or less stable although there are fluctuations because of the continuous of emerging and dying Web sites. Thus the global site ranking only needs to be computed periodically like what is done by modern search engines for computing the document rankings of the whole Web. The ranking algebra makes it possible to represent formally the decomposing the computation of the global document ranking to two step: the computation of local document rankings for every Web site; and then the combination of the computed local rankings. Swarm intelligence has been reported to have applications in many fields, such as combinatorial optimization, communication networks, robotics, etc.. As far as we know, nobody else or other research groups have tried to apply this idea for Web surfing to obtain ranking of documents for the purpose of information retrieval. By developing a swarm intelligent module, we turn a Web server into a self-organizing component of the aggregator graph in the Web. Combining the evaluation information implied in surfers' dynamic interactions with the Web server, we might have a ranking of local Web documents of better quality from the viewpoints of users. In short, by adopting my decentralized architecture, the computation cost of Web and P2P search engines will be reduced dramatically; the timeliness of search results of Web documents is enhanced a lot without suffering the crawling delay of nowadays Web search engines; potentially results fit more for the searchers' information needs will be returned thanks to the innovative way of computing local document rankings.

6.2 My Contributions

My work is original as the Swarm Intelligence-inspired way proposed by me of computing the document ranking of a Web site is an innovative new idea; and I also propose the completely new method of computing the global document ranking of the whole Web from the global site ranking and local document rankings in a totally decentralized way. Our work of ranking algebra is also original since it provides a formal framework which is absent in most ad-hoc systems for computing document rankings.

References

1. Karl Aberer. "P-Grid: A Self-Organizing Access Structure for P2P Information Systems". 2001.
2. Sergey Brin, Lawrence Page. "The Anatomy of a Large-Scale Hypertextual Web Search Engine". 2000.
3. Larry Page, Sergey Brin, R. Motwani, T. Winograd. "The PageRank Citation Ranking: Bringing Order to the Web". 1998.
4. Jon Kleinberg. "Authoritative Sources in a Hyperlinked Environment". 1998.
5. Karl Aberer, Jie Wu. "A Framework for Decentralized Ranking in Web Information Retrieval". APWeb2003.
6. Jie Wu, Karl Aberer. "Swarm Intelligent Surfing in the Web", submitted to ICWE2003.
7. C. Mic Bowman, Peter B. Danzig, Darren R. Hardy, Udi Manber, and Michael F. Schwartz. "The Harvest Information Discovery and Access System". In *Proc. 2nd Int. WWW Conf.*, pages 763-771, Oct. 1994.
8. Ricardo Baeza-Yates, Berthier Ribeiro-Neto. "Modern Information Retrieval". Addison-Wesley, 1999.
9. Duda, A. and Sheldon, M. (1994). "Content routing in a network of WAIS servers". In *Proceedings of the IEEE Fourteenth International Conference on Distributed Computing Systems*, pages 124-132.
10. Gravano, L. and Garcia-Molina, H. (1995). "Generalizing GLOSS to vector-space databases and broker hierarchies". In *Proceedings of the 21st International conference on VLDB Conference*, pages 78-89.
11. Luis Gravano, Chen-Chuan K. Chang, and Hector García-Molina. "STARTS: Stanford proposal for Internet meta-searching". In *Proc. ACM SIGMOD Int. Conf. on Management of Data*, pages 207-218, Tucson, AZ, May 1997.
12. NISO press, Bethesda, MD. *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification (ANSI/NISO Z39.50-200X)*, 2002. Accessible at http://www.niso.org/.
13. David A. Grossman, Ophir Frieder. "Information retrieval: algorithms and heuristics". Kluwer Academic Publishers, Boston, MA, 1998.
14. Eric Bonabeau, Marco Dorigo, Guy Theraulaz, "Swarm Intelligence: From Natural to Artificial Systems", Oxford University Press, 1999.

Footnotes

... engines.[1]

The work presented in this paper was supported (in part) by the National Competence Center in Research on Mobile Information and Communication Systems (NCCR-MICS), a center supported by the Swiss National Science Foundation under grant number 5005-67322.

... keywords[2]

The concept of *descriptor* is used for these keywords in some reference. Accordingly an *asciptor* is defined as a term that is a synonym of a descriptor. We do not cover the details here.

mSpace: interaction design for user-determined, adaptable domain exploration in hypermedia

m.c. Schraefel (1, 2)
(1) Dept. of Electronics and Computer Science, University of Southampton, Southampton, UK,
mc at ecs.soton.ac.uk

Maria Karam, Shengdong Zhao (2)
(2) Dept of Computer Science, University of Toronto, Toronto, Canada
{mc, mkaram, sszhao}@dgp.toronto.edu

Abstract

Adaptive Hypermedia systems have sought to support users by anticipating the usersí information requirements for a particular context, and rendering the appropriate version of the content and hypermedia links. Adaptable Hypermedia, on the other hand, takes the approach that there are times when adaptive approaches may not be feasible or available and that it would still be appropriate to facilitate user-determined access to information. For instance, users may come to a hypermedia and not have a well-defined goal in mind. Similarly their goals may change throughout exploration, or their expertise change from one section to another. To support these shifting conditions, we may need affordances on the content that a solely adaptive approach cannot best support. In this paper we present an interaction design to support user-determined adaptable content and describe three techniques which support the interaction: preview cues, dimensional sorting and spatial context. We call the combined approach mSpace. We present the preliminary task analysis that lead us to our interaction design, we describe the three techniques, overview the architecture for our prototype and consider next steps for generalizing deployment.

Categories and Subject Descriptors

H5.4 Hypertext/Hypermedia-Architectures, Navigation, User issues. H5.2 User Interfaces-Prototyping. General Terms Design, Experimentation, Human Factors

Keywords

Hypermedia, Task analysis, adaptable hypermedia, information visualization

1. Introduction

One of the promises of the Web, and more recently the Semantic Web is greater access to information. Make it digital: make it accessible. Representing digital information, however, is a huge challenge: make it digital, it often becomes invisible. The problem has been approached from multiple angles. Adaptive Hypermedia research in particular focuses on getting the right information about a given topic to the right user • whether that user is a beginner in a domain or an expert[1]. Information Visualization research focuses on the challenges of presenting rich information sources on (mainly) visual displays, dealing with challenges like finding the balance between how to balance contextual information with information currently in focus[7]. Both Human Computer Interaction (HCI) work and Hypermedia research has considered problems of navigating through hyperspaces.

Visualizing, navigating, and determining user level might each be seen as part of the information problem after the information source has been discovered. In this paper, we consider the part of the problem just before this: the issue of accessing the information in the first place. How can we reduce false steps in locating appropriate domain information? And then, in the sense of hypermedia affordances, how can we provide the support for the user to engage the information, to support their making their own connections about the materials discovered?

In this paper we propose an approach for domain interaction to support both discovery and connection building. We do this with three core interaction functions: preview cues, dimensional sorting and spatial context. We call this interaction approach mSpace (for representing multidimensional spaces). In the following sections we present the motivation for mSpace. We describe the task analysis evaluation we used to determine the interaction requirements, and detail the resulting prototype we have developed stemming from that analysis. We briefly present the architecture behind the current prototype, but emphasize that our interest is on the interaction approach: to design hypermedia-informed affordances for an information delivery system; we are architecture/backend agnostic. We discuss the related work as it applies throughout each section of the paper.

The techniques we describe afford the community new, additional methods by which better access to information can be achieved fro more users.

1.1 Motivation: Search, Browse, Exclude

Google has become the default method for accessing information on the Web. One enters their keywords for a search, a list of resulting links is returned, one browses through the results to find pages which best satisfy the query. For most cases, this is a highly successful method for retrieving information. There are some kinds of queries, however, that Google cannot support, such as when the users are domain naive. That is, these users do not have either the expertise to frame a query to begin with, or to interpret the results were they to be presented with them. For instance, people who know little about classical music, except that they know what they like when they hear it, cannot use Google (or any keyword search tool) because they do not know either the names of the pieces or the terms by which they are described. Similarly, if they do a generic search for Classical Music, they would find little value in browsing through sites with comprehensive listings of all recorded classical works: descriptions such as sonata, serenade, romantic or rondo are equally meaningless if all the user knows is what they like when they hear it. In effect, if they cannot name it or spell it or specify it, they cannot access it.

