Adaptive hypertext and hypermedia : workshop : proceedings, 3rd, Sonthofen, Germany, July 14, 2001 and Aarhus, Denmark, August 15, 2001

CS-Report 01-11

Third Workshop on Adaptive Hypertext and Hypermedia

Proceedings

Edited by
P. De Bra
P. Brusilovsky
A. Kobsa

Technische Universiteit Eindhoven
Department of Mathematics and Computer Science

Third Workshop on Adaptive Hypertext and Hypermedia

Sonthofen, Germany, July 14, 2001
Århus, Denmark, August 15, 2001

Proceedings

Edited by P. De Bra, P. Brusilovsky and A. Kobsa

01/11

Series editors: prof. dr. J.C.M. Baeten
prof. dr. P.A.J. Hilbers

This report is available at:
http://wwwis.win.tue.nl/ah2001/
Other reports are available at:
http://www.win.tue.nl/inf/onderzoek/

Computing Science Reports 01/11
Eindhoven, August 2001

Twelfth ACM Conference on Hypertext and Hypermedia

Hypertext'01

Århus, Denmark, August 14-18, 2001

Eight International Conference on User Modeling

UM2001

Sonthofen, Germany, July 13-17, 2001

Third Workshop on Adaptive Hypertext and Hypermedia

Pre-Workshop Proceedings

Note: For each paper it is marked whether it belongs to the UM2001 or the HT'01 session. Final (post-workshop) proceedings will be published by Springer Verlag, in the Lecture Notes in Computing Science.

Long Presentations

Short Presentations

Posters

Position papers

For More Information

Paul De Bra <debra@win.tue.nl>
Department of Computing Science
Eindhoven University of Technology
PO Box 513, NL 5600 MB Eindhoven
The Netherlands
Phone +31 40 2472733
Fax +31 40 2463992
WWW http://wwwis.win.tue.nl/~debra/

The Impact of Empirical Studies on the Design of an Adaptive Hypertext Generation System

Kalina Bontcheva

University of Sheffield, Regent Court, 211 Portobello St., Sheffield S1 4DP, UK
kalina@dcs.shef.ac.uk

Abstract. This paper presents two empirical usability studies based on techniques from Human-Computer Interaction (HCI) and software engineering, which were used to elicit requirements for the design of a hypertext generation system. Here we will discuss the findings of these studies, which were used to motivate the choice of adaptivity techniques. The results showed dependencies between different ways to adapt the explanation content and the document length and formatting. Therefore, the system's architecture had to be modified to cope with this requirement. In addition, the system had to be made adaptable, in addition to being adaptive, in order to satisfy the elicited users' preferences.

1 Introduction

The aim of our research was to design and implement an adaptive hypertext generation system, HYLITE+, which generates factual explanations of domain terminology. The corpus analysis of online encyclopaedia and previous empirical studies (e.g., [16, 5]) have shown the positive effect of additional information – e.g., definition of key vocabulary, less technical content, supply of background information and illustrations – on the subjects' reading comprehension and reading behaviour. On the other hand, hypertext usability studies [14] have shown that hypertext needs to be concise with formatting that facilitates skimming. Therefore, we performed empirical studies to test users' preferences and their perception of several adaptivity techniques. The results were used to establish a set of requirements for HYLITE+ and influenced the choice of adaptivity techniques adopted in the implementation.

For instance, the experiment showed that users prefer different additional clarifying information depending on the chosen formatting and desired explanation length. Another, somewhat unexpected, result was the strong desire of users to control the personalisation techniques applied by the system. Consequently, HYLITE+ was designed to be **adaptable**, in addition to being *adaptive*.

The main difference between our approach and other existing adaptive hypertext generation systems (e.g., [10, 12]) is the use of results from hypertext usability studies, user trials with similar software products, mockups and walkthroughs during system design. The use of these techniques, together with corpus analysis, which is traditionally used in the design of language generation systems [17], enabled the choice of adaptivity techniques, tailored to and by the user.

When compared to adaptive hypertext systems, not just generation-based ones, this work shares most similarities with the MetaDoc system [5], which is an intelligent document reading system. Apart from providing definitions of key vocabulary, it also facilitates readers' comprehension by offering less technical versions and background information. The evaluation results of MetaDoc confirmed the benefit of vocabulary definitions, which has been shown also in earlier studies [16], and together with our corpus analysis, motivated us to choose parenthetical definitions as one of the adaptivity methods to be explored. The main difference between the two systems is that the focus in HYLITE+ is on how to generate the definitions automatically and integrate them appropriately in the explanation. Also, our system is Web-based and produces conventional hypertext, instead of the stretchtext used in MetaDoc.

2 The Empirical Studies

Hypertext readability studies [14] have shown that people read 25% slower on the screen and dislike scrolling. Therefore, unlike printed material, people prefer hypertext with concise, objective content and scannable layout, i.e., the length and formatting of the hypertext are very important. For our system these requirements translate as:

- *brevity* – do not exceed one or, if a more detailed explanation is needed, two pages;
- *structuring* – use formatting that makes it easy to pick out the important information while skimming the text, e.g., bullet lists.

2.1 The First User Experiment

Research in usability engineering [13] has shown that empirical user tests on existing similar products are a productive way to elicit user requirements and facilitate system design. Therefore we performed a limited user trial with an existing electronic encyclopaedia: The Encyclopaedia Britannica CD-ROM [6]. The goal of the experiment was to gain insight into the ways users browse encyclopaedic hypertext, the types of information they prefer, and the best ways to present it.

8 subjects (4 male and 4 female) were asked to find and browse articles related to dispersion (physics sense) and computer memory[1]. The subjects were asked to think aloud and were also interviewed at the end of the session. Their path through hyperspace was logged using software that intercepts the Web browser, and the sessions were also recorded on audio tapes.

The subjects were not given a strict time limit because the idea was to let them decide when they had got enough information since encyclopaedia browsing often does not have a well-defined goal and different people might have different

[1] In this research we followed the discount usability engineering practices [13] which have shown that a small number of expert users is sufficient for this task.

strategies depending on their personalities and interests (e.g., skimming versus in-depth reading of all relevant articles)[2].

The interviews and transcript analysis showed a set of problems which was consistent among the users:

1. The multimedia software did not always show visited links in a different colour, so sometimes users could not recognise easily whether they have already followed a link.
2. Links need to be informative in order to help users decide whether they want to follow them or not.
3. Users do not like following long sequences of links away from the page they are reading since they feel distracted from the main topic.
4. Most users first skim a page to assess whether it is relevant and only afterwards read in detail the parts they are interested in. Consequently, they prefer formatting which facilitates skimming, not the usual mostly textual pages. Most users also decide which links to follow only after skimming the entire article first.
5. Most users found the articles too detailed and expressed a preference for having unimportant information on separate pages connected with links.

In addition, users with background or interest in the subject area (i.e., more familiar with the terminology) found it much easier to navigate through the hyperspace and looked at less pages since they ignored links to already known terms and also judged better whether a link is likely to lead to relevant material. Unlike them, novice users had problem navigating because most links contained unfamiliar specialised terms. They also showed a preference towards examples and figures which help them understand dry, abstract domains (physics, computers).

2.2 The System Mockup Experiment

The user study, the analysis of encyclopaedic texts[3] and previous research in dynamic hypertext (e.g., [10, 12]) suggested various ways for adapting the generated explanations:

1. Provide the user with *definitions of* important *unknown terms* in brackets (used in encyclopaedias to facilitate users' text comprehension; our study suggests it might also improve users' navigation);
2. Provide the user with a *familiar superconcept* in brackets to clarify unknown terms (same as above);
3. *Omit already known information*, e.g., omit mentioning computer parts when describing a computer if the user knows them already;

[2] Previous studies of hypertext usability have already established that most users fall into two broad categories – skimmers (79%) and word-for-word readers [14, p.104].

[3] We analysed a corpus that included texts from Encyclopaedia Britannica Online (`www.eb.com`), Microsoft Encarta (encarta.msn.com), and Harcourt Academic Press Dictionary of Science and Technology (`http://www.harcourt.com/dictionary`).

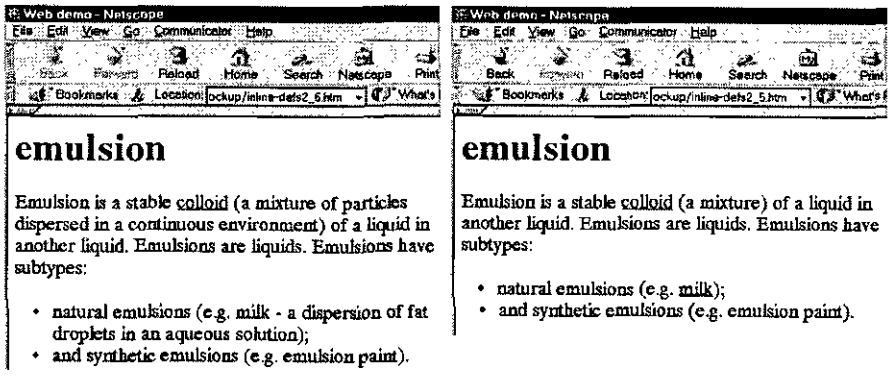


Fig. 1. The preferred version with the clarifying definition (left) and the one with a familiar supertype (right)

4. Contextualise the explanation by *referring to previously seen material*. For example, use phrases like 'As already mentioned' at the beginning of an already visited page or an already seen fact.
5. Use syntactic structures that refer explicitly to *previously seen material* when it is part of a sentence (e.g., 'Besides dispersion, other characteristics...');
6. When a user is returning to an already visited page, modify its content to take into account what was seen in the mean time;
7. Include *links to other related material* or if there is space, include this material on the page (e.g., information about the subtypes of the explained concept).

Before implementing these features in HYLITE+, we decided to test the user perception of their usefulness, that of adaptivity in particular, because many were derived from research on text/dialogue generation and might not fit well with the user expectations about encyclopaedic hypertext. Some of these alternatives have been explored in previous work on dynamic hypertext (e.g. [10, 12]). In particular, these systems explored the notion of hypertext as a dialogue between the system and the user. Therefore, the content and presentation of the hypertext pages, including those previously visited, were generated by taking into account the interaction history and the user model. Empirical evaluation, in the context of museum browsing [9], has shown that the participants did not report problems with the changing nature of previously visited pages. However, these results were obtained in a mixed-initiative application, which is different from the user-controlled interaction typical for information systems like ours.

Research in Human-Computer Interaction has shown that one fast, yet effective, way to test alternative designs is by using *predictive evaluation* techniques [15], which involve a small number of users that test a set of scenarios, realised as paper mockups and walkthroughs.

We created paper mockups for several user interaction scenarios which were used to test users' preference of different adaptivity techniques. Each scenario

consisted of one or more set(s) of hypertext pages which presented approximately the same information in alternative ways (see Figure 1 for an example of two such alternatives (out of 6 in total for this scenario)). The scenarios which tested different ways to adapt the presentation depending on previously seen material consisted of a sequence of pages which were given to users one after another as they pretended to follow a given link. The scenarios were desinged so that they only focus on one ot two adaptivity techniques at a time.

The participants from the previous study were asked to rank the different page versions according to their own preferences and explain the reasons for their choice[4]. All subjects were experts in hypertext and Web browsers and have already interacted with a similar online system, so they could successfully simulate computer interaction using the paper mockups. Again, we used two domains – chemistry and computer science, so the same users were both novices in the former and experts in the latter domain.

The mockup consisted of screen shots of the hypertext alternatives displayed in Netscape. The subjects were asked initially to customise the window and font size according to their preference and then the mockup material was produced to look exactly as it would on their screens. This was particularly important because, e.g., visually-impaired users use much larger fonts and their page ranking might have been affected if the experiment conditions did not match their everyday use.

The mockup experiment had to be performed on paper, because for most subjects it was not possible to show on the computer screen more than 2 windows in parallel. Most scenarios consisted of at least 4 alternatives, so we used a big table where the alternatives could be viewed simultaneously and compared. The order in which the alternatives were arranged was changed at random between the subjects.

The experiment differentiated two types of users with respect to content: those who always preferred the most *concise texts* with links which they can explore further; and those who rated higher texts with *additional information* which might even lead them to material they did not initially intend to read. These preferences were consistent in all scenarios.

Due to space constraints, here we will only discuss the three scenarios which tested users' attitudes towards clarifying information (e.g., definitions of important unknown terms and familiar supertypes); a detailed discussion of the scenarios on adapting previosly visited pages and presentation of previously seen material is available in [2].

One scenario tested the use of *clarifying information inside definitions*; the second tested its use in descriptions of *object parts*; the third one covered descriptions of *object subtypes*. All scenarios covered several alternatives:

1. provide *only a link* to the term (concise);
2. provide *a familiar supertype* in brackets and a link to the term;
3. provide *a definition* of the unknown term in brackets and a link to the term;

[4] We chose to use the same subjects since they already had some experience with the electronic encyclopaedias and were familiar with the problems of using such systems.

4. include a familiar supertype and no link to the term;
5. include the definition and no link to the term.

In all scenarios users always preferred to have the links included, because otherwise they would have to perform a search if they wanted to find further details about the term. Also, somewhat surprisingly, the experiments showed a connection between formatting and preferred alternatives. For example, definitions are acceptable when they are not too long (about 10 words), i.e., do not interfere with the flow of the main explanation. In parts and subtypes descriptions, definitions are preferred when list formatting is used because it makes it easier to ignore them when skimming the page (see the generated example in Figure 2).

For the first scenario, half of the experts preferred the text with the definition in brackets (Figure 1, left), whereas the other half rated the one with the familiar supertype best (right). The difference comes from overall personal preference for concise versus more informative texts but there is also a connection with the user's familiarity with the words used in the definition.

For the second and third scenarios the most preferred version was the one that used lists to enumerate all the parts/subtypes and provided short definitions of them (see the generated example in Figure 2). For terms where the system had further information (e.g., properties), links were also provided. The preference for definitions is not dependent on users' expertise in the domain, because the definitions can be easily ignored while skimming. In fact, one of the experts said she would rather have the short definitions there, rather than follow the link only to discover that the page contains just this information (i.e., is of no interest to her).

3 Summary

To summarise, all users exhibited strong preference for well-formatted, concise explanations, where further detail and additional information can be obtained from links and the form interface.

The scenarios which tested different ways of providing clarifying material showed that some users always preferred the shortest text with links to further detail, while others always chose relatively concise, but more informative, explanations. Therefore the system interface was designed to allow easy selection of different levels of explanation detail with further finer-grained tuning available from the user preferences page.

The results from these scenarios and those on adaptation of previously seen material showed that the participants had widely different opinions. For example, one of the users found phrases like as you have already seen and as you probably remember too patronising and would want to disable their use, although she liked the other adaptivity ideas. The fact that none of the other users disliked these phrases shows how individual these preferences can be.

Therefore, the polarity in the user preferences motivated us to adopt a more individualistic approach, where users can customise the system adaptivity be-

Type of adaptivity	Default behaviour	User choices
Links to related pages	after the explanation grouped as More general, More specific, Similar	disable
Return to a visited page - using Back - using a link	 show same page modify page opening	 disable modification customise the page opening phrase
Already seen material	include with a cue phrase	disable
Clarify unknown terms - short, to-the-point text - more informative text	 known superconcept in brackets and short definitions of parts/subtypes short definitions	switch to links only switch to definitions switch to superconcepts
Related information	only as a link to a related page	include as section in current page

Table 1. Adaptivity techniques summarised

haviour. The adaptivity techniques preferred by the majority of the mockup users are enabled by default in our implementation and the interface allows users to change them easily, including disabling all personalisation.

The users also expressed a desire to have control over the personalisation techniques applied by the system, i.e., customise the system behaviour with respect to both adaptivity and language use. Consequently, the system was designed and implemented to be adaptable in addition to being adaptive. In order to help users customise the system without interrupting the interaction, only adaptivity alternatives relevant to the current page are made available at the bottom of each generated page, with the full set of choices available from a separate preference page. For example, if the system has used definitions of unknown terms and links to related pages, only options related to these techniques (e.g., disable related links, switch to known superconcepts, disable all adaptivity) are displayed. This interface is based on HTML forms with check boxes and radio buttons for ease of use. These preferences are also stored in the user's model, so they can be retrieved and used in future sessions.

Finally, one must be aware of the possible discrepancies between user preferences regarding some system features and the impact of these features on users' performance. Therefore, when choosing the default system configuration or behaviour, it is also important to consider relevant results from existing task-based experiments. For example, the disabling parenthetical definitions for unfamiliar terms might lead to a decreased reading comprehension, because previous studies have shown their benefit [5]. On the other hand, denying users the possibility to change the system behaviour according to their liking could damage their acceptance of the adaptivity, because, as shown in our empirical evaluation (see

A DAT (Digital Audio Tape) is a tape drive, that is used to backup small to medium-sized network servers, personal computers, and UNIX workstations. A DAT has subtype(s)

- a DAT8 (a DAT, that is used to backup small network servers and personal computers),
- a DAT24 (a DAT, that is used to backup medium network servers and bigger personal computers), and
- a DAT40 (a DAT, that is used to backup network servers).

A DAT stores between 8GB and 40 GB of data.

More generic: tape drive

More specific: DAT24, DAT40, DAT8

Other related terms: DLT, Ultrium

Term: computer / cordless mouse / DAT / DAT8

Explain | Clear Data

Fig. 2. An automatically generated text with added definitions and links to related material

Section 6), users want to have the option to change system features they do not like.

4 The Implemented System

Based on these results, we implemented an adaptive hypertext system which, similar to [12, 10, 1] , uses Natural Language Generation (NLG) techniques to create dynamically the hypertext nodes and links. HYLITE+ generates factual explanations of domain terminology which have been developed and evaluated in the domains of chemistry and computer hardware. The need for such explanations, for example in e-commerce, has been proven in practice by the increasing number of online computer shops that provide reference guides and tutorials (see e.g. `www.action.co.uk`). Computer magazines like 'What laptop' and 'What PC' also have terminological glossaries, as part of their buyer's guides.

Following the distinctions made in [7], HYLITE can be classified as an *on-line information system* which provides *referential* information, without having educational goals as do, for example, intelligent tutoring environments. The infor-

mation is requested by users with different knowledge and interests and typically each hypertext node corresponds to a domain concept.

The user interacts with the system in an ordinary Web browser (e.g., Netscape, Internet Explorer) by specifying a term she wants to look up. Further information about subtypes, parts, and other related concepts is obtained by following hypertext links or specifying another query.

Similar to all Web applications, HYLITE+ needed to *(i)* respond in *real-time*, i.e., avoid algorithms with associated high computational cost; and *(ii)* be *robust*, i.e., always produce a response. Consequently the system uses some efficient and well-established applied NLG techniques such as text schemas and a phrasal lexicon (see [17, 12, 10]).

The system consists of several modules organised in two main stages: *(i)* content organisation, which includes *content selection*, *text organisation* and *semantic aggregation*; and *(ii) surface realisation* modules [3, 4]. The adaptivity is implemented on the basis of a user and a discourse models. The user model is updated dynamically, based on the user's interaction with the system. When a user registers with the system for the first time, her model is initialised from a set of stereotypes. The system determines which stereotypes apply on the basis of information provided by the user herself. If no such information is provided, the system assumes a novice user.

Unlike previous NLG systems which have their own, application-specific user models, our adaptive hypertext system has re-used a generic agent modelling framework (ViewGen) instead [3]. Apart from avoiding the delopment costs of a new model, this also enabled a more modular and extendable system architecture. As argued by [8], such modularity and re-use are much desired in adaptive hypertext systems and one way of achieving that is by using generic user models, such as BGP-MS [11] and ViewGen.

The user model is used to determine which concepts are unknown, so clarifying information can be provided if appropriate (e.g., parenthetical definitions for parts/subtypes). An example of a hypertext explanation generated by the adaptive system appears in Figure 2. In this example, the user model did not contain these concepts as known, so all three types of DAT drive are explained briefly in parenthesis. More detailed information about each one of them is available by following the hypertext links provided.

The model is also used to detect misconceptions, which might come either from a user stereotype or individual's beliefs. An example of a generated explanation addressing a common misconception follows:

> Emulsion is a stable colloid (a mixture of disperse phases in a continuous environment) of a liquid in another liquid. Emulsion has subtypes natural emulsion (e.g. milk) and synthetic emulsion (e.g. emulsion paint).
>
> Typically people believe that photographic emulsion is emulsion, whereas in fact, photographic emulsion is gel.

4.1 Design Impact 1: Adaptability

As discussed above, the mockup experiment revealed a lot of variation in user preferences which motivated us to adopt an individualistic approach, where users can customise the system's adaptive behaviour.

At present, the user has control over the user model and the adaptivity features, some of which are listed below. Table 1 showed the adaptivity features which are enabled by default.

- User modelling:
  - whether or not the system *updates their* VIEWGEN *model*;
  - whether or not the system *uses their* VIEWGEN *model*;
  - whether or not the system provides *information inherited* from more generic concepts;
  - whether or not the system uses *stereotypes*;
- Adaptivity features
  - whether or not the system generates *parenthetical definitions*;
  - whether or not the system presents *familiar supertypes* in brackets;
  - whether or not the system provides *links to related material*;
  - whether or not the system uses *contextualising phrases* like besides.
- *Disable all* adaptivity and user modelling.

When an adaptivity feature is disabled, this affects the generation algorithms. For example, if parenthetical definitions are disabled, but familiar supertypes are still enabled, then only the latter will be generated for unfamiliar terms, because the rules for the generation of the former will not fire.

The adaptability also proved useful during the development and testing of the generation algorithms, because it offered control over the corresponding functionality. In this way it was possible to examine the influence of each of these features on the generated hypertext (a kind of ablation experiment).

4.2 Design Impact 2: Recursive Architecture

As shown by our studies, there are several ways to provide additional information about unknown terms in generated encyclopedic entity descriptions. When such information is needed, the most appropriate clarification needs to be chosen depending on formatting, user knowledge and constraints (e.g., concise versus detailed pages). Each alternative requires different text content to be selected at the start of the generation process but the choice of alternative can only happen after the content and formatting for the main description have already been determined. Therefore, the typically used pipeline NLG architecture was extended to allow some module feedback. In the resulting *recursive architecture* additional content can be requested in later stages of the generation process, only if necessary. Below we provide an example to demonstrate how the recursion operates. A more detailed NLG-oriented description of the architecture is given in [4].

5 Generation Example

Let us assume a user who has looked up computer programs and then followed a link to personal computer; the user has not specified preferences for types of clarifying information, so definitions can be provided where appropriate. Following this request, the content organisation stage passes the following facts for realisation as text (the fact listing all parts is truncated here[5]):
[PC] <- (ISA) <- [MICRO_COMP *fs(um_state:unknown)*].

[PC] <- (PART_OF) <- [CPU *fs(um_state:unknown)*]
- (PART_OF) <- [MEMORY *fs(um_state:explained)*]
- (PART_OF) <- [HDD *fs(um_state:unknown)*]
- (PART_OF) <- [DISPLAY]...

First the realisation component determines the document formatting; in this case a bullet list is chosen to enumerate all parts. Then it starts generating text for the first graph. Because the introduced supertype is unknown, but important for the understanding of the text, the generator decides to provide clarifying information in parenthesis. Since it is not always appropriate to include parenthetical information, e.g., because the user has disabled this feature or because she has requested very short texts, such clarifications are generated recursively, only if they are appropriate.

An example of a recursively generated definition is shown in Figure 3: "a small computer that uses a microprocessor as its central processing unit". Similarly definitions are generated recursively for all unknown parts of the PC.

The only exception here is computer memory, because it has already been explained in a previous page (um_state:explained). In this case, the generator uses the discourse history to determine whether memory is recently explained. If so, additional information is not needed, so no recursive calls need to be made and the final text will only contain the term and a hypertext link, in case the user wants further detail.

6 Evaluation

The acceptability and utility of the adaptive features were evaluated by users who interacted with two versions of the system: a baseline one, with the user model and adaptivity disabled, and the default adaptive version (see Table 3). Since the goal was to evaluate the adequacy of the adaptivity methods, a small-scale formative evaluation was chosen, where the participants provided detailed feedback through questionnaires and semi-structured interviews. In this way,

[5] The facts are encoded as conceptual graphs, a type of semantic network, where concepts are written in square brackets and relations in round ones. Extra information (e.g., whether a concept is familiar to the user, as determined on the basis of the user model) can be associated with concepts and graphs as feature structures.

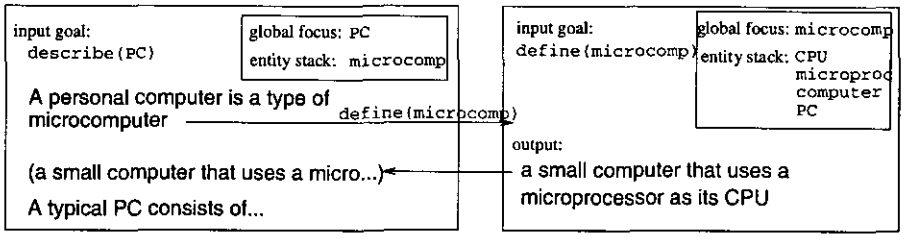


Fig. 3. An example of recursively generated parenthetical information

it was possible to gather qualitative results, which would help improving the system. Quantitative data was also gathered and analysed, but due to the small user sample (8 users), it was not possible to obtain statistically significant results.

The qualitative results showed that users prefer adapted explanations to the neutral version where information about unknown terms is not automatically provided but has to be accessed by following a link. If relevant to their task, the additional information provided by the adaptive system was also used by the participants to minimise the number of visited irrelevant pages. As stated by one of the users in their interview: "They [parenthetical definitions and examples] helped me to determine which pages are more relevant to me. In the non-adaptive system I had no choice but to visit them all." In addition, the users' subjective opinion showed that the adaptive system was easy to use and did not confuse them. Finally, 75% agreed that working with the adaptive system was more enjoyable than with the non-adaptive one, while only 12.5% disagreed.

The evaluation also showed that the acceptability of the adaptive system can be improved even further if its interface provided users with a way of changing the default system behaviour. For example, one of the users did not like the links to related information, included at the bottom of the page, while she liked the rest of the system. These results validated the conclusion from the mockup experiment, that people have different preferences, so the system needs to provide users with a way of controlling the default adaptive behaviour. Since the adaptability has already been implemented, it only remains to evaluate it by comparison with the default adaptive system, which does not allow user control.

Finally, probably the most important outcome of this formative evaluation was that it showed the need to control not just for user's prior knowledge (e.g., novice, medium, advanced), but also for reading style. Although previous studies of people browsing hypertext (e.g., [14]) have distinguished two types: *skimmers* and *readers*, in this experiment we did not control for that, because the tasks were concerned with locating information, not browsing. Still, the results obtained showed the need to control for this variable, regardless of the task type, because reading style influences some of the quantitative measures (e.g., task performance, mean time per task, number of visited pages, use of browser navigation buttons). Due to space constraints, we will refer the reader to [2] for

details on the evaluation experiment, including the exact quantitative results obtained.

7 Conclusion

The paper presented the *empirical studies* which were used to design the dynamic hypertext generation system HYLITE+. The results influenced two main aspects: (i) adaptable adaptivity: giving control to the users, so they can control the system's default behaviour; (ii) recursive architecture, which allows additional information to be generated only when necessary.

The formative evaluation of the implemented system showed that the adaptivity techniques designed on the basis of our empirical studies, were found acceptable and useful by the users.

While the empirical results on preferred adaptive behaviour are probably not applicable to applications other than intelligent online information systems, the low-overhead HCI techniques used in these emprical studies have shown their effectiveness and could be easily applied to facilitate the design of other adaptive hypertext applications. Meeting user expectations and designing the system with users in mind is particularly important for Web-based systems because users have strong preferences and if a Web site does not live up to their expectations, they can often go to other sites instead.

Acknowledgements

We wish to thank Yorick Wilks, Hamish Cunningham, Peter Brusilovsky, and the anonymous reviewers for their helpful comments and suggestions.

References

1. Liliana Ardissono and Anna Goy. Dynamic generation of adaptive web catalogs. In Peter Brusilovsky, Oliviero Stock, and Carlo Strapparava, editors, *Adaptive Hypermedia and Adaptive Web-Based Systems*, number 1892 in Lecture Notes in Computer Science, pages 5—16, Berlin Heidelberg, 2000. Springer Verlag.
2. Kalina Bontcheva. *Generating Adaptive Hypertext Explanations with a Nested Agent Model.* PhD thesis, University of Sheffield, 2001.
3. Kalina Bontcheva and Yorick Wilks. Generation of adaptive (hyper)text explanations with an agent model. In *Proceedings of the European Workshop on Natural Language Generation (EWNLG'99)*, pages 67 – 76, Toulouse, France, May 1999.
4. Kalina Bontcheva and Yorick Wilks. Dealing with dependencies between content planning and surface realisation in a pipeline generation architecture. In *Proceedings of the International Joint Conference in Artificial Intelligence (IJCAI'2001)*, Seattle, USA, August 2001. To appear.
5. Craig Boyle and Antonio O. Encarnaçion. Metadoc: An adaptive hypertext reading system. *User Modelling and User-Adapted Interaction*, 4(1):1 – 19, 1994.
6. Britannica Inc. *Encyclopaedia Britannica CD'99.* 1999. Multimedia Edition.
7. Peter Brusilovsky. Methods and techniques of adaptive hypermedia. *User Modelling and User-Adapted Interaction*, 6(2-3):87–129, 1996. Special issue on Adaptive Hypertext and Hypermedia.

8. Peter Brusilovsky. Adaptive hypermedia. *User Modelling and User-Adapted Interaction*, 11(1–2):87–110, 2001.
9. Richard Cox, Mick O'Donnell, and Jon Oberlander. Dynamic versus static hypermedia in museum education: an evaluation of ILEX, the intelligent labelling explorer. In S.P. Lajoie and M. Vivet, editors, *Artificial Intelligence in Education: Open Learning Environment: New Computational Technologies to Support Learning, Exploration and Collaboration*, pages 181–188. IOS Press, Amsterdam, 1999.
10. Alistair Knott, Chris Mellish, Jon Oberlander, and Mick O'Donnell. Sources of flexibility in dynamic hypertext generation. In *Proceedings of the 8th International Workshop on Natural Language Generation (INLG'96)*, pages 151 – 160, 1996.
11. Alfred Kobsa, Andreas Nill, and J. Fink. Hypertext and hypermedia clients of the user modelling system BGP-MS. In Mark Maybury, editor, *Intelligent Multimedia Information Retrieval*. MIT Press, 1997.
12. Maria Milosavljevic, Adrian Tulloch, and Robert Dale. Text generation in a dynamic hypertext environment. In *Proceedings of the 19th Australian Computer Science Conference*, Melbourne, 1996.
13. Jakob Nielsen. *Usability Engineering*. Morgan Kaufman, CA, 1993.
14. Jakob Nielsen. *Designing Web Usability: The Practice of Simplicity*. New Riders Publishing, 2000.
15. Jenny Preece, Yvonne Rogers, Helen Sharp, David Benyon, Simon Holland, and Tom Carey. *Human-Computer Interaction*. Addison-Wesley, Reading, MA, 1994.
16. David Reinking and Robert Schreiner. The effects of computer-mediated text on measures of reading comprehension and reading behaviour. *Reading Research Quarterly*, Fall:536—551, 1985.
17. Ehud Reiter and Robert Dale. *Building Natural Language Generation Systems*. Cambridge University Press, Cambridge, England, 2000.

User-Adapted Image Descriptions from Annotated Knowledge Sources

Berardina De Carolis and *Fiorella de Rosis*

Intelligent Interfaces, Department of Computer Science,

University of Bari, Italy

{decarolis, derosis}@di.uniba.it

Abstract. We present the first results of a research aimed at generating user-adapted image descriptions from annotated knowledge sources. This system employes a User Model and several knowledge sources to select the image attributes to include in the description and the level of detail. Both 'individual' and 'comparative-descriptions' may be generated, by taking an appropriate 'reference' image according to the context and to an ontology of concepts in the domain to which the image refers; the comparison strategy is suited to the User background and to the interaction history. All data employed in the generation of these descriptions (the image, the discourse) are annotated by a XML-based language. Results obtained in the medical domain (radiology) are presented, and the advantage of annotating knowledge sources are discussed.

1 Introduction

The amount of heterogeneous information available on the Web is growing exponentially; this growth makes increasingly difficult to find, access, present and maintain information. From research about how to make these tasks easier, methods for making machine understandable multimedia web resources have emerged: these methods require associating semantics to information, through the use of metadata. The description of such metadata is typically based on a domain conceptualization and a definition of a domain-specific annotation language. An annotation can be loosely defined as "any object that is associated with another object by some relationship" (from the W3C Annotation Working Group). In particular, XML is a standard, proposed by the W3C, to create mark-up languages for a wide variety of application domains; developing such languages favours universal storage and interchange formats, re-use and share of resources for web distributed knowledge representation and programming [11].

Metadata annotation of web resources is essential for applying AI techniques for *searching* and *extracting* relevant information (by improving a semantic contextualized search), for *maintaining* web resources (by keeping them consistent, correct and

up-to-date). More recently, there is a tend to employ it, as well, for *automatic document generation* especially when user-adaptation has to be considered.

Introducing annotations in a NLG systems requires two main steps:

1. *defining annotations for knowledge sources* in the application domain and for the intermediate results of the generation process; whenever possible, already existing and shared annotation languages should be employed (especially as far as application domain data are concerned);
2. *revising the NLG algorithms* so as to enable every generation module to read annotated data and to produce annotated results.

As far as user adaptation is concerned, annotating resources increases the possibility of finding information of interest to a particular user and to denote which particular piece of information is relevant for a particular interaction context. Annotating the steps of the generation process (for instance, the discourse plan) enforces a distributed vision of the process and enables rendering the final output as a function of the device through which the user interacts. This vision become particularly attractive when the resource to be described and explained to the user is an image. There are millions of images on the web that could be accessed for different uses and purposes, understanding their semantics would give the possibility of using them in several ways: for instance, for searching an image, for extracting useful information related to it, for creating image ontologies, for describing them, or relevant portions of them, verbally or textually, and so on.

In this paper, we will focus on this last aspect and in particular on the generation of user-adapted image descriptions in web-based consultation systems eventually accessible using different devices.

For this purpose, we need: i) to "understand" images, ii) to organize them into appropriate ontologies, iii) to define a user modelling component that formalizes the user features that are relevant for adapting the description, and iv) to generate the description more appropriate to the user and to the interaction context. This adaptation process may be seen as follows: given an knowledge base of images together with the related metadata describing them, a user model containing information about the user level of knowledge in the application domain, a list of already seen images during the interaction and the interaction context:

- select the attributes to include in the description and the level of detail of their description;
- select the appropriate description strategy" (an image can be described individually or by comparison with an image in the ontology that is known to the User);
- define the appropriate way to present the relevant information according to the context (i.e. web vs. wap);

To test our approach, we choose the medical domain in which image-based examples are very common to describes normal anatomical sites as well as particular pathologies. In particular, in order to show example of how it works in real domain application, we choose the context of hypertext for consulting medical guidelines ARIANNA (for more details see [3]). ARIANNA is a system aimed at dynamically generating user adapted hypermedia presentations of medical guidelines. Our medi-

cal partners envisage using this system to instruct students and to spread guidelines among general practitioners and specialists. In addition to the guideline, the prototype we developed is able to dynamically generate user adapted explanations of concepts involved in the clinical decision process: in this context, the User may ask to see some example about the explained concept, to better understand it; this example may be described either individually or by comparison with other cases, that the User is presumed to already know. As we work in the radiological domain, most of the examples to show are illustrated by images; therefore, our goal is to automatically generate context-dependent image descriptions, and we need "understanding" images to this purpose. The potential users may be classified as i) *students*, who may learn diagnostic and therapeutic procedures to follow in specific situations; ii) *doctors* with several degrees of competence, who may apply correct diagnostic and therapeutic procedures; iii) *patients*, who may get information about the scope and the efficacy of the health treatment they have to undergo.

In this context we used an image annotation tool to generate an XML structure correspondent to the metadata associated to it. For this purpose, we defined a XML-based mark-up language for radiological images and we developed an algorithm for interpreting its semantics. Starting from a set of annotated images, the information contained in the user model and given a communicative goal that formalises the User request of seeing an image example, a discourse plan is produced. This plan is built by taking into account the user's information needs and her background knowledge, and specifies the information content and the structure of the description text [3,10]: it is written also as an XML-structure, according to a mark-up language that we defined for this purpose. The annotated plan is the input of the surface generator that, according to the interaction context and to the User characteristics, decides how to render it.

In the following Sections, we will describe our approach by focusing, in particular, on how we use the annotation in the NLG process and by discussing the impact that an XML-based annotation may have on this process.

2 Generation of Image Descriptions

The explanation facility of ARIANNA uses two main strategies to generate the concept description that is appropriate in a given context: the concept position in a taxonomy of medical concepts and its relation with "similar" concepts that the User knows. If the User does not know other "similar" concepts, the generated text provides a complete description of the concept itself, in which its position in the taxonomy is specified by describing the relations with its ancestors. If, on the contrary, the User knows other concepts in the taxonomy (for instance because she has just seen their description), an explanation by comparison with the most similar of them is provided. To select the reference concept, a 'degree of similarity' between concepts is measured, by considering the attributes they have in common; the comparison then includes in the description the 'commonalities' and of the 'alignable' and 'non alignable' differences [9]. Only properties appropriate to the User level of knowledge

are mentioned in the text: commonalities are presented first, alignable differences second and non-alignable differences at the end. This strategy corresponds to what we consider a systematic description of concepts, which is typical of learning tasks, as opposed to information-seeking ones [7]

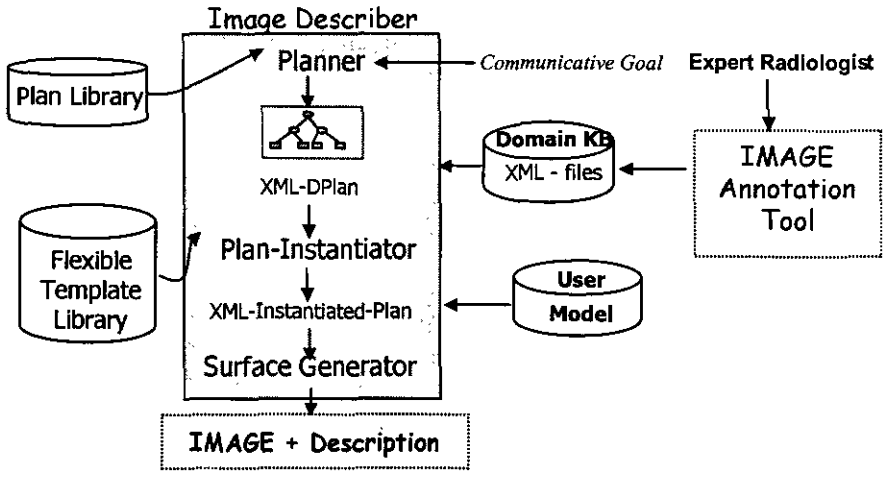


Fig. 1. The Architecture of Image Describer

As we mentioned in the Introduction, in the context of these explanations, the User may ask to see an example about the explained concept; these examples are in the form of radiological images, that have to be illustrated through some natural language text. In our first prototype of ARIANNA, image descriptions were pre-stored comments; this required our radiologists to provide a text for every example image and did not allow us to tailor it to the context. We therefore thought about applying, to produce image descriptions, strategies similar to those we applied in the case of concept explanations, so as to generate automatically texts by also taking into account adaptivity to the User knowledge. However, this goal required that our generator be able to "understand" images: let's see how we did it.

3. Understanding the Image

Understanding a image means extracting the features that characterize the information needed for its description: typically, these features are regions with their shape, texture, edges and so on. Since we do not use automatic image recognition techniques to extract these features, we use metadata to describe the image components, their attributes and the relationships among them. To build these metadata, we use an annotation tool (Inote [8]) in Java that is available on line and provides a way of annotating images with a XML-based mark-up language. Inote allows the User to attach textual annotations to an image and to store them in a text file as XML data,

through a XML structure that organizes them appropriately. With this tool, our medical partners can mark-up a digital radiological image by directly "writing on it" and without altering it; once a image has been loaded, the borders of one or more regions in the image may be outlined interactively, and a number of attributes may be associated with each of them. Regions are called "details" and attributes "annotations", and may be given a name; a text may be associated with every annotation of every detail, by filling a text field. The details may be organized into as many "overlays" as needed. Inote's mark-up language is very general, and may be applied to every kind of image. To tailor it to radiological images, we defined an ad hoc markup language that allows us to identify overlays and details in our images, with their attributes, in a univoque and unambiguously interpretable way. A radiological image has some "General Properties" that identify it: the technique with which the image was produced, the body region on which the exam was performed and the diagnosis. Its main information content then consists in a list of details that correspond to the regions of interest (anatomic structures); a set of characteristics (morphology, density, etc.) is associated with each of them.

```
<overlay>
    <title>parenchymal organs</title>
    <detail>
        <title>liver</title>
        <annotation>
            <title>position</title>
            <text>left</text>
        </annotation>
        <annotation>
            <title>morphology</title>
            <text>ellipsoidal</text>
        </annotation>
        <annotation>
            <title>volume</title>
            <text>normal</text>
        </annotation>
        <annotation>
            <title>margins</title>
            <text>regular</text>
        </annotation>
            </detail>
</overlay>
```

Fig. 2. An example of XML structure produced by Inote.

The first overlay in the Inote file then defines the "General Properties"; it is followed by other overlays, representing groups of visible details.

For instance, in the CT of abdominal organs, the following overlays may be defined:

- parenchymal organs
- hollow organs
- vascular structures
- muscular structures
- skeletal structures

The overlay named 'parenchymal organs' includes, as details, the organs in the image that belong to this category: the liver, the spleen and the lung parenchyma.

For each organ or detail, the following attributes may be specified: position in the image, relation with other parts, morphology, volume, density and margins. Each of them corresponds to an annotation. The example in Fig. 2 is a portion of the XML structure that was produced by Inote for a CT-scan (Computerised Tomography) of the abdomen. Figure 3 shows how this information was introduced, with Inote's graphical interface. In particular, the expert radiologist after the graphical marking of the liver, is entering the annotation for the 'morphology' attribute.

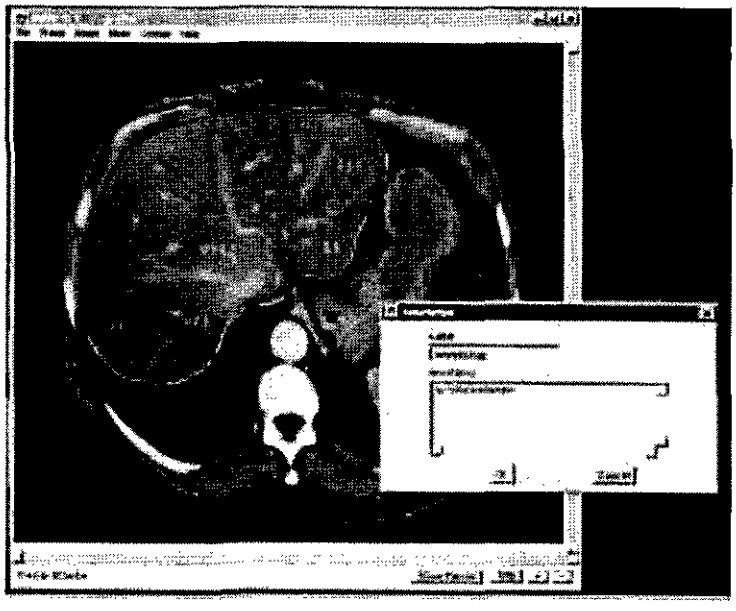


Fig.3: An example of a CT-scan annotation with Inote.