To begin to address the information invisibility problem, we proposed the following: first to represent information in terms of a spatially represented domain, an organized representation of an area where elements represented within the domain have defined associations to each other; and second, through the interaction, to leverage what a user already knows about that domain to support access to it. In the case of classical music for instance, users are presented with a structured representation of the classical music domain with categories such as Period, Genre, Composer, and Instrument. On their own, these categories may be meaningless if users only know what they like when they hear it. Therefore, we associate music with each attribute of the domain so that users can determine if they are interested in exploring this dimension of the domain based on what they can assess: whether or not they like the music associated with that attribute. We call this technique Preview Cues [11].

Similarly, if users are more familiar with instrument than composer they can rearrange the domain so that instrument is the primary dimension, with other dimensions projected as dependent levels of a hierarchy descending from instrument. We call this technique dimensional sorting. These techniques allow people to leverage what they know about a domain to enable discovery from their frame of reference rather than a particular information sourceís. In this approach, the domain is presented in a structured way, and the potentially unfamiliar structure is made explorable by preview cues and dimensional sorting that enable the user to make a judgement about how they wish to proceed through the space. We hypothesize that through the combination of these techniques • structure, cues and sorting - users will begin to learn the domain • gain domain expertise • simply by interacting with this domain interface. We further hypothesize that the ability to manipulate the domain from multiple user-determined perspectives supports discovery of knowledge within the domain that may not otherwise be readily apparent. In the classical music example, for instance, people simply using the interface would be able to learn implicitly that, while Bachís keyboard works are often recorded on the piano, the piano itself was invented after Bachís lifetime. One may then want to know what his music would sound like performed on the instruments of his time. With the interaction proposed here, the user could make these comparisons easily by moving from Works to Recordings. Thus, an initial discovery may lead to a new query and there to a new discovery, all of which can be satisfied from the usersí frame of reference.

We carried out an initial study to look at how combining a spatial domain representation with preview cues would improve access to classical music [11]. Based on the success of this work we considered a a more abstract problem space, JavaScript documentation for various levels of expertise, which lead us to the development of dimensional sorting. The combination of spatial context, preview cues and dimensional sorting is proving to be a gestalt technique for improved information access in adaptable hypermedia. We ground the presentation of the techniques first in a discussion of the task analysis we performed in the problem domain.

2. Task Analysis

We chose the JavaScript domain for two reasons: the classical music space trials proved to be expertise-neutral in terms of performance and enjoyment. Both experts and novices found the tool effective and satisfactory. We werenít sure if these results would carry through to a domain context where the goal is more complex information support, in other words, where more than simple selection based on preference is involved. In the case of JavaScript documentation, users from various expertise backgrounds would have distinct problems to solve, based on distinct levels of previous knowledge. We wanted to see if we could develop an effective adaptable interaction space that would support this range of expertise and complexity. Before we designed our interface, we needed to understand how JavaScript practitioners approached problem solving with current resources. It is absolutely critical to perform such preliminary evaluation of the problem domain, without which any model we proposed would be shooting in the dark. To this end, we performed a task analysis. Task analysis is a method in HCI [8, 14] which can be used to break down a task into goals and subgoals, show their order of execution, and make explicit any dependencies among goals. The resulting task model allows us to see where inefficiencies in the interaction exist. From this analysis, we can develop approaches to address the problem.

2.1 Task Domain

In our case we were interested in assessing how users used online resources to assist solving JavaScript problems. JavaScript is one of the most common scripting languages used for programming interactive effects into web pages. Based on our own experience with Web site design and Web application development, we predicted three user profiles: (1) Web site designers who use JavaScript as part of prewritten effects available in packages like GoLive and Dreamweaver. (2) casual JavaScript implementers who were either programmers, but without much JavaScript experience, or web site developers who were capable of editing prewritten JavaScript code, and (3) experts who used JavaScript coding professionally. After interviewing twelve participants who identified themselves as people who worked with JavaScript code, we found that these profiles were appropriate stereotypes, and could be further coded as Novice, Intermediate and Expert users. However, what was of particular interest to us is that each group required quite distinct JavaScript resources from the other. This diversity of user types gave us a different problem approach for user access to domain information than the user modeling case in Adaptive Hypertext (AH) research. In AH work, particularly educational AH [3], we are most interested in creating user models to adapt the same set of information in different ways as appropriate for different levels of expertise. In JavaScript, however, different user levels require different kinds of JavaScript information. Our challenge was to see if there would be value in (a) modelling a single site to support all these expertise resource needs and (b) interlinking all the information components in a manner to support a rich context for any user's access of the site at any entry point.

2.2 Task Evaluation

In order to understand both what kind of information requirements our three users profiles needed for the kinds of JavaScript problems they had to solve, we used two types of Task Analysis approaches: participant interviews about their work with JavaScript and an experimental evaluation of task performance.

2.2.1 Experiment

The study involved 12 JavaScript users, 4 novices, 4 intermediate users and 4 experts. Each participant was given 30 minutes in which to complete a task that required them to add a JavaScript feature to a sample web page. These tasks were based on a pilot study of 5 participants to determine what an appropriate task for each user level would be. Novice users were asked to add a script to a page that would make a banner appear with an imaginary company's name. Intermediates were asked to create a popup window that sizes itself to 60% of the available screen size of the user's system. The expert users were asked to create a trim function that removed all the leading and ending spaces from a string and displayed the new string. In each case, participants were shown a running version of a page using the code for their task, so that they could clearly see what they were being asked to do. They were also given a web page template into which they could insert their code for testing. The same computer was used for each session, and the same tools were provided: a text editor for code writing and editing and a web browser for web searching and code testing. A pre-task questionnaire helped determine the users' expertise level and a post-observation questionnaire discussed problems they encountered during their tasks. Participants frequently made suggestions for improving resources to solve those problems. Each participant was also encouraged to "think aloud" to describe what they were doing as they worked on their tasks. Each session was also taped.

2.2.2 Observations

2.2.2.1 Resource Use and Expertise Approach

We observed a consistent set of steps across user groups to solve the task to Get a Script Running. There were two large subgoals: (1) find the appropriate resources; (2) test the result.

There were multiple subgoals stemming from these two core goals. The model is presented in Figure 1, below.

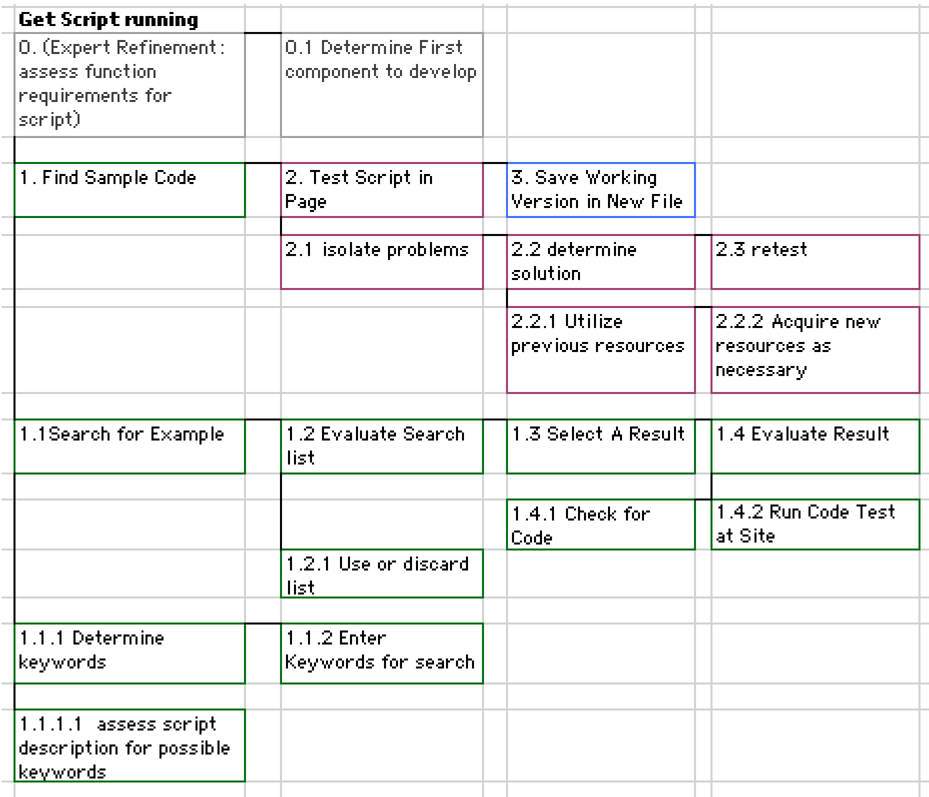


Fig. 1. Hierarchical Task Analysis of "Get Script Running"

The model is based on Hierarchical Task Analysis[4]. This approach lets us focus on the task and its subgoals without making assumptions about user knowledge structures. As can be seen from the diagram, most of the subgoals relate to the task of resource discovery. Subtask 1 "Find Sample Code" has ten subgoals compared to the five steps of Subtask 2 "Test Code"'. The model was the same for each user level, with the exception of one preliminary subgoal introduced by the expert users (labeled 0 and 0.1 above) to pre-assess the problem for what function calls it would likely require.

In subgoal 1, users either went directly to a source of information that they though had the answer to their questions, such as a JavaScript or design site, or did a more general key word search to begin. Key word searches took place either at the Web level, or within a JavaScript site. The second step, once the resource was located was to start coding, either by hand in the text editor, or by copying code directly into the web page template. If the code failed to produce correct results, users returned to step 1, either refining their search to find a better resource, or conducting a different search to address some problem they encountered while implementing their code.

Novices and Intermediate users searched for working examples of JavaScript that they could view and then copy into the Web page. Advanced users already had significant personal JavaScript knowledge, and found it more efficient to craft their own code. Generally, their need to consult resources was at the syntax level, so language references were important. They expressed wanting to understand how the code worked at that language level. If they encountered a bug in their code they could not solve from this approach, then they said they would look to Q&A sites to see if anyone had encountered the same error and what the solution was. None of the experts, however, found this necessary for the given task.

2.2.2.2 Number of Web Pages and User Level

We captured both sites visited and frequency of sites used by recording the browser history for each participant during the task.

Table 1. Pages per User Level During 30min. Task. U=Unique Pages; HI = Highest number of return visits; U to HI = ratio. Only results of .5 or higher are rounded up.

	Av Hits	Av U	Av HI	U to HI
Novice	35	24	4	6
Intermediate	26	18	6	3
Expert	26	11	5	2
Overall	29	18	6	4

Novices returned to the Web about 25% more than frequently than Intermediates or Experts, who used the web about equally. Experts, however, used about a third fewer unique sites. Also, as expertise increases, so does the ratio of reuse to discovered resources. Novices, on average, reused a sixth of their repeated resources, intermediates, a third and experts, a half.

2.2.2.3 Task Success

Those users who completed the task required about two-thirds the time allotted. Only 2 users did not finish the task: these were both novice users whose only previous experience of JavaScript had been using prewritten functions in a Web design program. Despite this, they expressed confidence at being able to complete the task, having viewed a considerable amount of HTML source with JavaScript embedded. Indeed, they successfully found code that provided the correct function, but the source did not provide a sample page, demonstrating the function call in the body of the web page.

2.2.2.4 Particular Domain Access Problems

Below, we summarize the observations we made during the task analysis. The following is based on observing participants while they worked, the comments they made while "thinking aloud" during the task, and the post task interviews.

1. Users experienced difficulty in articulating what they wanted for keyword searches within the Web resources.

Locating information about JavaScript can be an exercise in working around potentially ambiguous terms and concepts. Unless the user is experienced with the terminology or gets lucky, locating the exact concept is difficult. For instance, coded example titles are not necessarily associated with any JavaScript concept: searching Window Event does not necessarily lead to Resizing Windows as an associated topic. In other words, Google did not work effectively for retrieving ìan answerî

2. Users were not aware of useful resources that could help them solve their problems.

A common problem for novices and intermediates in particular was lack of awareness or discovery of available online resources. For instance, users across expertise levels stated that they thought a reference manual for JavaScript would have helped them with their task. Interestingly, they did not search for manuals during their task, and several exist, including Netscape's very popular JavaScript resource.

3. Users from different experience levels sought different kinds of resources for solving similar problems.

Although the tasks were very similar for the different expertise levels, the approaches that the users took to solve their problems varied with the amount of knowledge that each user had. Novices were aware that code examples existed online, so they sought these out for direct copy and paste. If however, the examples did not provide instructions about the function call, the novices were unable to succeed. Intermediate users were happy to discover working code samples, but also connected what they were attempting with previous knowledge about programming, and evinced little hesitation in attempting to alter the function given or its call. If the function did not work on first pasted into the Web template, they were able to hack the code to get it to work. If they had problems with the code, they would look for a resource to explain part of the code. This caused some difficulties narrowing down what part of the code was the problem and which reference was needed. Experts stayed consistently at the level of syntax in their searches, generally looking first to see if the function they needed was already part of the JavaScript set of objects. One expert only had some search time difficulty determining which object string trimming was part of (it's "string"). From there, they could hand build the code for the page.

4. Users were frustrated by the poor quality resources and code examples that they had to work with.

All of the JavaScript users began their task by searching for examples of code or functions that they could use as a reference. All of them also complained that much of the code that is available is poor quality and unreliable. Since grabbing code samples is one of the most common means of building JavaScript code for our novice and intermediate participants, those without backgrounds in programming usually have no way to check the accuracy of the code. Experts seemed to avoid most of this problem by searching for appropriate functions from which to build there own code. More so than intermediate users, they first analyzed the problem, and broke the problem down into likely component requirements. They then searched for those functions/objects first if no sample code was immediately available from a keyword search.

2.2.3 Analysis

With both the task model and the Pages per Use data, we are able to consider any correlation that may exist between number of iterations through any part of the model and user expertise. We can use these correlations as guidelines for where functional improvements in the interaction design need to be considered. The Pages per Use table (Table 1) above suggests, perhaps not surprisingly, that Experts are best able to discover and narrow in on a particular set of resources as they work, and reuse those resources (subgoal 2.2.1), flipping back and forth between a narrower set of mainly cached pages, than the other groups. Novices were least well able to do this. They cycled through the steps of subgoal 1 far more frequently before moving to subgoal 2, testing the code. Despite efficiency of resource use by experts, there was not much difference between the number of times all user levels reloaded pages. An average of 29 discrete reloads during a 30 minute task is considerably high. Each reload of a page replaces the previous window, the previous context, requiring the user to bear increased cognitive load to maintain that context. For novice users, based on the number of unique pages overall they touched, it is apparent that they had difficulty even establishing a context.

Not only do multiple page loads compromise a sense of persistent context for support in a task, they take time away from other subgoals. Even when pages only need be glanced from a list of returned search results to determine whether or not to pursue them, there is time involved in selecting a link to load, time for load, time for page scanning, dismissing and returning to the search. Overall, then, the main areas for task support improvement are in resource persistence to decrease searching through a stack for already discovered, preferred resources. Better presentation of the domain itself is also required to help novice and intermediate users in particular overcome barriers to access that key word matching imposes, causing them to look more frequently for new resources, rather than reuse existing ones. The following section presents an interface prototype and its design and function rationalization to address these problems.