4 Planning the Image Description

The XML structure produced by Inote represents the knowledge base for our description generator. Before generating texts, our XML-application has to interpret the Inote tags and the detail and the overlay to which every annotation belongs, so that sentences describing the image can be built correctly. The generation decides the text structure according to the discourse plan that corresponds to given communicative goal: for instance, "Describe (System User I)", where I denotes a specific image in the domain KB. According to this goal and to the User characteristics, a presentation plan is selected from a library of non-instantiated plans that are represented as XML structures too; the generic plan is, then, instantiated by filling the slots of its leaves with available data in XML-domain-files. The DTD definition of our Discourse Plan Markup Language is shown in Fig.4. In this specification, a discourse plan is identified by its name; its main components are the nodes, identified by a name, containing mandatory attributes describing the communicative goal and the rhetorical elements (role in the RR of its father and rhetorical relation) attached to it. Then the 'info' element, that is not mandatory, describes additional information, related to a node, concerning the focus of the discourse and the complexity of the sub-tree de-

parting from it. These optional information elements are not used in this particular application, but they are necessary in other NLG systems developed by our research group [4, 5]. The XML-based annotation of the discourse plan is driven by two reasons, the first is that in this way it is possible to build a library of standard explanation plan that can be instantiated when needed and used by several applications working in several contexts; the second one is that we have chosen to use XML has a standard interface between all the modules constituting our generators, favouring in this way the distribution of resources and computation.

DPML 1.0 – Discourse Plan Markup Language

```
<!DOCTYPE d-plan[
<!ATTLIST d-plan name CDATA #REQUIRED>
<!ELEMENT node (node*, info*)>
<!ATTLIST node name CDATA #REQUIRED goal CDATA #REQUIRED
        role (root|nucleus|sat) #REQUIRED RR CDATA #IMPLIED>
<!ELEMENT info EMPTY)>
<!ATTLIST info focus CDATA #REQUIRED compl (H|M|L) #REQUIRED >
]>
```

Fig4. Discourse Plan Markup Language DTD.

A small portion of the XML-Instantiated-Plan that was produced for describing the C.T. scan of the abdomen in Figure 3 is shown in Fig. 5. In this case, the XML-annotated plan has been instantiated according to the information relative to 'img1.xml' (as it is possible to notice from the goal of the tree root 'Explain(image, img1.xml)').

```
<d-plan name="CT-abdomen.xml">
  <node name="n1" goal="Explain(Image, img1.xml)" role="root" RR="Sequence">
    <node name="n2" goal="Describe(General Features, image)" role="nucleus" RR="ElabGenSpec">
      <node name="n4" goal="Inform(diagnosis,normal liver)" role="nucleus" RR="null"/>
      <node name="n5" goal=" Describe(Exam, C.T.)" role="sat" RR="Joint">
          <node name="n6" goal="Inform(name, C.T. Abdomen)" role="nucleus" RR="null"/>
          <node name="n8" goal="Inform(level, spleen)" role="nucleus" RR="null"/>
      </node>
    </node>
    <node name="n3" goal="Describe(Specific Features, image)" role="nucleus" RR="OrdinalSequence">
      <node name="n9"goal="Describe(ComplexStructure-1, parenchymal_organ)" role="nucleus"
      RR="OrdinalSequence">
          <node name="n10" goal="Describe(detail,liver)" role="nucleus" RR="ElabGenSpec">
            <node name="n12" goal="Describe(attribute,liver)" role="sat" RR="Joint">
              <node name="n13" goal="Inform(position,left)" role="nucleus" RR="null"/>
              <node name="n16" goal="Inform(rel_position,medialpart_abdomen)" role="nucleus" RR="null"/>
              <node name="n17"goal="Inform(morphology,ellipsoidal)"role="nucleus" RR="null"/>
              <node name="n18" goal="Inform(volume,normal)" role="nucleus" RR="null"/>
              <node name="n19" goal="Inform(margins,regular)" role="nucleus" RR="null"/>
            </node>
              <node name="n11" goal="Inform(name,liver)" role="nucleus" RR="null"/>
          </node>
          ...
</d-plan>
```

Fig. 5. An example of XML-Instantiated-Plan.

5 Rendering the Image Description

This functionality of our Image Describer is very simple; the XML-Instantiated-Plan is the input of a Surface Realizator that, using flexible templates, produces the image explanation as an HTML file. This process is mainly driven by the Rhetorical Relations (RR) between portions of the plan. The plan is explored in a depth-first way; for each node, a linguistic marker is placed between the text spans that derive from its children, according to the RR that links them. For instance, the following sentence: "Inside the parenchyma, tubular shaped, hyperdense and white images are visible (the superhepatic veins)." Is obtained from an template for the *ElabGenSpec* RR in which the satellite, corresponding to the application of the *Joint* template to the following attributes <position>, <shape>, <density> and <colour>, is followed by the nucleus stating the name of the object in focus that is put between brackets (the superhepatic veins, in this case). The decision of rendering this template in this way, is driven by common patterns we extracted from a corpus of explanation written by expert radiologists and, from the same corpus, we extracted also the generation rules for the templates corresponding to other RRs.

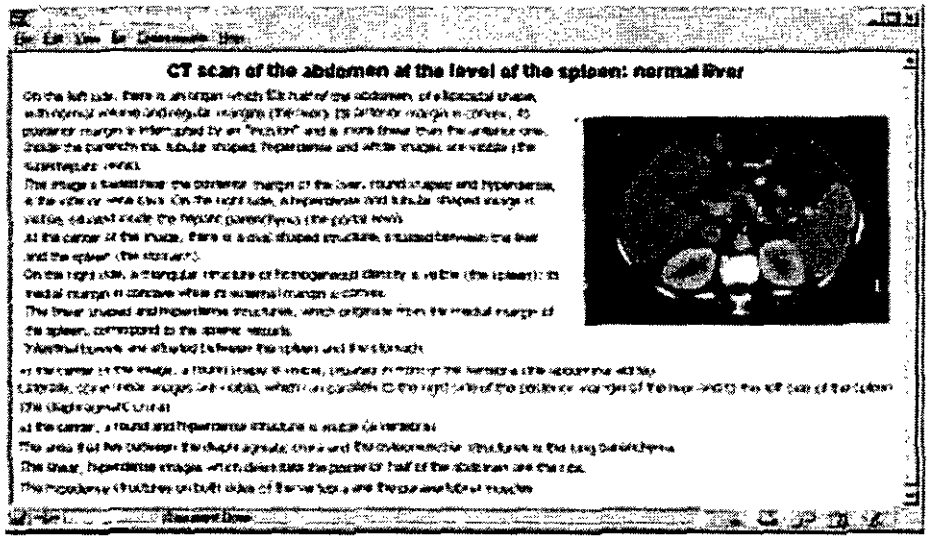


Fig 6. An example of image description.

At present, we generate the text in HTML; however, our approach is general enough to produce descriptions in different formats and, therefore, for different interaction contexts. It is also domain independent, since it is only driven by the Rhetorical structure of the discourse plan. We choose to develop our surface generator instead of using existing standard techniques, such as XSLT stylesheet templates, because these approaches did not allow us to produce complex textual description matching the style of the corpus that we analysed. This limit it is also underlined in

Cawsey and colleague papers [1,2], that used XML coupled with XSLT stylesheets for generating user tailored tabular presentations, from selected metadata, of online resources. In addition, in this system, as we will seen later on, we generates also description by comparison with other images, and this requires a more complex reasoning that is difficult to reduce to XSLT application to an XML file.Fig. 6 shows an example of the description that was generated from the discourse plan in Fig. 4.

6. Comparing Images

Let's now see how we generate the description of a image by comparing it with a reference image. The general strategy we apply is similar to the one we applied to compare concepts in ARIANNA. For every detail in a overlay, we mention first commonalities, second alignable differences and finally non-alignable differences. In the case of image descriptions, we distinguish, at the moment, three types of comparisons, that depend on what the User already knows and on the images she has already seen. Then, given a Image I to be described to a User U and a Reference-Image RI, three different comparison plans may be activated:

Comparison 1. KnowAbout(U, RI) AND Remember(U, RI) ⇒ Exec(S, cplan_1);
If the user, according to its background knowledge, profession and level of expertise or according to what she has already seen, knows RI and is presumed to remember its description, the first comparison plan (cplan_1) is applied. This plan corresponds to the following strategy: for each overlay and for each detail, only the attribute values of I that are different from the ones in RI are mentioned (alignable differences). After them, the values of the attributes that are not present in RI are presented (non-alignable differences). This plan is applied, for instance, to describe pathological cases to radiologists.

Comparison 2. KnowAbout(U,RI) AND ¬Remember(U,RI) ⇒ Exec(S, cplan_2);
If the user knows RI but does not remember it in all its details, the second comparison plan (cplan_2) is applied. This plan corresponds to the following strategy: for each overlay and for each detail, the attributes of I that take different values from those of RI are mentioned, by describing both values (for I and for RI). After them, also in this case, non-alignable differences are presented. This plan is applied, for instance, to general proactitioners.

Comparison 3. ¬KnowAbout(U,RI) ⇒ Exec(S, cplan_3);
If the user does not know RI, the third comparison plan (cplan_3) is applied. This plan corresponds to the following strategy: for each overlay and for each detail, all attributes in the two images are described, by emphasizing commonalities, alignable and not-alignable differences. This plan is applied, for instance, to students.

Let us see some examples of comparisons that were generated with our system: in all these examples, the reference image is a CT scan of the abdomen for a 'non-

pathological' case, while the image to be described is a case of hepatic cirrhosis, obtained with the same technique. The first text is generated by cplan_3: alignable differences are emphasized in italics, while there are no 'non alignable differences' between the two images; only the first part of the text is shown, for space reasons.

CT scan of the abdomen at the level of the spleen: hepatic cirrhosis.

As in the non-pathological case, the liver is the organ situated on the left side of the image, which fills half of the abdomen, of ellipsoidal shape. *In this case, however, its volume is reduced, its density is inhomogeneous and its margins, instead of being regular, are lobulated.* Like in the normal case, the anterior margin is convex while the posterior one is more linear and is interrupted by an incision. Superhepatic veins are visible inside the parenchyma; they are tubular shaped, hyperdense and white, like in the normal case.
As in the normal vein, the inferior vena cava is situated near the posterior margin of the liver, round shaped and hyperdense. The portal vein lies inside the hepatic parenchyma: it is hyperdense and tubular shaped like in the normal case, *but is enlarged.*
As in the normal case, the stomach is visible at the center of the image, between the–liver and the spleen, and is oval-shaped.
The spleen is visible on the right side; it is triangular and has a homogeneous density, like in the normal case, *but it is enlarged.* Also in this case, its medial margin is concave while the external one is convex.
The splenic vessels, which originate from the medial margin of the spleen, are linear and hyperdense, like in the normal case.
Also in this image, between the spleen and the stomach, intestinal bowels are visible.
.....

If c_plan2 is applied to the same case, the following text is obtained:

CT scan of the abdomen at the level of the spleen: hepatic cirrhosis.

If compared with a non-pathological case, the volume of the liver in this image is reduced, its density is inhomogeneous and its margins, instead of being regular, are lobulated. The portal vein is enlarged and the spleen is enlarged too.
.....

---

7 Conclusions and Future Work

In this paper, we presented the first prototype of Image Descriptor, a software to generate image descriptions from annotated knowledge sources: this prototype was built in Java using the IBM-XML4J parser and will be integrated in a system (ARIANNA) that dynamically generates hypermedia presentations of clinical guidelines; ARIANNA is already in use and an experimental evaluation study has been performed, to check how physicians react to it. The methods and the techniques we employed for generating image descriptions aim at favouring sharing and re-use of information. In particular, annotating images has several advantages: first of all, it enables retrieving images from Web databases according to ad hoc criteria; in addition, once a image has been retrieved, it may be described in a natural language text

whose content, structure, and style may be adapted to the context in which retrieval was made.

The annotation of linguistic resources favours, in general, their re-use and distribution: their semantics can be interpreted and rendered in different ways according to the interaction context; for instance, plain text, HTML or WML. Our research efforts go in this direction: we plan to introduce, in ARIANNA, a Conversational Agent with the role of an "Explainer" that supports the User at different levels; we already developed a similar Agent in another context, the generation of 'Animated User Manuals' for software applications [4]. In passing from hypertexts to Animated Agents, most of the techniques described in this paper will not change: for instance, the DTD for representing discourse plans is the same, and therefore also the planning component remains invaried; we only add a 'Sentence Planner' that revises the XML-plan files and substitute the surface text generator with a module that generates what we call the "Agent's behaviours".

We claim that, to enable sharing of resources and methods among various research centers and to produce outputs in context and application-dependent forms, establishing standards in the NLG field is a promising approach. This may foster reuse of methods in different applications and settings: let's think about new UMTS phones or wearable computers, whose particular graphical interface will require revising the generation methods that many of us developed so far. The work described in this paper is a step in this direction.

Acknowledgments

This work was founded by the CNR grant 21.15.01 on the topic: "Digital Processing of Radiological Images" and by the NationalCo-founded Project on "Intelligent Agents: Knowledge Acquisition and Interaction".

References

1. Cawsey, A. Presenting tailored resource descriptions: Will XSLT do the job? In Proceedings of the 9th International WWW Conference, 2000.
2. Cawsey A., Bental D., Bruce E. and McAndrew, P.: Generating resource descriptions from metadata to support relevance assessments in retrieval. Proceedings of RIAO 2000.
3. De Carolis, B., de Rosis, F., Andreoli, C., Cavallo, V. and De Cicco, M.L.: The dynamic Generation of Hypertext Presentations of Medical Guidelines. The New Review of Hypermedia and Multimedia, 67-88 (1998).
4. De Carolis, B., de Rosis, F., Pizzutilo, S.: Generating User-Adapted Hypermedia from Discourse Plans. Fifth Congress of the Italian Association of Artificial Intelligence (AI*IA 97), Roma , (1997).
5. B. De Carolis, C. Pelachaud, I. Poggi, Verbal and non verbal discourse planning. Workshop on Achieving Human-like Behaviors. Autonomous Agents 2000. ACM Press.
6. de Rosis, F., De Carolis, B., Pizzutilo, S.: Automated Generation of Agent's Behavior from Formal Models of Interaction. To appear in proceedings of AVI 2000, Palermo, Italy (2000).

7. Hammond, N. and Allinson, L.: Extending Hypertext for Learning: an Investigation of Access and Guidance Tools. People and Computers V, HCI 89, Cambridge University Press (1989).
8. Inote: Image Annotation Tool. http://jefferson.village.edu/iath/inote.html.
9. Markman., A.B. and Gentner., D.: Commmonalities and Differences in Similarity Comparisons. Memory and Cognition, 24, 2 (1996).
10. Moore, J., D. Participating in Explanatory Dialogues. Interpreting and Responding to Question in Context. ACL-MIT Press series in NLP, (1995).
11. W3C: eXtensible Markup Language (XML). http://www.w3.org/xml/
12. Marcu, D.: Extending a Formal and Computational Model of Rhetorical Structure Theory with Intentional Structures à la Grosz and Sidner. The 18th International Conference on Computational Linguistics COLING'2000, Luxembourg, July 31-August 4, 2000.

INSPIRE: An INtelligent System for Personalized Instruction in a Remote Environment

Maria Grigoriadou, Kyparisia Papanikolaou, Harry Kornilakis, George Magoulas

Department of Informatics, University of Athens, Panepistimiopolis,
GR-15784 Athens, Greece
{gregor, spap, harryk, magoulas }@di.uoa.gr

Abstract. In this paper we present the architecture of an Adaptive Educational Hypermedia System, named INSPIRE. This particular system, throughout its interaction with the learner, dynamically generates lessons that gradually lead to the accomplishment of the learning goals selected by the learner. The generated lessons are adapted to the learner's knowledge level, learning style and follow his/her progress. The adaptive behavior of the system, the functionality of its various modules and the opportunities offered for learner's intervention are presented.

1. Introduction

Adaptive Educational Hypermedia Systems (AEHS) [2][3] extending the benefits derived from the instructional use of the Web, incorporate the idea of offering learners personalised support and/or instruction in a distance learning setting. The adaptive characteristics of an Educational Hypermedia System usually aim to both *usability* and *learning*. Thus, the educational implications are very important and should be considered through the design and development stages of the system. Although many questions are still open in the area of Instructional Design about *instruction / learning* and how it is efficiently provided / attained [17], it is important to consider adaptation within the framework of current learning theories and models, and thoroughly plan the sharing of the task of adaptation between the learner and the system.

We have developed an AEHS, named INSPIRE. Based on the learning goal that the learner selects, INSPIRE generates lessons that correspond to specific learning outcomes accommodating learner's knowledge level and learning style. Thus, aiming to individualize instruction, the system generates lesson plans tailored to the needs, preferences and knowledge level of each individual learner by making use of information about the learner gathered through their interaction. Furthermore, aiming to engage learners in the learning process, the system provides them with the option to intervene, expressing their perspective about their own characteristics or about the lesson contents and accordingly formulate their interaction with the system.

2 INSPIRE's adaptive functionality

The proposed system aims to facilitate distance learners during their study, adopting a pedagogical framework inspired by theories of the area of Instructional Design and Adult Learning. In the beginning of the interaction, the domain knowledge presented to the learner is restricted and gradually it is enriched, following the internal structure of the domain (*curriculum sequencing technique*), while a navigation route is proposed based on his/her progress (*adaptive navigation technique*).

The main instructional outcomes of the generated lessons on learners' level of performance are to understand and to remember the most important instances and generalities associated with the learning goal they study (*Remember*), to be able to apply them to specific cases (*Use*) and finally to be able to generate new generalities (*Find*) [12]. The presentation of the educational material provided for each different

level of performance, i.e. Remember, Use and Find, is mainly determined by the learning style of the learner (*adaptive presentation technique*). Thus, learners' preferences, that usually guide systems' adaptation [2], are determined based on their learning style. Following the theory of learning styles [9][4][11], how much individuals learn, i.e. the effectiveness of instructional manipulations, is mainly influenced by the educational experiences geared toward their particular style of learning. This approach to learning emphasizes the fact that individuals perceive and process information in very different ways. In this paper we propose a framework for the system's adaptive behavior exploiting the learning style information. The learning style model that we adopted in the current implementation of the system is that of [8] where Honey and Mumford, based on Kolb's theory of experiential learning [9], suggested four types of learners: *Activists*, *Pragmatists*,*Re flectors*,*Theorists* .

The proposed system also supports end-learner modifiability offering opportunities to the learners to intervene in different stages of the lesson generation process, as well as on the construction of their learner model. Thus, learners have the option to activate or deactivate the lesson generation process of the system. In case they choose to activate it, they have the option to guide system's instructional decisions by updating accordingly their characteristics on their model, i.e. their knowledge level on the different concepts of the learning goal and their learning style. The externalization of the model to the learners is implemented in a manner that allows it to be understandable, transferable and usable [7].

3 The Architecture of INSPIRE

INSPIRE's architecture has been designed so as to facilitate knowledge communication between the learner and the system and to support its adaptive functionality. INSPIRE is comprised of five different modules (see Fig. 1): *(i)* the *Interaction Monitoring Module* that monitors and handles learner's responses during his/her interaction with the system,*(ii)* the *Learner's Diagnostic Module* that processes data recorded about the learner and decides on how to classify the learner's knowledge, *(iii)* the *Lesson Generation Module* that generates the lesson contents according to learner's knowledge goals, knowledge level,*(iv)* the *Presentation Module* whose function is to generate the educational material pages sent to the learner and (*v*) the *Data Storage*, which holds the *Domain knowledge* and the *Learner's Model.*

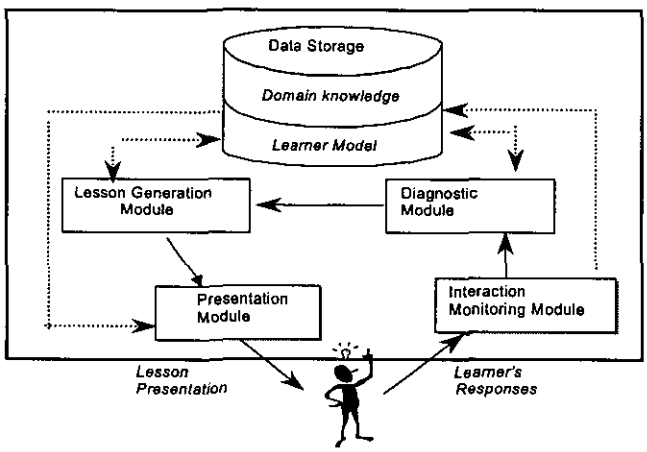


Fig. 1. INSPIRE's components and the interactions with the learner

3.1 Representing Knowledge about the Domain and the Learner: *Data Storage*

The domain knowledge of the system is structured in three hierarchical levels of knowledge abstraction: *learning goals*, *concepts* and *educational material* [14]. Every learning goal is associated with a subset of concepts. Assigning qualitative characterisations provides interrelation among the different concepts of a learning goal, i.e. *outcome concepts*, *prerequisite concepts* and *related concepts*. Note that the prerequisites and related concepts are linked to specific outcome concepts. The outcome concepts of a learning goal are further organized in a layered structure, i.e. the outcome concepts belonging to a specific layer are those that should be presented before the ones of the next layer.

The educational material related to each outcome concept consists of *knowledge modules,* developed according to three different levels of performance proposed in [12], *Remember*, *Use* and *Find*. Each different level of performance is associated with a different combination of multiple types of educational material aiming to increase learning efficiency as follows: (*i*) the *Remember* level of performance includes information necessary to present the concept, i.e. expository and inquisitory theory presentations and/or examples plus images and/or questions and self-estimation tests, assessment tests, (*ii*) the *Use* level of performance includes information necessary to apply the concept to specific cases, i.e. hints from the theory and/or examples and/or exercises and/or activities based on computer simulations, self-estimation tests, assessment tests, and (*iii*) the *Find* level of performance aims to the ability of the learner to find a new concept, principle, procedure, and thus the educational material provided includes activities on simulations, exploration activities, case studies. The representation of the knowledge modules in the database is based on the ARIADNE recommendation for educational metadata [1]. Metadata specify the attributes that fully and adequately describe the knowledge modules of the educational material.

The learner model

The learner model is the system's representation of the learner. It supports learner's communication with the system and reflects some of his/her features. It describes the learner (general information, learning style) and his/her "current state" (knowledge level on the different concepts and learning goals, performance on assessment tests, number, type and order of resources s/he has accessed etc.).

The knowledge level of the learner on a certain concept is assigned one of the characterizations {I, RS, AS, S} = {Insufficient, Rather Sufficient, Almost Sufficient, Sufficient}. This assignment is made based on learners' answers to assessment questions of different types. The diagnostic module uses the approach described in [15] for multicriterial decision-making in order to assess learner's knowledge level on each particular concept of a learning goal.

Currently, the learning style of the learner is initialised through the submission of the questionnaire developed by Honey & Mumford [8] or directly by the learner, who has the option to select his/her dominant learning style based on information provided by the system about the general characteristics of the different categories. In the first case, the learner, the first time s/he logs in the system submits the questionnaire and automatically according to the procedure defined in [8], his/her dominant learning style is determined and stored in his/her learner profile.

3.2 Monitoring the Learner: Interaction Monitoring Module

The function of the interaction-monitoring module is to log the requests made by the learner, as part of his/her HTTP request, and update the learner's model with the newly acquired information. Since the interaction-monitoring module is the only part of INSPIRE that receives direct input from the learner, it is responsible for collecting data concerning the learner's observable behavior and for notifying the other modules about any actions performed by him/her. Such actions are, the inspection or modification of his/her model, the selection of a learning goal and the activation/deactivation of the lesson generation process:

3.3 Planning the Lesson's Contents: *Lesson Generation Module*

The Lesson Generation Module realizes the lesson generation process, which plans the content and the delivery of each lesson. The outcome concepts of a learning goal are presented gradually according to the priority of the layer they belong to. The lesson generation process determines which layer of the outcome concepts (See in Sect. 3.1 the structure of the domain knowledge) should be proposed to the learner. This decision is mainly guided by his/her knowledge level on the outcome concepts of the previous layers.

Every outcome concept on the selected layer is accompanied by its prerequisites and related ones. In the proposed approach we use different strategies for planning the content of a lesson on each particular layer. This process, takes into account the relative importance of each concept on the learning goal as well as the knowledge level of the learner on those concepts. For example:

- If the knowledge level of the learner has been evaluated as {Insufficient} on a number of outcome concepts. Then, s/he has to study the educational material of the *Remember* level on these outcome concepts and their entire prerequisite ones.
- If the knowledge level of the learner has been evaluated as {Rather Sufficient} on a number of outcome concepts and {Sufficient} on several prerequisite concepts. Then, s/he has to study the educational material of the *Use* level on these outcome concepts and the rest of the prerequisite ones of the outcome.

The relative importance of the concepts included in a lesson determines the extent of their presentation. Thus, the generated lesson includes: *(i)* complete presentation of the outcome concepts (according to the three levels of performance), *(ii)* links to brief presentations of the prerequisite concepts focusing on their relation to the outcome and *(iii)* links to the definition of the related concepts in a glossary. The educational material associated to each of the concepts is predefined while its presentation to the learner is tailored to his/her learning style. Also, the results of the lesson generation process, on the contents and delivery of the generated lessons reflect on the navigational route proposed by the system (See Section 3.3. Adaptive Navigation).

3.4 Presenting the Lesson: *Presentation Module*

This module is responsible for the presentation of the lesson to the learner. The Lesson Generation Module has already determined the lesson contents based on the knowledge level of the learner, but the presentation module will decide on the appearance of the knowledge modules based on the learning style of the learner.

Adaptive Presentation

Learners with different learning style view different presentations of the educational material. The main objective is to support learners, following their preferred way of studying. To this end we exploit the information of their learning style in order to guide decisions on the instructional approach proposed to each individual learner.

According to the proposed framework, the multiple representations of the outcome concepts (expository and inquisitory presentations, examples, exercises, activities based on computer simulations, exploration of resources and group works) constitute different instructional primitives [10] which are combined to formulate alternative instructional strategies for the presentation of the educational material. The selection of the appropriate instructional strategy for each learning style category reflects some tendencies of the category in approaching information and is in accordance to related work proposed in the literature [6] [16]. Furthermore, empirical investigations on the learning preferences of learners have been realised during the first stages of the formative evaluation [13] of the system aiming to provide direct information from learners about their attitudes towards the instructional material while studying.

Thus, all learners are provided with the same knowledge modules. However, the method and order of the presentation of the different representations that they include, is adapted, implementing multiple instructional strategies that focus on different perspectives of the concept. This way, we attempt to maximise the benefit gained from style awareness [9][8]. Learners are motivated to pass through all the provided educational material exploiting their own capabilities and developing new ones. Consequently, for Reflectors who tend to collect and analyse data before taking action, *example-oriented* (see Fig.2), proposing him/her to start reading the example, continue with a brief theory presentation and then try to solve an exercise. Accordingly, for the presentation of material to Activists, who are more motivated by experimentation and attracted by challenge, the instructional strategy adopted is *activity-oriented* (see Fig.3), proposing him/her to start experiment with an activity designed for a computer simulation and providing him/her with the necessary information (examples and theory).

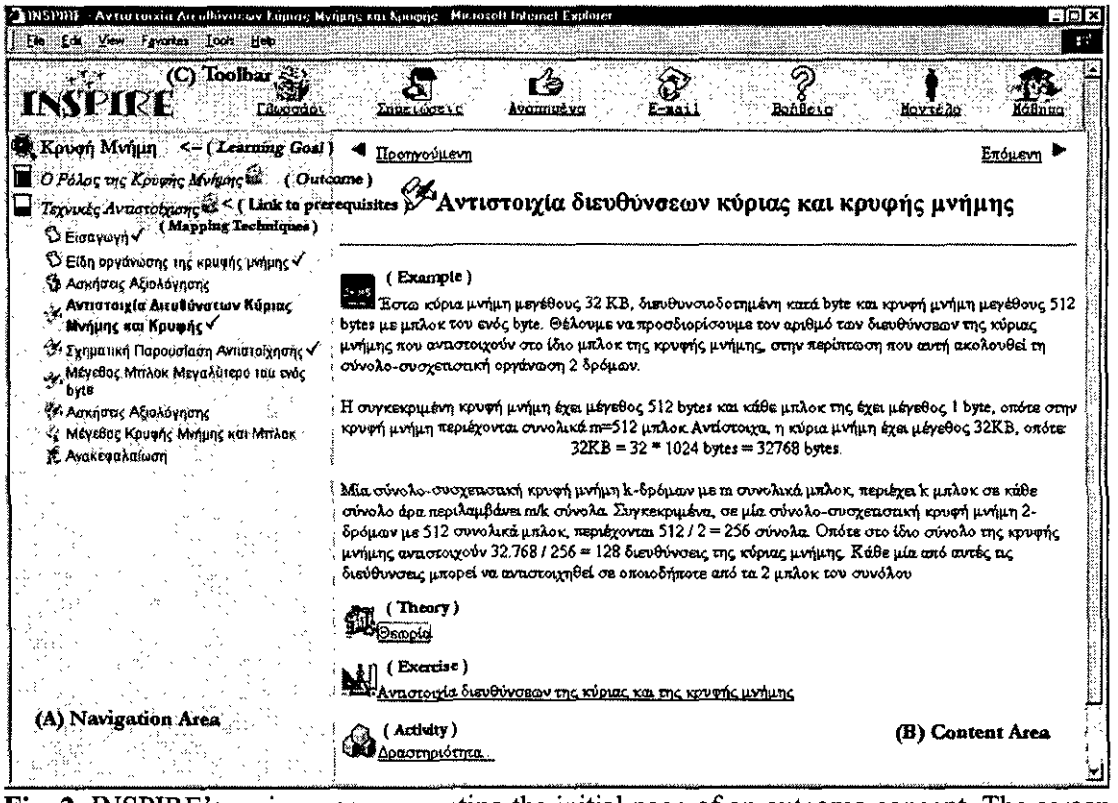


Fig. 2. INSPIRE's main screen presenting the initial page of an outcome concept. The screen is divided into three areas: (A) Navigation Area (B) Content Area, and (C) Toolbar. In the Content Area, different knowledge units comprising a page of educational material as viewed by Reflectors: (3) An application example (1) Link to hints from the theory (2) An exercise (3)Link to an activity on a computer simulation.

Concerning the implementation of the adopted instructional strategy, if it is example-oriented then the knowledge module "example" will be embedded at the beginning of the page while the rest of the modules will appear next as links; if the instructional strategy is activity-oriented then the knowledge module "activity" will be embedded at the beginning of the page while the rest of the modules will appear next as links;

In INSPIRE, the use of multiple representations in different instructional strategies formulating their presentation alleviates the problem of rewriting the same content tailored to each learning style category. The different knowledge modules are presented as different areas in the educational material pages. These areas are associated with a condition referring to the learning style of the learner and they are, either embedded in the page, or appear as links, or they do not appear at all. This way, the same knowledge modules can provide multiple alternative representations of the same concept through the adaptive presentation technique.

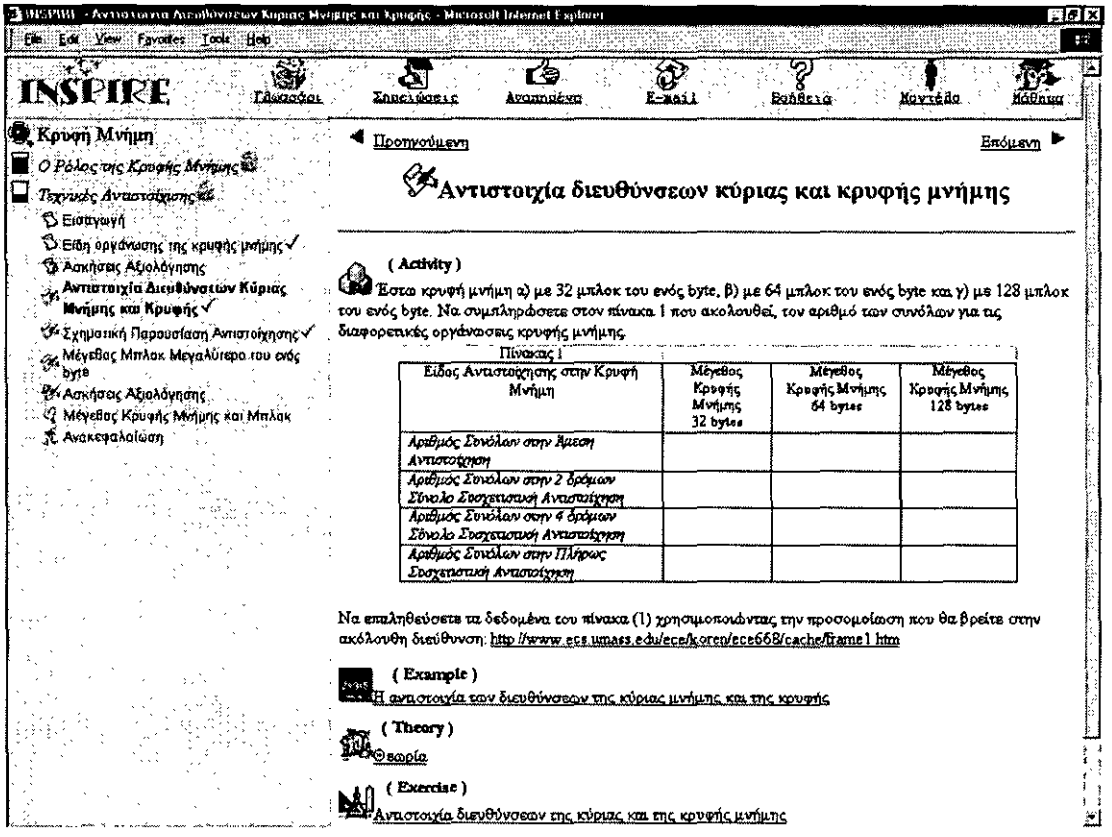


Fig. 3. Different knowledge modules comprising a page of educational material as viewed by Activists: (1) An activity on a computer simulation (2) Links to application examples (3) Link to hints from the theory (4) Link to an exercise.

Adaptive Navigation Support

The system supports learner's navigation and orientation in the lesson contents, by annotating the links that appear in the Navigation Area. Additional information is provided to the learner through the use of icons next to the names of concepts and the educational material, that distinguish the outcome from the prerequisite concepts as well as the educational material provided for each level of performance (see Fig.4 - Navigation Area). Especially on the outcome concepts, the filling of a measuring cup is used as a metaphor denoting learner's progress.

Furthermore, two state icons accompany the prerequisite concepts and the educational material of the outcomes reflecting the instructional decisions of the lesson generation process on the educational material that the learner should study next. Thus, coloured icons accompany the links that lead to the material that the system proposes the learner to study next, while black and white icons appear next to the rest of the links (see Fig.4 - Navigation Area).

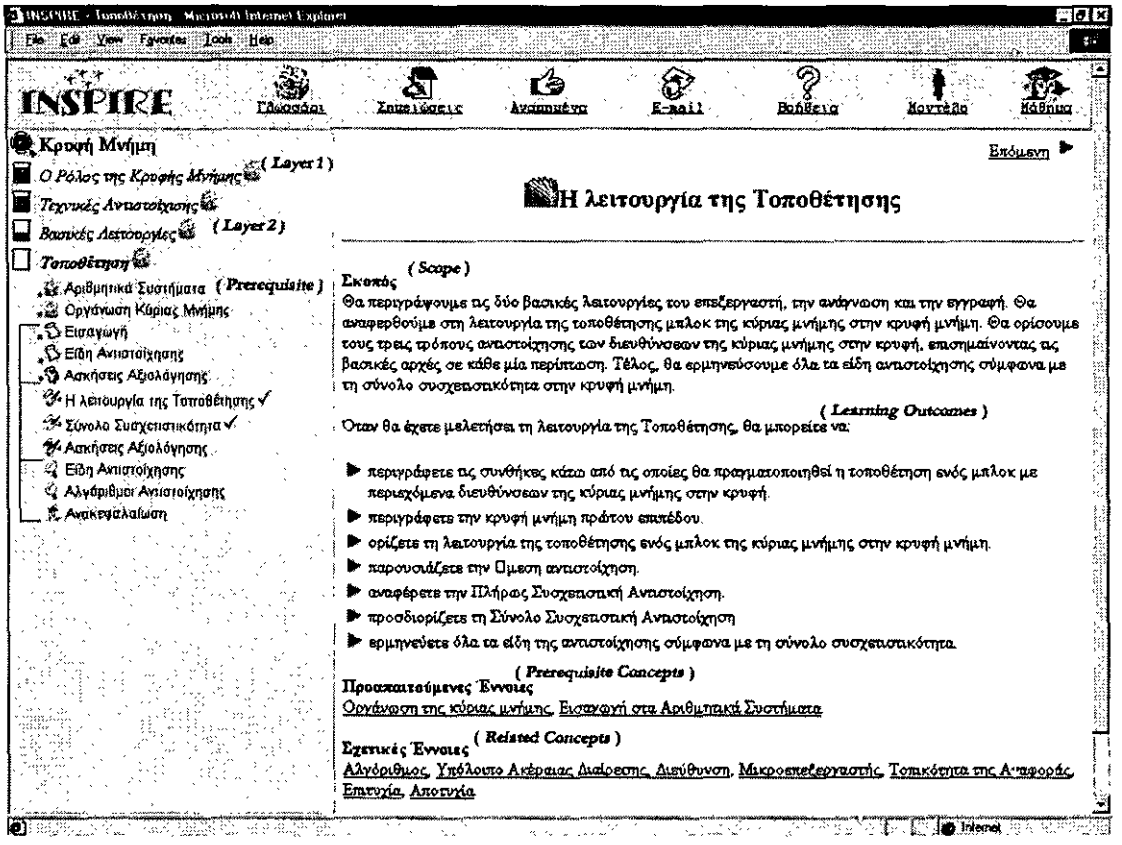


Fig. 4. INSPIRE's main screen presenting the initial page of an outcome concept.

4 Implementation Issues

INSPIRE is currently used to support an introductory course on Computer Architecture. In particular, educational material referring to the learning goal "Which is the role of cache memory and its basic operations" has been developed based on the chapter *Computer Memory* of the university module "Computer Architecture" [5] developed in the Computer Science department, University of Athens. The current implementation of the system is using an IIS web server running on Windows NT, which processes the requests made by the learners. The learner model and the educational metadata describing the educational material [1] are stored in a SQL Server database that communicates with the web server through use of the ActiveX Data Objects (ADO) technology. The education material itself is stored in the file system pages. We are making use of the Active Server Pages (ASP) technology developed by Microsoft, which allows the dynamic generation of HTML page, in order to implement the adaptive presentation technique.

5 Conclusions and Further Research

INSPIRE is an adaptive system that monitors learner's activity and dynamically adapts the generated lessons to accommodate diversity in learners' knowledge state and learning style. An experiment focusing on the evaluation of the instructional design of the system has been conducted with students of the department of Informatics of the University of Athens, attending the course on Computer Architecture and with participants of a seminar on the usability of educational software. The initial reactions towards the system have been encouraging while students' comments inspired several improvements on system's interface.

The system is both adaptive and adaptable, as it allows the learner to control the interaction and provides guidance or help. The learner model of the system provides a complete description of the current state of the learner and it is open to the learner to make changes and in this way allows him/her to intervene in the lesson generation process, supporting "end-learner modifiability". Further processing of the information stored in the learner model can be exploited by: (*i*) the system for the learner diagnosis process, (*ii*) the learner in order to be informed on system's decisions and intervene accordingly, as it will be described below, and (*iii*) the tutor for the evaluation of the provided material and for monitoring learners' progress and study attitude. From the various statistics stored in the learner model the tutor can have a quantitative estimation of the learners preferences on the educational material, in the sense of the time they spent on it, their performance, their requests to the system for help on specific pages etc. Furthermore, the tutor can examine the system's learner profile in order to get information about each learner's attitude while studying, and their progress.

The knowledge level and the learning style of the learner are used for the appropriate selection of the lesson contents and the presentation of the educational material. In the current implementation of the system, the learning style of each individual learner is recognized through the submission of the appropriate questionnaire or by the learner. Further research is on progress concerning the estimation of the way each learner uses the educational material in order to identify inconsistencies in the association of the learning style of the learner (already known) with the different types of educational material. For example, it is expected that the Activist will spent most of his/her time on activities and exercises, while the Reflector on theory presentations and examples. The way a learner uses the educational material in conjunction with his/her progress is valuable information denoting how successful is the selection of particular type of educational material for the particular learner. Furthermore, this information can also be used for the dynamic adaptation of the instructional strategy adopted for presentation of the educational material during learner's interaction with the system.

Acknowledgement

This work was partially supported by the Greek General Secretariat for Research and Technology of the Greek Ministry of Industry under a ΠΕΝΕΔ 1999 grant No 99ΕΔ234.

References

1.AR IADNE project. Available at http://ariadne.unil.ch

2.Brus ilovsky, P. (1996). Methods and Techniques of Adaptive Hypermedia. Learner Modeling and Learner-Adapted Interaction, Vol.6. Kluwer Academic Publ. Netherlands

3.Brus ilovsky, P. (1999). Adaptive and Intelligent Technologies for Web-based Education. In: C.Rollinger and C.Peylo (eds.) Kunstliche Intelligenz, Special Issue on Intelligent Systems and Teleteaching

4.Ent wistle, N.J. (1988). Styles of Learning and Teaching. London: David Fulton.

5.Grigori adou M., Papanikolaou K., Cotronis Y., Velentzas Ch., & Filokyprou G. (1999). Designing and Implementing a Web-based course. In Proc. of Int. Conf. of Computer Based Learning In Science, Enschede, Netherlands, H5.

6.Groat, A, & Musson, T. (1995). Learning Styles: individualising computer-based learning environments. *ALT – Journal, 3*(2), 53-62.