3. Interaction

From the analysis above, we see the global requirement is, as suspected, to improve access to domain information, but also to provide a way to support the particular usersí perspectives, or frame of reference, on the data. Our interface proposal, therefore, is first and foremost to represent the JavaScript domain in a more accessible way for the range of users. Since much of the time in the above task model is spent simply attempting to locate resources, providing a single resource as an information broker and domain context should improve task performance for help with this part of the interaction. However, within this representation we must also support the userís context.

3.1 Improved Context = Spatial Context

Research in information visualization, referenced below, suggests that spatial, hierarchical visualizations of information are more effective than list views • the default views of the Web for information discovery. We leverage this research and expand it to include hyperlinked supplemental information. In this way, we present more than a spatialized view of information contexts, but also in-context information about path elements to assist the user in making navigational decisions. This gives us a tool to investigate the combined approaches of spatial views with artifacts similar to link annotations.

3.1.1 Context for Context: Link Lists

Most Web searches return lists of linear results, and usually limit the maximum number of results displayed to some interval, as per Figure 2, below.

Expanding/contracting hierarchical list views, common in most desktop file system navigators (Fig 3), have been shown to be better for most kinds of information retrieval than the list [3].

As with [3], we have also found strong results with a multipane type display which expands the contents not into the same list, but into a new adjacent frame, where each frame can be scrolled independently [10]. A version of this kind of viewer was made popular in the NeXT OS, and is used more recently in the Mac OS X environment (Figure 4).

Expanding and collapsing lists have the advantage over lists of maintaining a controllable, persistent context for the user. This persistent context is also interactive: a much or how little associated information in a hierarchy appears in view. Chimera et al [2] found that this control reduced scrolling through data and improved discovery efficiency as users controlled the area of the data on which they wished to focus. We also found [11] that users reported having a better sense of context with multipane views than with single list views of part of a hierarchy, which placed path information above the list. This path above list approach is common in Web category sites (Figure 5).

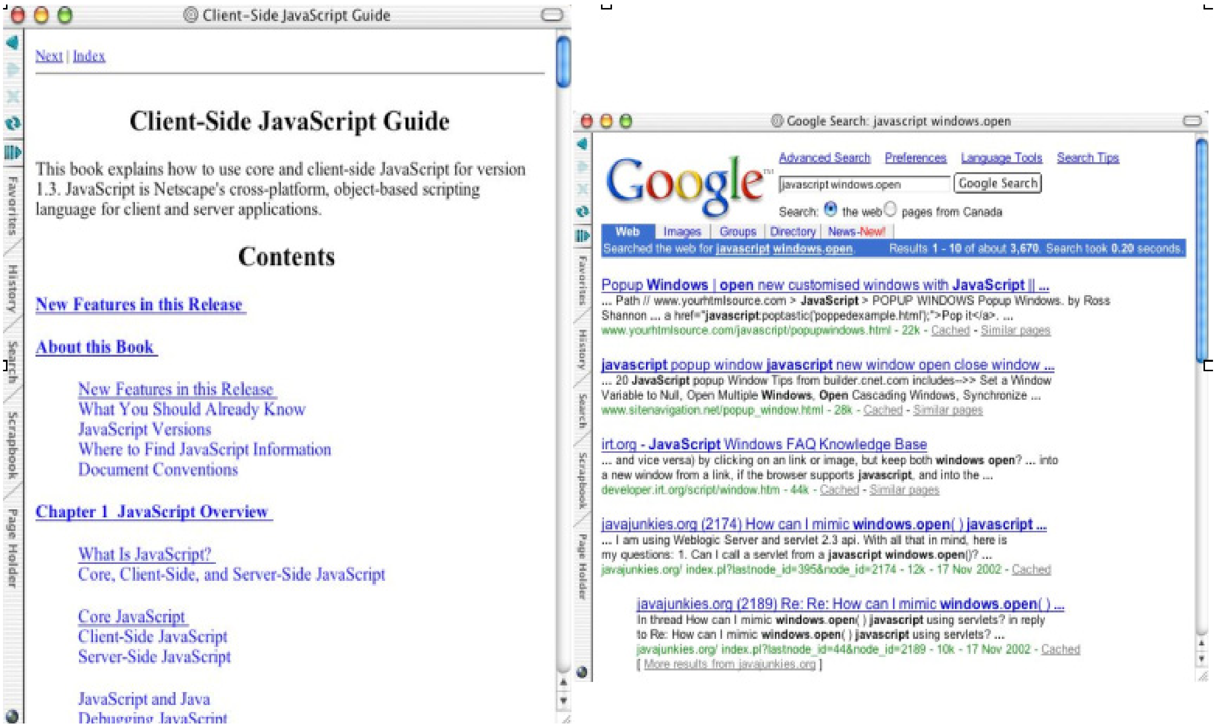


Fig. 2. Two common Web page examples of linear lists for accessing content: the left window is linked TOC for an online manual; the right is a search result window.

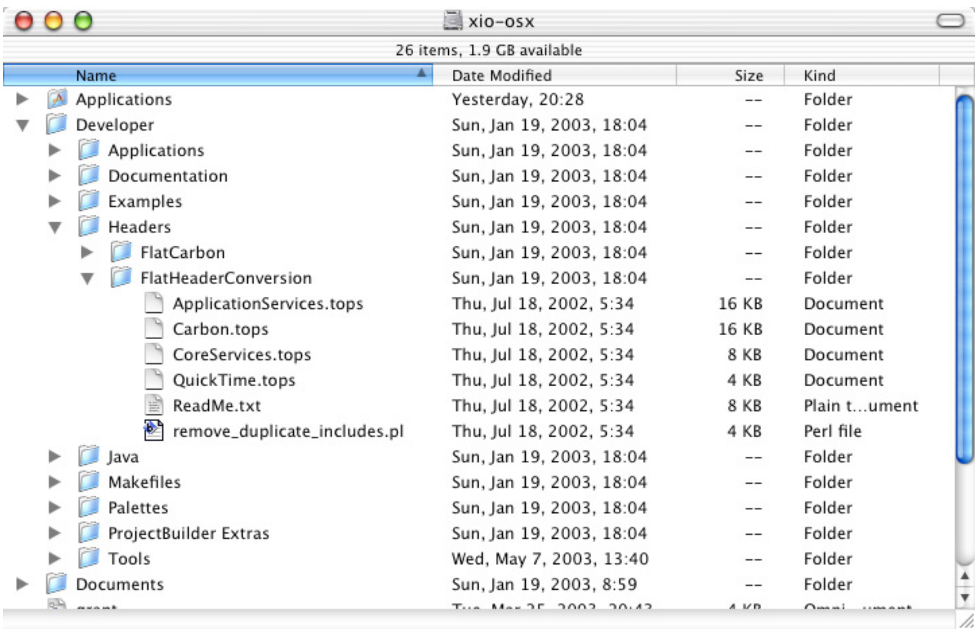


Fig. 3. Example of an Expanding/Collapsing list shown as indented) or collapsed (shown in line).