7.Hartl ey, R., Paiva, A., Self J.: Externalizing Learner Models. In: Greer, J. (ed.): Proc. of Int. Conf. on Artificial Intelligence in Education. Washington:AACE, (1995) 509-516

8.Honey, P. and Mumford, A. (1986, 1992). The manual of Learning Styles. Peter Honey Maidenhead
9.Kol b, D. A. (1984). Experiential learning. Englewood Cliffs, N.J.: Prentice-Hall.
10. Marcke, V. (1992). A Generic Task Model for Instruction. Instructional models for Computer-based Learning Environments. *Nato ASI Series F*, Vol. 104, Berlin: Springer-Verlag.
11. McLoughlin, C. (1999). The implications of the research literature on learning styles for the design of instructional material. Australian J. of Educational Technology 15:3
12. Merril, M.D. (1983). Component Display Theory. In C.M.Reigeluth (Ed.), Instructional design theories and models: An overview of their current status. Lawrence Elrbaum Association Hillsdale NJ
13. Nielsen, J. (1993). Usability Engineering. San Francisco: Academic Press.
14. Papanikolaou, K.A., Magoulas, G.D., Grigoriadou, M.: A Connectionist Approach for Supporting Personalized Learning in a Web-based Learning Environment. In: Brusilovsky, P., Stock, O., Strapparava, C. (eds.): Adaptive Hypermedia and Adaptive Web-based Systems. Lecture Notes in Computer Science, Vol. 1892. Springer-Verlag, Berlin (2000) 189-201
15. Papanikolaou, K. A., Magoulas, G. D., Grigoriadou, M. Computational intelligence in adaptive educational hypermedia. In Proc. of Int. Joint Conf. on Neural Nets, Italy, 2000
16. Stoyanov, S., Aroyo, L, & Kommers, P. (1999). Intelligent Agents Instuctional Desiogn Tools for a Hypermedia Design Course. In S.P.Lajoie and M.Vivet (eds.) Artificial Intelligence in Education. IOS Press
17. Vosniadou, S.: Towards a revised cognitive psychology for new advances in learning and instruction. Learning and Instruction 6(2) (1996) 95-109

Developing Adaptive Internet Based Courses with the Authoring System NetCoach

Gerhard Weber, Hans-Christian Kuhl, and Stephan Weibelzahl

Pedagogical University Freiburg, Germany
[weber,kuhl,weibelza]@ph-freiburg.de

Abstract. Developing adaptive internet based learning courses usually requires a lot of programming efforts to provide session management, keeping track of the learners current state, and adapting the interface layout to specific requirements. NetCoach is designed to enable authors to develop adaptivelearnin g courses without programming knowledge. In this paper, we describe the adaptive, the adaptable, the interactive, and the communicative features of NetCoach. Both authors and tutors are supported in many ways to develop and manage courses via an online interface. Experiences with NetCoach courses in different domains and settings have shown that learners profit from the adaptive features.

1 Introduction

Internet based instruction has grown strongly during the last years. While in the beginning most internet based courses consisted only of a collection of static HTML-pages (mostly simple translations of already existing scripts and papers), a lot of sophisticated internet based learning systems emerged in recent time. The former systems could be easily created by authors using simple authoring tools, but these systems were not much more than copies of textbooks and lacked any adaptivity and guidance that would be needed to support learners when learning a new topic on their own. On the other hand, most current more sophisticated learning systems are proprietary solutions and can only be built by experienced programmers and skilled web-based instruction authors.

This puts high demands on authoring tools to create adaptive internet based instruction courses. In this paper, we will introduce NetCoach, an authoring system that meets the needs to create adaptive learning courses in the internet. Creating adaptive courses with NetCoach is very easy and can be done without being a skilled programmer. NetCoach is derived from ELM-ART[1], one of the first and by now most comprehensive adaptive web-based educational systems (Weber & Specht, 1997).

2 Features and Adaptivity in NetCoach-Courses

NetCoach is an authoring-system which allows to create adaptive and individual course modules without programming-knowledge. This section describes four characteristics that are common to all courses that have been developed with NetCoach. The Courses are adaptive, interactive, adaptable, and communicative.

[1] cogpsy.uni-trier.de/projects/ELM/elmart.html

2.1 Adaptive Elements in NetCoach Courses

According to Brusilovsky (1996), adaptive learning systems may adapt to the learners experience, knowledge, goals, or preferences. NetCoach adapts to the last three aspects of the user. This information can be used either to adapt the presentation of the content or to support the navigation (Brusilovsky, 1996). NetCoach implements two adaptive navigation techniques: *curriculum sequencing* and *adaptive annotation of links*. The goal of *curriculum sequencing* is to provide the student with the most suitable, individually planned sequence of knowledge units to learn and the sequence of learning tasks (examples, questions, problems, etc.) to work with. In other words, it helps the student to find an "optimal path" through the learning material. The goal of *adaptive annotation of links* is to support the student in hyperspace orientation and navigation by changing the appearance of visible links.

These adaptation techniques require a content specific knowledge-base and a user model that allows the system for responding individually to learners' interactions with the system.

The Knowledge-Base as Basis for Adaptive Behavior In NetCoach, the knowledge base of a course consists of concepts. These concepts are internal representations of pages that will be presented to the learner.In many domains the different concepts are related in many ways. To build up this knowledge-base, which is the basis for adaptive navigation support, the author can create many content-specific relations for every concept. However, the author is not forced to specify any relation. Default values that retain the sequential order of the concepts will be applied otherwise. Note, that specifying the concept relations is a simple procedure (as will be shown in Section 3) and is basically a content-specific task.

Normally the contents of a domain are related and interdependent. There are two relations between concepts: prerequisites and inferences.

First, the author can decide which other concepts are required to be learned to understand the current concept. These prerequisites can be chosen in the concept-list as shown in Figure 1 (*a*). The system will guide learners to theses prerequisite pages before suggesting the current concept. Because prerequisite concepts might have prerequisite concepts themselves there are also indirect prerequisites (*b*). In our example-course the system will recommend the following sequential order of concepts in case *Chapter-2-1-2* is the current learning goal: *Chapter-1* (indirect prerequisite), *Chapter 1-2*, *Chapter-2* (prerequisites), and finally *Chapter-2-1-2*.

Second, the inferences (*c*) of a concept are in some way the opposite of prerequisites. Perhaps an user wants to learn *Chapter-3-1* first and solves the test items correctly. Because *Chapter-2-1-2* is marked as inferred by *Chapter-3-1*, the system will assume that the user already knows *Chapter-2-1-2* as soon as *Chapter-3-1* has been worked at successfully. Note that prerequisites and inferences are related but not equal. E.g., knowing A might be required to understand B, but if one knows B this does not necessarily imply that A is known.

In addition to these relations between concepts the knowledge base contains relations between test items and concepts. Sets of test items (so called test groups) assess the user's current learning state of a concept (*d*). However, test items may not only test

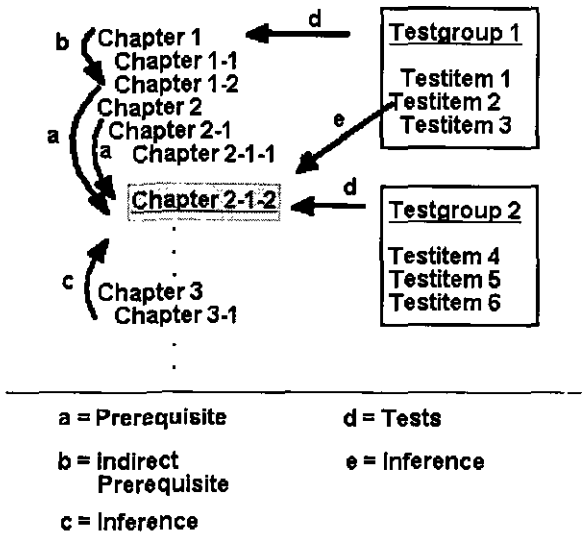


Fig. 1. Example of the relations of concept *Chapter-2-1-2.*

one concept but also assess aspects of other concepts. Thus, it is possible to quantify the inference of test items to other concepts. If the learner solves *testitem 2* in Figure 1 correctly (*e*), she has understood some important aspects of *Chapter-2-1-2*. A concept is supposed to be learned if one has reached a critical value. If there are already some inferences from test items of other concepts, the learner is closer to this critical value and has to solve less test items in *Testgroup 2* correctly.

The User Model. Based on the descriptions in the concepts, all pages are computed individually with respect to the learner's user model. The user model used in NetCoach is a multi-layered overlay model (Weber, 1999). Individual information about each learner is stored with respect to the concepts of the course's knowledge base (as described in the previous section). The first layer describes whether the user has already visited a page corresponding to a concept. The second layer contains information on which exercises or test items related to this particular concept the user has worked at and whether he or she has successfully worked on the test items up to a certain criterion. The third layer describes whether a concept could be inferred as known via inference links from more advanced concepts the user has already worked on successfully. Finally, the fourth layer describes whether a user has marked a concept as already known. That is, the user model can be inspected and edited (Bull & Pain, 1995). Sometimes, this is called a cooperative user model (Kay, 1995). Information in the different layers is updated independently. This leads to the fact that information from each different source does not overwrite others. E.g., if a student unmarks a concept because she realized that she has not enough pre-knowledge about it, the information about tests on this concept is still available.

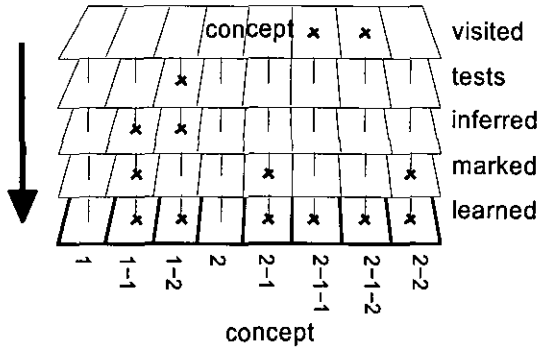


Fig. 2. Example of a student's overlay model. NetCoach infers the student's current learning state from four independently updated layers. Concepts without tests are treated as learned if they have been visited.

See Figure 2 for an example of a student's overlay model. A concept is assumed to be learned if it is either tested to be known, inferred from other learned concepts, or marked by the user. In case no test group is available the concept is assumed to be learned if it has been visited. I.e., the visited layer and the test layer are applied alternatively.

Curriculum Sequencing and Link Annotation. The multi-layered overlay model supports both the adaptive annotation of links and individual curriculum sequencing. Links that are shown in an overview on each page or in the table of contents are visually annotated in correspondence to the user's current learning state. Individual curriculum sequencing means that the system's suggestion which page is best to be visited next is computed dynamically according to the general learning goal and the user's learning state of the concepts. Users get a warning if they visit a page with missing prerequisites. However, access to that page is not restricted and the warnings can be turned off. See Figure 3 for an example of a warning due to unfulfilled prerequisites, the corresponding page suggestion, and the link annotation in the overview frame on the lefthand side.

Learning Goals. In addition, NetCoach supports the specification of learning goals. A goal consists of a set of concepts that have to be successfully worked on by the learner. All (direct and indirect) prerequisites are computed automatically and corresponding pages are suggested. Thus, learning goals are especially useful for learners that do not want to complete the whole course. E.g., the goal "I want to get an introduction on this topic" might include the introductory chapters only, while the second goal "I am familiar with ..., but I want to know more about ..." would leave out the first chapter and suggest to go to the advanced sections directly.

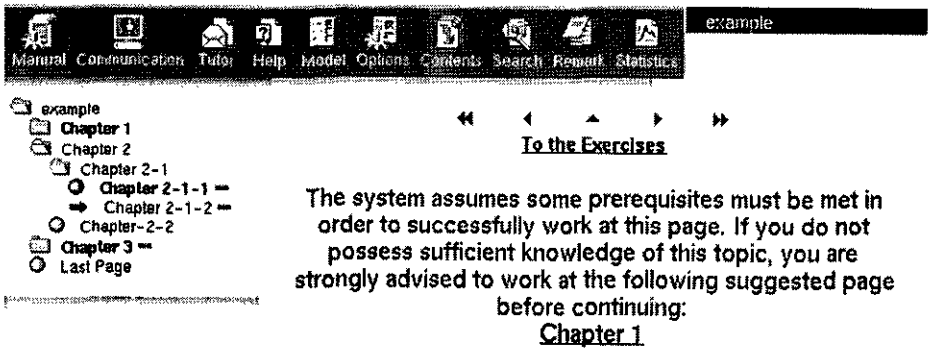


Fig. 3. Screenshot of the adaptive learning environment including curriculum sequencing and link annotation.

2.2 Interactive Elements in NetCoach Courses

Online-presented and evaluated exercises and tests are central features of interactive courses. NetCoach provides the possibility to present exercises and tests in different formats. These are multiple choice, forced choice, gap filling tests, open questions and e-mail-questions. While the e-mail-questions will be evaluated individually by human tutors, open questions have an example-answer as feedback, so that the learners can compare their solution by themselves. The feedback for the remaining item formats consists of a hint which answer is correct and an explanation why the answer was false or correct. Moreover, it is possible to give at first a hint only, before the correct answer is given.

NetCoach Courses can be additionally shaped highly interactive by connecting animations (e.g., flash-animations). These interactive animations can be contained like in every normal web-page. Animations can provide interactive work in simulated scenarios with multiple interactive mouse-events.

A glossary and a page with references can be accessed by the users with direct links in the text or a button. Finally, a search-tool and a notice-board are available.

2.3 Adaptable Elements in NetCoach Courses

Web-based courses are used by users with very different knowledge and different computer skills. Because of that it is useful if learners can adapt the learning environment to their own needs.

Not only the developed courses adapt to the user, but also the users themselves can adjust many features for their own preferences. Especially the kind of presentation, warnings and recommendations can be changed or switched off. The model behind the adaptive functions is not incomprehensible but can be investigated and changed by the learners themselves. The manner of annotation and the feedback can be adjusted, too.

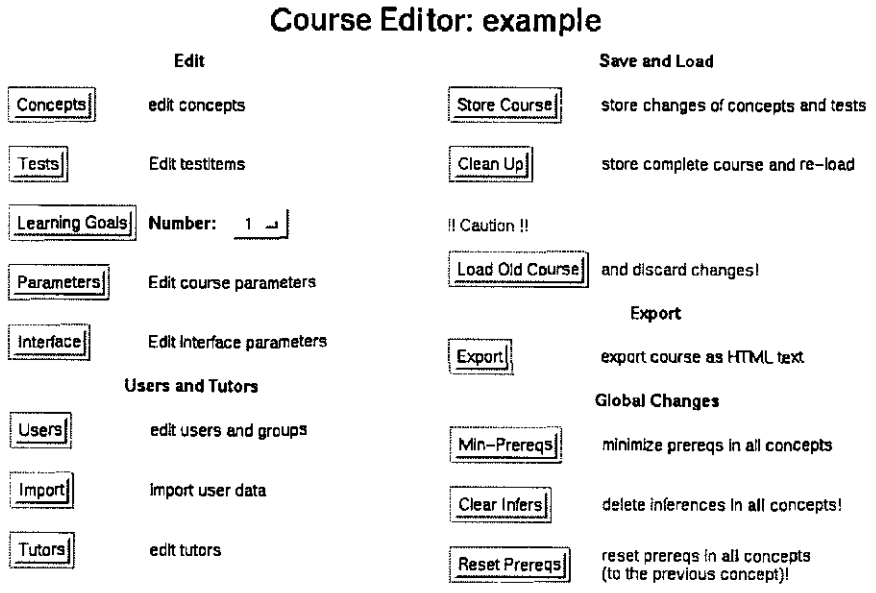


Fig. 4. Screenshot of the online-interface for authors

2.4 Communicative Elements in NetCoach Courses

Courses developed with NetCoach provide different syncronous and asyncronous communication tools. Questions, proposals etc. can be sent via e-mail to human tutors. A chat module provides direct communication between students. Besides, there is a possibility to discuss the contents in different discussion lists where the learners can exchange opinions and ask questions. It is also possible for every learner to exchange documents (e.g., word documents or pdf) with other learners.

All these communicative features enable lectures and teachers to organize complete virtual courses where students can interact with each other, but are still free to learn at their individual speed.

3 The NetCoach Authoring System

The NetCoach authoring-system bases on a LISP-server (CL-HTTP[2]) / web-browser-client technology. NetCoach[3] is available for Windows, Apple, and Linux operating-systems. Learners, tutors and even authors just need a standard-web-browser to work with the corresponding interfaces.

The goal of developing NetCoach was to provide authors with a tool to create highly adaptive courses without being required to program user models or interactive tests.

[2] www.ai.mit.edu/projects/iiip/doc/cl-http/home-page.html

[3] www.net-coach.de

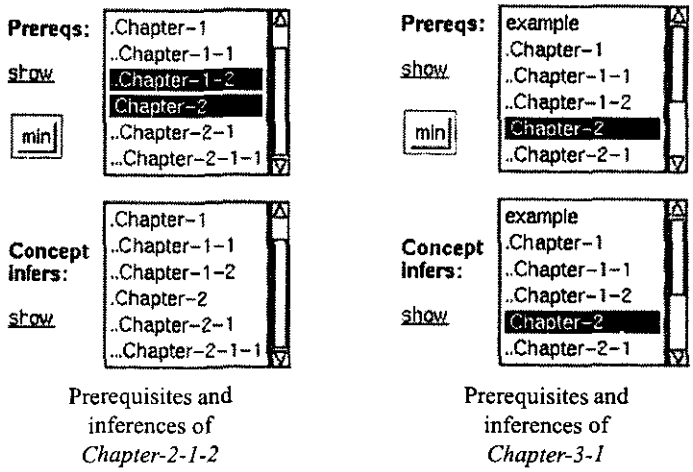


Fig. 5. Screenshot of a the author interface for specifying the prerequisites and the inferences of the concepts *Chapter-2-1-2* and *Chapter-3-1*.

The NetCoach authoring tool supports the complete developing process of adaptive web-based courses which includes authoring the learning material (e.g., texts, pictures), composing tests, defining learning goals, and adapting the layout and behavior of the interface. Figure 4 shows a screenshot of NetCoach's online-interface for authors.

In the concept editor, the concepts of the learning course (corresponding to pages presented in a browser) are described. Concepts may be arranged hierarchically similar to chapters and subchapters in a book. Authors may simply type in plain text, paste code from a HTML-editor, or even import an already existing HTML-file. In addition, animations that are created with Java, JavaScript or common plug-ins (e.g., flash animations) are supported as well.

The prerequisites on a concept and the inferences that can be drawn from successfully learning a concept are described by selecting the corresponding concepts from a table. See Figure 5 for a screenshot of how an author specifies the prerequisites and inferences of the concept *Chapter-2-1* which has been described in Section 2.1. To achieve perfect adaptivity effects authors have to define these concept relations very carefully in dependence of the domain structure.

In the test editor, test items can be defined and tested. The test editor offers templates for all test types, so authors are not required to program complicated cgi- or Java-scripts. In fact, NetCoach presents the test-questions, evaluates the answers, and observes the learning state of each user automatically, while the author can focus on the contents of the test items. Each item consists of three parts: First, the question that will be presented to the user.Secon d, the correct answers that have to be filled in by the user (gap filling, free input) or that have to be marked (forced choice, multiple choice). Third, authors can provide an explanation for the solution to help learners in understanding why they were wrong. Some of our courses include item pools of more than thousand items.

Test items are collected in test groups that are assigned to concepts. These test groups can be used as exercises or as introductory and final questionnaires. The learner's success on working on these test items is used to compute the user's learning state in the multi-layered learner model as described above.

Finally, a course might implement different learning goals. A learning goal consists of a set of concepts that has to be completed. Learners who decide not to work on the complete course but to fulfill a subgoal will receive individually recommendations which concept to visit next to complete this goal. The author has just to specify the concepts that are necessary to complete a goal. The prerequisites are computed automatically based on the hypertext model.

NetCoach not only is highly flexible in presenting different contents but also in adapting the course layout. Many optional parameters specify which buttons are presented how and where, which services are available (e.g., communication, search, or manual), and which components are adaptable by the user. This flexibility makes it easy to meet the requirements of different settings and even experimental studies. Moreover, NetCoach supports multiple languages, so courses can be developed in different target languages at the same time.

The NetCoach editors work in direct interaction with the NetCoach server so that it is possible to see effects of changing parameter settings, concepts, or test items directly in the course under development. This makes the creation of very sophisticated courses easy without requiring programming knowledge. A short tutorial on creating courses with NetCoach is described at http://art.ph-freiburg.de/NetCoach-Tutorial.

4 Tutors in NetCoach Courses

Courses created with NetCoach are guided by tutors that aid users on help requests, inspect user data, edit discussion lists, send messages to users, and manage user accounts and user groups. This is done via an online-interface which is shown in Figure 6.

Authors of a course can register tutors via the main course editor. Tutors have their own access rights and are able to inspect the course and users in the course.

First, tutors can get into contact with users. In a tutor help window, users can ask questions to tutors or give remarks on the course. The text directly typed into the tutor help window is sent by the server to the tutor (or tutors) by e-mail. Tutors can respond by e-mail in case the user has provided his or her e-mail address or send a message that is stored with the learner's user model and will be displayed to the user with the next page the user opens in the course.

Second, tutors can observe users in the course. They get a list of all currently active users and have access to all users in the course. Tutors can inspect the current learning state of a user. That is, they can see how long users have been working at the course, which messages they have sent to tutors, which concepts they have worked at, how many errors they have made, and some more interesting information that may help a tutor to understand the difficulties learners have with the course in order to help these learners.

Third, tutors can manage a discussion list. They can provide new topics in the discussion list, change trees of contributions to the discussion list to a new topic, and

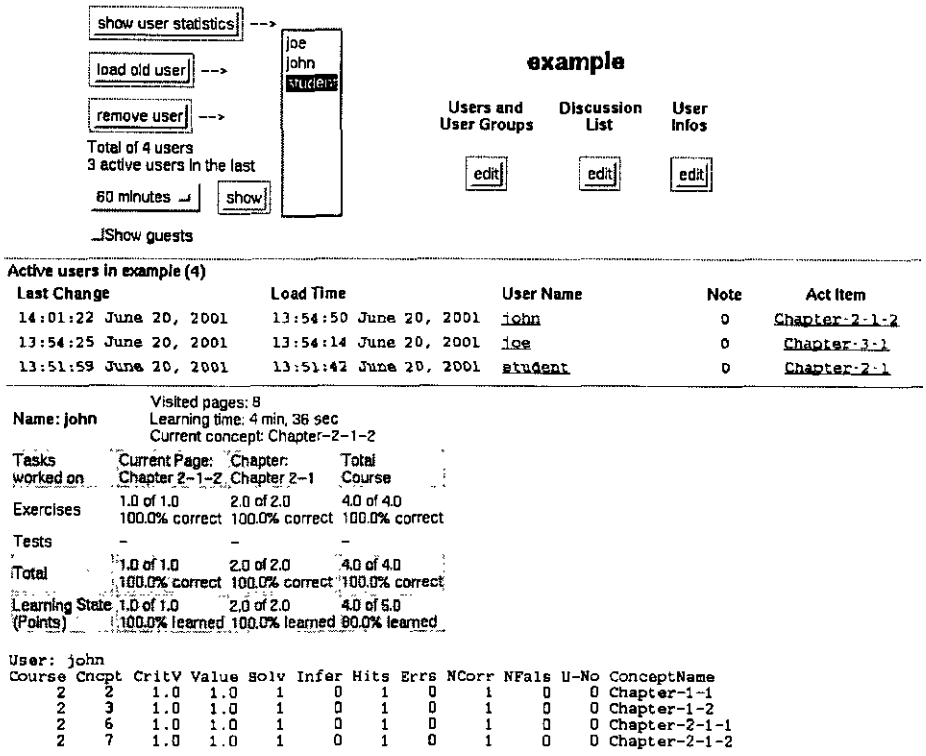


Fig. 6. Screenshot of the online-interface for tutors

remove parts of the discussion list. In the discussion lists, learners can place statements that can be read by all other learners from the same course or make remarks on those statements. Tutors can watch these discussion lists and remove contributions that are not related to the topics of the course.

Fourth, tutors can put information messages to all users of a course or to members of a user group. These messages have an expiration date so that messages that remind of an important date, for example, will be removed automatically after expiration. These information messages will be displayed to a user when he or she (re-)enters a course or even during working at the course.

Fifth, tutors have to manage user accounts and user groups. In case of closed courses (that is, the courses are restricted to users that have access rights), tutors can add new users or remove users, give access rights to users and assign users to working groups. Users assigned to working groups can exchange documents via the server. Additionally, users communicate directly with other members of the group via e-mail or chat.

5 Comparison with other Authoring Systems

5.1 General Features of Authoring Systems

Presently there are many authoring systems to create virtual learning environments. Most of them are non-adaptive or just adaptable like WebCT. In the following, NetCoach will be compared to the well distributed authoring systems WebCT (Goldberg, Salari, & Swoboda, 1996), Learning-Space[4], and TopClass[5]. These three authoring-systems where chosen because they are also based on a server/Web-Browser-Client technology and are already in use. In Section 5.2 we will compare NetCoach to other authoring systems that are designed to deliver adaptive hypermedia courses.

WebCT can be used flexibly to create entire online courses, or to publish materials that supplement existing courses. All interaction with WebCT takes place through a web browser. Essentially a WebCT course consists of a series of linked HTML pages that define a path or "road-map" through the course material. The course content is supplemented by WebCT tools which can be built into the course design by simply dragging the appropriate tool icon onto the web page.

TopClass courses are constructed of Units of Learning Material (ULMs). These ULMs can consist of pages, exercises, or further ULMs themselves. ULMs can be freely exported and imported from course to course. In addition to course management, TopClass also manages student progress, user-tracking, and access to course materials.

Learning Space is based on Lotus Notes and uses Notes Server technology to provide a secure environment with a rich set of tools. Learning space includes tools for browsing the web and inserting multimedia material into learning space documents. Links can be defined from Learning Space to multimedia content on the web. Additionally resources and other content may be exchanged via the Media Center. Completed courses may be archived by the instructor for future use. A Portfolio is contained in every participant's Profile. This is a secure area for returned assignments and assessments which can only be viewed by the participant and the tutor.

As shown in Table 1 there is much conformity comparing NetCoach and the other authoring-systems described above regarding the functionality for authors, tutors and learners. Authors can import contents or contain multimedia elements. Tutors can investigate, add or delete user data and provide discussion lists. Learners can use different asynchronous and synchronous communication tools like e-mail, discussion-lists, file-exchange, and chat. Whitebord, video conferences and homepage-authoring are not implemented in all systems and as well not in NetCoach. A web browsing tool only exists in Learning Space.

The main differences are the adaptive possibilities in course-development with NetCoach. These features like curriculum-sequencing and dynamic link-annotation are described in section 2.1. The course-management (e.g. registration, examinations, calendar) which is more central in the other systems is less important in NetCoach. NetCoach is mainly a system to develop entire, adaptive courses. For this reason, NetCoach will be compared with other intelligent systems in the following section.

[4] www.lotus.com

[5] www.wbtsystems.com

Table 1. Comparison of authoring systems for web-based training

	NetCoach	WebCT	TopClass	Learning Space
Author Tools				
adaptive guiding	yes			
adaptive link annotation	yes			
creating / importing content	yes	yes	yes	yes
add / play multimedia content	yes	yes	yes	yes
Tutor Tools				
store & view learner data	yes	yes	yes	yes
add / remove learners	yes	yes	yes	yes
performing assessments	yes	yes	yes	yes
create discussion groups	yes	yes	yes	yes
Student Tools				
adaptable preferences	yes	by tutor	yes	Yes
web browsing				Yes
creating / importing content		yes		Yes
store bookmarks		yes	yes	Yes
play multimedia	yes	yes	yes	Yes
homepage authoring		yes	yes	Yes
calendar tool		yes		Yes
searchable resource archive		yes		Yes
Communication				
e-mail	yes	yes	yes	Yes
noticeboard	yes	yes	yes	Yes
file exchange	yes	yes	yes	Yes
asynchronous discussions	yes	yes	yes	Yes
chat	yes	yes	Add. module	Yes
whiteboard		yes		Yes
video conferencing			Add. module	Yes
Technology				
server/web-browser-client	yes	yes	yes	Yes

Table 2. Comparison of authoring systems for adaptive hypermedia courses in reference to the user's features that the systems adapt to and the methods that are used for adaptation.

	Net-Coach	AHA	ECSAI-Web	Inter-book	Meta-Links
user features: to what?					
goals	yes		yes		yes
navigation history	yes	yes	yes	yes	yes
tested knowledge	yes		yes		
preferences	yes		yes		yes
methods: how?					
adaptive guidance	yes	yes	yes	yes	yes
adaptive annotation	yes	yes	yes	yes	yes
adaptive hiding of links		yes			
adaptive navigation maps					yes
adaptive text presentation		yes			yes

5.2 Adaptivity Features of Authoring Systems

Several other authoring systems aim at delivering adaptive web-based courses. We selected four of them to highlight the strengths and weaknesses of adaptivity in NetCoach: AHA (De Bra & Calvi, 1998), ECSAIWeb (Sanrach & Grandbastien, 2000), Interbook (Brusilovsky, Eklund, & Schwarz, 1998), and MetaLinks (Murray, Shen, Piemonte, Condit, & Thibedeau, 2000).

See Table 2 for a comparison of the user's features that the systems adapt to and the methods that are used for adaptation. The comparison is based on the categorization of adaptive hypermedia systems introduced by Brusilovsky (1996).

NetCoach implements most commonly used adaptive features, but does not adapt the text presentation (as e.g., AHA) and refrains from hiding links. We argue that the student should have full freedom of navigation and content access while the adaptive system should provide hints and suggestions only. However, NetCoach's adaptations are based on a diversity of user data. Especially knowledge assessment with tests is only found in ECSAIWeb and NetCoach. The overlay model in ECSAIWeb is slightly less sophisticated as it does currently not consider the inference relation (section 2.1) for adaptation purposes.

6 Conclusion

Several courses have been developed with NetCoach. They are used at different universities in Germany and in some companies. Up to now, most courses are written in German, though some are written in English (ELM-ART) or in French. Because NetCoach does not require any programming knowledge, many different authors from many disciplines developed courses in different domains including programming, spelling rules, cognitive and pedagogical psychology, and product presentation. At the Pedagogical

University in Freiburg, students develop simple courses on their own and test these courses with pupils in secondary schools. NetCoach has been used for "learning on demand" settings, as well as for supplementing courses at universities and adult education. E.g., several courses on pedagogical psychology are used by students to prepare lessons and exams, while two courses on programming LISP and HTML are available world-wide for training purposes. Accordingly, the courses differ widely in structure and features to suit the specific settings. Experiences with these courses show that users can learn easily and successfully. Results of several investigations support the usefulness of the adaptive features of NetCoach (Weber & Specht, 1997; Weibelzahl, 2001).

References

Brusilovsky, P. (1996). Methods and techniques for adaptive hypermedia. *User Modeling and User-Adapted Interaction, 6*(2-3), 87–129.

Brusilovsky, P., Eklund, J., & Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. In *Computer Networks and ISDN Systems. Proceedings of Seventh International World Wide Web Conference, 14-18 April 1998* (Vol. 30, pp. 291–300).

Bull, S., & Pain, H. (1995). "Did I say what Ith ink I said, and do you agree with me?" Inspecting and questioning the student model. In J. Greer (Ed.), *Artificial Intelligence in Education, Proceedings of AI-ED'95, 7th World Conference on Artificial Intelligence in Education, 16-19 August 1995. Washington, DC, AACE* (pp. 501–508).

De Bra, P., & Calvi, L. (1998). AHA! An open adaptive hypermedia architecture. *The New Review of Hypermedia and Multimedia, 4*,1 15–139.

Goldberg, M. W., Salari, S., & Swoboda, P. (1996). World Wide Web - course tool: An environment for building WWW-based courses. *Journal of Computer Networks and ISDN Systems, 28*(7-11), 1219–1231.

Kay, J. (1995). The UM toolkit for cooperative user models. *User Models and User Adapted Interaction, 4*(3), 149–196.

Murray, T., Shen, T., Piemonte, J., Condit, C., & Thibedeau, J. (2000). Adaptivity in the MetaLinks hyper-book authoring framework. In *Workshop Proceedings of Adaptive and Intelligent Web-Based Education Systems workshop at ITS 2000.*

Sanrach, C., & Grandbastien, M. (2000). ECSAIWeb: A web-based authoring system to create adaptive learning systems. In P. Brusilovsky, O. Stock, & C. Strapparava (Eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems. International Conference, AH 2000* (pp. 214–226). Berlin: Springer.

Weber, G. (1999). Adaptive learning systems in the World Wide Web. In J. Kay (Ed.), *User modeling: Proceedings of the Seventh International Conference, UM99* (pp. 371–378). Vienna: Springer.

Weber, G., & Specht, M. (1997). User modeling and adaptive navigation support in WWW-based tutoring systems. In A. Jameson & C. Tasso (Eds.), *User Modeling: Proceedings of the Sixth International Conference, UM97* (pp. 289–300). Vienna: Springer.

Weibelzahl, S. (2001). Evaluation of adaptive systems. In M. Bauer (Ed.), *User modeling: Proceedings of the Eighth International Conference, UM2001.*

Link Augmentation: A Context-Based Approach to Support Adaptive Hypermedia

Christopher Bailey[1], *Samhaa R. El-Beltagy*[2] *and Wendy Hall*[1]

IAM Research Group
Department of Electronics and Computer Science
University of Southampton, UK
[1]E-mail: {cpb99r,wh}@ecs.soton.ac.uk
[2]E-mail: samhaa@computer.org

ABSTRACT
In today's adaptive hypermedia systems, adaptivity is provided based on accumulative data gained from observing the user. User modelling, the capturing of information about the user such as their knowledge, tasks, attitudes, interests etc., is only a small part of the global context in which the user is working. At Southampton University we have formed a model of one particular aspect of context that can be applied in different ways to the problem of linking in context. We report how that context model has been used to provide link augmentation; an existing open hypermedia technique, which has a direct application in adaptive hypermedia systems. This paper presents a technique for cross-domain adaptive navigational support by combining link augmentation with a model of the user's spatial context.

KEYWORDS: (adaptive) link augmentation, linkbase, context, information finding, adaptive hypermedia

INTRODUCTION
One of the main goals of any adaptive hypermedia (AH) system is to increase user efficiency. This efficiency is usually measured either in the time spent searching for information or increasing the amount of information absorbed by the user. The work presented in this paper is built around the philosophy of providing the user with greater access to information through link augmentation – a technique whereby external links are inserted directly into the body of a document. There are already several systems that provide link augmentation such as Microcosm [9], Personal WebWatcher [18], and WBI [17], but they base their insertion algorithms on individual keywords or phrases in the document. However because the English language contains multiple uses for individual words, a simple augmentation algorithm like this can lead to out-of-place or irrelevant links and in addition to being frustrating, can also lower a user's confidence in the system.

Out-of-place links are added when the component that adds those links fails to recognize a document's context. This contextual information can be obtained by analysing the text in a document and comparing it against previously visited documents. This history information can then act as a filter to remove or ignore those links that fail to match the current context.

While user modelling involves capturing some contextual information such as a user's knowledge in a particular area, their tasks, goals and interests etc., this information is often obtained using explicit feedback which can distract the user away from their original task [15]. One advantage of the technique used in this paper is the lack of explicit user feedback required. All information about the user is obtained implicitly from the user's trail and the contents of each page the user views. This removes the need to question the user and although not exploited in this paper, this data can be employed in other user modelling environments to infer details such as user interests, hobbies, skills and tasks.

Another advantage of this system is that it works without making any pre-defined assumptions about its users, thereby removing the need to bootstrap the system with user data. Additionally, since the trail capturing component is located on the end user's machine, adaptive link augmentation can also be provided across any hypermedia web page that the user visits.

This paper focuses on the extraction and analysis of a document's spatial context as viewed from a user's perspective. Spatial context has been referred to elsewhere as the user's browsing context [16]. A method is presented for obtaining history information and using it as the basis of a linkbase-filtering algorithm. A linkbase is simply a database of links and the filter results in a single 'active' linkbase that contains a set of context dependent links. These links can be dynamically inserted into the current document. A new linkbase will be activated whenever the system determines that the user has entered a new context and as associated linkbase is available. This link augmentation technique has been shown to provide users with a viable means of decreasing search times [12], and combining this with the concept of spatial context, will benefit users and provide a new research area for adaptive

systems.

BACKGROUND

The role of linking has long been established in the hypermedia community where its primary use has been as a mechanism for navigation. Since the early days of adaptive hypermedia systems, links have been employed in many systems as a means of adaptive navigational support [5,10,18], and adaptive presentation [13].

The importance of links was reinforced in Brusilovsky's seminal paper [4] where he defined several subcategories of adaptive navigational support: direct guidance, link sorting, link hiding, link annotation and map adaptation. However it seems that a sixth category can be added – (adaptive) link augmentation – which we define as the process of dynamically inserting additional links into existing web page. This differs from link annotation, which concerns the visible properties of hyperlinks, although these techniques can be combined to provide annotated, augmented links. The advantage of augmented linking is that the underlining navigational structure of the web page remains unaffected as all the original hyperlinks remain intact. However, the danger lies in information overload, which results when too many links are added, possibly leading to the situation where 'every word becomes a link'.

While there are several link augmentation systems, the earliest occurrence was seen in Microcosm [14,9], which was first developed in 1990 as a distributed open hypermedia environment that provided the user with dynamic, cross-application hyperlinks on the fly. These links were inserted (augmented) into the user's existing application and selecting one of these links issued a request to the Microcosm link service. This link service maintained a set of link databases (linkbases) and each link had one of three start point types: generic, specific or local. Specific links originate from an object at a specific point in the source document; local links originate from an object at any point in a specific document, and generic links, link from an object at any position in any document. Microcosm also provided text retrieval links where the user could highlight any text and ask the system to supply a set of related links.

One of the follow up projects to Microcosm, the DLS (Distributed Link Service) [7], is a link delivery system that operates in an open hypermedia environment. The DLS was aimed at bringing the concepts from the open hypermedia community to the Web. It acts as a link service providing other applications with hyperlinks on demand. These links are stored in multiple linkbases maintained by the DLS. However, by removing the need for hard-coded hyperlinks, the responsibility of determining link context fell on the shoulders of the user. So the major limitation of the system is in its inability to automatically switch between linkbases depending on the context of documents [11].

Today, many of the features found in both Microcosm and the DLS can be seen in Active Navigation's Portal Maximiser[1]. Essentially a web site server engine, Portal Maximiser provides many features such as document recommendations, contextual and relevance ranked search results, document categorization and theme-based dynamic (augmented) linking.

Another link augmentation system, WBI (WeB Intermediaries) [17], has been written as a web proxy that adds intermediary functions to the World Wide Web (WWW). WBI sits between the user and the outside WWW, analysing every page a user visits. It has a set of knowledge bases (KB's) hand authored for a specific subject. When the user views a related page, it replaces any known word or phrase with a hyperlink from the KB. If the user clicks on this hyperlink, a further information dialog box pops up on the client's machine offering additional resources (like word definitions or links to external web pages).

The PWW system [16] offers an approach to implicit (zero-input) personalization that is similar to the one taken in this paper. The system described by Kushmerick et al. is a server side system that offers recommendations to pages in the web site based on the URL and/or content of the referring page. User evaluations indicate that the performance of PWW gives a value 77 times more effective then random guesswork. Our approach significantly differs from PWW by moving the architecture away from server side scripting thus allowing the system to gain a context of the user that extends across their entire browsing history and not just the referring page.

More recently, the Web, aided by improvements in browser technology, has seen the development of knowledge delivery systems that implement features similar to AH systems. While such system lack any formal user modelling component, knowledge delivery systems, such as Flyswat[2], Zapper[3] and Atomica[4], provide resources such as link augmentation, keyword lookup, recommender functionality and shopping facilities to provide additional information to their users.

The current systems that employ link augmentation do so on the basis of the individual text of each word or phrase. If there is a match with a known link, then the word is replaced with the corresponding link. However this causes a problem when words have different meanings in different contexts. For example, the word 'java' may refer to the programming language Java, the country or the coffee bean and it is only by analysing the context in which the word is used that it is possible to distinguish between these meanings.

CONTEXT

Context is an important concept that has been examined in

[1] http://www.activenavigation.com/PortalMax/default.htm
[2] http://www.flyswat.com/
[3] http://www.zapper.com/
[4] http://www.atomica.com/

many different fields and for various tasks. It is also an involved issue, as it depends on the task at hand and the available variables that can be modelled in relation to that task. In the case of 'linking' the primary entities involved in this particular task, are those of the user and the document. There are many factors that can affect the context of the user. These include the user's role in an organisation/group/etc, their physical location, level of expertise in various topics, browsing history, interests, tasks, etc. Many of these user features are already captured in existing user models. The context of a document can be defined in many different ways such as by its content, its format (html, pdf, gif, etc), its purpose, the date it was created, the server on which it resides on, its download speed, etc. [11]. One particularly relevant system to the work presented in this paper and which has addressed the issue of 'linking in context' is the QuIC system.

QuIC

The work in this paper has drawn on work undertaken for a project called QuIC (Queries in Context). QuIC is a multi-agent system that was developed at the University of Southampton with the overall goal of utilizing concepts from the open hypermedia community to help users with their navigation and information finding activities. One of the issues addressed by the QuIC system is the use of linking in context as a way of assisting users in their information finding activities. This specifically targets a failing associated with traditional information retrieval models which is attributed to the isolation of these systems from the context in which queries are made [6].

THE QuIC APPROACH TO CONTEXT

The model used by the QuIC project defines two factors for context: the interests of a user, and the contents of the document within which the links are to be rendered. A number of methods have been developed for using the content of unstructured information resources for inferring user interests for the purpose of constructing user or filtering models. In these models, the capture of user context or document context for the achievement of a specific task, is one of the goals. Depending on the specific task at hand, a number of techniques have been employed to build such models or profiles. Examples of techniques employed by Web agents to learn or capture a document or user profile include Decision trees, Neural Nets, Bayesian classifiers, Nearest Neighbour and TF-IDF (Term Frequency, Inverse Document Frequency) [19].

The decision was made to adopt a technique that would be capable of accurately capturing document context. TF-IDF is a very well studied and widely used information retrieval technique [22]. The technique is used to derive weights for terms in a way that would reflect their importance in a given document. TF-IDF is based on the vector space model where a vector is used to represent a document or a query. The cosine angle between different document vectors is a measure of how similar the documents are, and is used as a similarity function. Used in conjunction with a similarity function and other text processing techniques such as stop word removal and stemming, TF-IDF can be employed to distinguish between documents. The model has been used successfully for document ranking, document filtering, document clustering, and as the basis for relevance feedback [1]. One of the advantages of using the TF-IDF method is that unlike many other machine learning algorithms, it does not require large training data sets in order to distinguish between various documents. By representing a document through a vector space model computed via TF-IDF, comparing a document to other documents or queries is simply achieved through the application of a similarity function. The technique has therefore been employed by a large number of Web assistants, examples of which are: FAB [3], WebMate [8], and Margin Notes [21]. To further increase the accuracy of this method in distinguishing between different contexts, conditional heuristics, as described in [12], have been introduced to enable better determination of context.

The work resulting from the QuIC project has obvious applications for adaptive hypermedia systems, specifically adaptive navigational support such as link annotation and augmentation. To this end, the idea of context has been extended within this work to include the concept of a user's spatial context within an information domain.

SPATIAL CONTEXT

The goal of this work is to use the context technology described above to introduce the concept of spatial context into adaptive hypermedia systems. Such a system would work alongside existing user models providing another level of contextual information for the modelling component to draw upon.

A document's spatial context represents its location within the surrounding information domain. This spatial context is refined further by a user's path through the hyperspace before arriving at the current document. By doing this, a user is effectively selecting one path out of many other possibilities. In a hypermedia environment like the Web, the number of possible paths to a web page is continually changing, making it impossible for page designers to cater for the needs of all possible visitors. As a result, hard coded hyperlinks tend to cater for the 'average' user who has followed the most logical path to arrive at the current document.

SPATIAL CONTEXT IN AN EXAMPLE SITUATION

Figure 1 shows three separate paths (starting from documents labelled 1,2&3) taken to reach the central document (shaded grey). The black arrows represent the traversed navigational hyperlinks, while the grey lines represent other pre-defined (hard coded) hyperlinks. In a real-world scenario, the central document might be a review of an XML book in an online bookstore such as Amazon[5]. Trail 1 would be a single link from one XML book review to other similar books. Trail 2 represents a direct link from the author's external homepage to the current book. Finally,

[5] http://www.amazon.com/

trail 3 represents a user interested in Computer Science technologies, who visits the bookstore and searches through reviews of Network and Programming language books before finally arriving at the XML book review.

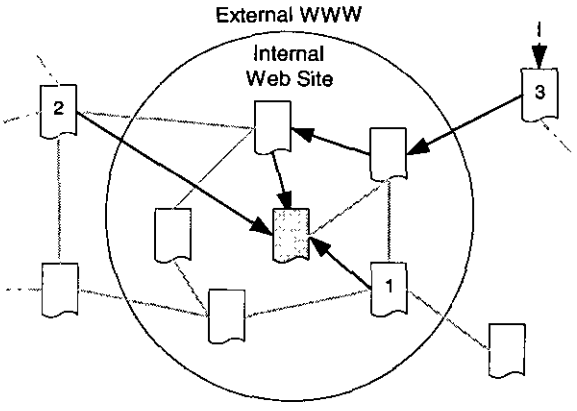


Figure 1. Examples of a document's spatial context

Each of these three trails represents a different context. If link augmentation were to be provided, each user would require a different set of hyperlinks. Situation 1 would require links to other XML books, the user in trail 2 should be presented with links to books by the same author, while in the third example, the augmented links should point the user to books on a wide range of computer science subjects. It is this issue of knowing when to activate these 'dynamic linkbases' that can be overcome by understanding the user's spatial context in arriving at the current document.

The following system uses the contextual component of QuIC to provide a means of capturing a user's spatial context and, using this information, select and apply a dynamic linkbase.

SYSTEM ARCHITECTURE

The spatial context analyser has been implemented as an agent-based system built on top of an agent framework developed at Southampton University called SoFAR (Southampton Framework for Agent Research) [20]. SoFAR is a Java implemented framework designed as a test bed for agent research. SoFAR provides performatives for communication between agents, and ontologies for defining the contents of this communication. The decision for building an agent system arose due to the modular nature of the system's components, which are well suited to an agent environment [2] and the desire to distribute the linking mechanism (which is essentially a DLS).

The network structure has been designed as a client-server approach. The agents (which run within the SoFAR environment) and user data all reside on a single server. In contrast to this, and in following with the architecture of the DLS, the linkbases used by the Linkbase Agent are fully distributed and can reside on any web server.

The user interface is packaged as a downloadable Perl application and executed on the client's machine. This interface communicates with the agent server through the use of sockets and also hooks into the client's Internet Explorer Web browser through the Microsoft OLE (Object Linking and Embedding) automation feature. This allows the system to receive browser events such as OnLoad, DownloadComplete and DocumentComplete. However as a result, the system has been restricted to machines running Microsoft Windows with Internet Explorer 5.0 or later.

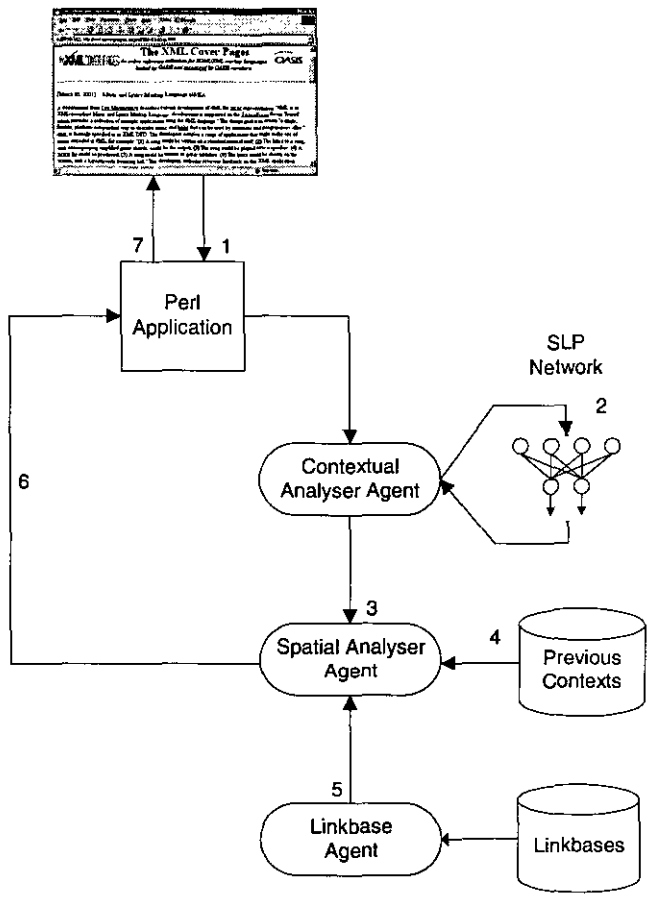


Figure 2. The System Architecture

The system is triggered whenever the Perl application receives a DocumentComplete event. This event is fired whenever a new page has been fully loaded by the browser. For each page the user views, the following steps are executed.

1. The Perl program first captures the URL of the current page and forwards this information onto the context agent.
2. An important issue for the context agent is that it needs

to be aware that not every page is a possible candidate for context analysis. While some pages, such as Shockwave Flash sites, will produce empty context sets when processed which the system will ignore, others will produce misleading information, such as the '404 Not found' and '301 Document Moved' pages created automatically from broken hyperlinks. In order to identify these pages, the context agent employs a Single Layer Perceptron network, which extracts the key features from the page and applies a set of weights to them. These features include identifying adjoining words like 'Document Moved', the amount of text in the page, the number of hyperlinks and the frequency of keywords such as 'broken', 'error' and '404'. The neural network produces a probability, which determines its confidence in the page belonging to the 'Page Not Found' category. If this is high, the page is ignored; otherwise the context agent applies the context algorithm to the page and produces a 'context model' for this document.

3. This model is then passed to the spatial context analyser to determine the context of the current user. Here there are three possibly outcomes: the user is in the existing context, the user has returned to a previous context or the user has entered a new context. Firstly, this agent compares the new context model against the current model using a cosine similarity function. If a match is found, then the system moves onto step 5, otherwise step 4 occurs.
4. The spatial analyser agent compares the current context model against the entire set of previous context's that the user has experienced since they last logged in. This comparison uses the same similarity function. If the highest match exceeds a given threshold then the system assumes that the user has returned to a previous context. This could occur for a number of reasons. For instance, the user could press the back button after arriving at an irrelevant page or the user has finished following a search thread and is returning to an old topic, this could also occur if the user has several browser windows open and is switching between them. If no match is found, then the current document's context forms the start of the user's new browsing context (and a record of it is stored in the 'Previous Contexts' database).
5. When the system has calculated the current context of the user, a request is sent to the linkbase agent for a matching linkbase. This agent applies keyword matching to each known linkbase to find the highest similarity match with the context. The linkbase agent returns the highest matching linkbase (or null if no match is found).
6. The system passes all the links from the matching linkbase onto the Perl application.
7. The last job of the Perl program is to extract the text from the web, page search through it and replace all the matching words with a hyperlink. Matches are found through simple keyword comparison. The resulting text is re-inserted back into the web page. The text extraction is achieved by an OLE call to the browser to return all the text in the current web page. This request ignores the page type giving the link augmentation process the ability to operate on any type of web page including dynamic pages such as ASP, JS, CGI, PHP and SHTML.

This whole process is executed in real time and the user will see the requested web page displayed, and then a second later, the new links will appear. Because the page is held in memory, there is no visible refresh; all that is apparent is that certain words are instantly transformed into hyperlinks. Often there is no visible delay between loading the page and inserting the links.

Context dependent links work due to the property that all the links reside in a set of linkbases and each linkbase has been hand crafted to contain relevant links on specific subjects. For instance, a linkbase on the subject of XML might contain individual links to:

- W3C XML specification
- XML parsers
- XML technologies
- XML books.

The authors of these linkbases have the freedom to make each linkbase as detailed as desired, so for example, there could be separate linkbases for each of the above sub-topics or even linkbases of each sub-sub-topic.

Figures 3 shows the system in operation. It shows the same page is viewed from two different spatial contexts[6]. Page (a) appears as viewed by a user who has been reading articles about XML before arriving at the current document. In this instance, the system has determined that the user's spatial context best matches the 'XML' linkbase and so links from the keywords 'XML', 'DTD', 'standard' and 'developer' have been inserted. Although it appears that this page exists in a predominately XML context, the user viewing page (b) has previously been looking at music related sites and therefore their spatial context best matches the 'Music' linkbase and so the page has been augmented with the links 'lyrics' and 'music'.

LINKBASES AND CROSS-DOMAIN SUPPORT

By abstracting the links and storing them in a set of linkbases, the system gains cross-domain support for free. This one-size-fits-all approach will provide link augmentation to any information domain contained within the linkbases. However the system relies heavily on both the quality and quantity of these linkbases. Absent linkbases will simply lead to a lack of augmented links for that domain, however badly authored linkbases can lead to unhelpful or irrelevant links. While the existing linkbases have all been hand authored, it is desirable to find an automated link extraction algorithm that could be used to create linkbases to cover a variety of topics.

[6] In these screen shots, the original document contains the hyperlinks: 'sponsored', 'Leo Montgomery' and 'SoundForge'.

24. Rafter, R., Bradley, K., & Smyth, B. (2000). "Automated Collaborative Filtering Applications for Online Recruitment Services". Adaptive Hypermedia and Adaptive Web-Based Systems ItalySpringer-Verlag
25. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). "GroupLens: An Open Architecture for Collaborative Filtering of Netnews". Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work (pp. 175-186).
26. Rocchio, J. (1971). "*Relevance Feedback in Information Retrieval*". The SMART Retrieval System: Experiments in Automatic Document Processing, 313-323Notes: No lo tengo aun.
27. Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). "A bayesian approach to filtering junk e-mail". AAAI-98 Workshop on Learning for Text Categorization
28. Singh, M., & Provan, G. M. (1996). "Efficient learning of selective Bayesian network classifiers". . Proceedings of the 13th International Conference on Machine Learning
29. Vel, O., & Nesbitt, S. (1997). "A Collaborative Filtering Agent System for Dynamic Virtual Communities on the Web". URL= http://citeseer.nj.nec.com/de-collaborative.html.
30. Versteegen, L. (2000). "The Simple Bayesian Classifier as a Classification Algorithm". URL= http://www.cs.kun.nl/nsccs/artikelen/leonv.ps.Z.

Finally, while the system does indeed produce real-time context-dependent augmented links, there has yet to be any formal evaluation of the system. When the improvements stated above have been introduced, the final stage will need to involve a system evaluation.

CONCLUSIONS

While link augmentation is not a new technique and has been used in many systems before, applying context analysis as a means of filtering out irrelevant links is entirely new. The work here shows that link augmentation, when applied to a document's spatial context, is a highly significant area for further exploratory study. To support this claim, user evaluations of the QuIC project (as reported in [12]) and PWW [16] already show that link augmentation and recommendation are viable means of enriching the existing hypermedia domain with effective user-centric information which can be used as a means of decreasing search times.

The flexible design of the system allows link augmentation to be provided across a variety of information domains, dependent only on the availability of linkbases. In addition to this, the authors feel that adaptive link augmentation, when implemented with care, is a worthy addition to Peter Brusilovsky's list of adaptive navigation technologies and warrants further research.

ACKNOWLEDGMENTS

The work presented in this paper has been supported by the QuIC project, ESPRC grant GR/M77086

REFERENCES

1. Baeza-Yates, R. and Ribeiro-Neto, B. (1999). *"Modern Information Retrieval"*. Addison-Wesley/ACM Press.

2. Bailey, C and Hall, W. (2000) *"An Agent-Based approach to Adaptive Hypermedia using a link service"*. P. Brusilovsky, O. Stock, C. Strapparava (Eds.): Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, Trento, Italy, August 2000, pp. 260-263

3. Balabanovic, M. and Shoham, Y. (1997). *"Fab: content-based, collaborative recommendation"*. Communications of the ACM, 40(3), pp. 66-72.

4. Brusilovsky, P. (1996). *"Methods and Techniques of Adaptive Hypermedia"*. Journal of User Modelling and User-Adaptive Interaction, UMUAI6.

5. 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), pp. 291-300.

6. Budzik, J. (2000). *"User Interactions with Everyday Applications as Context for Just-in-time information Access"*. In Proceedings of Intelligent User Interfaces (IUI), ACM, New Orleans, LA USA, pp. 44-51.

7. Carr, L., De Roure, D., Hall, W. and Hill, G. (1995). *"The Distributed Link Service: A Tool for Publishers, Authors and Readers"*. World Wide Web Journal 1(1), O'Reilly & Associates (1995). pp 647-656.

8. Chen, L. and Sycara, K. (1998). *"Webmate: A Personal Agent for Browsing and Searching"*. In Proceedings of the Second International Conference on Autonomous Agents, ACM, Minneapolis, pp. 132-139.

9. Davis, H.C., Hall, W., Heath, I., Hill, G. and Wilkins, R.J. (1992) *"Towards an Integrated Information Environment with Open Hypermedia Systems"* In Proceedings of ECHT'92, ACM Press, pp. 181 – 190.

10. De Bra, P. and Calvi, L. (1998). *"AHA: a Generic Adaptive Hypermedia System"*. In Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia, HYPERTEXT'98, Pittsburgh, USA, June 20-24, 1998.

11. El-Beltagy, S. (2001). *"An Agent Based Framework for Navigation Assistance and Information Finding in Context"*. PhD thesis. University of Southampton, 2001.

12. El-Beltagy, S., Hall, W., De Roure, D. and Carr, L. (2001). *"Linking in Context"*. To appear in the Twelfth ACM Conference on Hypertext and Hypermedia (HT'01), Denmark, 2001.

13. Espinoza, F. and Höök, K. (1997). *"A WWW Interface to an Adaptive Hypermedia System"*. Presented at PAAM (Practical Applications of Agent Methodology), April 1996, London. http://www.sics.se/~espinoza/pages/PAAM_submission .html

14. Fountain, A., Hall, W., Heath, I. and Davis, H. C. (1990) *"Microcosm: an open model with dynamic linking"*. In Hypertext: Concepts, Systems and Applications (A. Rizk, N. Strietz, and J. Andre, eds.), (France), pp. 298-311, European Conference on Hypertext, INRIA, November 1990.

15. Kim, J., Oard, D. W., and Romanik, K. (2000) *"Using implicit feedback for user modeling in Internet and Intranet searching"*. Technical Report, College of Library and Information Services, University of Marylandat College Park.

16. Kushmerick, N., McKee, J. and Toolan, F. (2000). *"Towards Zero-Input Personalization: Referrer-Based Page Prediction"*. P. Brusilovsky, O. Stock, C. Strapparava (Eds.): Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, Trento, Italy, August 2000, pp. 133-143

17. Maglio, P., P. and Farrell, S. (2000). *"LiveInfo: Adapting web experience by customization and annotation"*. In Proceedings of the First International

Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2000). LNCS Series, Springer-Verlag.

18. Mladenic, D., (1996). *"Personal WebWatcher: Implementation and Design"*. Technical Report IJS-DP-7472, Department of Intelligent Systems, J.Stefan Institute, Slovenia.

19. Mladenic, D. (1999). *"Text-Learning and Related Intelligent Agents: A Survey"*. IEEE Intelligent Systems, 14(4), pp. 44-54.

20. Moreau, L., Gibbins, N., De Roure, D., El-Beltagy, S., Hall, W., Hughes, G., Joyce, D., Kim, S., Michaelides, D., Millard, D., Reich, S., Tansley, R. and Weal, M. (2000). *"An Agent Framework for Distributed Information Management"*. In The Fifth International Conference and Exhibition on The Practical Application of Intelligent Agents and Multi-Agents, Manchester, UK.

21. Rhodes, B. J. (2000) *"Margin Notes: Building a Contextually Aware Associative Memory"*. In Proceedings of Intelligent User Interfaces (IUI '00), ACM, New Orleans, LA USA, pp. 219-224.

22. Salton, G. and McGill, M. J. (1983). *"Introduction to Modern Information Retrieval"*. McGraw Hill, New York.

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

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

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

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

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

1 Introduction

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

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

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

The basic components of Adaptive Hypermedia Systems are the *Application Domain Model*, the *User Model* and the *techniques to adapt presentations* to such model. The Application Domain Model is used to describe the hypermedia basic contents and their organisation to depict more abstract concepts. In addition to traditional models, such as those developed in the Human-Computer Interaction and Database fields, the modelling of AHs requires to consider the different

sources that affect the adaptation process. The approach that seems to be the most promising for modelling the Application domain is data-centric, and many researches employ well-known database modelling techniques [1].

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

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

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

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

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

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

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

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

2 Adaptive Hypermedia Modelling

This Section presents XAHM, our approach to the modelling of Adaptive Hypermedia. Note that in the rest of the paper the term Application Domain will refer to an Adaptive Hypermedia in a particular Application Domain. We first introduce our proposed multidimensional adaptation scheme, and then we show a graph-based layered model for the Application Domain and a probabilistic interpretation of the hypermedia structure, modelling respectively the logical structure and the "intrinsic properties" of the adaptive hypermedia. XAHM adopts XML essentially because

of its flexibility and data-centric orientation: it makes possible to elegantly describe data access and dynamic data composition functions, allowing the use of pre-existing multimedia basic data (e.g. stored in relational databases and/or file systems) and the description of contents in a terminal-independent way.

2.1 Adaptation Space

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

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

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

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

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

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

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

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

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

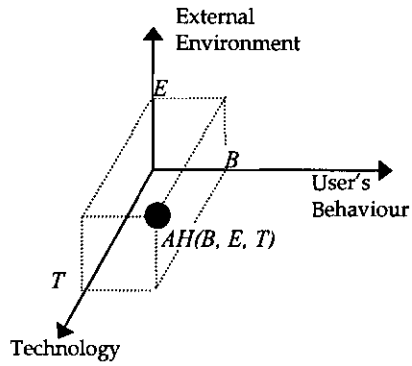


Fig. 1. Adaptation space and adaptivity dimensions

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

2.2 A Layered Model for the Logical Structure of the Hypermedia

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

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

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

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

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

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

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

2.3 Probabilistic Interpretation of the Adaptive Hypermedia Schema

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

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

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

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

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

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

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

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

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

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

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

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

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

2.3.1 Static Properties of Hypermedia Structure

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

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

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

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

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

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

3 User Classification

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

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

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

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

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

On this basis, the system constructs three PDFs:

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

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

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

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

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

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

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

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

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

4 System Architecture

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

4.1 XML Metadata and Presentation Descriptions

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

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

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

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

As said before, in our system the Presentation Descriptions are XML documents whose key parts are the *content*, *fragment* and *embedded-code* elements. The *content* element is used to include text in the page. The *fragment* element is useful for including basic multimedia fragments referenced by their aliases. Finally, the *embedded-code* element increases flexibility allowing the insertion of

terminal-dependent code (e.g. WML, HTML) in the page (obviously, wrapped by an XML CDATA section).

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

4.2 The Run-Time System

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

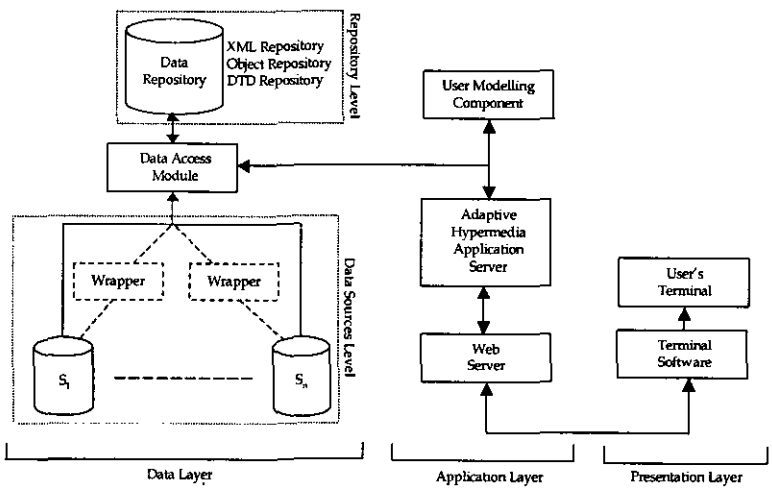


Fig. 2. Run-time system architecture

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

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

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

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

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

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

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

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

4.3 The Java Adaptive Hypermedia Suite (JAHS)

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

4.3.1 Multidimensional Adaptive Hypermedia Authoring

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

High-Level Structure Definition

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

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

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

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

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

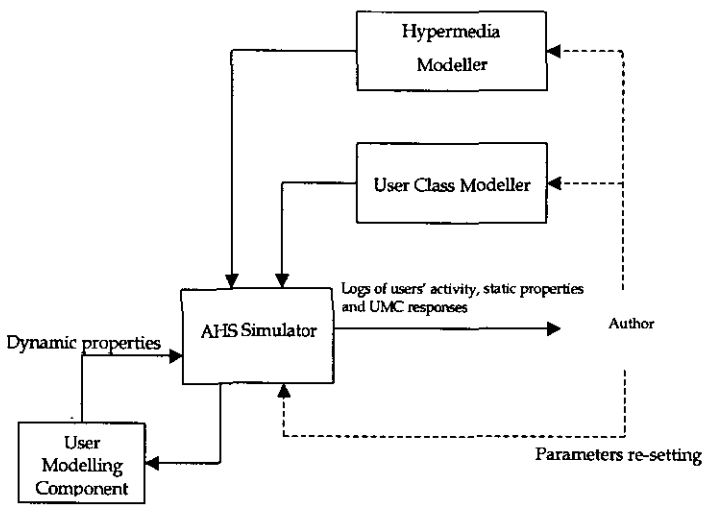


Fig. 3. The Simulation Tool

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

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

5 Concluding Remarks and Future Work

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

We believe that the main contributions of this paper are:

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

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

References

1. Abiteboul, S., Amann, B., Cluet, S., Eyal, A., Mignet, L., Milo, T., "Active views for electronic commerce", in *Proceedings of the 25th VLDB Conference*, 1999.
2. Adaptive Hypertext and Hypermedia Home Page, http://wwwis.win.tue.nl/ah/
3. Apache Software Foundation, "The Apache XML Project", http://xml.apache.org.
4. Ardissono, L., Goy, A., "Tailoring the Interaction With Users in Web Stores", in *User Modeling and User-Adapted Interaction*, 10(4), Kluwer Academic Publishers, 2000.
5. Brusilovsky, P., "Methods and techniques of adaptive hypermedia", in *User Modeling and User Adapted Interaction*, v.6, n.2-3, 1996.
6. Cannataro M., Carelli G., Pugliese A., Saccà, D., "Semantic lossy compression of XML data", *8th Int. Workshop on Knowledge Representation meets Databases* (KRDB-2001) in conjunction with *VLDB 2001*.

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

Revisiting and Versioning in Virtual Special Reports

Sébastien Iksal and Serge Garlatti

Department of Artificial Intelligence and Cognitive Sciences
ENST Bretagne
Technopôle de Brest Iroise
B.P. 832, 29285 Brest Cedex
FRANCE
Tel. : (+33) 2 29 00 14 28
Fax : (+33) 2 29 00 10 30
{sebastien.iksal, serge.garlatti}@enst-bretagne.fr

Abstract. Adaptation/personalization is one of the main issues for web applications and require large repositories. Creating adaptive web applications from these repositories requires to have methods to facilitate web application creation and management and to ensure reuse, sharing and exchange of data through the internet/intranet. Virtual documents deal with these issues. In our framework, we are interested in adaptive virtual documents for author-oriented web applications providing several reading strategies to readers. These applications have the following characteristics: authors have know-how which enables them to choose document contents and to organize them in one or more consistent ways. A reading strategy and the corresponding content are semantically coherent and convey a particular meaning to the readers. Such author's know-how can be represented at knowledge level and then be used for generating web documents dynamically, for ensuring reader comprehension and for sharing and reuse. Then an adaptive virtual document can be computed on the fly by means of a semantic composition engine using: i) an overall document structure – for instance a narrative structure - representing a reading strategy for which node contents are linked at run time, according to user's needs for adaptation, ii) an intelligent search engine and semantic metadata relying on semantic web initiative, and iv) a user model. In this paper, we focus on a semantic composition engine enabling us to compute on the fly adaptive/personalized web documents in the ICCARS project. Its main goal is to assist the journalist in building adaptive special reports. In such a framework, adaptation, personalization and reusability are central issues for delivering adaptive special reports.

Keywords : Semantic Composition Engine, Adaptation/Personalization, Semantic Web, Ontology, Virtual documents, Revisiting, Versioning

Introduction

Numerous applications are available on the Web today and their size and volume are increasing. For instance, portals, e-learning, problem solving systems, decision support systems, digital libraries, on-line information systems, virtual museums, e-business and digital newspapers are current applications. On the World Wide Web, we have large distributed information repositories which convey large amounts of knowledge for internet users as well as for companies which are the owners of this knowledge. Adaptation/personalization is one of the main issues for web applications. 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 applications - also often called Adaptive Hypermedia Systems - can provide different kinds of information, different layouts, different navigation tools according to users' needs [1]. Creating adaptive web applications from these repositories requires the following features: i) methods to facilitate web application creation and management and ii) reuse, sharing and exchange of data through the internet/intranet.

The notion of flexible hypermedia and more particularly of virtual documents can lead to methods facilitating web application 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 at run time" [2]. Virtual documents have grown out of a need for interactivity and individualization of documents, particularly on the web. Virtual document and adaptive hypermedia are closely related – they can be viewed as the two faces of the same coin. Reuse, sharing and exchange through the internet/intranet require to have a precise search engine. 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 [3] and semantic metadata, overcome these limitations [4, 5].

In our framework, we are interested in adaptive virtual documents for author-oriented web applications providing several reading strategies to readers. These applications have the following characteristics: authors have know-how which enables them to choose document contents and to organize them in one or more consistent ways – author reading strategies. Content and organizations are "semantically" related to ensure reader's comprehension. In this paper, we focus on organizations called narrative structure. The reader has the ability to recognize – sometimes unconsciously – these structures. For instance, scientific papers, courseware, report, special report in journalism, etc., have each of them a distinct narrative structure. At present, the narrative structure is implicit in printed document, but also in digital one. Such author's know-how and skills can be represented at knowledge level and then be shared and reused among authors, used for generating web documents dynamically and for enhancing reader comprehension. A narrative structure provides an overall document structure which is a declarative description of web documents. Then, a web document can be computed on the fly by means of a semantic composition engine using: i) an overall document structure – for instance a narrative structure - representing a reading strategy for which node contents are substituted at run time, according to user's needs for adaptation, ii) an intelligent

search engine, iii) semantic metadata, and iv) a user model. An authoring tool is provided for creating narrative structures, specifying their content and associating metadata.

Numerous web sites offer tools to users for personalizing their information space and the presentation. Two projects aims at supplying personalized news on a Web site. Sistemi Telematici Adattativi [6] is a project which aims at filtering and displaying news and advertisement according to users' preferences and characteristics. The system selects a relevant set of news according to the reader's interests via probabilities. Then, some rules are applied for choosing the relevant presentation for a single news (abstract or full text, image or video, etc.). Personalization is done by a filtering process on news and a presentation selection. KMI Planet [7] is a kind of private on-line newspaper where all readers and writers are in a same group – university. It collects news through e-mail, processes them and sends the result to the most interested readers. The tool is able to order articles for filling in gaps, and after to inform the reader when the news is available. It supplies with an advanced interface for searching documents. Each news is annotated via an academic ontology, and then the query interface uses the same ontology for writing queries. The system uses the annotations and all queries given by a user to find out the most relevant news. Personalization mainly consists of a filtering process based on user queries and ontology annotations. In this two projects, personalization is mainly based on a filtering process using user's preferences for selecting the most relevant news.

Our project aims at delivering adapted special reports to news readers. They consist of a set of news – selected by authors - and several semantic structures organizing them. These semantic structures provide different reading strategies to the readers. A reading strategy and the corresponding news collection are semantically coherent and convey a particular meaning to the readers. This meaning can be viewed as a viewpoint on this collection. Some reading strategies are created by authors and others by the system. The reading strategies and their content may be adapted to users' needs. According to C. Watters, revisiting, versioning and reusability are some of the main issues for virtual documents. In this paper, we focus on a semantic composition engine enabling us to compute on the fly adaptive/personalized special reports in the ICCARS project. Its main goal is to assist the journalist in building adaptive special reports. In such a framework, adaptation, personalization and reusability are central issues for delivering adapted/personalized special reports.

Firstly, we present the context of our approach: journalism and reporting on the web via the ICCARS Project. Next, we will give a summary of the architecture of our semantic composition engine. We will present how we will manage revisiting and versioning of dynamic documents according to our context and our composition engine. Then, our adaptation policies is presented. Finally, some directions for the future will be proposed.

ICCARS Project

ICCARS is the acronym for INTEGRATED AND COOPERATIVE COMPUTER ASSISTED REPORTING SYSTEM. It is a joined project between the IASC Laboratory, a SME called Atlantide and a regional daily newspaper called Le Télégramme. It is funded by Brittany Regional Council. The ICCARS prototype will be a computer assisted reporting system. Its main goal is to assist the journalist in creating adaptive special reports. These documents are able to include audio and video material, links, and they are no longer limited in size.
Due to internet features, numerous web sites are offering news. Then, It is not sufficient to filter news. We need to go beyond a news delivery service and to provide new services around information. Special reports seem to be the most representative journalists' task. A special report offers news as well as analysis, debate, synthesis and/or development. It can be viewed as an organized collection of articles offering a viewpoint on events. It is a matter of journalist know-how for creating such type of document.

The digital special report

A special report is a synthesis made by one or more journalists on a particular topic, for instance a yachting race. We consider a special report as a collection of articles with a given narrative structure. In a paper version, there is a single organization which appears through the sequence of pages and the page layout. A digital special report may naturally offer different narrative structures.

New features

Digital special reports can provide new services to the reader. We present some of them: such as various reading strategies for a single special report, enrichment, reusability and adaptation. It is important to notice that as in the printed version, the journalist chooses the set of articles belonging to a special report. That is to say, in a digital special report we can find only the articles that the journalist wants to provide his readers with.

The notion of reading strategy

In a special report, a set of articles can be read in different ways, according to a reader's or author's viewpoints. We call a particular sequence of articles a reading strategy. We distinguish two kinds of reading strategies:

1. An author strategy: This is a narrative structure designed by a writer for presenting a particular angle on a set of articles. One of the main roles of journalists is to analyse events and report them in a consistent and synthetic way. A narrative structure is composed of nodes and semantic relationships. Nodes are spans of texts. Relationships belong to those analyzed by Rhetorical Structure Theory (RST) [8]. RST defines relations between spans of text, each span have a role inside the relation (nucleus and satellite). Each relation is defined by some constraints on the nucleus, the satellite, the

combination of the nucleus and the satellite, and an effect to the reader. Among these relations, we can find are antithesis, restatement, summary, interpretation and so on.

2. A reader strategy: This is an overall document structure computed from a reader's goals. For instance, it can be based on geographic, history or topic criteria – a domain model - organizing the access to articles. The structure delivered is computed on the fly and controlled by the computer according to a generic structure. Nevertheless, journalists are aware of such structures because they associate metadata with articles and special reports for these services.

Special report views

In a digital document, three different views coexist: semantic, logical and layout/presentation [9]. For each view we have a specific structure. The semantic structure of a document conveys the organization of the meaning of the content of a document. This view fits the semantic level of the semantic web architecture. Indeed, it can be represented by ontologies. Ontologies are used to model types of fragment as well as their relationships. For instance, "The fragment A which is an *interview* is the volitional cause of the fragment B which is an *analysis*", the underlying relationship cannot be represented by a syntactic structure [10]. Interview and analysis are types of fragment. The interview is the satellite and the analysis is the nucleus of this rhetorical relation. This relation is oriented and encode a particular reading guide. In this case, the fragment B will be better understood if the fragment A is read before. It could be interesting to show the type of relation to the reader as explanations or for increasing the comprehension.

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). These can also be broken down into components (titles, paragraphs, figures and so forth). It turns out that just about every document can be viewed this way. The logical view fits the syntactic level of the semantic web architecture. A logical structure can be encoded in XML [11]. The layout/presentation view describes how the documents appear on a device and a physical structure describes it, (eg. the size and colour of headings, texts, etc). The layout/presentation view may be processed by an XSLT processor [12] for transforming an XML document into an HTML document that can be viewed by any web browser. It can also be processed by a java engine able to compute an XML document for presenting by a web browser.

In a printed document, these three views are intertwined and are not separable. There is no straightforward mapping between the semantic and the logical structure, that is to say, for instance, a paragraph does not correspond to a particular content's meaning. On the other hand, the logical and physical structure are closely related. Indeed, the physical structure encodes the logical structure. For instance, each section element has a particular presentation – font, size, colour, etc. The semantic structure is implicit and so it can be analyzed and/or recognized by a reader. Moreover, it is a key issue for reader comprehension. In a digital document, these three views may be represented and managed.

Reusability
It is very important for journalists to be able to reuse at least an article or a part of a special report in more than one special report and why not in the same report. Indeed, an article and a part of a special report generally concern more than one topic. Watters [2] as well as the Semantic Web Community argues that allowing reusability of fragments – articles, special report parts - leads to associate metadata with fragments. Moreover, narrative structures can also be reused for new special reports.

Enrichment
A special report can be updated. The journalist may add a new article in his base and modify the organization in order to insert this new article. A journalist can organize a subset of articles in order to develop various viewpoints (economic, ecological, for example.). Enrichment is a very important possibility with digital documents, but in order to not disturb the reader, the system must be able to rebuild the same special report, that is to say, the same version of the report. Enrichment leads to the management of versioning in special reports.

Adaptation/Personalization
The digital special report as a particular type of web site is a good candidate for adaptation. As a digital document may be managed at three different levels: semantic, syntactic and layout/presentation, adaptation can take place on each level. For instance, the content and the overall document structure may be adapted to the reader's preferences, knowledge and goals. At a syntactic level, different logical structures may be chosen to fit user needs. At a layout level, the presentation may be adapted to the current device and/or the user stereotype.

Moreover, we have to deal with versioning purposes due to enrichment purposes. Then, personalization involves annotating the documents already read, or those which have been added since the last visit.

Virtual Document for Special Reports

A journalist organizes a set of articles according to one or more reading strategies, but it is necessary to prepare the special report which is relevant to a particular reader and a specific device. First of all, we give a definition of a virtual document in our framework:

- An adaptive/personalized virtual document consists of a set of information fragments, ontologies and a semantic composition engine which is able to select the relevant information fragments, to assemble and to organize them according to an author's strategy or user's goals by adapting various visible aspects of the document delivered to the user.

Fragments can be atomic or abstract information units. The latter are composed of atomic and abstract information units. In ICCARS, fragments are articles – atomic - , special reports and sub-reports – abstract. A special report and corresponding reading strategies are modelled as follows in figure 1.

In order to facilitate comprehension, we prefix all the elements of the special report model with the I of ICCARS. An I-SubReport is composed of a set of articles selected by the author – explicitly associated with it in order to define its relevant information space -, and one or more narrative structures – Reading strategies - between these elements. When a journalist considers that a particular I-SubReport is relevant enough to be a special report, an I-PublishedReport is created and gives a user access to this I-SubReport, that is to say, a document ready to be delivered to readers.

An I-SubReport can be organized according to one or more I-Structures – Reading strategies. An I-Structure is a collection of I-Components among which one is the root of the I-Structure. An I-Component is an abstract object, which exists only inside a particular I-Structure. An I-Component is linked to others through a semantic relation belonging to those of RST. This relationship gives the organization of the I-Structure. That is to say, each I-Component in the structure which is the source of a relationship, is a nucleus in RST and the corresponding destination (an I-Component also) is a satellite. So, we use RST as a basis to build a narrative structure in which nodes are different categories of fragments. If RST is very far from the journalists viewpoint, we will use journalists' relations in the future. The set composed of I-Components and the relationships is one narrative structure of the special report. An I-Component is a kind of information retrieval service which uses a description given by the author according to metadata, in order to send a query to the intelligent information broker. It is able to use the user model to filter the small set of answers. So the I-Component is able to deliver the most relevant articles or sub-reports. Then, the I-SubReport is a graph where the nodes are I-Components and the vertices are relations between I-Components. Several special reports can be generalized to provide special report templates. So, reusability concerns these templates as well as all the instances of the special report model. For instance, in the case of the wreck of the Erika, the structure can be reused for other wrecks – super tanker oil slick - (Tanyo, Amoco Cadiz, etc.).

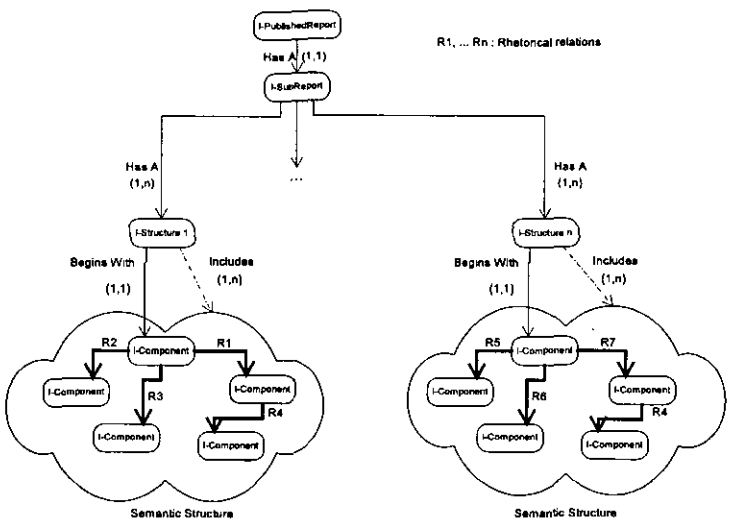


Fig. 1. Model for a special report

This special report model is an input for the semantic composition engine which computes an adaptive/personalized special report for a given reader.

Semantic Composition Engine Architecture

Our semantic composition engine relies on OntoBroker for ontology management and intelligent search engine. OntoBroker is a knowledge management engine which is useful for filtering and information retrieval in a large amount of data as well as in the model specification – ontologies [5, 13, 14]. OntoBroker contains four ontologies and facts 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 and a special report ontology which represents the author's competences and know-how for creating special reports. The domain ontology defines a shared vocabulary used in the metadata schema for the content description of data. It will also be used by the semantic composition engine as an overall document structure, by the user as an information retrieval tool because the user often has difficulty in defining his/her interests, and it is easier for him/her to recognize required information in a domain model than to specify it.

According to the three views of a document, our semantic composition engine architecture is described below (cf. fig. 2).

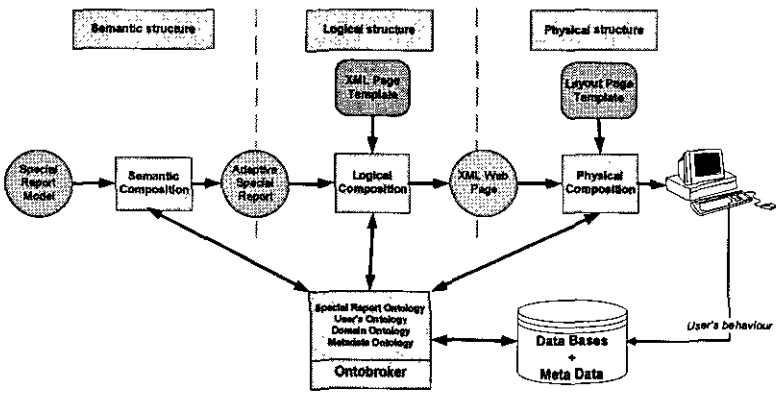


Fig. 2. The Semantic Composition Engine Architecture

One of the main ideas behind the notion of semantic composition engine is to declare as much as possible all the user's tasks and interactions. The semantic composition engine is composed of three different stages: a semantic composition which manages the semantic structure of a special report model for defining a user adapted special report and selects its contents, a logical composition which computes an XML web page from the user adapted special report and a physical composition

which computes the current web page layout from the XML structure[1]. This architecture is based on two different studies: ICCARS Project and CANDLE Project (Collaborative and Network Distributed Learning Environment) which is an european project.

Semantic Composition: for an author's reading strategy, the main role of the semantic composition is to define the special report content and to adapt the chosen I-Structure to user needs. Indeed, each I-Component has only a specification of the content. From this specification, one or more fragments may be selected from the relevant set of articles associated to the considered special report. Indeed, only a subset of metadata entries are used for content specification by the authoring tool. The others are used for defining variants of fragments - according to adaptation policies. The special report structure may also be adapted: an I-Fragment or an I-SubReport may be hidden according to reader's needs, that is to say topics, its professional activity and/or expertise level. Articles may be dedicated to specific professional activities and/or to some levels of knowledge. The semantic composition produces a user adaptive special report – user document - in which all I-components have contents according to their specification and to the user model.

Logical Composition: the logical composition aims at computing an XML page with a content and navigation tools for accessing the different fragments of a special report, by means of a template. A web page, represented as an XML structure [11], is generated from a particular template. A template describes the logical structure of a web page but without any content or navigation tools. It has queries for computing navigation tools and for loading the content via OntoBroker. The content is given by the current node in the narrative structure. For accessing the other nodes, the logical composition engine has to browse the narrative structure. The logical composition has also to associate properties to hyperlinks – Xlinks - for managing annotation, hiding, sorting and direct guidance. All interactions between a user and the web page are also represented in this template in order to respond to users' requests. The database will provide several templates which will be indexed with user's tasks, type of fragments, user's category, user's level of expertise, etc. Then, several templates can be available for a given type of content. By means of these templates, adaptation may take place at this stage.

Physical Composition: finally, the physical composition has to map some presentation rules on the web page. The final process of this architecture is concerned by the design of the web pages of a special report. The final layout of each page may be tailored to the user's preferences: print size, color, and so on and/or use standardized styles from corporate, SMEs or institution style sheets. The physical composition has also to manage the adaptive navigation. From author specification or user stereotypes or user preferences, he has to hide, to annotate, etc; the different types of hyperlinks in a web page. There is a style sheet for each template. It is one way for mapping a presentation on a web page. Indeed, a java process can be applied

[1] A logical structure is associated with one web page and not with the entire document – it cannot be relevant in such a framework.

to an XML structure to provide a web page.

The above description has presented the semantic composition engine architecture and the process for delivering a special report to a user. Now, we present the management of revisiting and versioning in a virtual special report.

Revisiting and Versioning

Computing a document on the fly is very interesting because it is cheaper to produce a single virtual document which can be the source of numerous real documents according to a user's requests than to prepare all these documents in advance. Nevertheless, a real document is ephemeral. Readers have an expectation that documents found once will be available on a subsequent search. In fact, they expect to retrieve the same document in the same state. So the system needs enough information to recreate the document as it was. It is called revisiting purposes which may lead to bookmark an I-Component. In fact, there is no URL to store because of the dynamic generation of the document, so we need to recreate the document and to display the same I-Component.

Thanks to the enrichment capabilities of virtual special reports, it is necessary to manage versioning [15]. Version control is a central issue for special report management, readers need to retrieve a former version of the document, may be because they have already read it or because they want to be aware of the life cycle of this report. They also need to go forward and backward in time through changes in order to develop their own analysis of the situation.

Revisiting and Versioning are very closed together but these are two different issues. Even if we often intertwine them. The main difference between them concerns the domain, providing revisiting features requires the storage in a user model of all the data enabling the semantic composition engine to compute the same real special report again. On the other hand, the versioning process is related to the metadata schema and the special report ontology, that is to say it works directly on documents, structures and so on. According to versioning purposes, revisiting requires to have in the user model the version number of the last visited special report.

Versioning management

Versioning management is related to enrichment features and to the metadata schema. The author will be able to add or remove all the elements of the special report model – I-SubReport, I-Structure, I-Component - and also the relationship between I-Components. We provide, by default, the latest version of the element except if the user asks for a particular version.