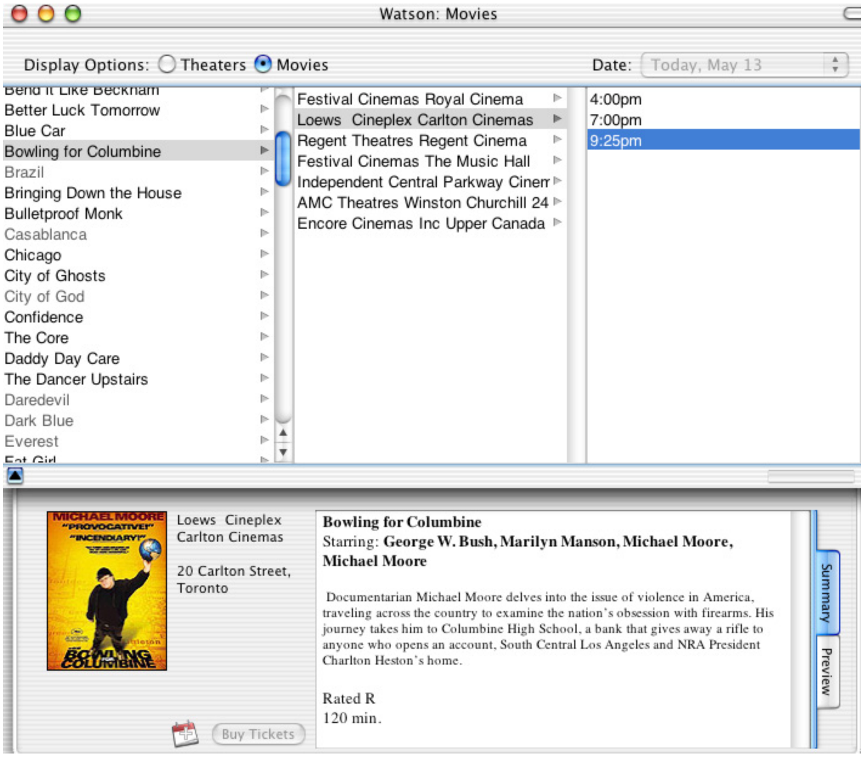


Fig. 4. Example of a multipane file browser in which items expand into the adjacent pane (Watson, Carelie Software). A type of detail rather than preview cue is also available in the lower pane description of the film. In this case, the detail reflects a specific instance, (the theater location and the film description). The detail here is very specific: films showing in Toronto, without associating information that describes kinds of films,etc. That said, the power of bringing this information together in one location, with ease of exploring options like locations and times, is undeniable.

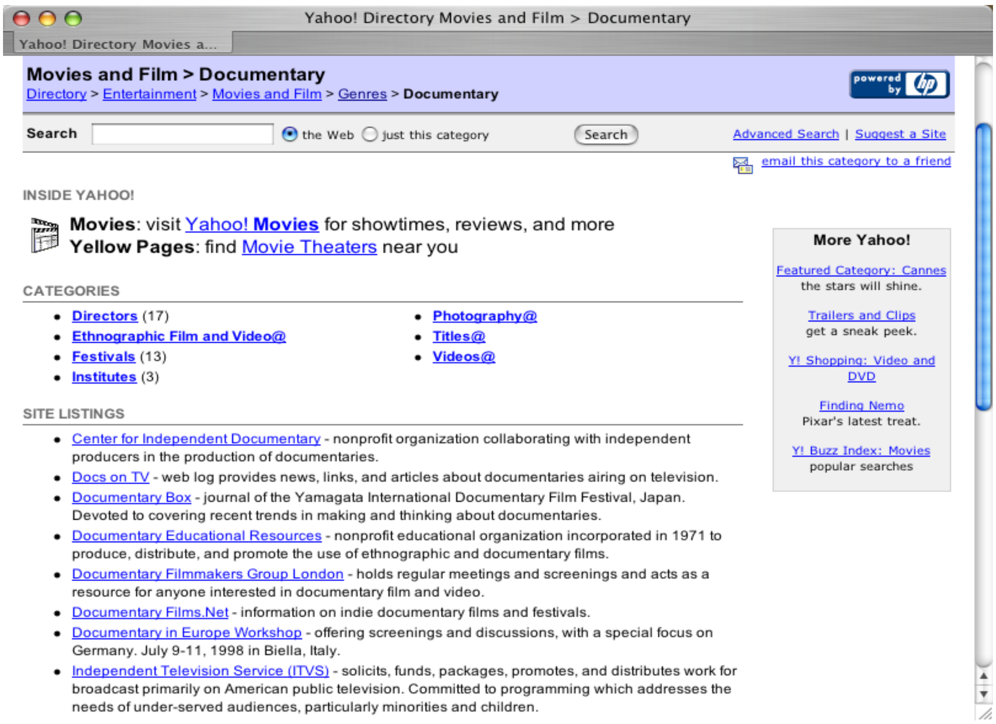


Fig. 5. Sample Web page in which path to current node in hierarchy is given via a category list (page top). Links for the current node are viewable in the vertical list beneath.

We also found that, especially if users have discovered an item of interest within a hierarchy, they prefer to see the context for each part of the path to their current location. A multipane viewer such as Fig. 4 provides this context by highlighting the previous node, situated among the visible list of nodes/leaves in that pane/at that level of the hierarchy. While we have not tested this explicitly, observations suggest that this persistent, node-level context assists in building a mental map of the domain. Such a map assists in information discovery and rediscovery.

We have extended the multipane browser to work as a hypermedia multipaned browser. For these reasons of demonstrated value for context support and efficient information discovery, we adopted a multipane hierarchical topic browser approach for our JavaScript domain interface prototype (Figure 6, below).

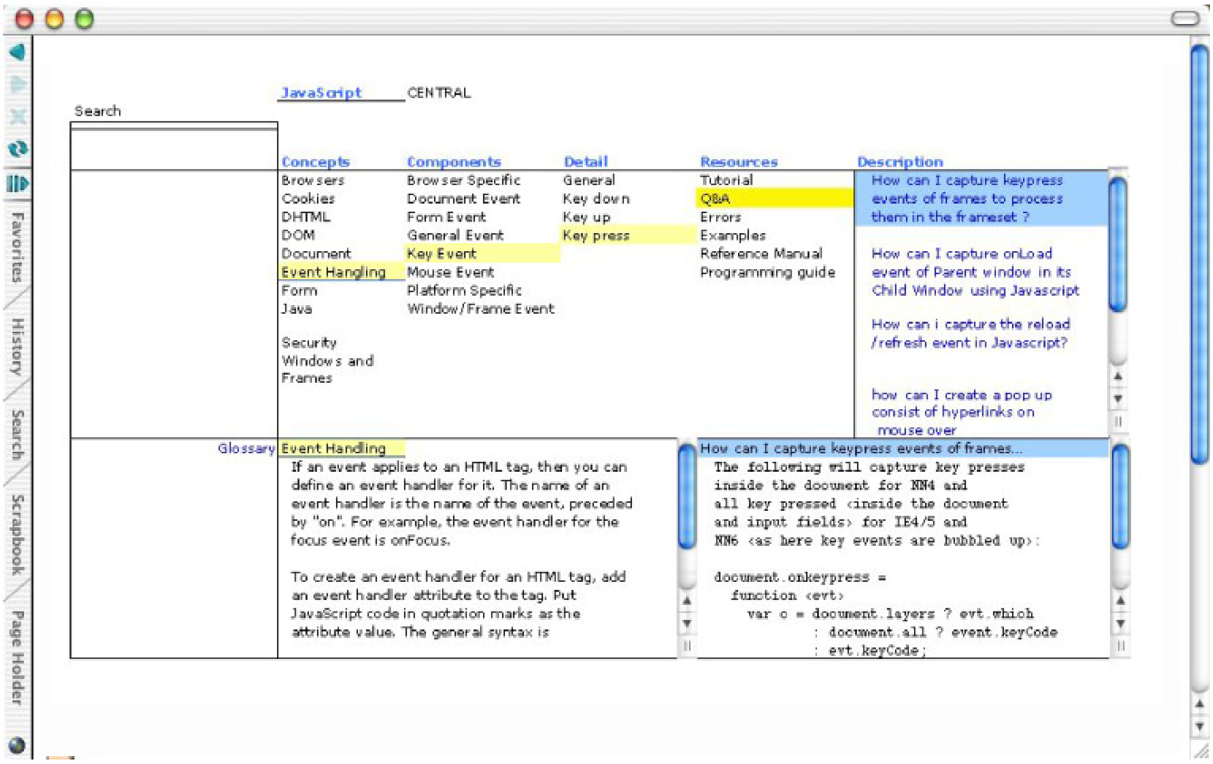


Fig. 6. JavaScript Central mSpace prototype

Context is maintained several ways: in the main area of the browser, topics in JavaScript are revealed in a multipane hierarchal view. As the user clicks a term in one pane, two events occur: (1) the next layer of hierarchy opens in the adjacent pane and (2) the glossary pane beneath the browser is populated with the entry for the selected term. By option clicking on the term in its pane, the user can "fix" that term in the glossary. A fixed term is underlined in the pane (as is Event Handler in Figure 6). If the term is not fixed, the glossary will reflect the last level of the hierarchy selected which has an available entry. In the example in figure seven, this would be the term "key press." In the final pane of the browser, the user is presented with a list of discrete entries for the specific topic. Clicking on one of the entries opens that item in the detail pane beneath that list.