The metadata schema (Table 1) provides metadata information about fragments. The semantic composition engine uses the schema for information retrieval. It

matches content specification against metadata. The current version of the metadata schema consists of six parts:

MD.1	General	General information about the resource	Unique Instance
MD.1.1	Identifier	Unique Identifier	Single Value
MD.1.2	Title	Title given by the author	Single Value
MD.1.3	Authors	One or more authors for the resource	Unsorted List
MD.1.4	Description	Short description of the resource	Single Value
MD.1.5	Language	Language of the resource	Single Value
MD.2	Life cycle	Entries for versioning purposes	Unique Instance
MD.2.1	Version	Version number of the resource	Single Value
MD.2.2	Status	Status of the resource (draft, final ...)	Single Value
MD.2.3	Authors	Authors of the version	Unsorted List
MD.2.4	Date	Date of the version	Single Value
MD.3	Meta Metadata	Information about metadata	Unique Instance
MD.3.1	Creator	Author of metadata	Single Value
MD.3.2	Validator	Reviewer of metadata	Single Value
MD.3.3	Language	Language of metadata	Single Value
MD.3.4	Date	Date of metadata	Single Value
MD.4	Technical	Technical information about the resource	Unique Instance
MD.4.1	Location	Where the resource can be found?	Single Value
MD.4.2	Format	Format of the resource	Single Instance
MD.4.2.1	Type	Type of the resource (ppt, doc, html, ...)	Single Value
MD.4.2.2	Size	Size of the resource in Kbytes	Single Value
MD.4.3	Requirements	What is required to read the resource?	Single Instance
MD.4.3.1	Software	Software names and version	Unsorted List
MD.4.3.2	Hardware	Hardware description	Unsorted List
MD.5	Classification	Data about the content and reporting features	Unique Instance
MD.5.1	Domain	Description related to the domain	Unsorted List
MD.5.1.1	Concept	Concept name	Single Value
MD.5.1.2	Level	Level of knowledge required	Single Value
MD.5.2	Reporting	Reporting classification	Single Instance
MD.5.2.1	Resource Type	Type of resource (Interview, report ...)	Single Value
MD.5.2.2	Edition	Edition concerned	Single Value
MD.5.2.3	City	City concerned	Single Value
MD.6	Rights	Use conditions of the resource	Unique Instance
MD.6.1	Copyrights	Copyrights or licenses	Single Value
MD.6.2	Access	Access restrictions	Single Value
MD.6.3	Cost	Cost needed	Single Value
MD.6.4	Publisher	Publisher identity	Single Value
MD.6.5	Remarks	Usage remarks	Single Value

Table 1. The metadata schema

Now, according to the special report model and the metadata schema, we manage versioning features as follows :

- The I-PublishedReport does not have versioning purposes, because if the author changes the main I-SubReport, we create a new I-PublishedReport.
- An I-Component: This exists only inside an I-Structure and it works as a small information retrieval service. We do not have versioning for this kind of element, because the internal information retrieval features are not updated. If needed, a new I-Component is created. But relationships between I-Components can change, so we add a validity period (cf. fig. 3,

R4 (m-n)) to relations. This property corresponds to the list of the I-Structure's version in which this relation is valid.

- The I-Structure is composed of a set of I-Components and a link to the first I-Component which is the root of the I-Structure. Two changes can be applied to an I-Structure. First, the root can be replaced, and next the set of I-Components can be updated. An I-Component can only be added. Indeed, an I-Component may be not relevant on a new version, but has to be present for a previous one. For each new version we have to increase the set of I-Components and to give the validity period of the root (cf. fig. 3 *(k-n)*).
- The I-SubReport is composed of a set of I-Structures and a data collection (in our case, this is a list of articles selected by the author). Between two versions of I-SubReport, these two sets can be updated (inserts). For each new version we store the new sets.

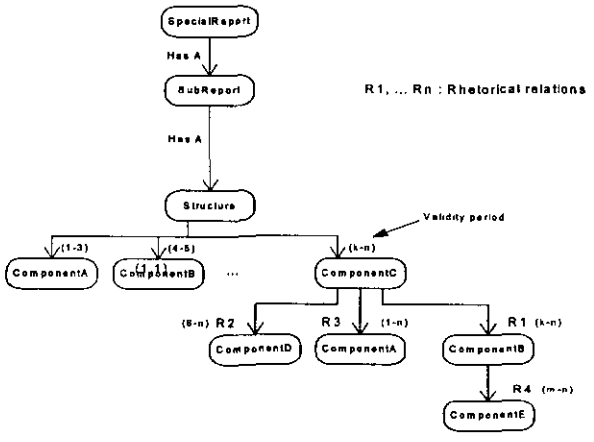


Fig. 3. Versioning application on an instance of the special report model

For each new version, we instantiate a new element of the special report model. We associate the metadata information related to the "Lifecycle part". This will enable us to use information retrieval methods to find the relevant version. Instead of managing versions via numbers like many versioning systems, we manage version via the date. In Press Institutions, especially daily local newspapers, several editions are printed every day according to different geographical areas. It is therefore more relevant for a reader to ask for the special report of a particular date, than the third version of the special report. According to this versioning management, retrieving the correct version is in fact a filtering process.

History and revisiting

As well as the metadata schema useful for versioning features, the user model is necessary for history and revisiting issues. The current version of the user model (Table 2) is an individual model because it deals with individual features, unlike

stereotype models which are about user class features. The user model consists of five parts: personal, preferences, knowledge, history and session.

UM.1	Personal data	Personal data concerning the user	Unique Instance
UM.1.1	Identity	His identity	Single Instance
UM.1.1.1	Last name	His last name	Single Value
UM.1.1.2	First name	His first name	Single Value
UM.1.1.3	Age	His age	Single Value
UM.1.2	Login	Unique identifying data	Single Instance
UM.1.2.1	Login name	The login name	Single Value
UM.1.2.2	Password	The corresponding password	Single Value
UM.1.3	Classification	Classification data	Single Instance
UM.1.3.1	Location	Where does he live?	Single Value
UM.1.3.2	Professional Activity	What kind of job? (economist, fisherman, student, etc.)	Single Value
UM.1.3.3	Role	The role in the application (author, reader ...)	Single Value
UM.2	Preferences data	Data about the preference of the user	Unique Instance
UM.2.1	Interest	Topics of interests	Single Instance
UM.2.1.2	Topic	A list of topics	Unsorted List
UM.2.2	Adaptation	Adaptation preferences	Unsorted List
UM.2.2.1	Element	An element which can be adapted (link ...)	Single Value
UM.2.2.2	Rule	A method of adaptation (annotation ...)	Single Value
UM.3	Knowledge	Data about the knowledge of the user	Unique Instance
UM.3.1	Domain	Knowledge about the domain	Unsorted List
UM.3.1.1	Element	A domain concept	Single Value
UM.3.1.2	Level	A level of knowledge	Single Value
UM.4	History data	Data about access to special reports	Unique Instance
UM.4.1	Access	Access log	Unsorted List
UM.4.1.1	Report ID	Which special report have been accessed	Single Value
UM.4.1.2	Structure ID	Structures used	Single Value
UM.4.1.3	Component ID	Component read	Single Value
UM.4.1.4	Date	Date of access	Single Value
UM.4.2	Bookmark	Bookmark data	Unsorted List
UM.4.2.1	Report ID	Identifier of the special report	Single Value
UM.4.2.2	Structure ID	Identifier of the structure in the special report	Single Value
UM.4.2.3	Component ID	Identifier of the component in the structure	Single Value
UM.4.2.4	Date	Date of the storage, for retrieving the relevant version	Single Value
UM.5	Session data	Data concerning the current session	Unique Instance
UM.5.1	Choice	Stores the first choices of the user	Single Instance
UM.5.1.1	Special Report ID	The Special Report chosen	Single Value
UM.5.1.2	Structure ID	The structure chosen	Single Value
UM.5.2	Current	Stores current data about the reading path	Single Instance
UM.5.2.1	Special Report ID	The current Special Report	Single Value
UM.5.2.2	Structure ID	The current structure	Single Value
UM.5.2.3	Fragment ID	The current fragment	Single Value
UM.5.3	Device	Description of the current device	Single Instance
UM.5.3.1	Software	Software names and version	Unsorted List
UM.5.3.2	Hardware	Hardware description	Unsorted List

Table 2. The user model

Comments: Personal: The geographical area is relevant for a local daily newspaper, and the professional activity (economist, fisherman, student, etc.) will be used to

provide some stereotypes of needs in the future; **Knowledge**: The estimated level of knowledge about some domain concepts. At present, it is given by the reader; **History:** It stores all features required to retrieve the last special report(s). Then, the semantic composition engine will be able to deal with revisiting and versioning.
We plan to add two new entries about readers' Behavior and Background.

We have to store all data needed to recreate a web page. According to our special report model, these data are: the I-PublishedReport ID, the I-Structure ID, the I-Component ID and the date for retrieving the relevant version. The I-SubReport ID is not necessary because we can obtain it via the date and the I-PublishedReport ID. Concerning the versioning issue, the I-PublishedReport ID and the date chosen by the user are sufficient to recreate the same special report. Then we will store the user's bookmark in the appropriate section of his individual model. Concerning history purposes, these are necessary for two reasons. First, the system has to propose navigational guides to the reader. In this case, it has to show the path covered by the reader. Next, it is interesting to be able to show, on another visit, the element already visited or elements newly added since the last version. In the first case, we have to store the path exactly, that is to say, an ordered list of elements where something can be found more than one time. In the second case, we just need a list of elements without duplicates, but with the date of the last visit. This date is very useful in order to re-open a document in the same configuration the user left it. We will use in the user model, the history section, and the current session.

Adaptation

Our semantic composition architecture is composed of three engines. Each one is able to offer different types of adaptation to readers.

1. **Semantic Composition:** at this level, the engine adapts the special report structure and the content to the user. By means of the authoring tool, the author is able to select the I-Component content. The authoring tool chooses some metadata entries for specifying this content, at least the classification section (5.1.1 concept name, 5.2.1 resource type) implicitly. Moreover, it has to ensure the consistency of the special report. In other words, an article may be referred in several I-Components according to the author, but not elsewhere. An I-Component may have several contents which are variants due to knowledge level, technical requirements or versioning. As soon as a new article is included in a special report, the authoring tool has to check if another specification matches this article. In this case, more metadata entries has to be added in the specification. We shall have to learn what are the relevant implicit metadata entries in order to help the authors. The composition engine can match the following user model entries: topics of interests and knowledge against the classification entries in metadata. If there are not compatible with, the I-Component is deleted from the I-Structure. The resulting structure with the corresponding content becomes the user adapted special report.
2. **Logical Composition:** the logical engine may select the relevant template according to the resource type, the user task – for instance reading -, the device, etc. for computing the next XML web page. The templates will have metadata

entries which will be used to select the most relevant one. They will be indexed with user's tasks, type of fragments, user's category, user's level of expertise, etc.The composition engine determines the XML page content and the corresponding navigation tools. Hyperlink annotation will be defined at this level by means of properties associated to hyperlinks - Xlink.

3. **Physical Composition:** the layout engine selects a style sheet, at least, according to the template. This style sheet has to manage adaptive navigation according to user's preferences (annotation, hiding, direct guidance).

Perspectives

In this paper, we have presented our framework which consists in delivering adaptive special report according to an author oriented viewpoint. Authors have know-how which enables them to choose document contents and to organize them in one or more consistent ways by means of narrative structures. Authors can share and reuse these narrative structures. A particular knowledge elicitation method is used to formalize this knowledge because they are unable to explicit this knowledge. This method relies on theories and methods stemming from cognitive psychology and psycholinguistics. The reusability of this knowledge leads to Knowledge management. It aims to exploit an organization's intellectual assets for greater productivity, new value, and increased competitiveness[2].

We have proposed a semantic composition engine which delivers a user adapted special report by means of a user model and metadata. This composition engine is also studied for another European project called CANDLE which concerns with distance learning. In this paper, we have considered the management of revisiting and versioning because of the non persistent state of virtual documents. Our system has to ensure that a dynamic document can be recreated every times in the same state for a particular reader.

We plan to offer a kind of free browsing mode which will use a narrative structure as a guide. In other words, the intelligent search engine will not be limited to the information space dedicated to the special report. Indeed, the content specification of each I-Component will be applied to the entire database. A reader will be able to access all articles fitting the different content specifications and then to get articles closely related the current I-Component.

In the future, readers' strategies will be managed by adding enough metadata to compute their structures on the fly according to user's goals. For instance, a clustering process can be applied to geographical (or temporal) criteria present in the metadata and according to a domain ontology. For a local newspaper, which has several issues organized by editions (geographical areas), readers use this criteria for information retrieval from its website.

[2] http://www.aifb.uni-karlsruhe.de/WBS/ontoknowledge/

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. Berners-lee, T., Weaving the Web. 1999, San Francisco: Harper.
4. Decker, S., et al., Knowledge Representation on the Web. 2000, On to Knowledge Project: http://www.ontoknowledge.org/oil/papers.shtml.
5. Decker, S., et al. Ontobroker: Ontology Based Access to Distributed and Semi-Structured Information. in Conference on Database Semantics. 1999. Rotorua, New Zealand: Kluwer Academic Publishers.
6. Ardissono, L., L. Console, and I. Torre. Exploiting user models for personalizing news presentations. in 2nd Workshop on Adaptive Systems and User Modeling on the WWW, AH'99 and UM'99. 1999: Einhdoven University of Technology, Computer Science Reports.
7. Domingue, J. and E. Motta. A Knowledge-Based News Server Supporting Ontology-Driven Story Enrichment and Knowledge Retrieval. in 11th European Workshop on Knowledge Acquistion, Modelling, and Management (EKAW '99). 1999.
8. Mann, W.C. and S.A. Thompson, Rhetorical Structure Theory: Toward a functional theory of text organization. Text, 1988. 8(3): p. 243-281.
9. Christophides, V., Electronic Document Management Systems. 1998, UCH/FORTH: http://www.ics.forth.gr/~christop/.
10. Decker, S., et al., The Semantic Web - on the respective Roles of XML and RDF. 1999. http://www.ontoknowledge.org/oil.
11. Bray, T., et al., Extensible Markup language (XML) 1.0, (Second Edition). 1998, W3C: http://www.w3.org/TR/2000/REC-xml-20001006.
12. Adler, S., et al., Extensible Stylesheet Language (XSL) Version 1.0. 2000, W3C: http://www.w3.org/TR/xsl/.
13. Fensel, D., et al. On2broker in a Nutshell. in the 8th World Wide Web Conference. 1999. Toronto.
14. Fensel, D., et al. On2broker: Semantic-Based Access to Information Sources at the WWW. in World Conference on the WWW and Internet, WebNet 99. 1999. Honolulu, Hawai, USA.
15. Vitali, F. Versioning hypermedia. in ACM Computing Surveys. 1999.

Tailoring the recommendation of tourist information to heterogeneous user groups

L. Ardissono, A. Goy, G. Petrone, M. Segnan and P. Torasso

Dip. Informatica, Università di Torino, Corso Svizzera 185, I-10149 Torino, Italy
e-mail: {liliana,goy,giovanna,marino,torasso}@di.unito.it

Abstract. This paper describes the recommendation techniques exploited in INTRIGUE (INteractive TouRist Information GUidE), an adaptive recommender system that supports the organization of guided tours. This system recommends the places to visit by taking into account the characteristics of the group of participants and addressing the possibly conflicting preferences within the group. A group model is exploited to separately manage the preferences of heterogeneous subgroups of people and combine them, in order to identify solutions satisfactory for the group as a whole.

1 Introduction

Web-based information systems have become very popular tools to retrieve specialized information. In particular, the provision of tourist information is extremely appealing, as it supports the search for up-to-date information about services and attractions, without relying on books or travel agencies. The development of Web-based tourist guides is however challenged by the variety of user needs to be satisfied during the presentation of the information. Users are typically interested in different types of attractions (pieces of art, scientific attractions, natural parks, etc.). Moreover, most people do not travel alone, so that possibly conflicting requirements have to be taken into account when recommending the places to visit. Therefore, the presentation of tourist information requires personalized travel guides, satisfying individual information needs; e.g., see [13, 18, 19, 16].

In this paper, we present INTRIGUE (INteractive TouRist Information GUidE), a Web-based adaptive system that provides information about tourist attractions and services, such as accommodation and food. In the following presentation, we focus on the techniques for the generation of the personalized recommendations tailored to the preferences of the group of participants.

INTRIGUE supports the user in a combined search for tourist attractions, based on orthogonal criteria, such as category-based and geographical search. Moreover, the system dynamically generates multilingual presentations, by exploiting efficient template-based NL generation techniques. Our current prototype presents information about the city of Torino and the surrounding Piedmont area, in Italian and in English. The system assists the user in the organization

of a tour by providing personalized recommendations about attractions and services. Moreover, it offers an interactive agenda that supports the scheduling of the tour, by considering both the user's needs and the constraints concerning the opening hours, the average visit time for the selected places, and so forth.

As far as the recommendation functionality is concerned, INTRIGUE deals with a structured model of the group of people traveling together, to manage the possibly conflicting preferences of the subgroups. This approach supports alternative recommendation criteria, which the user can select to receive suggestions customized according to specific viewpoints. For instance, the system supports the suggestion of a solution that satisfies all the participants in a more or less a uniform way. However, also the search for solutions focused on the preferences of specific subgroups is available (in our prototype, we consider children and disabled people). In this way, the user can ask for suggestions focused on particular perspectives. As noticed in [7], an essential feature for an intelligent system is the explanation of the reasons for its own suggestions. This aspect becomes even more crucial for group recommendations, where there is no immediate correspondence between the user's preferences and the system's decisions. For this reason, we have developed an explanation technique which supports a clarification of the evaluation strategies adopted by the system in the recommendation. This technique enables the system to specify which properties are most suitable for the characteristics of the various subgroups, therefore helping the user to select the items to include into the agenda in a very informed way.

The paper is organized as follows: Section 2 describes the structure and the management of the group model; section 3 describes the recommendation criteria used in the system to provide users with personalized suggestions tailored to a possibly heterogeneous tourist group. Section 4 describes some related work and section 5 closes the paper, outlining some future work.

2 Management of group models in INTRIGUE

2.1 User groups

The management of heterogeneous groups having different and potentially conflicting preferences is essential to enhance the recommendation task in several application domains, other than the tourism one. For instance, consider the management of personalized television services and the suggestion of items to purchase in a group environment.

In some cases, group preferences have been managed by exploiting stereotypical models that describe the preferences of the group as a whole. For instance, in the TV domain, family models have been used to customize Electronic Program Guides to the standard preferences of groups formed by adults and children. Although this approach suits the applications having a small number of typical user groups to be considered, it is not flexible enough to manage the cases where the groups can be highly heterogeneous and there are many possible combinations of user classes. Tourist services represent an interesting example of this case,

because people having very different preferences and requirements may join the same tour, which should be organized by taking into account the interests of all the participants. To this extent, a different group model is needed, in order to avoid a combinatorial explosion in the number of models to be described. In particular, the management of the user group should be considered as orthogonal with respect to the management of different user classes.

Ideally, a group model could be managed as the integration of one individual user model for each member of the group. However, this approach would not scale up when large groups are considered. Moreover, to address individual preferences, the system would need individual descriptions of the group members, therefore imposing strong overhead on the user interacting with the system.

We adopt a different approach, which overcomes the drawbacks of the previous ones. We consider a group as a set of people that can be partitioned into a limited number of subgroups.[1] Each subgroup is modeled as a class of users having similar preferences and the preferences of the whole group can be inferred by combining the preferences of the its subgroups. Moreover, an influence on the system's recommendation is evaluated for each subgroup, in order to support recommendation strategies tailored to the preferences of specific subgroups. A subgroup could be particularly influent either because it represents a very significative portion of the tourist group, or because its members belong to a class having special needs (e.g., children and disabled people).

The types of (sub)group to be considered and their relevance are domain dependent and have to be defined on the basis of the personalization requirements to be addressed. For instance, in a tourism domain, groups with special needs, such as children and disabled, could be given maximum relevance in order to take their needs in particular account. However, other groups could be considered; for example, animals could be "members" of the group, with specific preferences, e.g., the one for places where they are accepted. In the next sections, we describe the management of group models adopted in our system.

2.2 Structure of the group model

INTRIGUE exploits a structured group model, where uniform subgroups are represented as distinct entities. A subgroup model is associated to each *homogeneous subgroup* of people planning the tour together. Each subgroup model is structured in three portions:

- The "Characteristics" section provides information about the characteristics of the participants, acquired by the system by questioning the user via a registration form. For instance, Figure 1 represents a subgroup of people aged between 46 and 55, with a human science background, full mobility capabilities, partial vision capabilities and interested in arts and history.
- The "Preferences" portion specifies the system's predictions for the subgroup preferences. Each preference is represented as a slot and includes:

[1] In the simplest case, the group is formed by homogeneous people and consists of a single subgroup.

Characteristics:
Age: 46-55;
Background: human_science;
Mobility: full;
Vision: partial;
Interest_for_arts: yes;
Interest_for_history: yes;
Interest_for_science: no;
Preferences:
Special_transportation_systems:
Importance: 0; *Values*: missing: 0.3; some: 0.4; present: 0.3;
Special_facilities_for_vision:
Importance: 0.8; *Values*: missing: 0.1; some: 0.4; present: 0.5;
Historical_value:
Importance: 1; *Values*: low: 0.05; medium: 0.1; high: 0.85;
Artistical_value:
Importance: 1; *Values*: low: 0.05; medium: 0.35; high: 0.6;
Scientific_value:
Importance: 0.5; *Values*: low: 0.3; medium: 0.4; high: 0.3;
...
Group Information:
Cardinality: 5;
Relevance: 0.4

Fig. 1. An example subgroup model for the tourism domain.

- An "Importance" facet, which specifies the importance of the preference to the subgroup. For instance, the travelers represented by the model shown in Figure 1 have no interest for special transportation systems (the importance is 0). In contrast, their interest for the historical value of tourist attractions is extremely strong (importance = 1).
- A probability distribution over the values of the preference. For instance, the described tourists very likely prefer attractions having high historical value ("high" is dominant in the distribution), while the probability that they prefer attractions with low historical value is almost null.

– The "Group Information" section stores general information about the subgroup. Each subgroup has a *cardinality*, specifying the number of people forming it, and an *relevance*, representing an estimate of the weight that the preferences of a prototypical member of the subgroup should have on the selection of tourist attractions to be recommended. The relevance ranges from 0 (null relevance), to 1 (maximum one). In our example, the subgroup is formed by 5 people and has a medium relevance (0.4).

2.3 Knowledge about user classes

We manage the presence of users characterized by different preferences and requirements by exploiting stereotypical information that describes the charac-

teristics of the various user classes. The tourist population can be segmented according to different perspectives; for instance, we can consider their interests for tourist attractions, their knowledge about arts and other topics, or their mobility and vision capabilities. These perspectives have influence on different sets of preferences, which the system can exploit to evaluate a tourist attraction. For our prototype, we have specified the following perspectives, each one including a set of stereotypes that represent the tourist classes described from that viewpoint:

- "Age_range" perspective: the traveler population is segmented by ranges of age. The main goal is to distinguish children from adults and to model interests for special activities (such as playing) accordingly. The only relevant characteristic is "age", but the stereotypes predict specific interests for activities and types of documentation about tourist places.
- "Interests" perspective: in this cluster of stereotypes, the tourists' interests are modeled. People are segmented into groups characterized by different educational backgrounds (e.g., historical, technical, etc.), also depending on the explicit interests declared in the registration form. This cluster makes predictions on the preferences for different types of tourist attractions: e.g., some places may have a noticeable value from the historical point of view, others may excel in scientific or technological aspects.
- "Mobility_capabilities" perspective: the population is segmented to characterize different mobility capabilities. The preferences concern the reachability of places and the availability of special transportation systems.
- "Vision_capabilities" perspective: this segmentation concerns the travelers' sight and makes it possible to describe the preferences of people having full, partial, or null vision capabilities.

2.4 Representation of user classes

The stereotypical information is stored in a knowledge base, where it is organized as a set of clusters, each one associated to a different perspective: e.g., age, mobility and vision capabilities. A cluster contains a list of stereotypes, representing the classes of tourists forming the partition: e.g., the "vision_capabilities" cluster includes three stereotypes, representing the people having complete, partial and null vision capabilities. Similar to the representation defined for the SeTA system [3], a stereotype includes a set of classification data, describing characteristics of travelers belonging to the represented class, and a set of preferences, describing the typical requirements of such people for properties of the tourist attractions. For instance, the stereotype describing people with null vision capabilities has only one significant classification data, i.e., the "vision", which is a trigger for the stereotype. Moreover, the main predicted preference concerns the availability of vocal presentation devices.

As clusters represent different viewpoints for describing people, their stereotypes may be based on different classification data, although some data may be exploited by more than one cluster. Moreover, the stereotypes belonging to different clusters make predictions on distinct sets of preferences (those significant

PRIMARY-SCHOOL
Classification data:
Age: up_to_5: 0.0; 6-11: 0.1; 12-14: 0.0; ...; 46-60: 0.0; more_than_60: 0.0;
Preferences:
Play_activities:
Importance: 1; *Values*: null: 0.0; low: 0.05; medium: 0.45; high: 0.5;
Reading_material:
Importance: 0.8; *Values*: null: 0.05; introductory: 0.2; specialized: 0.05; scholastical: 0.7;
Length_of_visit:
Importance: 1; *Values*: short: 0.6; medium: 0.35; long: 0.05;
Background_knowledge:
Importance: 1; *Values*: low: 0.9; medium: 0.1; high: 0.0;
...
Relevance: 1;

Fig. 2. Stereotype describing primary-school children.

from the described viewpoint). For instance, the stereotype described in Figure 2 belongs to the Age_range cluster and describes children aged from 6 to 11, i.e., studying at primary school.

The stereotypes also specify the relevance of the represented group ("Relevance" slot). The relevance predicted by a stereotype S represents the weight that the preferences of a prototypical tourist belonging to S should have on the selection of tourist attractions. This parameter ranges in [0..1], where 0 denotes null relevance and 1 represents the maximum relevance. In order to take into account the fact that some subgroups, such as children, have strong requirements on the organization of a tour, the related stereotypes predict a relevance equal to 1 (e.g., see the "PRIMARY-SCHOOL" stereotype in Figure 2). Instead, most of the other stereotypes have an medium or low relevance.

2.5 Management of subgroups within a tourist group

At the beginning of the interaction, the user visiting the Web site is asked how many people are going to travel together. Then, the system asks her to distribute such people into relevant subgroups, on the basis of a set of pre-defined features. Currently, the main user features which we have considered concern the range of age (to deal with children and elderly) and the mobility and vision capabilities; however, the system can be configured to take into account other features, such as social and cultural aspects. For each subgroup, a registration form is displayed, in order to provide the system with information about the interests of the related travelers, the cardinality of the subgroup, and other similar information. The fields of the forms are not mandatory, but the system's suggestions can be more focused, if more information about each subgroup is provided. Each subgroup model is initialized with the preferences of a very generic traveler, corresponding to an adult with average interests and without special requirements.

This initialization enables the system to have a basic description, in case the direct user does not appropriately fill in the forms.

The system initializes each subgroup model by exploiting stereotypical information about tourists, according to the techniques developed for the SeTA system: the subgroup characteristics (provided by the user in a registration form) are matched against the stereotypical information. The stereotypes best matching the characteristics of the subgroup are then used to make predictions on the subgroup preferences and on its importance. Details about these techniques can be found in [3].

3 Generation of recommendations for a tourist group

3.1 Evaluation of tourist attractions

The evaluation of items for a heterogeneous tourist group is achieved in two steps. First, items are separately evaluated and ranked with respect to each subgroup. Then, the subgroup-related rankings are combined to obtain the overall ranking, from the viewpoint of the whole group. In the following, we will focus on the subgroup-related evaluation of items. Then, in section 3.2, we will discuss how the separate rankings are combined to generate the system's recommendations.

Given the preferences of a homogeneous subgroup, items are ranked by exploiting the same techniques used in the SeTA system for the recommendation of products. As such techniques are extensively described in [3], we only sketch them in the following.

Tourist attractions are represented as entities described by features providing different types of information: e.g., geographical information, category, logistic information, and so forth. In the evaluation of a tourist attraction, the system exploits the *properties* og the item. Properties are features that provide a qualitative evaluation of the attraction. For instance, we take into account the historical or artistic value of an attraction, how much background knowledge is required to appreciate it, or whether the place offers play areas for children ("play_activities" in Figure 2).

The degree of matching (henceforth, *satisfaction score*) between an item and a subgroup model is evaluated by analyzing the preferences of the subgroup towards the properties of the item, stored in the subgroup model. Each property is matched against the related preference to establish an individual score.[2] The overall satisfaction score of the item results from the merge of the individual scores of its properties. Two individual scores, X and Y, are combined by exploiting the following formula:

$$SATISFACTION\_SCORE(score_X, score_Y) = score_X * score_Y / (score_X + score_Y - score_X * score_Y)$$

[2] This score is a decimal value in [0..1], where 1 represents perfect compatibility with the preference, while 0 represents total incompatibility. In this evaluation, the importance of the preference in the subgroup model is used to tune the influence of less relevant properties, when they are not compatible with the subgroup preferences.

This formula, described in [2], is additive and therefore supports and incremental evaluation of the overall satisfaction score of an item. The formula takes values in the [0..1] range; moreover, it is particularly selective: being based on the product operator, it returns a 0 satisfaction score for any item having at least one null individual score. This selective power is essential in the tourism domain, as it enables the system to dramatically downgrade the evaluation of items incompatible with basic requirements of the traveler subgroups. For instance, a tourist place without a transportation system suitable for disabled people cannot be recommended as a good solution to a group of tourists with mobility problems.

3.2 Recommendation criteria

INTRIGUE provides the user with alternative recommendation criteria to support her in the selection of the attractions for the tour. The reason for providing different recommendation criteria is that no specific recommendation method can satisfy all the possible requirements. For instance, the user may want to see separate recommendations for each subgroup and compare the lists by themselves. Alternatively, she may prefer to be provided with a single recommendation list, representing a synthesis of the suggestions, in order to avoid the analysis of multiple and possibly long recommendation lists. However, even in this case, different criteria could be applied to generate the list. For instance, items unsuited for at least one subgroup might need to be ignored, although they are interesting options for other groups. Moreover, the recommendations could be fair, trying to satisfy all the subgroups in a uniform way, or they could be biased towards the preferences of the most influent subgroups.

In our system, we have included three recommendation modalities, which the user can choose from by clicking on buttons available in the user interface. See the buttons at the top of Figure 3: "Separate listing by groups", "Unique listing (method 1)", "Unique listing (method 2)". We describe these recommendation criteria referring to a scenario where a user similar to the one described in Figure 1 inspects the civil buildings in Torino. The user is organizing a tour with some children and impaired people. In this case, the system generates three subgroup models: one for the subgroup including the direct user, the others for the two homogeneous subgroups traveling with her.

Separate listing by group. In this modality, the system shows separate lists, one for each subgroup. with items sorted on the basis of the rankings previously evaluated for each subgroup. The best ranked elements for each subgroup are at the top of the lists, while the worst ones are at the bottom.

Figure 3 shows the system's recommendations, reporting the suggestions for the subgroup including the direct user in the first column, for the children in the second one and for the impaired people in the third one. Notice that each item is associated with an icon (stars), which represents the satisfaction score obtained by the item and supports an easy identification of the best items for each group.

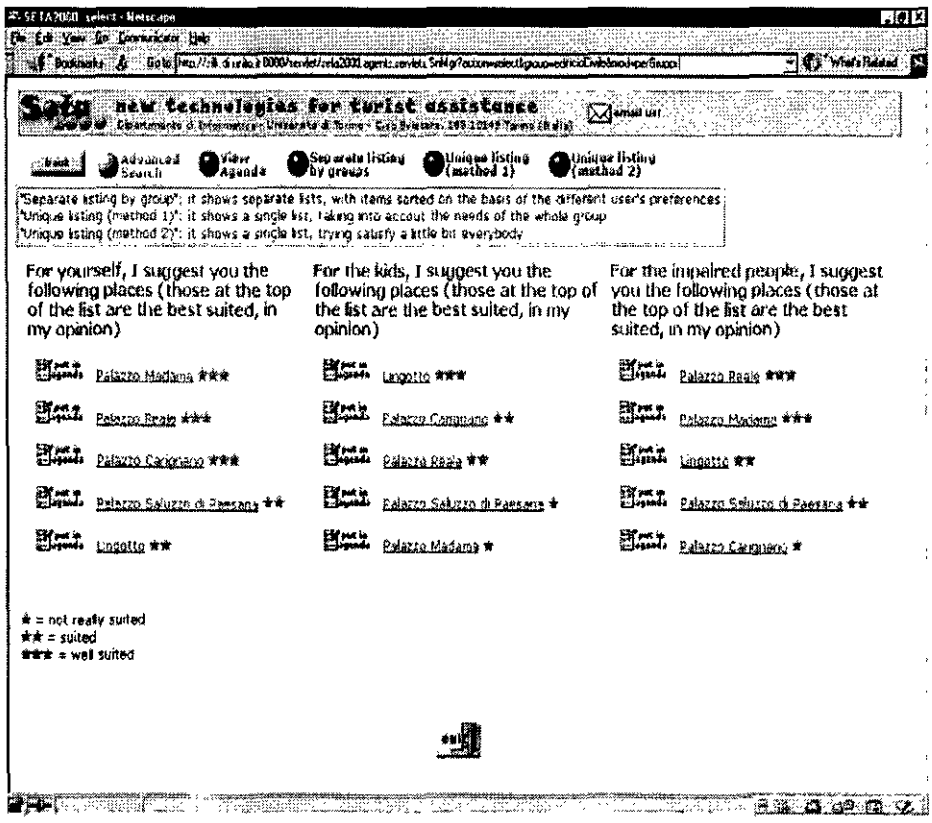


Fig. 3. Separate listing of tourist attractions

This type of recommendation explicitly presents the suggestions for each subgroup and enables the direct user to select the interesting attractions in an informed way, taking into account the ranking of each item in each subgroup-related list. However, it may be confusing if there are too many subgroups to be considered, because the user should compare several permutations of the items, which may result in a certain overhead.

Preferential satisfaction listing (method 1). In this modality, the system displays a single, sorted list of items representing the suggestion for the whole group. The overall ranking of items is obtained by merging all the subgroup-related rankings in a weighted way, depending on the cardinality and the relevance of the various subgroups. Thus, the system's suggestions take into account the presence of large homogeneous subgroups, and that of subgroups with special needs. The overall score S of an item is evaluated by combining the satisfaction scores S_{UM} associated to the item for each subgroup. In our example:

$$S = inf_{direct\_user} * S_{direct\_user} + inf_{children} * S_{children} + inf_{impaired} * S_{impaired}$$

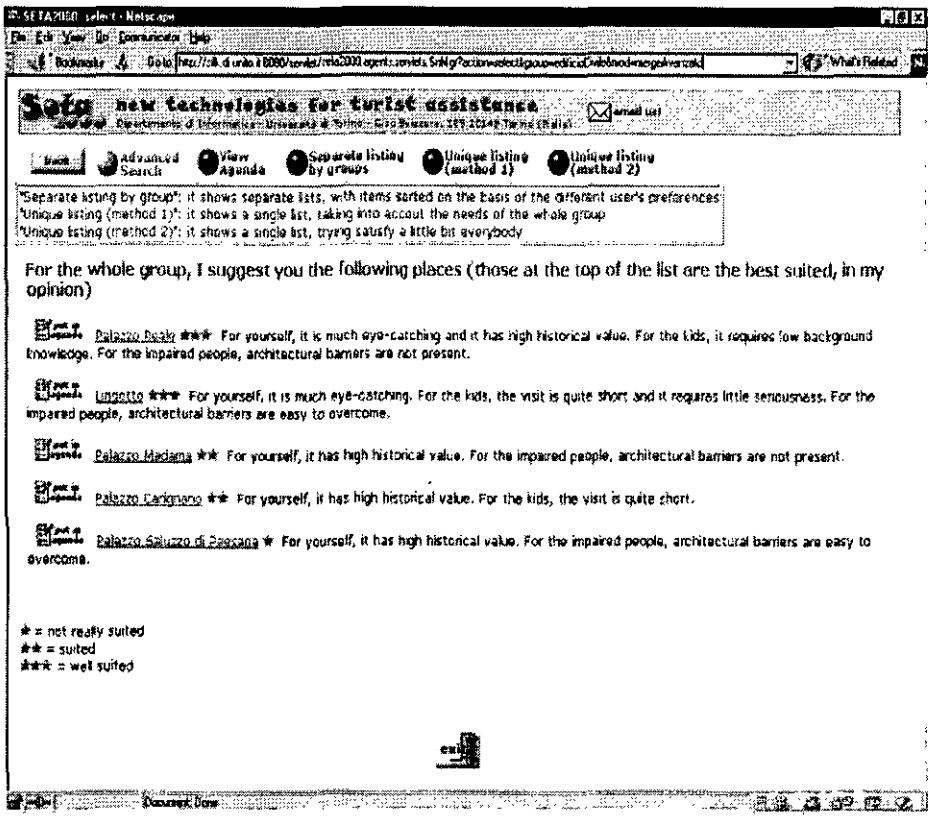


Fig. 4. Unique listing of tourist attractions

For each subgroup, the associated weight (inf_{UM}) represents the influence of the subgroup's preferences within the whole group and is evaluated according to the following formula:

$$inf_{UM} = relevance_{UM} * cardinality_{UM} \ / \ total\_cardinality$$

where $relevance_{UM}$ is the relevance of the subgroup, $cardinality_{UM}$ is the number of its members[3] and $total\_cardinality$ is the total number of tourists forming the overall group. In this way, special subgroups can be privileged in a flexible way, depending on the portion of the overall group they represent.

Figure 4 shows the preferential listing recommendation for the same group of tourists considered in Figure 3. The tourist attractions are sorted according to the overall ranking, resulting from the weighted merge of the individual group rankings. In this case, the stars next to the items represent their overall ranking, related to the whole group of tourists.

This type of recommendation is useful to provide the user with suggestions supporting her in the selection of the items to add to the agenda. In fact, the first

[3] Relevance and cardinality are retrieved from the subgroup model.

items in the list are the best ones for the group considered as a whole, possibly with special attention to the preferences of the most influent subgroups. However, other strategies could be exploited to weight the influence of the subgroups; for example, the group relevance alone could be used, without considering the cardinality, in order to maximize the influence of special subgroups (or, in alternative, the cardinality alone could be used). We are investigating the possibility of providing the direct user with methods to select the parameters to be considered for the evaluation, therefore supporting further personalization forms in the recommendations.

Uniform satisfaction listing (method 2). As an alternative to the preferential satisfaction listing, which sorts tourist attractions in a biased way, the user may be interested in receiving a fair recommendation, where the suggestions uniformly satisfy all the subgroups. To this extent, we have introduced a third modality, where the system shows a single recommendation list whose items are sorted to achieve uniform satisfaction for each subgroup. In this case, the satisfaction score of an item is explicitly used as a *degree of satisfaction* for the related subgroup of tourists: an item with a satisfaction score equal to 1 increases the satisfaction of a subgroup in a maximal way, while an item with a 0 satisfaction score does not modify the satisfaction degree at all. The recommendation list is generated in order to achieve uniform satisfaction for all the subgroups. We do not describe this method in detail, but the idea is the following: at each step, the system selects, out of the set of items to be sorted, the one maximally increasing the satisfaction degree of the subgroup that has received minimum satisfaction in the previous selections. The aim is to raise the group satisfaction degree as much as possible.

3.3 Explanation of the system's suggestions

The explanation capability is desirable for any interactive system. In our case, two types of explanation are particularly important: the first one is the specification of the target of the recommendation. The second one concerns the reasons for suggesting the various items. In INTRIGUE, we have addressed both types of explanation, by supplementing the recommendations with textual descriptions, which explain the main reasons for suggesting an item to a subgroup.

As shown in Figure 4, in the unique listing modalities each item is coupled with a sentence specifying, for each subgroup, the most important properties determining the suggestion of the item.[4] For instance, "Palazzo Reale" (Royal Palace) is a good suggestion for the subgroup including the direct user since it is much eye-catching and has high historical value; moreover it is also good for children because its visit requires low background knowledge. Finally, it suits disabled people because it has no architectural barriers.

[4] The presentation of the most suited properties is not displayed in the separate listing by group due to space constraints on the screen. However, information about items and their properties can be retrieved by asking for their detailed presentation pages.

Notice that the explanations generated by the system for a subgroup do not include the whole list of properties satisfying the preferences in the subgroup model: in fact, they only report the most relevant preferences of the class characterizing the subgroup, in order to produce maximally useful information for the user. For instance, the recommendations for children in Figure 4 are focused on properties such as the background knowledge required to appreciate the attraction, the length of the visit, and so forth. In contrast, they do not mention any property such as being eye-catching because most people (and also children) like eye-catching places, and so this property is not particularly relevant for the specific tourist class addressed in the explanation.

The contextually relevant properties for a subgroup are selected by exploiting the stereotypical information: given the set of properties satisfying the preferences of a subgroup model, the properties to be mentioned are selected by taking into account the importance of the related preferences in the stereotype describing the tourist class characterizing the subgroup. For instance, as shown in Figure 2, children are mostly interested in properties such as the availability of play areas, length of the visit and required background knowledge.

The linguistic form of the explanations are automatically generated by exploiting template-based Natural Language Generation techniques (see [2]), on the basis of a language independent internal representation of the item properties. In this way multilinguality is supported.

4 Related work

The typical tourist information Web sites are static hypertexts and provide non-personalized information about attractions and services available in a town, or in a region. These systems suffer from two major drawbacks: first, they cannot provide users with information focused on specific interests (the only way to search for specific information is typically provided as an embedded search engine). Second, they rely on static descriptions, which become obsolete in a short time and have to be manually revised by the site administrators. Another type of site are the e-travel agencies, like, for instance, Expedia [11]; their main goal is to offer discount airfare, flight, hotel, cars, vacation packages reservations. However, these sites only help the user to gather information or to make reservations and do not help her to organize a trip or a tour of a city.

Some dynamic hypermedia systems have been designed to generate the presentations "on the fly", possibly tailoring contents and styles on the basis of the application of personalization strategies; e.g., see [10,17] and [14] for an overview. For instance, AVANTI [13] was designed as a kiosk system which generates customized presentations of the services and tourist attractions available in a town. The goal was to support alternative interaction media and personalization strategies were exploited to tailor the presentations to the individual user's interests. More recently, intelligent virtual guides have been designed to personalize the visit of a museum, taking into account several factors such as the user's interests, domain expertise, the fact that the user was visiting the place

for the first time, and also the (physical) navigation style within the museum [18, 16]. In particular, there is a strong interest in the development of context-aware applications, supporting a selective presentation of information, based on the physical location of the user [15, 9].

From a related perspective, special attention has been paied to the development of systems supporting the individual user's search for information with personalized recommendations. For instance, see [8, 7, 12]. Finally, some researchers have defined techniques to support the user in the definition of her own search criteria, therefore leading to the configuration of her own information service [6].

5 Discussion

Our work is focused on the provision of multilingual, personalized recommendations for groups of people planning a visit to a given geographical area. Different from on-site kiosks and context-aware applications, the role of our system is in assisting the user to schedule the tour, not in guiding the group during the visit. Therefore, physical context has a marginal role and is exploited only when the schedule of the trip is considered, to estimate, for instance, the appropriate transfer times from one tourist attraction to another.