Each of the panes such as the glossary entry, detail view, and item list can be opened into their own windows. Clicking on the term "Glossary" also opens a complete glossary-listing window that the user can scroll through, using an expanding/collapsing list for topics/subtopics. In other words, users have control over how much information they want displayed at any time.

The Glossary and main browser are also interlinked: clicking on an entry in the glossary window highlights its path through the multipane browser. This allows users to move between glossary and context view to explore relations among data elements. Similarly, the search box in the upper left of the window produces a list of results, organized by topic, and displayed in the blank paned beneath the search box. Clicking on a result in that list highlights its path through the multipane space, leaving the final Description pane blank.

Finally, for context support, we have incorporated work from a previous project, Hunter Gatherer [12]. With Hunter Gatherer as part of the architecture, users can select any component within any area of the browser or any of its windows, and have those selections collected into a new window. These collected components can be saved and shared with others. They also act as a reference resource so that, rather than flipping back and forth through collected resources while working on a problem -- as we saw our experts do • users can have in one place the information they found useful. Users can create multiple collections of resource components for future reference as well. Each collected component in a collection window maintains a link back to its source information block. Users can return to this larger source any time they wish to peruse the larger context of a reference. Each source document (what we call and information block) has markers associated with it, identifying other areas in the domain to which it can be associated. Clicking on any of these codes again highlights that path through the multipane viewer so that users can quickly perceive associated contexts and explore these contexts should they wish.

In our prototype, a user does not click out of the main window. The data context remains in view. When a new window opens, it does not completely occlude any other, and the user can always resize or dismiss that window. Detailed views are available quickly, and in context, supporting faster evaluation of any list of entries returned in either the search pane or the description pane. Information found to be of value can also be readily collected and maintained in its own window for persistent availability.

3.2 Improved Access

Preserving context, we know, is beneficial for discovery and navigation through information sources. In the above section, we describe several techniques that we have employed to give users both control over how they see the information in the domain, as well as how they are able to access the larger context of any component from multiple information locations in the interface. Context preservation, however, is only valuable once a user gets to the domain and begins to investigate a path. How does a user • a domain-naÔve user in particular • determine which path to explore, especially if the domain terminology is unfamiliar? Part of the task breakdown for novice users in our study was lack of term knowledge: they spend considerable time hacking around web pages looking for jargon that matched what they saw before them on the screen. Few of their discovered resources assisted them in narrowing their focus, since few of these examples showed any relation to any other example/context. Even presented with our multipane view, a domain naÔve user may spend time doing a brute force search through nodes of the tree [2]. Arbitrarily going down paths is hit and miss, not informed exploration, particularly if the domain is large. To reduce this kind of domain hacking, we developed Preview Cues and Dimensional Sorting.

3.2.1 Preview Cues for Pre-Path Evaluation

Preview Cues reveal information to describe a range of values referenced by its label. The cues are triggered by a user holding down a command key while brushing their cursor over a term in one of the browser panes. In this way, users get a sense of a range of a domain before they commit to exploring that part of a domain. Unlike tool tips or Fluid Links[18], domain preview cues do not have a one to one correlation between themselves and their trigger. A preview cue for Event Handler is not a definition of a term (that is provided in the glossary pane). It is a representation of a range of values, from which a sense of that part of the domain can be derived. Showing domain range through preview cues is not unlike multi-pointing links: rather than have a link go to only one document at its end, multiple associated documents can be pointed to. Such lists, however, represent all the possible links the system has available. They are also possible paths from which users can head out further; preview cues point in: they are representative samples of all available information in that part of the domain. For instance, a user control-brushing over the label ìEvent Handlerî causes a new small window to open to the right and top of that term. The window contains (in the current prototype) three entries representing event handlers: a representative question asked about events, a code description of a specific kind of event and an example of working event code which the user can run in that preview window. No one had to handwrite these domain previews; they are taken directly from information available to the interface from the local domain database. We will come back to this briefly in the Architecture section below. By clicking on any of the elements in the preview window, the user will see where in the path the preview is from. Along with the glossary entry below the multipane browser, the preview cue quickly provides information to users for immediate evaluation of that node in the hierarchy. Users can then decide whether or not they wish to explore further along that path. If the user brushes on a deeper part of that same path, such as "Key Press" all the preview cues would reflect the particular part of the domain relating to the Key Press event. The location of the brushed element in the path acts as the lower bound on the region for preview cue building.

In this respect Preview Cues are closer to Side Views [16] than to Query Previews[6], but both are worth mentioning. In Side Views, users of image editing software can preview simultaneously the effects of a filter: multiple versions of the image are given which show what it would it would look like with different settings of the filter applied. Seeing many versions at once saves the user from going back and forth between the image, trying one filter setting, discarding it and trying another and so on. With multiple views, the user has more data available in context to determine whether or not the filter is appropriate and with what settings. Preview Cues act likewise to give a sense of a range of values for determining whether a path is the right one to explore. Query Previews, on the other hand, provide users with preset sliders and buttons on data which the users operate to see a all the data that matches the input values, should a result exist. Preview cues are situated not to afford specific results but to act as domain indicators.

In our study of domain preview cues [10] we found that while 95% of the 82 participants in our gender/age balanced study found preview cues useful for refining path selection, and that they wanted them available. 50% of users were satisfied with preview cues being turned on with any brush over a term. 45% of participants, however, wanted to be able to turn them on or off as they chose. This is consistent with Zellweger who also found in the display of annotations on mousing over links in documents that user control annotations was critical [18]. That is why in this prototype we use a control key to modify the brush over.

Preview cues give users examples of domain content that they can quickly evaluate whether or not that area is of interest for perusal. Because these cues are assembled from pre-existing components in the domain, no special content needs to be written to describe the range of possible values in the domain. At this time, we are working on developing specific heuristics for determining best or most representative samples for a domain elementís preview cue. In the interim, what we have found is that any preview cue available on request is significantly better than no preview cue.

3.2.2 Dimensional Sorting for Orientation

While preview cues help users make decisions about exploring part of a domain, dimensional sorting lets the user organize the domain in a manner that suits their current interests/knowledge. Providing multiple perspectives on an information space has long been valued by hypertext researchers not only for access for but the value of learning more about a domain [5, 15]of building knowledge by seeing associations that would not otherwise be apparent. It also lets the explorer move into the domain from a locus of familiarity. For the domain-naive user, dimensional sorting may provide an approach into a domain, without which they would not be able to enter. In the JavaScript example, an expert user may find it effective to explore the domain in the way it is organized in Figure 6, privileging Language. A domain-naive or novice user may want to see the domain organized to privilege Examples, in order to see simply what the language can do. Our interface supports such rearrangement.