The main contribution of this paper concerns the management of a *group model*, where the characteristics, interests and preferences of the various components of the group are taken into account to tailor the recommendations in a suitable way. The management of a group model and the group-oriented personalization distinguishes our system from the other recommender systems, which tailor the suggestions to the individual user. In the case of a single user, her preferences may be more or less articulated, but are unique. In contrast, a group of people traveling together may have conflicting preferences (and needs) and the generation of a recommendation which addresses the requirements of all such people is much more complex. In order to address this issue, we have considered the group as composed of subgroups, having homogeneous preferences and needs. Moreover, we have designed different recommendation criteria, which the user can select to get different types of suggestion from the system. These criteria include the separate ranking for subgroups, a uniform merge of the preferences of all the homogeneous subgroups, and a weighted recommendation, where the subgroups have different influence on the system's rankings. Another important aspect is the capability to explain why a recommendation has been made: when the system presents the lists of tourist places to be visited, a sentence specifying the most relevant properties determining the suggestion is generated.

All the Web pages of the INTRIGUE user interface are dynamically generated, accordingly to the following steps: first, the information to be displayed is selected; then, the linguistic descriptions are generated, by exploiting a template-based Natural Language Generator (see [2]) and an XML object, representing the "content" of the page, is produced (see [5]). Finally, the XML object is transformed into a HTML page to be interpreted by a standard browser, by exploiting XSL transformations. Our goal is to have a representation of the personalized

content of each page independent from the actual user interface implemented by the system. For instance, the XML object could be fed to a different module, that generates a user interface for a different medium (e.g. a mobile), or stores the personalized content in a database for further processing.

We have not yet focused on the adaptive presentation of tourist attractions. However, the generation module could be easily extended with the techniques developed for the SeTA system, in order select the features of an attraction to be presented, on the basis of the user's interests [2, 3]. We are also working on improving the scheduling facility to realize a virtual "travel-agent" able to help the user to organize her day tour, including hotel and meal arrangements.

INTRIGUE is based on the multi-agent architecture of SeTA, instantiated on the tourism domain and updated with the introduction of the interactive agenda agent. Details about this architecture can be found in [1] and [4].

As our system is still under development, no evaluation has been carried on up to now. However, we plan to test the recommendation and explanation functionalities of the system with users in the next future. In particular, we would like to evaluate the system on some typical scenarios, such as a family visiting a town, a tour to be organized for the a group of students or the organization of a spare day for some people visiting a town for business reasons. In all these cases, we will evaluate the recommendation capabilities of the system by comparing the tourist attractions selected by the user with those suggested by the system. Moreover, we will collect, from a selected number of users, individual feedback on the usefulness and effectiveness of the system's alternative recommendation criteria and explanation capabilities.

References

1. L. Ardissono, C. Barbero, A. Goy, and G. Petrone. An agent architecture for personalized web stores. In *Proc. 3rd Int. Conf. on Autonomous Agents (Agents '99)*, pages 182–189, Seattle, WA, 1999.
2. L. Ardissono and A. Goy. Dynamic generation of adaptive Web catalogs. In *Lecture Notes in Computer Science n. 1892: Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 5–16. Springer Verlag, Berlin, 2000.
3. L. Ardissono and A. Goy. Tailoring the interaction with users in web stores. *User Modeling and User-Adapted Interaction*, 10(4):251–303, 2000.
4. L. Ardissono, A. Goy, G. Petrone, and M. Segnan. A software architecture for dynamically generated adaptive web stores. In *Proc. 17th IJCAI*, page to appear, Seattle, WA, 2001.
5. L. Ardissono, A. Goy, G. Petrone, M. Segnan, and P. Torasso. Dynamic generation of personalized tourist information on the web. In *Proc. of 12th ACM Conference on Hypertext and Hypermedia (PERTEXT 2001)*, page to appear, Aarhus, Denmark, 2001.
6. M. Bauer and D. Dengler. InfoBeans - configuration of personalized information assistants. In *Proc. 1999 Int. Conf. on Intelligent User Interfaces (IUI'99)*, pages 153–156, Los Angeles, 1999.
7. D. Billsus and M. Pazzani. A personal news agent that talks, learns and explains. In *Proc. 3rd Int. Conf. on Autonomous Agents (Agents '99)*, pages 268–275, Seattle, WA, 1999.

8. R.D. Burke, K.J. Hammond, and B.C. Young. The FindMe approach to assisted browsing. *IEEE Expert*, pages 32–39, 1997.
9. K. Cheverest, N. Davies, K. Mitchell, and P. Smith. Providing tailored (context-aware) information to city visitors. In *Proc. International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2000)*, pages 73–85, Trento, Italy, 2000.
10. R. Dale, S.J. Green, M. Milosavljevic, and C. Paris. Dynamic document delivery: Generating natural language texts on demand. In *Proc. 9th Int. Conf. and Workshop on Database and Expert Systems Applications (DEXA'98)*, Vienna, 1998.
11. Expedia.com. Expedia. http://www.expedia.com, 2001.
12. J. Fink and A. Kobsa. A review and analysis of commercial user modeling servers for personalization on the World Wide Web. *User Modeling and User-Adapted Interaction, Special Issue on Deployed User Modeling*, 10(2-3):209–249, 2000.
13. J. Fink, A. Kobsa, and A. Nill. Adaptable and adaptive information for all users, including disabled and elderly people. *New review of Hypermedia and Multimedia*, 4:163–188, 1999.
14. A. Kobsa, J. Koenemann, and W. Pohl. Personalized hypermedia presentation techniques for improving online customer relationships. *The Knowledge Engineering Review*, page to appear, 2001.
15. P. Marti, A. Rizzo, L. Petroni, G. Tozzi, and M. Diligenti. Adapting the museum: a non-intrusive user modeling approach. In *Proc. 7th Int. Conf. on User Modeling*, pages 311–313, Banff, Canada, 1999.
16. L. Marucci and F. Paternò. Logical dimensions for the information provided by a virtual guide. In *Proc. International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (AH 2000)*, pages 359–362, Trento, Italy, 2000.
17. M. Milosavljevic, R. Dale, S.J. Green, C. Paris, and S. Williams. Virtual museums on the information superhighway: Prospects and potholes. In *Proc. In Proc. Annual Conference of the International Committee for Documentation of the International Council of Museums (CIDOC'98)*, Melbourne, 1998.
18. D. Petrelli, A.De Angeli, and G. Convertino. A user centered approach to user modelling. In *Proc. 7th Int. Conf. on User Modeling*, pages 255–264, Banff, Canada, 1999.
19. R. Wilkinson, S. Lu, F. Paradis, C. Paris, S. Wan, and Mi Wu. Generating personal travel guides from discourse plans. In *Lecture Notes in Computer Science n. 1892: Adaptive Hypermedia and Adaptive Web-Based Systems*, pages 392–85. Springer Verlag, Berlin, 2000.

Application of ART2 Networks and Self-Organizing Maps to Collaborative Filtering

Guntram Graef and Christian Schaefer

Telecooperation Office (TecO), University of Karlsruhe, Vincenz-Priessnitz Str. 1, 76131 Karlsruhe, Germany, Tel.: +49 (721) 6902-89, Fax: -16,
E-Mail: {graef}@teco.edu

Abstract. Since the World Wide Web has become widespread, more and more applications exist that are suitable for the application of social information filtering techniques. In collaborative filtering, preferences of a user are estimated through mining data available about the whole user population, implicitly exploiting analogies between users that show similar characteristics. These preferences are then normally used to filter content or functionality of an application. Two important factors for the quality of the filtering process are the number of users and the amount of information (such as observed behaviors) available about each user. Another factor is the number of objects in the pool of the application that can be considered during the filtering process. Today in most cases memory based approaches to collaborative filtering are used. Unfortunately with O(#users * #items) those do not scale well. Therefore we implemented a model based approach using two different types of neural networks and benchmarked them against a widely used memory based approach. Especially with ART2 networks we obtained some encouraging results.

1 Introduction

The World Wide Web has been established as a major platform for information and application delivery. The amount of content and functionality available often exceeds the cognitive capacity of users. This problem has also been characterized as information overload [15].

Various approaches exist that address this issue, such as search engines [7], web catalogs or filtering techniques based on user profiles, such as collaborative filtering [21].

In collaborative filtering, user profiles are generated that describe user preferences in relation to items within a specific domain. Depending on the application, items can e.g. be Web resources, components [14], services [12], or products [1]. Initial knowledge about user preferences can be obtained either explicitly such as from ratings by users [21] or implicitly through behaviour analysis [19] [13]. For each user a vector is generated with one entry for each known item. The profile vectors are then used as input for either a memory based or a model based method to compute item recommendations by exploiting information stored in profiles that show similarities to a given profile. An often used memory based method is the *Mean Squared*

Differences Algorithm [21]. As illustrated in figure 1, a given profile D is compared to profiles of other users to find the n nearest neighbours i.e. the n most similar profiles that are not equal to D. For this purpose a vector distance metric is employed. In the example, n=2 and thus profile A and C are selected. A target vector is then computed as the average of the neighbour vectors. Depending on the type of the application, the target vector is used to recommend items or to find estimates for blank parts of the original user profile.

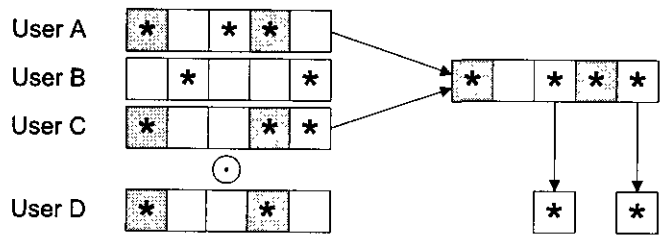


Fig. 1. Memory based collaborative filtering

In model based approaches such as [10] [6], all available profile vectors are first learned by a model. Later single profile vectors can be applied to the model to either obtain a target vector or directly receive recommendations. Figure 2 illustrates this.

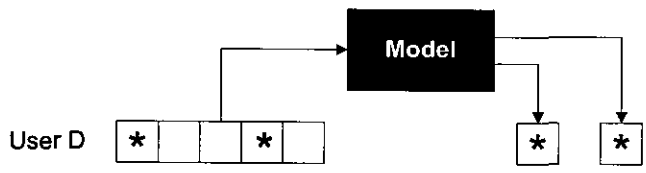


Fig. 2. Model based collaborative filtering

A model usually is smaller in size than the whole set of profile data and no any-to-any matching of profiles is necessary which often leads to performance problems in memory based approaches. On the other hand, an abstraction is performed that usually leads to an information loss and the adaptability of the model to changes to a profile is an issue.

Some applications of collaborative filtering can be found in [21], [17], [22] and [14].

In the first part of this contribution we describe two memory based approaches to collaborative filtering using self-organizing maps (SOMs) [16], and Adaptive Resonance Theory (ART) networks [9]. In the second part of the paper we perform an evaluation with two sets of test data from real world applications and compare the two approaches with a widely used memory based approach.

2 Using Neural Networks for Collaborative Filtering

A large variety of neural networks have been described [20]. Neurons are modeled after nerve-cells in animals. Quite popular is the McCulloch-Pitts Neuron [8] shown in figure 3 (left). The activation of a neuron j is computed by comparing a threshold value θ_j to the weighed input $w_{ij}s_i$. In effect, the neuron performes pattern recognition with the angle between input and weight vector being a measure of conformity. To obtain an adaptive filter this angle is also used during learning, instead of the output value that is used in supervised learning. The weight vector of the neuron is then adjusted towards the input vector as shown in figure 3 (right). This is called unsupervised learning. An important parameter is the learning rate δ that determines how quickly the weight vector is adjusted. Usually a high learning rate is used at the beginning which is then decreased when more input vectors are learned.

Fig. 3. McCulloch-Pitts Neuron (left), Adjustment of a weight vector to an input pattern (right)

Neurons are connected to form networks where the output signal of one neuron is used as input signal for other neurons. Competitive learning can be realized by organizing McCulloch-Pitts Neurons in a layer and presenting an input pattern to each neuron and then only allowing the neuron with the highest activation to produce an output signal. Thus the neuron layer responds to each input pattern with a specific neuron. It independently classifies all patterns into clusters.
Two types of competitive learning are described with the Adaptive-Resonance-Theorie [9] and Self-Organizing Maps [16].

2.2 Self-Organizing Maps

In Self-Organizing Maps (SOMs) [16] not only weights are important but also the location of neurons within the layer. Similar neurons that classify similar patterns are closely located. During the learning process a spacial distribution is created so that neighbour neurons are activated by similar signals. Following the biological example, neurons are arranged in spaces of low dimensionality. This is why the term "map" is used. A topology preserving mapping is attained that maps highly dimensional input patterns to few spacial dimensions. Thus, a strong compression of dimensionality is performed.
Usually, two dimensional maps are used for classification tasks since they allow for greater flexibilty in neighbourship relations than one dimensional maps while allowing for easier computations than maps with more than two dimensions. As

shown in figure 4 (left) for an example of three neurons, each neuron of the map is fully connected to the input layer. The weight vectors are thus of the same dimensionality as the input vectors. The neurons at the border of the map have less neighbours than those in the center. This leads to an undesired preferential treatment of some neurons. To avoid this, the map is projected to the surface of a sphere [3]. In figure 4 (left) the neighbourhood of a neuron is shown in the case of a spherical mapping.

During the learning phase competitive learning is used to determine a winner neuron for each pattern in the training set. But now not merely the winner neuron is adjusted according to the training pattern but also all neighbour neurons. The learning process of a neuron is not independent from other neurons which is why a global learning method must be used [5].

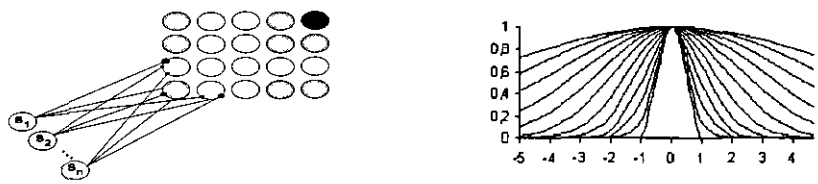


Fig. 4. 2-dimensional SOM with input signal (s{n}), weighted connections showing neighbourhood relations (grey) of a neuron (black)

In each learning step the neuron with the strongest activation by the external signal is determined. If for the McCulloch-Pitts neurons the additional conditions are met that the sum of all weights is constant and the input signal are normed, then the winner neuron k can be determined by applying the euklidian norm.

The weight adjustments of the winner neuron are performed according to the formula that describes the learning rule for competitive learning: $w_j(t+1) = w_j(t) + \delta(t)*(s-w_j(t))$. The adjustment of the neighbouring neurons j depends on their distance d_{jk} from the winner neuron k. The strength of the adjustment is described as a function of the distance. With this function $h_{jk}(t)$ and learning rate $\delta(t)$ the following learning rule is true: $w_{ij}(t+1) = w_{ij}(t)+ \delta(t) * h_{jk}(t) * (s_i - w_{ij}(t))$.

In [16] and in [5] the calculation of the adaptation strength is performed with the Gauss bell: $h_{jk}(t) = -d_{jk} / e^{2\sigma(t)* 2\sigma(t)}$. At the beginning of the learning phase the general structure of the map must develop. After some time smaller details should manifest themselves on the map. Therefore at the beginning even neurons that are relatively distant are adjusted, while later only local adjustments are performed. Figure 4 (right) shows how through the dependance of the Gauss function on the learning duration the neighbourhood is affected less and less by the learning process.

Stability and Plasticity

The application of a SOM to a classification problem is only interesting if there are more patterns than neurons. A neuron must be able to represent more than one pattern of a cluster. If the patterns greatly differ, after each learning step the weight vector is adjusted towards the applied pattern but away from previously learned ones. Thus the neuron "jumps" within the cluster. To avoid this problem and to obtain a stable net,

the learning rate is reduced in order that later changes only slightly modify the weight vector. This way a convergence is forced.
To create a clustering for a set, representative samples must be presented to the map several times during the learning phase. If the net shall be able to learn new patterns even after the learning phase, the learning rate can't be reduced too much and the influence on the neighbourhood during the learning process must stay rather strong. Thus, the ability of the net to be shaped after the learning phase can only be maintained if stable weights are abandoned. A stable net can't be adjusted. This problem is called stability-plasticity problem and has been addressed in [9] with the Adaptive-Resonance-Theorie.

2.3 Adaptive Resonance Theorie

On the SOM the number of neurons is fixed and can't be changed. If during the learning phase a pattern is applied to which no neuron strongly responds, a neuron is selected pretty much by random and adjusted to that pattern. In the worst cases that neuron had already been adjusted optimally to a class of patterns from the training set. By adjusting the weights to the new pattern the weight vector representing that class is changed which results in the classification being unlearned. In that case weights won't stabilize.
The Adaptive-Resoncance-Theorie solves this problem by making the neuron layer adaptive. This means new neurons can be added step by step if a pattern does not match any existing neuron closely enough. Furthermore, a winner neuron can reject the pattern if the similarity is too low. The winner neuron therefore sends its weight vector back to the input layer and only learns the new pattern if it lies within a cone around the weight vector [20]. The size of the cone is determined by the vigilance parameter ρ. Figure 5 (right) shows the attentiveness cone.

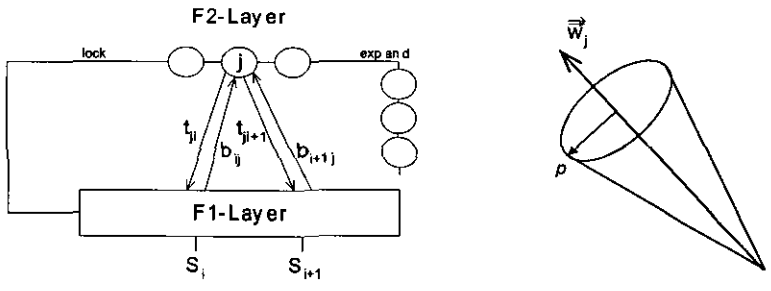


Fig. 5. Set-up of an ART network (left) and attentiveness cone of a weight vector (right)

The net contains two layers F1 and F2. F2 is the competitive layer and consists of a set of ordered neurons that do not consitute any neighbourhood relations. For each input pattern a winner neuron is determined in that layer. The F1 layer controls the classification. If a pattern is not within the attentiveness cone the F1 layer blocks the winner neuron for that pattern. Between the layers weighted connections exist in both directions. Figure 5 (left) shows the set-up.

During the learning process a pattern is applied to the F1 layer. Through the weights that are directed upwards the neuron with the strongest activation is determined by computing the scalar. By applying the weight vector that is directed downwards, the F1 layer checks if the pattern lies within the attentiveness cone. If that is the case, both weight vectors are adjusted to the pattern. Otherwise the F1 layer blocks the winner neuron during the further processing of the pattern and tries to locate another winner neuron. If none of the existing neurons fits, then a new neuron that matches the pattern is added. A detailed description of the F1 layer and the learning process can be found in [11].
The descriped approach makes sure that the net always converges to a stable state and still maintains plasticity. A potential disadvantage is that under certain circumstances the optimal number of classes can be greatly surpassed [5]. Thus the extension of the competitive layer F2 is limited to a maximum number of neurons. After the limit has been reached, no new neurons will be added and rejected patterns must be discarded.
The Adaptive Resonance Theorie exists in two versions. One version is restricted to binary input patterns while the ART2 networks that we selected for our research have been designed to process analog input patterns.

3 Applied Model

In the previous sections two neural networks have been introduced that are able to cluster a training set without supervision and to classify patterns applied after a learning phase. The nets can therefore be used to assign a profile to a class of similar profiles. Information is stored in the weight vectors of the neurons. Besides classification this allows for another application of the nets, as described in the following section.
The weight vectors of the neurons in the competitive layer are also called reference vectors. During each step of the learning phase, the weight vectors are adjusted towards an applied pattern. In the case of competitive learning a weight adjustment is only performed if a similar enough pattern is used. Then the weight vector converges towards the average of all learned patterns. It more or less describes each learned pattern. The weight vector is thus a *"codebook"* for the represented class.
In an ART2 network, the weight vectors that are directed downwards are the reference vectors. For the learning process in an ART2 net two different learning methods can be applied. But only the slower one of the two methods produces a reference vector. In contrast to a SOM the weight vectors are not normed. Still, they converge i.e. the vector norm converges towards a constant value $1/(1-d)$ whereas d is the constant output of a winner neuron. To reach that limit a large number of learning steps is required which is why usually convergence is forgone.
In the SOM the normed weight vector also describes the "learning history" of a neuron. However, in that case during the learning phase the reference vector is also influenced via neighbourhood relations. But since the neighbourhood contains similar neurons this does not necessarily have a negative effect. It is rather assumed that this modification of the reference vector is desirable since it prevents neurons to specialize too strongly on a small number of similar patterns.

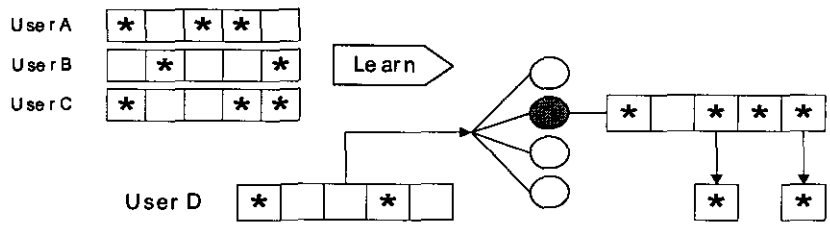


Fig. 6. A neural network as a model for generating predictions. Asterixes mark user interest.

The ability to classify and the formation of reference vectors form the basis for using neural networks as a model in collaborative filtering. Figure 6 shows the set-up.
The first step is to determine the network parameters. The learning rate and the number of learning steps are empirical values and can be chosen similar for every application. In contrast the network size must be estimated individually for different data sets. If the network is too large, neurons specialize too much on a few patterns. Is the network too small the neurons generalize too much and the reference vector becomes useless. As an indicator for estimating the network sizes the number of profiles is used. It is assumed that with a large number of profiles more different behavioral patterns i.e. more independent profiles exist than with a smaller number.
In an ART2 net the similarity parameter must be specified. It can be set automatically during the learning process. A maximum value is used for initialization. If during the learning process a pattern of no neuron can be accepted the parameter is decreased until the pattern can be assigned to a neuron.
After the determination of the profile and the initialization of the weights, the network learns the existing profiles. After the learning process has been completed the network is able to classify profiles and to assign them to a reference vector that can be used as a basis for compiling recommendations.

Learning
From the database a number of representative and expressive profiles are compiled to a training set that is then learned by a neural network. Less expressive profiles are those that are only sparsely populated.
The higher the dimensionality of the input vectors the more difficult it becomes for a neural network to a achieve a useful clustering because the number of independent patterns that do not show similarities increases. For each of those patterns an individual neuron must be learned.
If in addition to this, the training set contains sparsely populated profile vectors, the network must use a large part of its resources or neurons for the classification of vectors with low expressive value. The result is a network that does badly represents the training set. Thus only profiles that contain a minimum number of entries should be selected for training.
Both SOMs and ART networks possess the ability to learn after completion of the initial learning phase. If the profile of a user changes it can be learned again. However, this leads to an advantage for profiles that change frequently over profiles

that seldom change because it is learned more often. In the worst case, other profiles might be unlearned as the weight vector shifts towards the re-learned profile.
The re-learning of changed profiles can be handled best by the SOM because in that case the global structure of the map needn't be modified. A changed profile usually is still similar enough to the original profile to be represented by the same neuron or at least one of its neighbours. Thus it is unlikely that existing patterns are un-learned. Changes can be performed localy. On the other hand the ART2 network has problems if the changed profile lays no longer within the similarity cone of the neuron that it had been assigned to before. In that case the profile will be assigned to a new neuron and over time the optimal clustering of the training set will be destroyed.
If a new profile is added that has no similarities to existing profiles, it can easily be added to the ART2 network while the SOM would have to adjust its global structure, which would have to be done by re-learning the whole training set. Thus a SOM tends to forget patterns when new patterns are learned.
It is also possible to add new items after a network has been trained. For this purpose the number of input neurons must be increased because a new dimension is added to the profile vectors. In both cases the extension is easily performed by adding a new initial weight to each neuron. To avoid violating the requirement of equal weight sums it only needs to be ensured that the initial weight values stay within a small interval.

Requests
After the training phase the network is ready to classify profiles. To generate recommendations for a user his or her user profile is applied to the network. Each neuron of the competitive layer computes its activation and a winner neuron is determined. This neuron represents the class of the profile. Instead of generating a target profile from all profiles of this class, the reference vector of the neuron is used i.e. the target profile has already been generated when computing the reference vector during the learning phase. As in the case of the *Mean Squared Differences Algorithm*, the target vector is compared to the applied profile and recommendations are made for items that appear to have been underused by the user.
When determining the winner neurons, a scalar product must be computed for each neuron of the competitive layer. The complexity of a request for recommendations thus depends on the number of items and the number of neurons: O(#neurons * #items). The number of neurons in the network does not depend on the number of users but on the number of relevant user clusters. This factor tends to be smaller by orders of magnitude and can be treated as a quality of service parameter.

Limitations
A winner neuron is determined by choosing the neuron with the highest activation. This means, only the similarity between input vector and weight vector is recognized. But if two vectors are opposite to each other this can also carry potentially useful information.

4 Evaluation

Three requirements for neural networks can be identified: Requests for recommendations must be processed quickly, the recommendations should be of high quality and the network should be able to adapt to changes to profile data during run-time. The first experiments presented here have been performed to test if those requirements were met when using two test data sets. The performance of the neural networks has been compared to memory-based Collaborative Filtering based on the *Mean Squared Differences Algorithm*. After presenting our first experiments, the learning duration, selection of parameters and the dependance of the quality of recommendations on the training set will be discussed. Based on those criteria SOMs and ART2 networks will be compared.

4.1 Test Data

We used two test data sets. Most results presented here have been obtained by using the publicly available EachMovie dataset [18]. It contains 2.811.983 ratings on a scale from 1 to 5 for 1628 movies by 72.916 users. As in [4] we randomly selected a subset of 2000 users that had a minimum of 80 entries in their profiles. In each profile 30 entries are then randomly selected as control set while the others are used as input for the filtering algorithms. To test the performance of the different methods with sparse profile data from a different domain, we used data that we obtained in a large intranet E-Commerce application, Eurovictor II [14]. The data describes the intensity with which employees have accessed different services of the application.

4.2 Response Time and Quality of Predictions

For our experiments we used a 10x10 SOM and an ART2 network for a maximum of 50 neurons. We also used *Mean Squared Differences Algorithm* with a neighbourhood size of 20. We used the EachMovie-dataset. To simulate a growing database, we did three experiments using 20%, 60% and 100% of available profile entries, with 30 control set entries in each case that we used to evaluate the computed recommendations.

The three different subsets have been used as training sets for the neural networks and as input for the memory based method. Afterwards one, three, five and 30 ommendations have been computed and compared to the control set of 30 hidden profile entries. The results of those experiments can be found in figure 7 (left). As can be seen, the ART2 network performed significantly better than the memory based method while the SOM produced results of the lowest quality.

Each time recommendations were computed the response time has been measured. As expected and shown in figure 7 (right), the performance of the neural network based methods mostly scales with the number of items, while the number of users has a much less significant influence. The three test sets contain profile entries for 400, 1200 and 2000 of users. While the memory based approach has to calculate scalars for all users, the neural networks only need to compute 50 and 100 scalars respectively

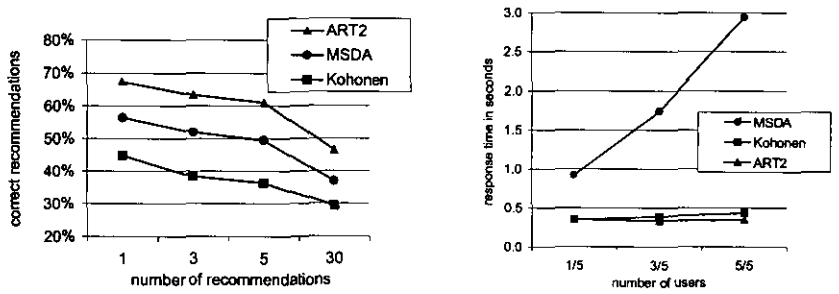


Fig. 7. Quality of results (left) and response time with growing number of users (right)

4.3 Adaptivity

A third important requirement besides quality of recommendations and performance is the ability of the neural networks to adapt during run-time. The networks can't relearn all profiles whenever a single profile changes since the complete learning process takes a considerable amount of time, in the case of our test sets several minutes. Therefore new or changed profiles must be (re-)learned by the network without repeating the initial learning phase.

We performed an experiment to test the ability of our networks to adapt to changes or additions of profiles. The same test data as in the previous experiments was used. This time we started with 400 profiles that we randomly selected from the complete set of 2000 user profiles and used them as training data on the networks. Then we randomly selected 800 profiles that were learned by the networks, then 1200, 1600 and finally all 2000. Please note that those data sets were not disjunct and some profiles were learned more often than others. Then, recommendations were generated and compared to the control set of 30 profile entries that were not used for training.

Figure 8 (left) shows the results of the experiment using a SOM. To un-learn as few patterns as possible when updating the network, the map's learning rate is reduced from 0.8 to 0.5. There is only a slight difference in the quality of the obtained results. The SOM shows a high degree of adaptivity.

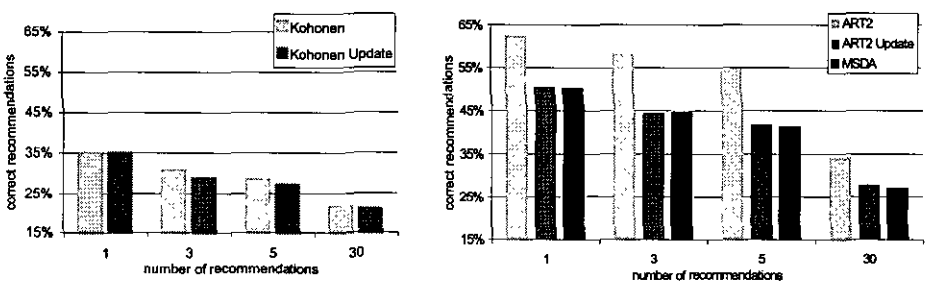


Fig. 8. Quality of results when gradually updating the network compared to relearning all data

Figure 8 (right) shows the corresponding results for an ART2 network. To preserve the knowledge of previously learned patterns the number of learning steps has been reduced from 10 to 5. The quality of predictions decreased in our experiments and after (re-)learning all profiles from the 5 training subsets the quality of the recommendations fell to the level of what we obtained using our *Mean Squared*

Differences Algorithm implementation. The neurons have not been adapted to profile changes in a step by step manner. Modified profiles have been rejected as dissimilar to existing neurons and have been assigned to new neurons. This prevented an optimal clustering of the input data.

4.4 Learning Speed

The decrease of prediction quality that occure after updates of an ART2 network can be countered by occasionally relearning the whole data set. Figure 9 shows that compared to a SOM an ART2 network can be trained fairly quickly. The same test data and networks as in the previous experiments were used. The comparatively slow learning speed of the SOM results from the complex computation of neighbourhood relations using the Gauss bell. The learning speed can be increased by using a slightly less exact linear function. The duration for (re-)learning a single profile from our training set was on average 0.06 seconds with an ART2 network and between 0.5 and 1 second for a SOM.

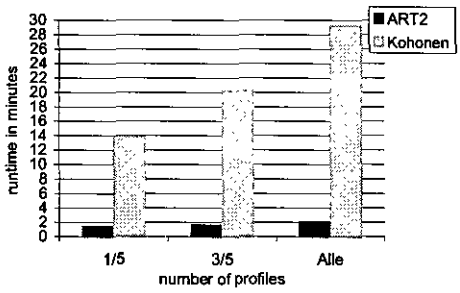


Fig. 9. Learning duration for 400, 1200 and 2000 profiles

4.5 Choice of Parameters

For most parameters of the neural networks standard values can be used. Two important parameters that have to be set are the number of learning steps (*epoch*) and the size of the network In the case of our ART2 networks ten learning steps have been enough, often the network stabilized even faster.

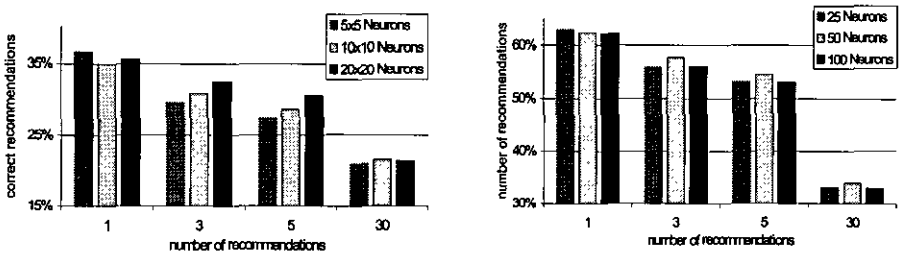


Fig. 10. Quality of results in relation to network size

For the SOM we also used ten learning steps. The map needs a global learning process and can then make more and more fine grained local weight adjustments

while decreasing the learning rate. This fine grained learning phase needs much longer and can't be completed after ten learning steps. We performed experiments with up to 100 steps without being able to observe any measurable improvements. The most important parameter is the network size which means the number of neurons in the competitive layer. In order to decrease run-time during profile evaluation the network size should be constant and as small as possible. But the quality of recommendations also depends on the number of neurons. Thus, depending on the number of relevant clusters in the profile set more or less neurons are necessary.

When using SOMs, too few neurons lead to an instability while too many neurons prevent a useful classification. The ART2 network only uses new neurons when a pattern does not lie within a cone of attentiveness. Thus the number of neurons in an ART2 network can never be too high. The number of neurons in an ART2 network is only limited for performance reasons. If the set maximum number of neurons has been reached and a new pattern does not lie within the attentiveness cone of a neuron then the cone radius is increased automatically until the pattern can be assigned to a neuron. Figure 10 (left) shows that in the case of a SOM in the range of 25 to 400 neurons only small differences in quality can be observed with our test data. Figure 10 (right) shows that there is also no significant difference when increasing the number of neurons in an ART2 network from 25 to 100.

4.6 Influence of Training Set

To test the influence of the population of the training set on the quality of results obtained with the three collaborative filtering methods, we performed two more experiments using the Eurovictor II data set. Since that data set is smaller and very sparsely populated we chose 16 neurons as the network size for both the ART2 network and the SOM and only computed 5 recommendations per profile. The Eurovictor II data set contains 770 profiles, only 91 with 5 or more entries.

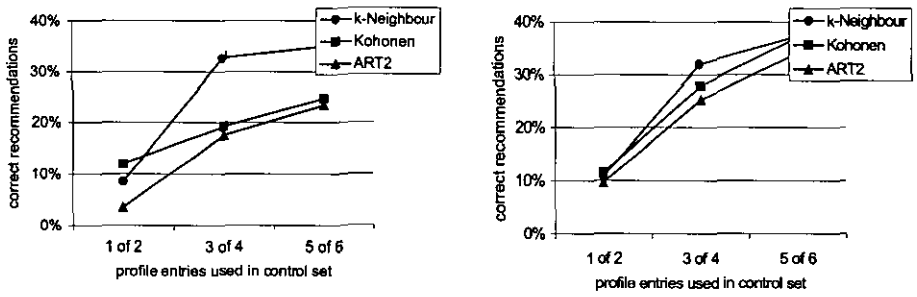


Fig. 11. Results with E-Victor data set: all profiles (left), only profiles with 5+ entries (right)

Figure 11 (left) shows results using all profiles as training set and figure 11 (right) visualizes the results that were obtained only using profiles with 5 or more entries. In both cases three different query sets have been created: These are sets that contain profiles with two or more, four or more and six or more entries respectively and use

one, three or five of those entries for the control set. The result shows that the neural networks' performance is affected negatively by the sparsity of the training data. With very sparse data our memory based implementation produced better quality results.

6 Conclusion

We have described two model based approaches to collaborative filtering based on Self-Adaptive Maps and ART2 networks. When evaluating them with test data from two real life applications we found out that ART2 networks produce even better results than a widely used memory based approach if the profile vectors in the training data are not sparsely populated. The model based algorithms needed by several orders of magnitude less memory and less computations to produce recommendations. The ART2 network proofed to be adaptive but the quality of predictions still degraded slowly after more and more changed profile vectors were (re-)learned suggesting that it is necessary to relearn the complete data set once in a while.

In the near future we want to include other memory based algorithms into our comparison, such as suggested in [10] or [6]. We also plan to explore wether it is possible to improve results further by combining several methods, including the neural network models explored in this contribution.

References

[1] AMAZON.COM, Amazon Homepage: http://www.amazon.com (accessed: May 2000)

[2] R. ARMSTRONG, D. FREITAG, T. JOACHIMS, T. MITCHELL, WebWatcher: a learning apprentice for the World Wide Web, in: AAAI Spring Symposium, Stanford, U.S., pp. 6-12.

[3] W. BENN, O. GOERLITZ, Semantic Navigation Maps for Information Agents, in: Proceedings of the 2nd Intl. Workshop on Cooperative Agents, Lecture Notes in Artificial Intelligence 1435, Springer-Verlag, Heidelberg, 1998,

[4] D. BILLSUS, M. J. PAZZANI, Learning Collaborative Information Filters, in: Proceedings of the 15th Intl. Conference on Machine Learning, Morgan Kaufmann Publishers, 1998

[5] H.BRAUN, J.FEULNER, R.MALAKA, Praktikum Neuronale Netze, Springer Verlag, 1996

[6] J. S. BREESE, D. HECKERMAN, C. KADIE, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, in: Proceedings of the 14th Annual Conference on Uncertainty in Artificial Intelligence, pp. 43-52, 1998

[7] S. BRIN, L. PAGE, The Anatomy of a Large-Scale Hypertextual Web Search Engine, in: Proceedings of the 7th World Wide Web Conference, 1998

[8] G. A. CARPENTER, Neural Network Models for Pattern Recognition and Associative Memory, 1989, in: G. A. Carpenter and S. Grossberg, Pattern Recognition by Self-Organizing Neural Networks, MIT Press, 1991

[9] G. A. CARPENTER, S. GROSSBERG, ART2: Self-Organization of Stable Category Recognition Codes for Analog Input Patterns, 1987, in: G. A. Carpenter and S. Grossberg, Pattern Recognition by Self-Organizing Neural Networks, MIT Press, 1991

[10] M. K. CONDLIFF ET AL., Bayesian Mixed-Effects Models for Recommender Systems, in: Proceedings of the ACM SIGIR Workshop on Recommender Systems: Algorithms and Evaluation, 22nd Intl. Conf. on Research and Development in Information Retrieval, 1999

[11] L. FAUSETT, Fundamentals of Neural Networks, Prentice Hall International Inc., 1994
[12] M. GAEDKE, C. SEGOR, H.-W. GELLERSEN, WCML: Paving the Way for Reuse in Object-Oriented Web Engineering, ACM Symposium on Applied Computing, Italy, 2000
[13] G. GRAEF, Adaptation of Web-Applications Based on Automated User Behaviour Analysis, 2nd Annual Conference on World Wide Web Applications, Johannesburg, 2000
[14] G. GRAEF, M. GAEDKE, Construction of Adaptive Web-Applications from Reusable Components, in: Lecture Notes on Computer Science, 1st Conference on Electronic Commerce and Web Technology, Springer-Verlag Heidelberg, 2000
[15] D. K. HAWES, Information literacy in the business schools, in: Sep/Oct Journal of Education in Business, 70, pp 54-62, 1994
[16] T. KOHONEN, Self-Organizing Maps, Springer-Verlag, Heidelberg, 1997
[17] J. A. KONSTAN ET AL., Applying Collaborative Filtering to Usenet News, in: Communications of the ACM, Vol. 40, No. 3, pp. 77-87, 1997
[18] P. MCJONES, EachMovie collaborative filtering data set, DEC Systems Research Center, http://www.research.compaq.com/SRC/eachmovie/.
[19] D. M. NICHOLS, Implicit Rating and Filtering, in: Proceedings of the Fifth DELOS Workshop on Filtering and Collaborative Filtering, Budapest, 1997
[20] R. ROJAS, Theorie der Neuronalen Netze, Springer Verlag, Heidelberg, 1993
[21] U. SHARDANAND, P. MAES, Social information filtering: algorithms for automating "word of mouth", in: Human factors in computing systems (CHI'95), Denver, USA, 210-217
[22] A. M. A. WASFI, Collecting User Access Patterns for Building User Profiles and Collaborative Filtering,in: Proceedings of the International Conference on Intelligent User Interfaces, pp. 57-64, 1999.

METIOREW: An Objective Oriented Content Based and Collaborative Recommending System

David Bueno[1], Ricardo Conejo[1], Amos A. David[2]

[1] Department of Languages and Computer Science, University of Málaga, 29071, Málaga, Spain.
{bueno, conejo}@lcc.uma.es
[2] LORIA, BP 239, 54506 Vandoeuvre, FRANCE.
adavid@loria.fr

Abstract. The size of Internet has been growing very fast and many documents appear every day in the Net. Users find many problems to obtain the information that they really need. In order to help users in this task of finding relevant information, recommending systems were proposed. They give advice using two methods: the content-based method that extracts information from the already evaluated documents by the user in order to obtain new related documents; the collaborative method that recommends documents to the user based on the evaluation by users with similar information need. In this paper we will present our approach through the employment of a user model and analyze some existing Web recommending systems and identify some problems that we try to solve in our system METIOREW. Some of the problems in document recommendation are: a) how to begin with document recommendation to users at the beginning of interaction when there is little or no knowledge on the user, b) how to make document recommendation to the user with changing information needs (objectives) without employing the general preferences of all the users but employing explicit individualized user model that integrates the user's objectives, c) how to provide access to the user's past history in order to review interesting documents related to specific objectives. The algorithms that we propose for calculating the degree of relevance of documents based on our user model is also explained.

1 Introduction

The size of Internet has been growing very fast and many documents appear every day in the Net. Users find many problems to obtain the information that they really need. In order to help users in this task of finding relevant information *recommending systems* are proposed. Due to this great amount of information that comes from everywhere, recommending systems are needed to filter junk e-mail [27], to obtain only the relevant news from Usenet, like GroupLens [25] or URN [10], to get only the interesting World News for the user [7] and probably the most important, to find information in the WWW. In this article we will concentrate on this last type of systems.

In the next subsections we will present the three methods that exist to make recommendations on the Web. In section 2 we will see the work that has been done in this area focusing on the advantages and disadvantages of each. In section 3 we will describe our Web Recommending system METIOREW that tries to solve most of the identified problems in the systems of section 2. Section 4 describe the state of development of this system and the lines that we are following.

1.1 Content-Based Recommendations

In order to recommend a document to a user some systems use only the content of the documents. To do so, documents are represented with a set of features like title, author, keyword, etc. When a user evaluates a document as interesting this set of features is used to look for similar documents. A user model or profile is constructed from different evaluations that lead the systems to know more about the user preferences. This model has been constructed only using the features of the documents. That's why they are called *content-based recommending systems.*

1.2 Collaborative Recommendation

The activities of many users on an Information Retrieval System (IRS) are often very similar because they have similar preferences or related interest. This means that the difficulties to find interesting information are repeated for each of these users. A possible solution to avoid work already done is to share the result of a user between other users with "similar interest".