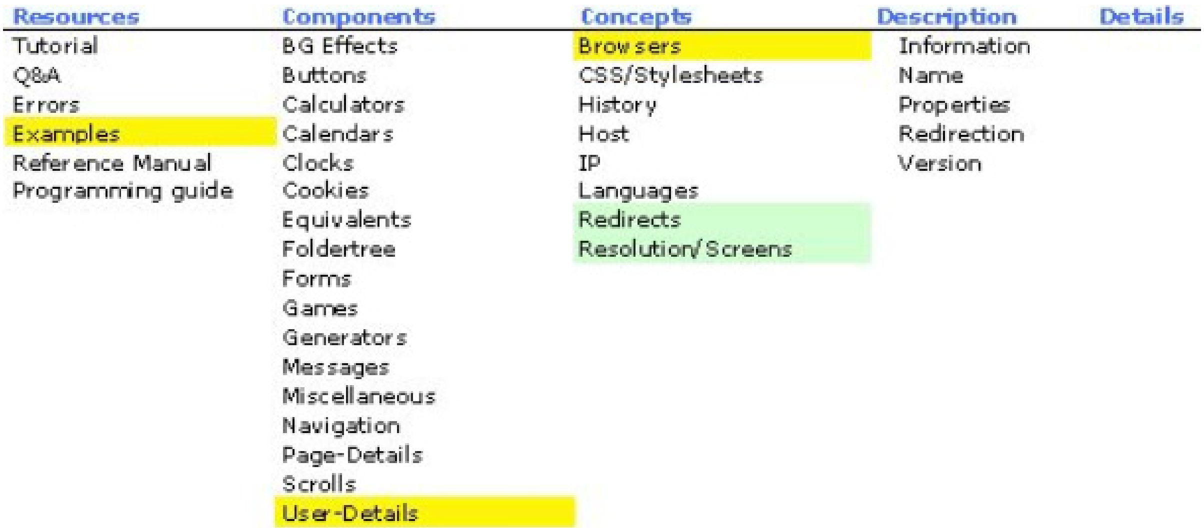


Fig. 7. The JavaScript domain hierarchy reorganized.

Hierarchical rearrangement is not to be confused with sorting order within a fixed hierarchy. Rearranging a hierarchy causes elements to be regrouped. Indeed, dimensional sorting is named for seeing these domain spaces as n-dimensional spaces. In order to visualize these n-dimensions, we flatten the space as a projection along a dimension. This allows us to render the space as a hierarchy. Flattened in this way, we can apply hierarchical visualization techniques to render the hierarchy in multiple ways. The multipane view we have chosen here is one among many possible visualizations that could extend to two or three dimensions. The mSpace model does not insist on any particular visualization; the only constraint we impose on the visualization is that it supports persistent context. In the case of dimensional sorting, the multipane view affords a clear persistent context that makes explicit the effects of reorganizing the categories of the domain. In the above example, for instance, we have pulled Resources from the fourth position to the first and moved the Concepts category from the first to the third position in the order. This rearrangement supports users who primarily want to consider code examples in the domain. Users could get at all the examples in the domain in the previous hierarchical organization, but to do so, they would need to move through almost each element of the hierarchy iteratively to gather all the examples they would want. By supporting dimensional sorting to improve access from multiple perspectives, we are not explicitly modeling users for a particular version of a domain, but are letting users determine the domain version for their needs/tasks.

3.2.3 Adaptable vs. Adaptive Hypermedia

In terms of their hypermedia context, both Preview Cues and Dimensional sorting can be considered as types of adaptable [9] rather than adaptive hypermedia techniques. They emphasize user-directed support to adapt domain representation rather than system-determined adaptive representations based on user-modeled domain interactions. The approach provides tools for users to determine directly the perspective from which they wish both to explore as well as to orient a domain. Preliminary assessment suggests that it is well worth considering integrating such adaptable direct-manipulation techniques with adaptive approaches.

3.3 Preliminary Assessment

3.3.1 Design Reviews and Predictions

The above prototype is the result of cognitive walkthroughs and design reviews. The walkthrough let us test the design flow and prototype responses to user actions via evaluations of paper-based mockups before coding the above prototype. We have only just started testing the prototype with real users at multiple expertise levels. In casual presentation with domain users, feedback has been positive, with participants keen to know when the site will "go live." Once we have a larger data set behind the prototype, we will be able to do the next round of user evaluations to see first, if our new model is more efficient and second, if we have broken the reliance on having to formulate a keyword search and improved efficiency/effectiveness in accessing this domain space.

4. A Word on Architecture

In order to support the above interaction prototype, we developed a functional back end that let us demonstrate the interaction for evaluation. We do not postulate this as a ìsolutionî for supporting the described interface in general practice.

The architecture, however, has let us process and repurpose a considerable amount of Web-based data from multiple sources. In future, this processing is an ideal job for the Semantic Web. Indeed, we have already taken the lessons learned here and applied them to a Semantic Web ontology/backend [13]. That said, the architecture is functional for repurposing existing web-based information sources with minimal manual involvement.

The main purpose of the architecture is to gather JavaScript information from a variety of sources which can be represented and interacted with in one place. This domain-based approach allows users to maintain a persistent context, rather than the user having to jump from site to site to find what they may be looking for. The associated domain approach supports evaluating any particular information component within a variety of contexts.

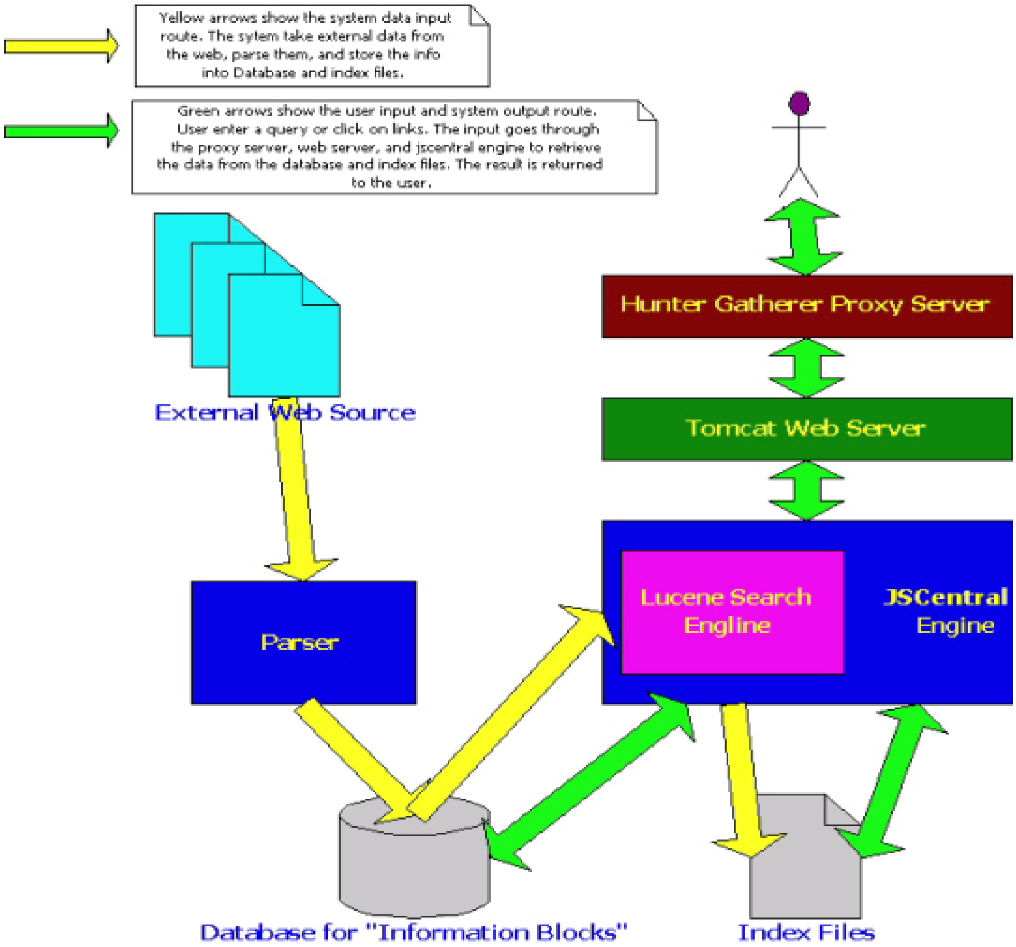


Fig. 8. Architecture diagram

The architecture is designed around the concept of (a) making data available to the interface as information blocks and (b) supporting strong interlinking among components. As Figure 8 above indicates, the current architecture reflects a largely two-stage process: preprocessing of web data for storage into a local database; post processing requests of that data for display. In the first stage, what we call our JSCentral engine gathers information from predetermined JavaScript Web resources, processes it (stores meta data like date, author, original source) into an *information block* a single, meaningful element of information. A block could be the answer to a question, a page of code that solves a problem or represents a function, a glossary description of an object. The engine then parses each block against recurrent key words to tag it for association with categories (a block can be associated with multiple categories). The blocks are then stored into our main database. The JSCentral Engine then parses and analyzes interactions by the user with the interface, and finds the best match among information blocks that relate to a particular selection. The search result is packaged into the specific format for its presentation in the interface. The presentation layer is where our main program logic lives. This component focuses on presenting the information retrieved from the database and processed by the JSCentral Engine to the user. The Hunter Gatherer component, which allows users to collect elements from within the domain, resides on top of the presentation layer. Its architecture is described in [12].