A pure collaborative recommending system offers documents to the user not because of its content but because there is a similar user who has evaluated them as interesting. This means that in this case the similarity is between people that evaluated in the same way some documents without taking into account the content of those documents.

The problem with this kind of systems relates to new documents. Until somebody evaluates them the system has no information regarding their relevance and the system will not be able to recommend them. Another important problem relates to the number of users of these systems. The fewer the number of users the lower the probability of evaluating the documents with the same interest. In this case, the system will not be efficient.

1.3 Hybrid Solution

The content-based method has been used in IR area with interesting results. But the idea of completing the recommended documents, that has been evaluated by the user, with other documents that has been retrieved by other users with similar

characteristics looks like very promising. This hybrid solution is more efficient than each method applied separately.

2 Related Work

In this section we will analyze the most representative recommending systems that apply some of the methods explained above. We will concentrate on the main aspects that differentiate the systems from each other.

WebWatcher[1]. [1] This system makes only content-based recommendations. The user expresses what he's looking for using keywords that define his goal. The goals are restricted to technical reports and the keywords can be on author, title, etc. The user navigates through the Web under the supervision of WebWatcher that will assist him by highlighting the links that are closer to the keywords of the goal. To calculate the degree of relevance they use the methods of Winnon [20], Wordstat, TFIDF [26] and Random.

Letizia. [19] This is another content-based system that recommends web documents. The user doesn't need to enter information about his information need. Letizia supervises his actions and uses some heuristics to determine what's interesting for the user. For example, if a user makes a bookmark of a document, it means that he's interested in it. Other less strong heuristic is that if a user analyses the links of a document, the document is most likely related to his information need. Documents are represented by list of keywords.

Syskill & Webert. [23] Using content-based recommendation this system predefines some topics that can be the possible goals of the users. An index for each topic has been manually created. When the user evaluates some documents of this index the system can recommend the most related pages with the pages already evaluated. The algorithm to select relevant documents is a Bayesian classifier. Also LYCOS' queries can be constructed by the system.

FAB. [2-4] FAB is an adaptive collaborative web recommending system. It has different kind of agents: collection agents (to look for new information of a limited number of topics), selection agents (one for user who has a model of him in order to recommend the most interesting documents) and a central router (who send pages obtained from the collection agents to selection agents of users with a similar profile to the content of the pages). The user regularly receives a list of pages to evaluate. This information is used to update the original collection agent (that is not attached to this user) and his selection agent. This agent uses the TFIDF [26] to obtain the keywords of the document and the cosine similarity measure to calculate the

[1] WebWatcher http://www.cs.cmu.edu:8001/afs/cs.cmu.edu/project/theo-6/web-agent/www/project-home.html

similitude between the user profile and the document. The best-evaluated documents are sent to other users with similar profile.

PTV. [12] This system recommends Television programs through the WWW and the WAP. There is a user profile composed of the channels, keywords, programs, etc., that interest the user. He can update the model but the best way is through relevant feedback. The system selects the k users most similar to the actual one and offers the *r* best programs for this user. When a recommendation is asked by the user a list of programs is shown, some of them selected from those *r* programs and others from content recommendations.

MOVIELENS[2]**.** [13] It recommends films to the users. For this, it recommends films using the information of other users with similar video preferences and also information based on the user's previous evaluations. They use different agents to collect information using different methods and combine them to obtain better results. In their experiment they make comparison when using only content-based information or possible combination with one or more agents, and they conclude that the best solution is the mixture of several agents and the information based on the user feedback.

Casper/Jobfinder. [8; 24] Casper helps to find a new job. It works making case based reasoning. It evaluates each possible new job comparing it with the jobs already evaluated and proposes it if it is the most similar to one that has interested to the user. With this idea they restrict the selection problem to a classification one. They use a standard weighted-sum metric to calculate the similarity, and as features they use the kind of work, salary, experience, etc. Casper is also collaborative because it makes recommendations from similar user, where the calculation of similarity is done using the number of different jobs that they have evaluated in common.

GASs. [5] It pretends that a group of people with the same goal looks for information in the Web, and that this information will be shared between them. For this they need to have a group model besides a user model.

WebCobra. [29] It's also a recommending system where initially the user evaluates a set of documents from where a vector of keywords is extracted that will be used to identify this user. This vector is sent to a server that uses the simple cosine method to calculate the similarity and assigns a user to a group. When the user evaluates other documents he selects which of them are the best to send to the partners of the group. The subjects for the groups are concentrated in very specific domains to facilitate the task of a group. The user can ask for recommendations and he will receive the documents marked as interesting by other partners.

We will resume here some limitations of these systems and present how we have tried to solve the problems in our system METIOREW.

---

[2] MovieLens http://movielens.umn.edu/

The first problem relates to the creation of a model of a user on which the system has little or no information. This is a typical problem in User Modeling. One of the solutions to solve this problem is through the use of stereotypes [17] [15]. Letizia, Syskill, Webert and Casper have difficulties at the beginning to give advice to the user because the model is empty. In MOVIELENS, a little training is needed by the system that offers a list of films to evaluate to create the initial model. In FAB, at the beginning, a list of documents is given to the user to evaluate. FAB has an 'amalgamated profile' with the documents that are most interesting for the actual population of the system and from which *n* documents are offered to the user to begin to create the model. WebWatcher and WebCobra begin to create the model using some initial keywords that nearly fix the model to them.

Other important problem that we have found in those systems is the global vision of the people's interests. It looks like if somebody will always want to find the same information in the web. Letizia, FAB, PTV, MOVIELENS or WebCobra don't define the concept of goal or objective. But even the systems that define it are very restrictive or let the user to have only one goal. For example WebWatcher allows one goal restricted to technical reports, Syskill & Webert allows the selection of a limited number of topics for which some index has been manually constructed, GASs supposes that many people have the same one single goal.

The last important concept is related to the manipulation of the history. Everybody has bookmarks in his Web browser with the most relevant URLs. Many users group them by topics. But why can't we find the documents that we have already evaluated as FAB does? From the systems analysed only[3] MOVIELENS allows the review of the evaluated documents.

3 METIOREW description

METIOREW is a collaborative and content-based Web recommending system. It recommends documents to the user by trying to solve the problems presented in section 2. The first aspect of METIOREW is that it is objectives oriented where an objective expresses an information need. We can relate users and objectives in the following manners in an Information Retrieval System a) a user's information need can evolve, b) the user can have the same information need at different times and c) different information needs can be related. Related information needs does not have the same degree of similarity. The same methodology is used in METIORE [11].

METIOREW allows the user to review the documents already evaluated through the user's history. The user also has the possibility to modify the evaluations attached to the documents. This can be interpreted as an "intelligent bookmark" organized by

[3] The information we have from these systems is based on the content of their publications, perhaps there are other features that haven't been documented

objectives and evaluated documents sorted in the order of their relevance to the objectives.

The problem of how to make recommendations to users at the beginning has also been carefully studied in METIOREW. The user inserts an objective (In natural language, but it's used only like a label) and a list of keywords that helps to create the initial model for this objective. As this model is not strong enough, the system looks for other user with a similar objective (initially using this list of keywords). Then for the new user two models are managed in parallel to give him recommendations: his own model and the most similar model found in the system. The second one is used until the new user's model is significant.

3.1 Architecture

Fig. 1 we represent the general architecture of METIOREW. The final objective of METIOREW is to find the most relevant Web pages for the current user's objective. The pages will come from Web robot search, supervised navigation and collaborative retrieval. In the following paragraphs we explain the architecture of the system.

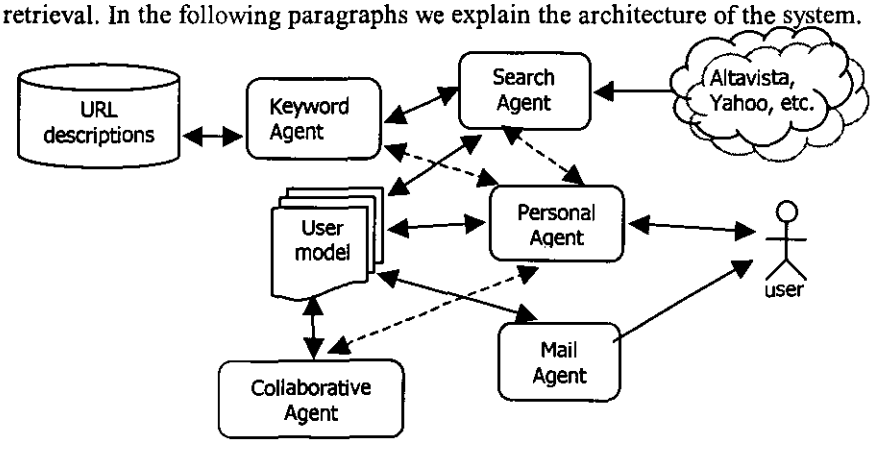


Fig. 1. Architecture of METIOREW

Personal Agent. For the development of METIOREW we have adopted an approach based on agents [22] with specific goals and who share information. The *Personal Agent* controls the users' identification, management of objectives, supervised navigation, history of activities, generation of recommendations and the reception of user feedback.

Search Agent. There is a search agent for each user objective that will look for information in a web index like Altavista, Yahoo and Ya.com. It constructs the queries using the most representative keywords of the user model for this objective. This agent will get a list of documents that will be kept for recommendation to the

user when needed. The *keyword agent* does a description of documents content in order to know the real relevance of these documents for the user.

Keyword Agent. This agent receives a Web page and generates a set of keywords that describe it. In 3.4 we have the algorithm to select the features that represent a document.

Collaborative Agent. Its goal is to offer relevant information to the user taking as reference documents that have already been evaluated by other users with similar objective with the current user's objective. Only documents with degree of similarity superior to a predefined threshold are proposed. The agent searches for the most similar objectives by comparing the models and for each of the objective retrieves a list of pages that will be sorted according to the degree of relevance.

Mail Agent. There is one mail agent for each user. It's activated with a timer defined by the user (for example once day or once week). Its mission is to examine the list of recommendations generated by the collaborative and search agent and send the N best links for each objective to the user through mail. This lets the user define different objectives, improve their model in different sessions, thus allowing the system to retrieve relevant documents without the user's interaction.

3.2 Functionality

When the user begins a session with a new objective the *personal agent* asks the user to insert a textual description of his current objective and a list of initial keywords that will describe it. Then a *search* and a *collaborative agent* are initiated to look for related pages. The user can also begin to navigate freely on the Web in a supervised way. If he finds by himself relevant documents he will give a feedback that is used to update the model. Whenever the user asks for recommendation the *personal agent* will look for the new documents that have been found for this objective and they are proposed in a list.

As the initial model (real model) is only restricted to a list of keywords, METIOREW tries to improve it using the model (external model) of the user with the most similar objective to the current one. The possible recommendations of the two models are used to make new recommendations. Each relevant feedback serves to improve the real model. The external model is used until the real one is enough independent (at least 10 positive feedback). Also the possibilities of disable or change the external model for another one are contemplated. This will be use when a high percentage of the recommendations of the external model are rejected.

After evaluating some documents the user model will be refined and composed of the initial keywords (that will have an important weight because they have been directly selected by the user) and new ones obtained from the documents evaluated by the user. The *search* and *collaborative agents* use the current information of the model to search for new related documents that are kept in a repository for this user.

The *personal* and *mail agents* consult it to generate recommendations. The documents of the repository are sorted by degree of relevance to the objective.

3.3 User Model

The user model keeps all the information needed to personalize the interactions with the user. In METIOREW we keep diverse information that we resume in Table 1. The model is objective oriented. This means that for each user we can have several models depending on the different information needs. With this representation the same user will be able to work in different sessions with different objectives, but having the possibility of review past sessions through the information acquired by the system.

Documents Revised	URL, keywords, evaluation	Used to regenerate the keyword synthesis, and for the review of the user's history
Keyword synthesis	Keyword, ev1, ev2, ev3, ev4	Each relevant keyword and the frequency for each of the four kinds of evaluations are kept in the system
Related objectives	User, objective, % similitude	List of objectives found by the collaborative agent as similar to the current objective
Documents to recommend	URL, keywords, %similitude	Documents obtained by the search and collaborative agents that hasn't been evaluated

Table 1. Content of the user model

The user's relevant feedback is fundamental to make the personalized recommendations. In METIOREW we use four kinds of feedback. *OK* (The document is interesting for the user), KNOWN (The document is interesting for the user but he already knew it), BOF (With the current knowledge of the user, he can not determine if this document is interesting or not), ERROR (This document is not relevant to user's objective).

3.4 Calculation of the degree of relevance

In this section we describe briefly the methods used to calculate the degree of similarity between a document and the user model (keyword synthesis), between two objectives and how we extract the relevant keywords from the documents and the model.

Obtaining relevant keywords. The keyword agent will obtain the features to describe the web page. It makes it using the Term Frequency TF [14] by applying some heuristics such as "remove the most and least repeated words". It is expected that this will provide the best *m* words that describe the document.

Classifying new pages in the user model. After each evaluation by the user, the synthesis of the keywords in user model is updated. Increasing by one the frequency of the evaluation for each keyword that represents the document evaluated does this. When a new page arrives the system must predict how the user will evaluate it. To do that we compare the vector of features of this document with the user model for the

current objective using an adaptation of the Naive Bayes [18] that has been proved to be a good classifier in [18] [28] [21] [16] [30].

Objectives similarity. To find similar models is needed to compare different objectives. For this we use the Pearson Correlation [6] that we adapt to the representation of our synthesis model. In the eq. 1 *w(a,i)* is the similitude between the objectives *a* and *i*. $v_{i,j}$ is the probability that the user with objective *i* (u_i) evaluates as interesting the element *j*. Where interesting means classify as *ok(c1)* or *known(c2)* and I_i is the set of features on which the user has given a feedback[4].

$$w(a,i) = \frac{\sum_j (v_{a,j} - \overline{v_a})(v_{i,j} - \overline{v_i})}{\sqrt{\sum_j (v_{a,j} - \overline{v_a})^2 \sum_j (v_{i,j} - \overline{v_i})^2}} \quad (1)$$

$$v_{i,j} = P(c1, c2 / u_i \wedge j) \quad (2) \qquad \overline{v_i} = \frac{1}{|I_i|} \sum_{j \in I_i} v_{i,j} \quad (3)[9]$$

Obtaining representative features for searching. To create a vector that represents the model of the current objective we use the eq. 2. It gives the probability of evaluating as correct each feature. Sorting this we obtain the *n* best keywords to be used by the *search agent.*

4 Future Work

The system presented here is in the phase of development and we are planning its experimentation in a real situation. This experimentation will be composed firstly by the analysis of information collected by the system, basically efficiency in recommendations and percentage of correct prediction of feedback. Besides that, we elaborate a questionnaire to be filled by the users in order to make a correlation between the system's proposal and the user's opinion. We are also working on the improvement of the *personal agent* as a 3D agent in the style of Pazzani [7] that makes the user feel a more human interaction with the computer.

5 References

1. Armstrong, R., Freitag, D., Joachims, T., & Mitchell, T. (1995). "Webwatcher: A learning apprentice for the world wide web". AAAI Spring Symposium on Information Gathering from Heterogeneous Distributed Environments
2. Balabanovic, M. (1997). "An Adaptive Web Page Recommendation Service". Proceedings of the First International Conference on Autonomous Agents Marina del Rey, CA

[4] The user doesn't gives feedback for features but for documents. The main features (or keywords) of a document inherit its feedback.

3. Balabanovic, M., & Shoham, Y. (1995). "Learning Information Retrieval Agents: Experiments with Automated Web Browsing". AAAI Spring Symposium on Information Gathering, Stanford, CA
4. Balabanovic, M., & Shoham, Y. (1997). "Combining Content-Based and Collaborative Recommendation". Communications of the ACM, 40(3)
5. Barra, M. (2000). "Distributed Systems for Group Adaptivity on the Web". Adaptive Hypermedia and Adaptive Web-Based Systems Springer-Verlag
6. Billsus, D., & Pazzani, M. (1998). "Learning Collaborative Information Filters". Proceedings of the International Conference on Machine Learning Madison, Wisc.Morgan Kaufmann Publishers
7. Billsus, D., & Pazzani, M. (1999). "A Hybrid User Model for News Story Classification". Proceedings of the Seventh International Conference on User Modeling (UM '99) Banff, Canada
8. Bradley, K., Rafter, R., & Smyth, B. (2000). "Case-Based User Profiling for Content Personalisation". Adaptive Hypermedia and Adaptive Web-Based Systems Springer-Verlag
9. Breese, J., Heckerman, D., & Kadie, C. (1998). "Empirical Analysis of Predictive Algorithms for Collaborative Filtering". Proceedings of the Fourtheenth Conference on Uncertainty in Artificial Intelligence Madison, WI.Morgan Kaufmann Publisher
10. Brewer, R. S., & Johnson, P. M. (1994). "Toward Collaborative Knowledge Management within Large, Dynamically Structured Information Systems". University of Hawaii, Dpt. of Information and Computer Science. Honolulu:

11. Bueno, D., & David, A. A. " METIORE: A Personalized Information Retrieval System". 8 International Conference on User Modeling.UM'2001
12. Cotter, P., & Smyth, B. (2000). "WAPing the Web: Content Personalisation for WAP-Enabled Devices". Adaptive Hypermedia and Adaptive Web-Based Systems Springer-Verlag
13. Good, N., Schafer J. , Konstan, J., Borchers, A., Sarwar, B., Herlocker, J., & Riedl, J. (1999). "Combining collaborative filtering with personal agents for better recommendations". In Proceedings of the Sixteenth National Conferenceon Artificial Intelligence
14. Joachims, T. (1997). "A Probabilistic Analysis of the Rocchio Algorithm with TFIDF for Text Categorization". Proc. of the 14th International Conference on Machine Learning ICML97 (pp. 143-151).
15. Kay, J. (1995). "Vive la difference! Individualised interaction with users". IJCAI'95
16. Keogh, E., & Pazzani, M. (1999). "Learning augmented Bayesian classifiers: A comparison of distribution-based and classification-based approaches.". Uncertainty 99, 7th. Int'l Workshop on AI and Statistics, (pp. 225-230). Ft. Lauderdale, Florida
17. Kobsa, A. (1993). "User Modeling: Recent Work, Prospects and Hazards".
18. Kononenko, I. (1990). "Comparison of Inductive and Naive Bayesian Learning Approaches to Automatic Knowledge Acquisition". Current Trends in Knowledge Adquisition, 190-197
19. Lieberman, H. (1995). " Letizia: An Agent That Assists Web Browsing". International Joint Conference on Artificial Intelligence Montreal, CA.
20. Littlestone, N. (1988). ", Learning quickly when irrelevant attributes abound , .". Machine Learning, 2:4, pp. 285--318Notes: Pedido a biblioteca
21. Mitchell, T. M. (1997). "Machine Learning". The McGraw-Hill Companies, Inc.
22. Nwana, H. (1996). "Software Agents: An Overview". Knowledge Engineering Review
23. Pazzani, M., Muramatsu, J., & Billsus, D. (1996). "Syskill & Webert: Identifying interesting web sites". AAI Spring Symposium on Machine Learning in Information Access . URL= http://www.parc.xerox.com/istl/porjects/mlia/papers/pazzani.ps.

24. Rafter, R., Bradley, K., & Smyth, B. (2000). "Automated Collaborative Filtering Applications for Online Recruitment Services". Adaptive Hypermedia and Adaptive Web-Based Systems ItalySpringer-Verlag
25. Resnick, P., Iacovou, N., Suchak, M., Bergstrom, P., & Riedl, J. (1994). "GroupLens: An Open Architecture for Collaborative Filtering of Netnews". Proceedings of ACM 1994 Conference on Computer Supported Cooperative Work (pp. 175-186).
26. Rocchio, J. (1971). "*Relevance Feedback in Information Retrieval*". The SMART Retrieval System: Experiments in Automatic Document Processing, 313-323Notes: No lo tengo aun.
27. Sahami, M., Dumais, S., Heckerman, D., & Horvitz, E. (1998). "A bayesian approach to filtering junk e-mail". AAAI-98 Workshop on Learning for Text Categorization
28. Singh, M., & Provan, G. M. (1996). "Efficient learning of selective Bayesian network classifiers". . Proceedings of the 13th International Conference on Machine Learning
29. Vel, O., & Nesbitt, S. (1997). "A Collaborative Filtering Agent System for Dynamic Virtual Communities on the Web". URL= http://citeseer.nj.nec.com/de-collaborative.html.
30. Versteegen, L. (2000). "The Simple Bayesian Classifier as a Classification Algorithm". URL= http://www.cs.kun.nl/nsccs/artikelen/leonv.ps.Z.

Integrating User Data and Collaborative Filtering in a Web Recommendation System

Paolo Buono, Maria Francesca Costabile, Stefano Guida, Antonio Piccinno, Giuseppe Tesoro

Dipartimento di Informatica, Università di Bari, via Orabona 4, 70125 Bari, Italy
{buono, costabile, piccinno, tesoro}@di.uniba.it, stpag@libero.it

Abstract

Web-based applications with a large variety of users sufferf rom the inability to satisfy heterogeneous needs. A remedy for the negative effects of the traditional "one-size-fits-all" approach is to enhance thes ystem's ability to adapt its own behaviour tot he users characteristics, such as goals, tasks, interests, that are stored in user profiles. Filtering techniques are used to analyse profile data and provide recommendation to the users to help them navigating in the site and retrieving information of interest. However, techniques such as collaborativef iltering are very time consuming and for thisa re not suitable to be implemented in small systems. We describe here the approach we have adopted in FAIRWIS (Trade FAIR Web-based Information Services), a system that offers on-line innovative services to support the trade fair business processes and a great number of exhibitors organized in a Web-based virtual fair. The approach is based on the integration of data the system collects about users, both explicitly and implicitly, and a classical collaborative filtering technique in order to provide appropriate recommendations to the user in any circumstances during the visito f the on-line fair catalogue.

1. Introduction

Web-based applications with a large variety ofu sers suffer from the inability to satisfy heterogeneous needs. For example, a Web bookstore offers the same selection ofb est sellers to customers with different reading preferences. A Web museum offers the same "guided tour" and the same narration to visitors with very differentg oals and interests.

A remedy for the negative effects of the traditional "one-size-fits-all" approach is to enhance the system's ability to adapt its own behaviour to the goals, tasks, interests, and other features of individual users. In the last years, many research teams have been investigating ways of modelling features oft he users ofh ypermedia systems. This has led to a number of interesting proposals of adaptationt echniques and adaptive hypermedia systems, which are especially challenging for the Web and therefore are pushing many researchers to work on this topic [1].

Personalisation is a process ofg athering and storing information about visitors of a web site, analysing thes tored information, and, based on this analysis, delivering the right information to each visitor at the right time. A personalisation component should be capable to recommend documentsa nd/or other web sites, promote products, make appropriate advice, target e-mail, etc. Personalisation is increasingly used as a mean to expedite the delivery of information to a visitor, making the site useful and attractive so that the visitor is stimulated to return to it. For this, personalisation is becoming an expected feature of e-business web sites.

A personalisation component builds ande xploits models or profiles of the users interacting with the system. A user profile isa (possibly structured) representation of characteristics of that user, in order to take into account hiso r her needs, goals, and interests. The term user profiling is also used to refer to a software module that acquires personal data ofa user, process these data to obtain additional information, and uses it to modify either contenta spects or navigation capabilities of web pages.

Big companies use different methods to personalize their Web sites. Many successful sites, sucha s Amazon.com, Yahoo.com, and CNN.com, use richp rofile information as the basis for providing valuable services and they are considered models for those who want to personalise their site. However, most Web sites do not provide yeta ny personalisation feature.

The work presented here is related to the project FAIRWIS (Trade FAIR Web-based Information Services), founded by EU. This project aims at offering on-line innovative servicest o support the trade fair business processesa nd a great number of exhibitors organised in a Web-based virtual fair. The whole concept of trade fairsi st ransferred into an electronic form, and visualisation techniques, including virtual reality, are used in order to provide "reality" feelings to the users of trade fair information systems. In recent years, some Web-based information sites have been made available, providing information both on trade fair events ando n companies participating in these fairs. However, these data are not organised in an integrated, homogeneous and comprehensive way, since are usually presented in a rigid pre-designed company oriented style. Moreover, currently available Webs ites exploit static data that it is difficult to update and to put on-line in an appropriate format. FAIRWIS has a real time connection with an underlying database to guarantee coherence ofd ata and up-to-date

status. Another unique feature of FAIRWIS is provided by the User Profile Engine (UPE) that is the personalisation module. In the analysis, we have performed of fair web sites worldwide, none was found to show any personalisation feature. Therefore, UPE provides a significant added value towards a system that can fit the users needs as better as possible.

In order to build the user profilesa nd provide recommendations, the approach we have implemented in UPE is based on the integration of data the system collects about users both explicitly and implicitly and a classical collaborative filtering technique in order to provide appropriate recommendations to the user in any circumstances during the visito f the on-line fair catalogue.

The paper is organized as follows. In section 2, we describe general approaches for collecting and exploiting user information. Section 3 discusses the ratings provided by the FAIRWIS users about the web pages they visit. Section 4 presentsU PE in more details, and in Section 5 the current prototypei s described. Section 6 concludes the paper.

2. Collecting and exploiting user information

Theo bjectiveo f collecting user information is to create a profilet hat describes user characteristics. The more common techniques are explicit profiling, implicit profiling, and use of legacy data:

- *Explicit profiling*: each user is asked to fill in a form when visiting the web site; this method has the advantage of letting users specify directly their interests.
- *Implicit profiling*: the user's behaviour is tracked automatically by the system. This methodi s generally transparent to the user. Often, user registration is saved in what is called a cookie that is kept at the browser and updated at each visit. Behaviour information is generally stored in a log file.
- *Legacy data*: they provide a rich source of profile information for known users.

The above methods can be combined to produce comprehensive profiles. This is whatw e have done in FAIRWIS.

The generated user profiles are analysed in order to present or recommend documents, items, or actions to the user. Making recommendations is a very challenging

step. Rule-based and filtering techniques are the best known for analysing profile data and making appropriate recommendations.

Rule-based techniques exploit a set of rules specified in the system in ordert o drive personalisation. Cross-selling is an e-business example of ther ule-based technique: a rule could be specified to offer product X to a customer who has just bought product Y; for example, a customer of a book might be interested in current or previous books by the same author or in books on the same subject.

Filtering techniques employ algorithms to analyse profile data and drive presentations and recommendations. The three most common filtering techniques are: simple filtering, content-based filtering, and collaborative filtering [2].

- *Simple filtering* relies on predefinedc lasses of users to determine what content should be displayed orw hat service should be provided. Fore xample, employees of the Research department may have access to somef unctionality that may not be available for employees of other departments. Therefore, specific pages will be presented to the employees of the Research department.

- *Content-based filtering* works by analysing the content of the objects to generate a representation of the user's interests. The analysis needs to identify a set of key attributes for each object and then fill in the attributev alues. For example, in e-commerce users are often asked to provide ratings for each attribute of a product. In this way, content-based filtering analyses the ratings provided by the users to determine, for any product, which other product of the same category has the closest ratings and could then be recommendedt o a user who got interest in the first product. This technique is most suitable when the objects are easily analysed by the computer and the user's decision about object suitability is not very much subjective. However, recommendations arel imited to objects related to thoset he visitor has considered during his or her navigation, and there is no provision for user qualification.

- *Collaborative filtering* collects visitor opinions on a set of objects, using ratings provided explicitly by theu sers or implicitly computed, to form peer groups and then learns from thep eer groups to predict ap articular user's interest in a item. Instead of finding objects similar to those a user liked, as in content-based filtering,

collaborative filtering develops recommendations byf inding visitors with similar tastes. Recommendations produced by collaborative filtering are qualified based on the peer group's response and are not restricted to a simple profile matching. However, thism ethod requiresa large user base in order to find a peer group for each visitor. This might imply a long learning curve, because at the beginning, when the number of participating visitors is small, the quality of recommendations will be low. The results improve gradually ast he number of participating usersi ncreases. The more objects two users have rated similarly, the closer the two users are.

For examples of systems incorporating a personalisation components based on content-based and/or collaborative filtering see [3-10].

3. User ratings for providing recommendations

UPE (User Profile Engine) is the Personalisation Module implemented in FAIRWIS. In the current implementation, UPE works as a recommender system that provides personalizeds uggestions about pages users might find interesting in the on-line fair catalogue available in the system. The recommendations are generated on the basis of different types of ratings that thes ystem gets from theu ser interaction or computes trough an algorithm of collaborative filtering. As illustrated in the previous section, collaborative filtering workso n the idea of analysing "human" evaluations (also called ratings) on items of certain domain and join users who share same tastes.

The ratings collectedb y the system may be both implicit and explicit. They are explicit if users tell the system what they think about an item. For example, the user may give a rate of 5 toa n element of the fair catalogue he or she has found very interesting by filling an appropriate form shown on the screen.

Even ife xplicit rating is fairly precise, it has disadvantages, such as: 1) stopping to enter explicit ratings can alter normal patterns of browsing and reading; 2) unless users perceive that there is a benefitp roviding the ratings, they may stop providing them.

Implicit rating is much more difficult to determine. Oard and Kim dividei mplicit ratings into three categories [11]: rating based on examination, when a user examines an item; rating based on retention, when a user saves an item; rating based on reference, when a user links all or parto f an item into an other item.

How can user preferencesw ith implicit ratingsb e determined? Some criteria were established [12]. In FAIRWIS, byt aking into account the structure of the system currently developed by the other partners of the project, we may only consider the following events, and each oneh as been associated with a weight that highlights the importance of thate ventf or collecting information about the user interests:

- access to a web page (we gave differentw eights to the home page and the other pages);
- printa nd/or save action (the user thatd oes this is highly interested on thatp age or item);
- download of specific files included in download areas;
- image zoom;
- access by search (if the system includes a search engine).

Even if implicit ratings are difficult to determine, they have the following advantages: 1) every interaction with the system (and every absence of interaction) can contribute to implicit rating; 2) can be gatheredf or free; 3) can be combined with several types of implicit ratings for a more accurate rating; 4) can be combined with explicitr atings for an enhanced rating.

Indeed, the method that is quite effective is the mixed technique implicit/explicit rating and we implemented it in FAIRWIS, as it will be described in the next section. However, especially in the case ofs ites with many pages, we can be in a situation that some pages have not been evaluated by the current user (neither explicit nor implicit ratings are available). To overcome this situation, algorithms of collaborative filtering may be used. They predict user interests on an item not evaluated by taking into account the historical data set on rates ofa users community stored into a database ofe xisting rating provided by other users. Such a database is a seto f rates $u_{i,j}$ corresponding to the evaluation of user i on the item j. If I_i is the set of items on which user ih as expressed a rate, then it's possible to define the average rate for user i as:

$$\bar{u}_i = \frac{1}{|I_i|} \sum_{j \in I_i} u_{i,j} \qquad (1)$$

It is also possible to compute the evaluationo f the current user (indicatedw ith a subscript a)b ased on information on the current user and on a set of weights calculated

from the user database. We can assume that the predicted rate of current usere xpected for item j, i.e. $p_{a,j}$, is a weighted sum of rates of other users:

$$p_{a,j} = \bar{u}_a + k\sum_{i=1}^{n} w(a,i)(u_{i,j} - \bar{u}_a) \qquad (2)$$

where n is the numbero f useri n the database with non zero weight. Weights $w(a,i)$ can reflect distance, correlation, or similarity between each user i and the current user. k is a normalisationf actor sucht hat the absolute values of the weights sum to unit. The expression thati dentifies the weight$w(a,i)$ which relates the currentu ser a with user i is:

$$w(a,i) = \frac{\sum_j (u_{a,j} - \bar{u}_a)(u_{i,j} - \bar{u}_i)}{\sqrt{\sum_j (u_{a,j} - \bar{u}_a)^2 \sum_j (u_{i,j} - \bar{u}_i)^2}} \qquad (3)$$

where summations over j are calculated on all items which have been evaluated by the users a and i.

4. The User Profile Engine

We describe in this section the current use in FAIRWIS of the User ProfileE ngine (UPE). UPE has been developed as an independent component so that it can be portable to other systems. Itu ses Windows NT, Java and SQL-Server 7.0.

The main types of users addressed by FAIRWIS are: i) fair organisers;i i) exhibitors, namely responsibles of companies that exhibit their products or activities in the fair; iii) professional visitors, who visit the fair for business reasons rather than fun.

The userp rofiles managed by UPE have a static component and a dynamic one. The static component consistso f a set of information that identifies each user and doesn't change (or change rarely). For example: name, nationality, type of users. The information sources come primarily from the registration forms that someF AIRWIS users, namely exhibitor and professional visitors, are required to fill.

The dynamic component of userp rofile is the changing part of userd ata. The set of user preferencesi sp art of the dynamic profile. UPE obtains this information by using different type of ratings: explicit ratings, for instance interest ratings for catalogue pages; implicit ratings, obtained by tracking usern avigation;c omputed ratings, obtained by collaborative filtering techniques to supply the preferencesn ot expressed by users, either implicitly or explicitly.

UPE stores individual user profiles, but also assigns the users to a kind of stereotype [13], each stereotype characterised by specific values of the attributes considered in the user profile; these stereotypesa re called "segments". Segmentsa re used because they group users with similar characteristics, so that if the individual user profile is non yet complete, it is still possible to provide recommendations to a user based on his or her stereotype. Indeed, the recommendations foru sers of a segment are calculated using all ratings available for every user belonging to that segment. In this way, UPE is able to provide recommendations even to a userw ho just registered, because in the registration form the user provided enough data to be assigned to a segment, and thus UPE gives as recommendations those relative to the segment the user belongs to.

It may also happen that the segment doesn't contain enough information for recommendations. In this case, UPE provides recommendations based onlyo n the current page the user is visiting, calculating them by taking into account the behaviour of the other userst hat have already visited that page. In other words, the system will suggest the pages indicated as interesting by most users who were also interested in the page the useri s currently looking at. This peculiarity of UPE is suitable also in the case of an unknown user, i.e. user noty etr egistered or any user who is surfing in the site.

The integration in UPE of a collaborative filtering algorithm permits to predict possible preferences of the current user on the basis of the evaluations provided by other usersa nd stored in the UPE DB. As it is well known, these algorithms are useful but also very time consuming. We are trying to reduce the algorithm complexity by using some heuristics. For example, UPE doesn't re-calculate all the weights in formula (3), but it does it only for those pairs ofc orrelated users, in which at least one oft he two users has interacted with the web site and has modified a numbero f ratings highert han a certain threshold.

The rates in UPE DB are updated automatically in a scheduled way, or with an explicitr equesto f the system administrator.

Fig. 1 shows the main modules of UPE. The predictive algorithm is responsible of computing the ratings the user did not provide (explicitly or implicitly). The recommendation module is the UPE main component, which manages the user profiles and computest he recommendations. To do this, UPE needs to communicate with the FAIRWIS Core system, which manages the web interface.

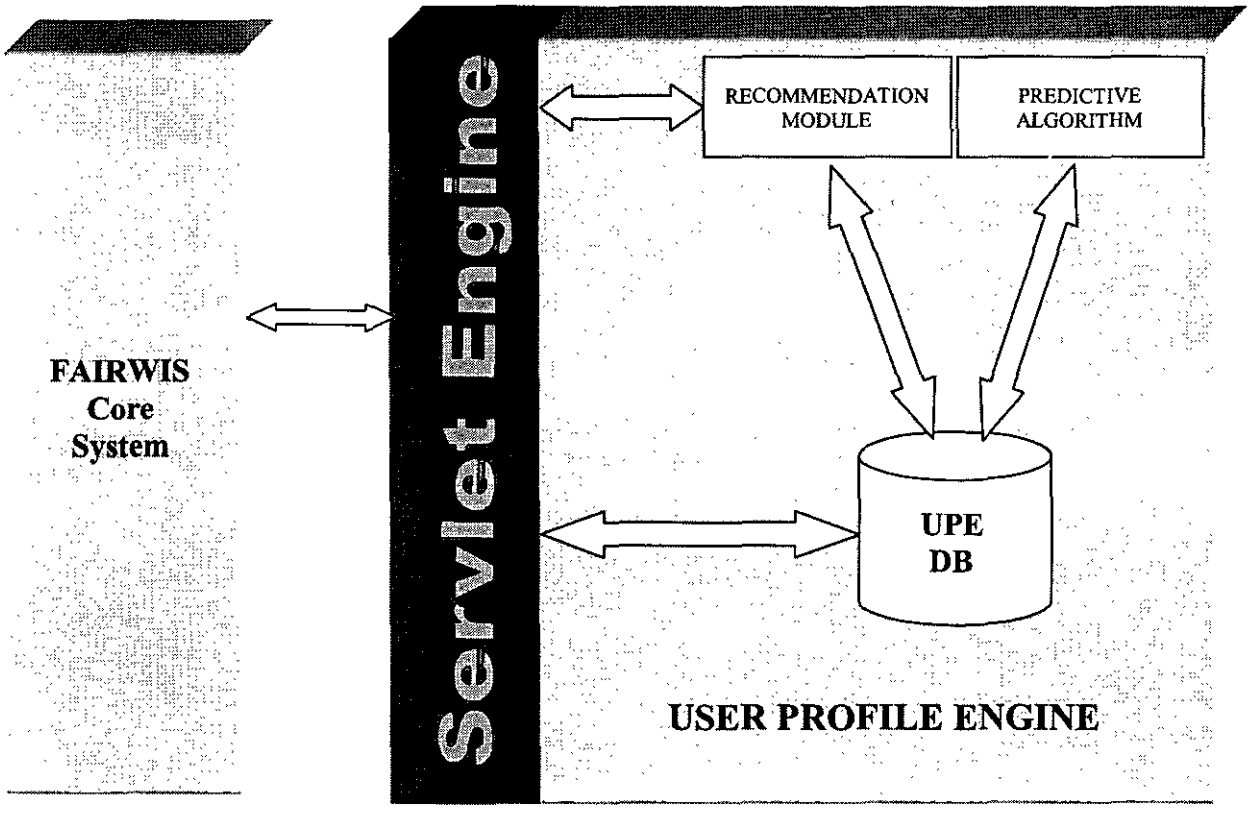


Figure 1. UPE architecture and the communication with FAIRWIS Core system.

The communication between FAIRWIS Core and UPE is based on Java Servlet technology. Servlets are programs that run on a Web server; they are designed to work according to a request/response processing model. In this model, a client sends a request message to a servera nd the server responds by sending back a reply message. Requests can come in the form of an HTTP, URL, FTP, or a custom protocol. As shown in Fig. 1 the communication module is called Servlet Engine. It is composed by servlets devoted to the communication between FAIRWIS Core and UPE.

More specifically, servlets running on the web server perform the following tasks:

1. insert static useri nformation into UPE DB; this information is obtained from the registration form filled by the user in the FAIRWISw eb site and sent by FAIRWIS Core to UPE ServletE ngine;
2. update UPE DB with the dynamic information coming from the useri nteraction, sentb y FAIRWIS Core to UPE ServletE ngine;

3. give the recommendation for au ser, by replying to as pecificr equest coming from FAIRWIS Core.

5. The prototype

We here report some examplest hat describe the interaction of users with a prototype web site in order to show the UPE behaviour. The recommendationsp rovided to the user are shown in an area of the fair web pages. In the current prototype, which refers to the fair on Information Technology MITE 2001, this area is located at the bottom-left area of all web pages but the home page, as shown in Fig. 2.

As we already said, exhibitors and professional visitors are required to fill the form for user registration thati ncludes data required by UPE.

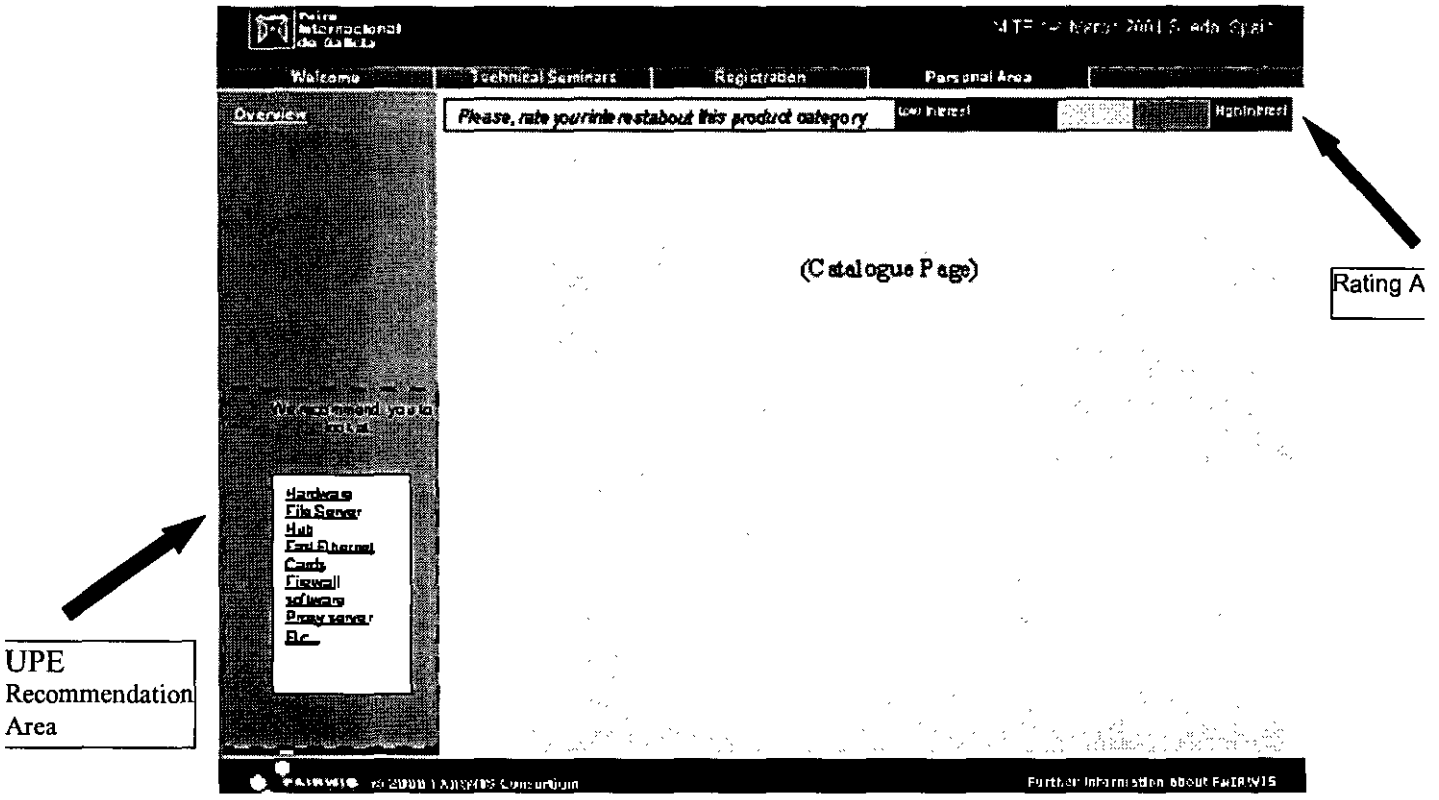


Figure 2. Example from the MITE 2001 prototype.

Example 1: Explicit rating request

When the registered user is visiting a catalogue web page (see Fig. 2), FAIRWIS asks him or her to rate the currentp age, showing a frame at the top of the web page.

Users can rate the web page by acting in an explicit rating box located at the top of

the main window (see Fig. 2): he or she selects the vote for the current page by clicking on a specific area of the rating box, according to his or her own interest. Of course, the user is free to ignore the request of explicit rating and keep navigating through the catalogue.