Currently we use the following technologies for our system: mySQL is our open source database solution and Java as the implementation language. We have reused considerable code from the parsing and term extraction components of the Apache Lucene Open Source project for the query interaction part of the service. We use Tomcat as our Web server + servlet/JSP engine. For the presentation layer, we use a combination of Java applets, JavaScript, and HTML to show the content to the users. We use JavaScript and Frames in the presentation layer to implement the spatial view.

This is only a brief overview of the current architecture. For a detailed description of how the engine parses and categorizes information for presentation, please see [19]. We return to architecture issues in the Future Work section below.

5. Conclusions and Future Work

We have seen from our previous work, and the task analysis presented here that for some kinds of information problems, Google is not enough. Further, non-adaptive web sites, such as the various JavaScript sites, cost users time in learning the set up of the site and then how to navigate within this new space. To address these problems, we presented an approach that brings information in a domain into a coordinated spatial context and affords several interaction methods to support users interacting with the content such that they can adapt the content to support their perspectives. We use dimensional sorting in particular for this assist, so that users can reorganize the domain itself to support their perspective. This particular technique was developed as a direct outcome of our task analysis to support usersí current knowledge about a domain, and to endeavour to leverage that knowledge by letting the user see the domain oriented from that perspective. We have run preliminary reviews of our interaction design, but plan a more formal, large scale evaluation to help test the robustness of our current approach is planned. In the preliminary reviews, however, dimensional sorting was frequently deployed, and, consistent with previous evaluations, preview cues enhanced efficiency of path deliberations when navigating through the domain. Participants across expertise ranges reported a desire to have access to the prototype ìas soon as possible,î saying that one of the largest benefits was coordinating multiple resources into a single, associated domain.

The larger goal of the project is to define a general mSpace interaction model for domain access and exploration. If the tests are successful, we can anticipate several parts behind the interaction to which we will need to return. For instance, we do not currently rate information blocks for user level suitability. The current prototpye is an opportunity to determine if such labelling is required or beneficial. It may be enough for the user to see a component both to glean its appropriateness, and to benefit from knowing that that information exists, even if it is not wholly understandable. This seems to be what users are doing with the prototype. We also have yet to incorporate any kind of ranking of the information components. The question remains if, when the user can see much of this information in context, and assess its intelligibility to them directly, whether ranking responses to queries is a priority for effective support. We have predicated that mSpace provides better support information access and exploration; we also wish to investigate how mSpaces may afford knowledge building by supporting persistent context and association building. As indicated above, since the mSpace approach seeks to represent a domain in a structured way (similar to the way structured hypertexts represent a document [17]) we postulate that mSpace can be an effective front end for Semantic Web queries over a particular domain space [13].

Overall, the benefit of this user-determined, adaptable approach to the Adaptive Hypermedia community is to propose a suite of successful interaction approaches that have been shown both to support the tenets of AH (that there is value in adapting content for various users and contexts), but that their particular success is in giving the adaptive control to the user rather than the system. It may be that we have reached a place in AHís evolution where we might be able to create a taxonomy of kinds of information that are more appropriate to adaptable hypermedia and those kinds of discoveries that are more appropriate to adaptive, and, of course, where the twine might meet.

References

1. Brusilovsky, P. Methods and Techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction, 6 (2-3). 87-129.
2. Chimera, R. and Shneiderman, B., An Exploratory Evaluation of Three Interfaces for Browsing Large Hierarchical Tables of Contents. In ACM Transactions on Information Systems (TOIS), (1994), 12.
3. De Bra, P. The Adaptive Web: Adaptive Educational Hypermedia on the Web Communications of the ACM: SPECIAL ISSUE: The adaptive web, May 2002, 60-61.
4. Diaper, D. (ed.), Task Analysis for Human Computer Interaction. Ellis Horwood, Chichester, 1989.
5. Neuwirth, C., Kaufer, D., Chimera, R., Gillespie, T., The Notes Program: A Hypertext Application for Writing from Source Texts, Proceeding of the Acm Conference on Hypertext. In Proceeding of the ACM conference on Hypertext, (November 1987), 121-141.
6. Plaisant, C., Shneiderman, B., Doan, K. and Bruns, T. Interface and Data Architecture for Query Preview in Networked Information Systems. ACM Transactions on Information Systems (TOIS), 17 (3). 320-341.
7. Schaffer, D., Zuo, Z., Greenberg, S., Bartram, L., Dill, J., Dubs, S., Roseman and M., N. Navigating Hierarchically Clustered Networks through Fisheye and Full-Zoom Methods. ACM TOCHI. 162 • 188.
8. Schraagen, J.M., Chipman, S.F. and Shalin, V.L. (eds.). Cognitive Task Analysis. Lawrence Erblaum Associates, Publishers, Mahwah, New Jersey; London, 2000.
9. schraefel, m.c., Contexts: Adaptable Hypermedia. In Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, (Trento, Italy, 2000), Lecture Notes in Computer Science, 1892, 369-375.
10. schraefel, m.c. An Evaluation of Preview Cues for Exploration of Information Hierarchies. Dept. of Computer Science, University of Toronto, CSRG-462, 2002, 1-8.
11. schraefel, m.c., Karam, M. and Zhao, S., Preview Cues for Exploring Domain Hierarchies. In Interact 2003, (Switzerland, 2003, forthcoming).
12. schraefel, m.c., Zhu, Y., Modjeska, D., Wigdor, D. and Zhao, S., Hunter Gatherer: Interaction Support for the Creation and Management of within-Web-Page Collections. In Proceedings of the eleventh international conference on World Wide Web, (Hawaii, 2002), 172-181.
13. Shadbolt, N., schraefel, m.c., Gibbins, N. and Harris, S., Cs Aktive Space, or How We Stopped Worrying and Learned to Love the Semantic Web. In International Semantic Web Conference, (Florida, USA, 2003 in submission).
14. Shepherd, A. Task Analysis in Hci. in Monk, A.F. and Gilbert, N. eds. Perspectives on Hci: Diverse Approaches, Academic Press, London, 1995.
15. Shipman, F., Moore, J.M., Maloor, P., Hsieh, H., Akkapeddi, R., Spatial Hypertext: Semantics Happen: Knowledge Building in Spatial Hypertext. In Proceedings of the thirteenth conference on Hypertext and hypermedia, (June 2002), 25• 34.
16. Terry, M. and Mynatt, E.D., Side Views: Persistent, on-Demand Previews for Open-Ended Tasks. In UIST, (2002), 71 - 80
17. Wang, W. and Rada, R. Structured Hypertext with Domain Semantics. ACM Transactions on Information Systems (TOIS), 16 (4). 372 - 412.
18. Zellweger, P.T., Regli, S.H., Mackinlay, J.D. and Chang, B.-W., The Impact of Fluid Documents on Reading and Browsing: An Observational Study. In CHI 2000, (The Hague, Amsterdam, 2000), CHI Letters, 2, 249 - 256.
19. Zhao, S. and schraefel, m.c. Information Blocks: Working Architecture to Support the Mspace Prototype. Dept. of Computer Science, U of Toronto, CSRG-464, 2002, 1-7.