Example 2: FAIRWIS provides recommendation for a registered user

When a registered user logs in the system, the page shown to the user is appropriately personalised and the system suggests some linksw orth to be examined. The recommendations are links to other web pages. The user can click on one recommendation or keep navigating among web pages in a "traditional" way.

6. Conclusions

In this paper, we have described the FAIRWIS personalization component that currently worksa sa recommendation system. The approach is based on the integration of data collectedb oth explicitly andi mplicitly from the user interaction anda collaborative filtering technique.

It is worth nothing that UPE has been designed to be an independentm odule thatca n be integrated in any system, with which the communication is done through Java Servlet technology. Moreover, UPE manages a complex user profile that can be better exploited to provide different types of personalisation. The current behaviour as a recommender system is due toc onstraints imposed by the actual structure of the FAIRWIS Core system.

Acknowledgement

The support of EuropeanC ommissiont hroughg rant FAIRWIS IST-1999-12641 is acknowledged.

References

[1] AH2000, Proceedings of the International Conference on "Adaptive Hypermedia and Adaptive Web-Based Systems", Trento, Italy, August 28-30, 2000.

[2] "Web Site Personalisation". IBM High-Volume Web site team, January 2000, http://www-4.ibm.com/software/developer/library/personalization/index.html.

[3] Balabanovic, M. Exploring versus exploiting when learning user models forn ext recommendation. UserM odeling and User-Adapted Interaction, 8(1), pp. 71-102, 1998.

[4] Basu, C., Hirsh, H., Cohen, W. Recommendation asc lassification: Using social and content-based information in recommendation. Proceeding of theF ifteenth National Conference on AI, Madison, WI, AAAI Press, pp. 714-720, July 1998.

[5] Boone, G. Concept features in Re:Agent, an intelligent e-mail agent. Proceedings of the Second InternationalC onference on Autonomous Agents, New York, ACM Press, pp. 141-148, 1998.

[6] Dent, L., Boticario, J., McDermott, T., Zabrowski, D. A personal learning apprentice. Proceedings of the Tenth National Conference on AI, San Jose, CA: AAAI Press, pp. 96-103, July 1992.

[7] Hinkle, D., and Toomey, C. N. CLAVIER: Applying case-based reasoning on to composit part fabrication. Proceeding of the Sixth Innovative Application of AI Conference, Seattle, WA, AAAI Press, pp. 55-62, 1994.

[8] Lang, K. NEWSWEEDER: Learning to filter news. Proceeding on the 12th International Conference on Machine Learning, Lake Tahoe, CA, Morgan Kaufmann, pp. 331-339, 1995.

[9] Pazzani M. J., Muramatsu J., Billsus D.:S yskill & Webert: Identifying Interesting Web Sites. Proceedings on Thirteenth National Conference on Artificial Intelligence, Portland, OR, AAAI/IAAI Press, Vol. 1, pp. 54-61, 1996.

[10] Shardanad, U., Maes, P. Social Information Filtering: Algorithms fora utomating "word of mouth". Proceedings of the Conference on Human Factors in Computing Systems, Denver, CO, ACM Press, pp. 210-217, 1995.

[11] Oard, D., Kim, J. Implicit Feedback for Recommender Systems. In Proceedings of AAAI Workshops on Recommender Systems. July 1998.

[12] Claypool, M., Le, P., Waseda,M ., Brown,D .I mplicit Interest Indicator.C omputer Science Technical Report Series, WPIC omputerS cience Department, Worcester, Massachussets, July 2000.

[13] E. Rich, "Stereotypes and user modeling", *User Models in Dialogue Systems*, A. Kobsa and W. Wahlster, eds. Springer-Verlag, pp. 35-51, 1989.

Adaptive Hypermedia System for Supporting Information Providers in Directing Users through Hyperspace

Yoshinori Hijikata, Tetsuya Yoshida and Shogo Nishida

Graduate School of Engineering Science, Osaka University, JP
E-mail:hijikata@nishilab.sys.es.osaka-u.ac.jp

ABSTRACT

This paper introduces an adaptive hypermedia system that provides a means for information providers to direct their users through hyperspace. With our system, the information providers are able to easily direct users to their own personalized navigation paths. Destination options are determined by hiding links and by applying rules so that the users are offered the best paths for them. In order to reduce the information providers' efforts in creating navigation rules, we simplified the format of these rules and offer an authoring tool that verifies the navigation rules and reports any errors if they exist. This tool not only allows the information providers to easily write navigation rules but also guarantees the adequacy of the navigation path.

KEYWORDS: adaptive hypremedia, navigation, rule, link hiding, authoring tool

INTRODUCTION

The WWW (World-Wide Web) has become popular as a tool for information retrieval. Furthermore its applications are diversifying into such areas as electronic commerce, marketing and education [14]. In these kinds of application, the information providers or the Web masters often want to direct their users through hyperspace as desired according to each user's preferences and status. For example, to offer teaching materials according to the learner's knowledge or study history, to regulate obscene contents to prevent children from viewing them, and to guide marketing campaign so that banner ads and item recommendations are in accord with each customer's preferences.

Some researchers have been studying these kinds of user navigation aids in the research field of adaptive hypermedia [1]. Adaptive hypermedia is hypermedia with functions to dynamically adapt to each user. A hypermedia document is the basic component of the WWW and allows users to freely move and retrieve information in hyperspace, which consists of nodes containing information and links relating the nodes. Adaptive hypermedia systems realize their user-adaptations based on various kinds of user information such as the user's prior knowledge, objectives and interests. Many adaptive hypermedia systems implement user-navigation guidance created by the information providers, using methods that allow the information providers to describe the navigation rules according to user categories and user behaviors defined in advance[2, 4, 5, 6, 7, 8, 9, 10, 13]. However this method leads to the following problems:

1. Information providers have great difficulty describing the navigation rules as they move towards fine-grained navigation control.
2. It becomes difficult to predict the resulting states for various kinds of users, because the navigation dynamically varies according to each user.

These problems become more critical as the need to direct users accurately increases.

This research aims to construct an adaptive hypermedia system that reduces the burden on information providers and prevents errors in the described navigation rules. We propose a system solving the above problems by the following mechanisms:

1. A simplified format for navigation rules.
2. An authoring tool that examines the navigation rules.

Existing hypermedia systems do not focus on providing mechanisms and functions that can reduce the burden on information providers for creating and verifying navigation rules.

In the proposed system, (1) destination options are determined by hiding links so that information providers can direct users accurately, (2) long-term and short-term user information can be used in the navigation rule, because it is generally important to consider users from both perspectives [12], and (3) the format of user information is also simple, because the format of navigation rule is simple.

This paper is organized as follows. First, we introduce existing navigation methods and user models, and explain the reason why we adopted link hiding and the type of user information used in our system. After that we describe the navigation method and rule format of our system and explain the authoring tool. Next we describe the implementation of the system and its evaluation. Finally we offer some conclusions.

ADAPTATION METHOD AND USER MODEL

Existing adaptive hypermedia systems construct a user model and use it for adapting to each user [1]. The user models describe information about the users such as the users' knowledge, objectives, and interests. This section describes the adaptation method and the user model.

Adaptation method

Adaptation methods can be classified into content-level adaptation and link-level adaptation[1]. Content-level adaptation adapts the displayed content of the node. Link-level adaptation adapts the links of the node. We adopted link-level adaptation, because the focus of this research is on how information providers can direct users through hyperspace.

Link-level adaptation can be classified into the four types (direct guidance, adaptive ordering, hiding, adaptive annotation) according to how the links are modified [1]. Direct guidance attaches an explanation to the link the user should follow or inserts a [Next Link] button for directing the user. Adaptive ordering sorts links in the order of the degree of suitability to the user. Hiding narrows the accessible hyperspace by hiding links. Adaptive annotation attaches additional decorations such as icons and colors to the links.

Out of these four kinds of link-level adaptation, direct guidance, adaptive ordering and adaptive annotation may display links that the information provider does not recommend. Although this leads to the problem the user may not always follow the information provider's intentioned navigation paths, it also gives the user the freedom to select links the information provider does not recommend. These approaches are especially suitable for applications like (1) information retrieval systems and (2) learning systems that focus on the learners' active information retrieval.

The link hiding method does not display any extraneous links. Although this forces the user to follow the information provider's navigation paths, it has the corresponding advantage that the information provider can always direct the users as desired. This constrained approach is suitable for applications such as (1) learning systems for business training where the learner follows the information provider's directions to acquire the knowledge quickly, (2) help systems for application software, and (3) information systems that screen obscene content from children. We adopted link hiding because our system focuses on the situation where the information provider wants to direct users precisely.

User model

As user models for adaptive hypermedia, there are overlay models, stereotype models, and models using keywords[1]. An overlay model is based on a structural domain model which is represented as a semantic network of domain concepts. A stereotype model assigns the user to one of several possible stereotypes for each dimension of classification. A model using keywords is represented as a vector or a matrix whose elements are the degree of interest in a keyword or topic. Although it is simple as a model, a browsing history expressed with the permutation of URL is also used for some Web applications. This is useful for describing the user's recent browsing action.

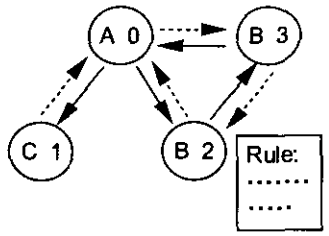


Figure 1: Hypermedia model.

Since it is generally important to consider long-term and short-term user information for designing user models [12], we also use both kinds of user information. Since our system simplifies the format of navigation rule, we also make our user model simple. We use pairs consisting of a property and a value (we call "user parameter") for modeling the user's long-term information. This approach is actually similar to all three kinds of user model described above. We also use a browsing history represented as the sequence of the node classifications (we call "path history") for modeling the user's short-term information.

NAVIGATION METHOD

Hypermedia model

Figure 1 shows the hypermedia model used in our system. The circles in the figure show hypermedia nodes. The content displayed for the users are contained in these nodes. The user can move between nodes by following a link represented as a straight line with an arrowhead. We also define a return link represented as a dotted line with an arrowhead. Although the user cannot follow this link, this is required for implementing the authoring tool (explained in the next section). The identity of the node is represented by a number and node classes (explained later) are represented by alphabetic characters. Some of the nodes have navigation rules (explained later) as created by the information provider.

Class

Every node has a "class" represented by symbols such as alphabetic characters. The reason why we introduced the notion of class is to allow information providers to describe the navigation rules by generalizing and specializing the characteristics and meanings of the navigation rules. Class is used for representing the user model and the navigation rules. The information provider defines a class in terms of the following distinctions:

1. Whether or not the system displays the contents for a specific kind of user.

2. Whether or not the node offers users an explanation, asks a4 question, or does something else for an educational purpose.

3. Which of several categories of contents the node belongs to (in cases where the information provider deals with information in more than one category).

Representation of user model

The user parameter is represented as a value from some range. The information provider assigns a meaning to the user parameter according to his/her navigation control. The user, the information provider, or some other person sets the value of a parameter. For example, they can be set by asking users to answer a questionnaire or by using the results of regular paper tests in academic environments.

The path history is represented concisely as the sequence of classes of the nodes the user has visited, and indicates the order of information the user has browsed. If the user browsed nodes in an order such that their classes were $C \rightarrow A \rightarrow B \rightarrow B \rightarrow A$, the path history is represented as $CABBA$.

Navigation rule

This system decides which links to hide based on a navigation rule that may be associated with the current node. There are four kinds of navigation rules: (1) node path rules, (2) general path rules, (3) node user rules, (4) general user rules.

A navigation rule that uses a path history is called a path rule and a navigation rule that uses a user parameter is called a user rule. The navigation rule can also be classified into two types, node rules and general rules. A node rule is defined and applied only for a specific node. A general rule is for describing frequently followed navigation paths in hyperspace and frequently used segmentation of the range of the user parameter. The information provider can apply it for any node.

In a navigation rule, the information provider should describe the links that should be displayed by the node ID or by the class of the node that is the target of the link. The system hides all links that are not referenced in a navigation rule as links to be displayed. The format of these four kinds of navigation rule is as follows:

1. Node path rule

$$C_{11} \cdots C_{1h} + \cdots + C_{m1} \cdots C_{mh} = D_1, \cdots, D_n \quad (1)$$

2. General path rule

$$C_{11} \cdots C_{1h} + \cdots + C_{m1} \cdots C_{mh} = C_1^f, \cdots, C_n^f \quad (2)$$

3. Node user rule

$$e_1 \# P_i \# e_2 : D_1, D_2, \cdots, D_n \quad (3)$$

. General user rule

$$e_1 \# P_i \# e_2 : C_1^f, C_2^f, \cdots, C_n^f \quad (4)$$

The above variables represent the following:

C: Class

D: Id of the node to be shown

C^f: Class of the node to be shown

h: Number of histories that will be referred to

m: Number of path patterns

n: Number of IDs or classes of nodes to be shown

P_i: The ith user parameter (property)

e: Boundary number of the user parameter

The symbol '#' represents one of the following three operators: $<$, $\leq$ or $=$.

$C_{m1} \cdots C_{mh}$ in the path rules (1) and (2) shows the path history pattern, which represents the order of the user's search in hyperspace as a permutation. The path rule means that the system displays links whose node ID or class is described on the right part of the rule if the user's path history matches one of the path history patterns which are described on the left side. In user rules (3) and (4), $e_1 \# P_i \# e_2$ specifies the user parameter P_i and the applicable range of the parameter values. The user rule means that the system displays links whose node ID or class is described on the right side if the user parameter specified on the left side is within the specified range.

If a node has several navigation rules, the system displays all links that any navigation rule accepts. This means that if at least one rule out of several rules approves the display of a specific link, the system displays the link regardless of the other navigation rules.

Example of navigation

Figure 2 shows an example of navigation using path rules and user rules. For an educational application, the classes are defined as follows:

- A: Nodes with a question.
- B: Nodes which display an appropriate response when the user answers correctly.
- C: Nodes which display an appropriate response when the user answers incorrectly.
- D: Nodes which display an explanation for students with good school records.
- E: Nodes which display an explanation for students with poor school records.

We assume that the user parameter stands for the knowledge level on a specific subject and is set based on the result of a standard paper examination at the school.

A node path rule is defined for Node No. 5. This rule is only applicable at this node. "$ACA = 7$" in this rule means that when the user comes to Node No. 5 and the user's path history is ACA, the system shows the link to Node No. 7 and

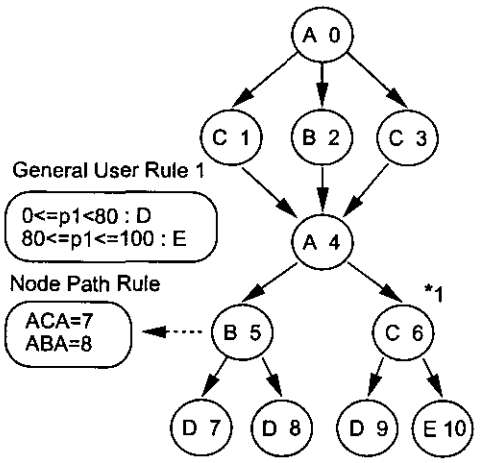


Figure 2: An example of the rules.

hides the link to Node No. 8. Because Class B means that the user answered correctly and Class C means that the user answered incorrectly, the history in this order means that the user answered the question in Node No. 0 incorrectly and answered the question in Node No. 4 correctly. "$ABA = 8$" means that if the user answered correctly for the question in Node No. 0 and also answered the question in Node No.4 correctly, then the system shows only the link to Node No. 8. That is to say the system changes the teaching materials according to the results of the previous questions.

A general user rule is also defined. This rule can be applied at any node in the hyperspace and in this case Node No. 6 uses it. In this general user rule, if the User Parameter 1 is 0, or more than 0 and not exceeding 80, the system shows any links to nodes whose class is D and ignores other links. If the user parameter is more than 79 and not exceeding 100, the system shows the link to nodes whose class is E and ignores other links. Because User Parameter 1 refers to the user's knowledge level of a specific subject, this means the system can offer suitable teaching materials based on the student's ability.

AUTHORING TOOL

Objective

Generally an authoring tool is important for an adaptive hypermedia system so that the information provider can direct users in hyperspace [3, 5, 6, 11]. Therefore our system provides an authoring tool that helps the information provider in describing the navigation rules. This tool examines the execution results of the navigation rule before they are incorporated into nodes. This aims for correct navigation with fewer errors and for simplification of the information provider's efforts to describe the navigation rules. We focused on detecting the following two kinds of problems because they can happen in any kind of content and are very likely to be related to navigation errors:

1. Dead end: There is a possibility that all links are hidden and the user cannot go anywhere after reaching a node with a navigation rule. This dead end problem could be caused by a bad navigation rule. Were a dead end to appear, it would force the user to stop searching in hyperspace. This may create obstacles to the user's progress.
2. Loop: In some navigation, the user may reach a node where the user has already been. We call this search looping. As seen in the WWW, we can use a loop effectively, for example as a link for returning to a top page, and it has an important role. However our concerns are that there may be unintended loops or the user may not be able to follow a loop that the information provider intended the users to follow. This is because the system hides links dynamically, which could cause problems for user navigation.

Dead end detection

A dead end can be caused by a path rule or a user rule or by a set of rules. This section describes an algorithm that checks if a dead end will happen in a node (or if there is a possibility a dead end can happen in the node) because of the path rules or user rules. In our system, if a node has several kinds of navigation rules, the system displays all links that any rule tries to display. It is possible that even if the tool detects a dead end caused by one kind of navigation rule (e.g. a path rule) in a node, the other kind of rule (e.g. a user rule) may try to display links in the node. Therefore when the tool detects an apparent dead end at a node, it checks whether another kind of rule is defined. If no other rule is defined for the node, it has detected a dead end. If another kind of rule is defined on the node, it has detected the possibility of a dead end.

Detection of dead ends caused by path rules A dead end caused by path rules happens when (1) the path history pattern the user has followed is not included in the path rules or (2) none of the links of the current node are described in the path rules as displayable, based on the path history pattern the user has followed to reach the current node. Here is an algorithm to check whether a dead end caused by path rules will happen or whether there is a possibility that it will happen in a specific node. This algorithm not only detects dead ends (dead ends possibility) but outputs the path history pattern that causes a dead end.

Detection algorithm for dead ends caused by path rules:

1. Node specification: The information provider specifies the node he/she wants to check.
2. Examination of displayable links: The system checks whether or not the links to be displayed according to the path history pattern described in the path rule defined at the specified node really exist in hyperspace. This is checked by comparing the nodes described as displayable in the path rules with the nodes that are the targets of the links of the current node.
3. Registration of live path: The system recognizes the path history pattern, which has links which should be displayed and

really exist, as a live path (If the user follows the live path to the specified node, there are links to proceed). It registers the live path in a list according to the length of the path history pattern. We call this list the live path list.

4. Depth-first search: The system executes a depth-first search from the current node (It is the specified node at first) using the return links mentioned in the last section and considering the current node to be the root of the inverted tree.
5. Path examination: The system refers to the live path list based on the length of the current depth-first search and checks whether or not the current path of the depth-first search is a live path for that node. If it is a live path, the system does not search deeper on this path, but returns to Step 4 for continuing the depth-first search from the upper node. If it is not a live path, the system continues to Step 6.
6. Detection of dead end possibility: If the length of the current depth-first search is the maximum length of the paths registered in the live path list, the system has determined that there is a possibility that a dead end happens when the user follows this path and the search continues to Step 7. If it is not the maximum length, the system returns to Step 4.
7. Decision on dead end: The system checks if a rule is defined at any of the nodes on the path. If no navigation rule is defined for any of these nodes, the system has determined that a dead end happens when the user has followed this path. If navigation rules are defined for at least one node, the system has determined that there is a possibility that a dead end happens when the user has followed this path. After that the system returns to Step 4.

Figure 3 shows an execution example of this algorithm. This example tries to detect a dead end at the shaded node in the figure. In this case, only the shaded node has navigation rules and the other nodes do not have any navigation rules. Out of 6 path history patterns in the navigation rule, only AB, AA, ABA, CCA have links which can be displayed and really exist. The system registers these path history patterns as live paths. After that, the system executes the depth-first search and dead end detection. In this example, the path CCB is not a live path. The system determines a dead end happens if the user follows this path, because the length of this path is the maximum length of the live paths in the live path list and there are no nodes that have a navigation rule in the path. CCA is an example of a path that does not cause a dead end, because it is a live path.

Detection of dead end caused by user rules A dead end caused by user rules happens when (1) the values of the user's user parameters are not within the range described in the user rules or (2) all displayable nodes described in the user rules do not exist as target nodes of links of the node where the user is. Here is an algorithm to check whether a dead end caused by user rules will happen in a specific node. This algorithm not only detects dead ends but also outputs the rule that causes a dead end.

Detection algorithm for dead ends caused by user rules:

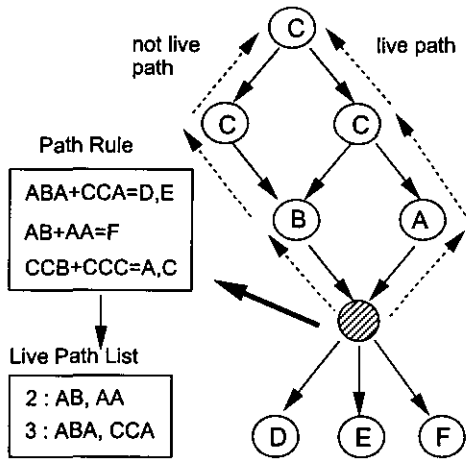


Figure 3: An example of dead end detection.

1. Node specification: The information provider specifies the node he/she wants to check.
2. Examination of displayable link: The system checks whether or not the displayable link for a specific range of the user parameter as described in the user rule of the specified node really exists. This is checked by comparing the displayable nodes in the user rules with the nodes that are the targets of the links of the current node. If such links exist, it recognizes the range as a live range (a range that has displayable links).
3. Examination of the range of the parameter: The system checks whether or not all ranges of the user parameter are live ranges. If there is a range that is not a live range, it has determined that a dead end will occur within that range.

Loops

Even if a path defines a loop without considering the effects of link hiding, it may not be a loop after link hiding is taken into account. It is necessary to set a specific user parameter and follow the path according to the navigation rule to check if the loop becomes a loop after link hiding. Here is an algorithm to detect loops by doing depth-first search from a specific node. This algorithm not only detects dead ends but also outputs the path of the loop.

Loop detection algorithm:

1. Node and maximum length specification: The information provider specifies the node he/she wants to start the depth-first search from and the maximum length of depth-first search.
2. Navigation rule execution: The system executes the navigation rules of the current node and hides links. After that it registers the displayed links in a displayed link list, which is necessary to execute depth-first search only using the displayed links.
3. Depth-first search execution: Perform one step in a depth-first search using a link registered in the displayed link list.

4. Loop examination: The system searches for the node ID of the node it has reached now in the path history of the depth-first search. If the same node ID exists in the path history, the system has determined the path from the previous node which has the same node ID to the current node is a loop.
5. Length check: If the search length of depth-first search is the length specified in Step 1, the system goes back to the previous node and returns to Step 3. Otherwise it returns to Step 2.

In Step 2 of the above algorithm, if a path rule is defined on the node where the system has reached and the search length of depth-first search is shorter than the length of the path history pattern described in the rule, the system cannot execute a path rule. In this case, the system does not execute the path rule and displays all links for detecting all possibilities for loops.

IMPLEMENTATION AND EVALUATION

Implementation of the system

We implemented the system using the C++ language. In the system, the information provider can use 16 kinds of classes. The maximum length of the path history is 16. There are no limits on the other parameters, the number of node, the number of links, the number of rules, and so on.

Figure 4-(a) shows an example of the system when the user searches the hyperspace. The user browses text information and searches by inputting the number of the link. Figures 4-(b,c) shows examples of the authoring tool. Figure 4-(b) is an example of dead end detection. It shows the sequences of the node IDs and the classes of the path history patterns leading to the information provider's specified node and causing dead end. Figure 4-(c) is an example of loop detection. It shows the sequences of the node IDs of the detected loop.

```
(1)
Learning System for Computer Science
1 System Development
2 Operating System
>1
(26)
Question 1 What is the name of system development
method using the following steps.
Analysis of Requirement -> Requirement Definition ->
System Design -> Program Design -> Programming -> Test
-> Employment -> Maintenance
1 Prototype model
2 User model
3 Water fall model
>
```

(a) Example of search by the user

```
Dead end detection mode
Input target node for detection>15
Input a rule type for detection
1 Path rule  2 User rule
>1
Dead end will occur in the following path.
3->6->9->10->11
C->A->B->A->C
```

(b) Example of dead end detection

```
Loop detection mode
Input start node for detection>83
Maximum length of loop>5
Following path will be a loop.
1->24->48->1
```

(c) Example of loop detetion

Figure 4: Output example from the system.

Objective of evaluation

We evaluated the system from the following viewpoints:

1. Qualitative evaluation of the entire system: This evaluation looks at how the features of the system, which are the simple rule format, the authoring tool, and adaptation by link hiding, appeared to the information providers. We asked some information providers to use this system and give us their subjective opinion on the effectiveness of the entire system.
2. Quantitative evaluation of the authoring tool: This evaluation examines whether the authoring tool succeeds in reducing the information providers' efforts to describe the navigation rules and insuring correct navigation. We quantitatively evaluated whether the authoring tool reduced the time that the information provider required for describing the navigation rules (the description time) and reduced the number of errors in the described navigation rules.

Evaluation method

Qualitative evaluation of the entire system Five information providers created content and described navigation rules for the content. After that they gave us their subjective opinions on the system's effectiveness and problems. They created the following content:

- Information Provider A: Educational content for science.
- Information Provider B: Educational content for English.
- Information Provider C: Content included obscene parts that children were not to see.
- Information Provider D: Content for marketing.
- Information Provider E: Content for software on-line manual.

Quantitative evaluation of the authoring tool Ten information providers participated in the experiment as subjects. These subjects were divided into two groups. The subjects of one group (Group A) described navigation rules without the authoring tool. The subjects of the other group (Group B) described navigation rules with the authoring tool. We evaluated the authoring tool based on the description time and the error ratio in the described navigation rules. The procedure of the experiment was as follows:

1. Experiment preparation: The experimenter prepared the experiment in the following way:

(a) Prepare content as hypermedia data.

(b) Assign meanings to the user parameters.

(c) Assign meanings to the classes.

(d) Define the class of every node.

(e) Create the task for the experiment (the navigation rules the subjects should create).

2. Explanation for the subjects: The experimenter explained how to describe the navigation rule to both groups, and how to use the authoring tool to Group B. The experimenter asked the subjects to work on a practice task for getting used to the system. After that the experimenter explained the task for the experiment. The experimenter sat by the subject and answered the subject's questions, but did not provide direct hints or solutions for the experimental task.
3. Experiment: Each test subject worked on the task and described all navigation rules. The experimenter observed the subjects working on the task during the experiment and measured the times taken for the descriptions.
4. Analysis: The experimenter measured the results of the experiment in the following way:

(a) Execute the navigation rule described by the subject and check (1) whether or not there is an error, (2) whether or not there is a dead end, and (3) whether or not there is an error in the loop when the navigation includes a loop.

(b) Calculate the following three evaluation parameters: (1) error ratio, which is the ratio of the tasks with an error in relation to all tasks, (2) dead end ratio, which is the ratio of the tasks with a dead end in relation to all tasks, and (3) loop error ratio, which is the ratio of the tasks with a loop and an error in relation to all tasks with loop.

(c) Determine the relative effectiveness of the authoring tool as it affects the above three evaluation parameters.

The content we created for the experiment are intended for students who study computer science. The size of the content, the usage of the user parameters and classes, and the contents of the navigation task are shown in the appendix.

Evaluation result

Qualitative evaluation of the entire system The information providers offered the following subjective opinions about the system:

1. I did not need programming knowledge and could describe the navigation rules easily because the rule format is simple.
2. Users will not hesitate to select links because I hid all the unnecessary links.
3. When I created the navigation with a loop, I had to check if the user can follow all of the paths in the loop. However the authoring tool showed all the paths of the loop and I did not have to follow all of the paths by myself.
4. The navigation rule with a dead end that was discovered by the authoring tool also had other errors.

Opinion 1 shows that even information providers who do not have the programming knowledge accepted the navigation rule description, because the rule itself is simple. Opinion 2 shows that link hiding reduced the users' hesitations in hyperspace and gave the information provider confidence that he could correctly direct the users. Opinion 3 shows that the information providers could recognize whether the loop paths they created were or were not accessible at a glance, because the authoring tool displays the sequences of the node IDs of all of the loop paths. From Opinion 4, we think that the navigation rule itself is more complex or the information provider created rules more carelessly in the node that has a dead end than in the other nodes.

The subjects also pointed out the following problems:

1. I have to describe the path history in a path rule even if I just want to create an easy navigation rule that only checks whether or not the user has passed a specific node.
2. I cannot change the user parameters while the user is searching in the hyperspace.
3. I think if the system had some general rules for frequent usage prepared in advance, I could describe the navigation rules faster.
4. I have to check not only dead ends and loops but also the detailed result of the navigation to see the sets of displayed links and sets of hidden links according to a specific path. Without this I have to do simulation by setting user parameters and following the paths.

Solving Problem 1 and Problem 2 has the advantage of strengthening the descriptive capability of the navigation rules. One of the solutions for Problem 1 is providing special user parameters for temporary flags and rules for updating the special user parameters. However this requires the information provider to manage the flags. The system should support the management of the flags. As regards Problem 2, we believe the system should not easily update the long-term user information (user parameters), because of the need to maintain the users' trust of our user model. However if the contents of the hypermedia are refined enough to manage changing the user parameters, there should be little problem when the navigation rules change them. In this case, the information provider should have the responsibility for the appropriateness of the content and the rules for updating the user parameters, because the system cannot guarantee the appropriateness of them.

The general rules for frequent usage mentioned in Problem 3 are important because the information provider can not only use them but also refer to them. We will provide them as a navigation rule library for our system. Providing other navigation rule checking functions besides dead end detection and loop detection would be a solution for Problem 4. However we only provided dead end and loop detection in the current version of the authoring tool for our system. The reason is that the information provider can perform simulations

by himself or herself, yet it is hard to manually detect dead ends and loops. We are considering functions besides dead end detection and loop detection to enhance the authoring tool.

Quantitative evaluation of the authoring tool As shown in Tables 1,2,3,4, Subjects a-e were in Group A and Subjects f-j were in Group B. Table 1 shows the description time. We did an analysis of variance to determine whether there is a significant difference in the description time between the two groups. However there was not a significant difference at the 5% level of significance.

Table 2 shows whether or not the subject described the navigation rules without an error, and the error ratio. Table 3 shows whether or not the navigation rule that the subject described included dead ends, and the dead end ratio. Table 4 shows whether or not the navigation rules (only for Tasks 5 and 6) that the subjects described included loop errors, and the loop error ratio. In each table, a circle shows that there is no error, there is no dead end, or there is no loop error. An X shows that there are errors, dead ends, or loop errors. Although the value of the error ratio, dead end ratio, and loop error ratio assumes discrete values because the number of tasks is small, we did an analysis of variance on these three parameters to get an idea of the effectiveness of the system. The result is that there is a significant difference between the two groups at the 5% level of significance in the above three parameters.

Figure 5 is a graph of the relationship between the description time and the error ratio. As regards the description time and the error ratio, the correlation coefficient of the group A is -0.67 and the correlation coefficient of the group B is -0.89. Although we cannot guarantee high and negative correlation, we see an apparent relationship that as the description time becomes longer the error ratio gets smaller.

The overall results, showing significant differences in the error ratio, dead end ratio, and loop error ratio, indicate that the authoring tool reduced the numbers of navigation errors. There is not a significant difference in the description time. We think the reason is that the Group B subjects tended to rely on the authoring tool to check the described navigation rule. However if the subject uses the authoring tool, the dead end and loop detection shows whether or not there is an error, and they repeatedly modified the navigation rules and checked them. The reason that there is not a significant difference in the overall description time is that (1) there are individual differences in the description times, (2) the authoring tool reduced the time to check the described navigation rule, and (3) the Group B subjects spent time repeatedly modifying and checking the navigation rules, thereby offsetting the time saved during each check. However we can recognize the effectiveness of the authoring tool also on the description time, because of the fact that the description time tends to get longer as the error ratio becomes smaller

Table 1: Description Time.

Subject	Time(min)	Subject	Time(min)
a	59.8	f	58.4
b	43.2	g	69.2
c	61.3	h	45.7
d	52.5	i	50.0
e	52.5	j	71.4

Table 2: Error in the navigation rule.

Subject	Task 1	2	3	4	5	6	Error ratio(%)
a	O	O	O	O	X	O	17
b	O	O	O	O	X	X	33
c	O	X	O	O	X	O	33
d	O	X	O	O	X	O	33
e	O	O	O	O	X	O	17
f	O	O	O	O	O	O	0
g	O	O	O	O	O	O	0
h	O	O	O	O	X	O	17
i	O	X	O	O	O	O	17
j	O	O	O	O	O	O	0

and the error ratio becomes smaller when the subjects use the authoring tool. This again shows that the authoring tool reduced the information providers' efforts in describing the navigation rules.

Discussion

We implemented an adaptive hypermedia system for information providers to properly guide users in hyperspace. Our system uses link hiding as the primary adaptation method and prevents users from selecting links that information providers do not make available. Our evaluation shows that adding a supporting tool that checks the execution of link hiding for this function enables information providers to direct users more reliably.

Because of the problems information providers have in describing the navigation rules, we focused on the following:

1. The effort required to express the navigation according to the format of the navigation rules.
2. The effort required to check the execution of the described navigation rules.

For the reduction of these efforts, the following devices and functions are effective:

1. Simple description of the navigation rules, which does not require programming knowledge for the information providers.
2. Providing an authoring tool, which detects errors in the described navigation rules.

The above devices and functions are effective for the reduction of the information providers' effort in describing the

Table 3: Deadend in the navigation rule.

Subject	Task 1	2	3	4	5	6	Dead End ratio(%)
a	O	O	O	O	X	O	17
b	O	O	O	O	X	X	33
c	O	X	O	O	X	O	33
d	O	O	O	O	X	O	17
e	O	O	O	O	X	O	17
f	O	O	O	O	O	O	0
g	O	O	O	O	O	O	0
h	O	O	O	O	X	O	17
i	O	O	O	O	O	O	0
j	O	O	O	O	O	O	0

Table 4: Loop errors.

Subject	Task 5	6	Loop error ratio(%)
a	X	O	50
b	X	X	100
c	X	O	50
d	X	O	50
e	X	O	50
f	O	O	0
g	O	O	0
h	X	O	50
i	O	O	0
j	O	O	0

navigation rules and reducing the errors in the described navigation rules for general hypermedia systems where information providers want to guide users.

CONCLUSIONS

This paper proposed an adaptive hypermedia system that reduces information providers' efforts to describe the navigation rules and leads to fewer errors in the described navigation rules while guiding users accurately. This system uses simple expressions for the navigation rules to reduce the information providers' efforts. It also adapts the hyperspace to the user by link hiding in order to achieve the desired user paths. We also offer an authoring tool for this system, which checks whether there are errors in the described navigation rules, with the aim of further reducing the information providers' efforts to describe the navigation rules and avoid errors in those rules.

The proposed system was implemented and evaluated qualitatively and quantitatively. In the qualitative evaluation, five information providers freely described content and navigation rules and gave the experimenter their subjective opinions. In the quantitative evaluation, ten information providers described navigation rules for the same navigation tasks and the experimenter measured the time required for describing the navigation rules and the error ratio, which is the ratio of

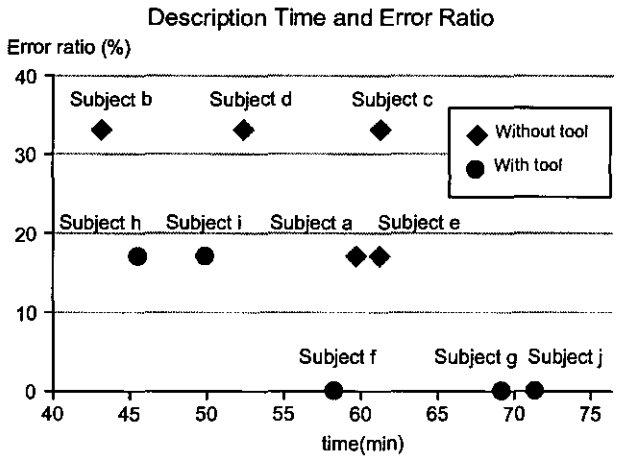


Figure 5: Time for describing and error ratio

the navigation tasks with errors in relation to all navigation tasks. The results of the experiments provide evidence supporting the effectiveness of the system in the reduction of information providers' efforts and in minimizing navigation errors. The proposed functions are effective for hypermedia systems in which information providers want to guide users properly.

Our future research will focus on an enhanced authoring tool and an enhanced rule function.

REFERENCES

1. Brusilovsky, P. L.: Methods and Techniques of Adaptive Hypermedia, User Modeling and User-Adapted Interaction, Vol.6, No.2-3, (1996), pp.87-129.
2. Brusilovsky, P. L.: Intelligent Tutor, Environment and Manual for Introductory Programming, Educational and Training Technology International, Vol.29, No.1, (1992), pp.26-34.
3. Brusilovsky, P. L., Eklund, J. and Schwarz E.: Web-based Education for All: A Tool for Development Adaptive Courseware, Computer Networks and ISDN Systems (Proc. of the 7th International World Wide Web Conference), Vol.30, (1998), pp.291-300.
4. Boyle, C. and Encarnacion, A. O.: MetaDoc: An Adaptive Hypertext Reading System, User Modeling and User-Adapted Interaction, Vol.4, No.1, (1994), pp.1-19.
5. De Bra, P. and Calvi, L.: AHA: a Generic Adaptive Hypermedia System, Proc. of 2nd Workshop on Adaptive Hypertext and Hypermedia (1998), http://wwwis.win.tue.nl/ah98/Proceedings.html.
6. De Bra, P., Houben, G. and Wu, H.: AHAM: A Dexter-based Reference Model for Adaptive Hypermedia, Proc. of Hypertext'99 (1999), pp.147-156.
7. Chittaro, L. and Ranon, R.: Adding Adaptive Features to Virtual Reality Interfaces for E-Commerce, Interna-

tional Conference on Adaptive Hypermedia and Adaptive Web-based Systems, LNCS 1892 (2000), pp. 86-97.

8. Gonschorek, M. and Herzog, C.: Using Hypertext for an Adaptive Helpsystem in an Intelligent Tutoring System, Proc. of AI-ED'95, 7th World Conference on Artificial Intelligence in Education (1995), pp.274-281.
9. Hohl,H. et al.: Hypadapter: An Adaptive Hypertext System for Exploratory Learning and Programming, User Modeling and User-Adapted Interaction, Vol.6, No.2-3, (1996), pp.131-156.
10. Perez, T. et al.: HyperTutor: From Hypermedia to Intelligent Adaptive Hypermedia, ED-MEDIA'95 - World Conference on Educational Multimedia and Hypermedia (1995), pp.529-534.
11. Rety, J.: Structure Analysis for Hypertext with Conditional Linkage, Proc. of Hypertext'99 (1999), pp.135-136.
12. Rich, E.: Users Are Individuals: Individualizing User Models, International Journal of Man-Machine Studies, Vol.18, (1983), pp.199-214.
13. Wu, H, De Bra, P., Aerts, A. and Houben, G.: Adaptation Control in Adaptive Hypermedia Systems, Proc. of International Conference on Adaptive Hypermedia and Adaptive Web-based Systems, LNCS 1892 (2000), pp. 250-259.
14. Yanagisawa, A. and Matsumoto, H.: Internet Value Chain Marketing, (1998), SCC

APPENDIX

Data for Experiment

We created educational content for learning computer science. As regards the size of content, there are 103 nodes with 177 links. Table 5 shows the meaning of the user parameters. Table 6 shows the meanings of the classes. Table 7 shows the contents of the task, the number of nodes where navigation rules should be defined, and the types of the rules. The abbreviations "nu", "gu", and "np" in Table 7 stand for the node user rules, the general user rules, and the node path rules.

Table 5: User parameters for the experiments.

User parameter	Meaning
1	The degree of interest in the area of networks
2	The degree of interest in the area of system development
3	The degree of interest in the area of hardware
4	The degree of interest in the area of operating systems
5	The degree of knowledge in the area of networks
6	The degree of knowledge in the area of system development
7	The degree of knowledge in the area of hardware
8	The degree of knowledge in the area of operating systems

Table 6: Classes for the experiments.

Class	Role
A	Offering a question
B	Offering a response when the user answers the question correctly
C	Offering a response when the user answers the question incorrectly
D	Offering an explanation
E	Topic-related class (networks)
F	Topic-related class (system development)
G	Topic-related class (hardware)
H	Topic-related class (operating systems)

Table 7: Tasks in the experiments.

Task	Contents	No. of node	Rule type
1	Hide links to the teaching materials that the user is not interested in. This is based on the degree of interest for the four areas.	1	gu
2	Provide questions first, then provide explanations for the user whose degree of knowledge is high. Provide explanations first, then provide questions for the user whose degree of knowledge is low.	6	nu
3	Provide three questions, then change the contents of the explanation according to the eight patterns that the users could answer.	2	np
4	Provide five questions, which are ordered from basic to difficult. After the user answers all questions, provide the same questions again beginning with the first question that the user answers incorrectly. If the user answers all questions correctly he/she is finished studying. However the user is only allowed to work on each question twice.	1	np
5	The user answers questions in three areas, which are hard disk, CPU, and memory, in this order. Each area provides two questions. If the user answers even one question in an area incorrectly, he/she has to answer the same two questions again for the topic. If the user answers both of the questions correctly, he/she goes forwards to the next area.	3	np

A Complementary Approach for Adaptive and Adaptable Hypermedia: Intensional Hypertext

W.W. Wadge[1] and m.c. schraefel[2]

1 Department of Computer Science, University of Victoria, Victoria, B.C., Canada
wwadge@csr.uvic.ca
2 Department of Computer Science, University of Toronto, Toronto, ON, Canada
mc@cs.toronto.edu

Abstract. In this paper describe a methodology and an authoring/publishing tool for adaptable Web documents (user-determined adaptable Web pages) as a complementary approach to Adaptive Hypermedia. Our approach is based on intensional logic, the logic of assertions and expressions, which vary over a collection of contexts or possible worlds. In our approach the contexts are sets of values for parameters which specify the current user profile as supplied by the current Web page URL, and the latest user input. The author produces generic (multi-version) source in the form of HTML with extra markup delimiting parts that are sensitive (in various ways) to the parameters. This source (in what we call Intensional Markup Language) is translated into a Perl-like language called ISE (Intensional Sequential Evaluator). To generate the appropriately adapted individual pages, the server runs the ISE program in the appropriate context. The program produces HTML that, when displayed in the user's browser, is rendered into the desired adaptation of the requested page. Although this intensional approach was originally designed to work without any explicit user model, we can extend it (and make the documents adaptive as well as adaptable) simply by incorporating a user model that monitors the user and computes some of the profile parameters.

1 Introduction

By the implementation of a user model, adaptive hypermedia systems select the appropriate components of a hypertext space to serve to their users. Such systems depend upon some kind of evaluation of an actual user against a user profile in the model [3]. This means that users need to identify themselves to the system in terms of traits the system will recognize (age, grade in a course, sex, favorite TV program, performance on an online test, etc) before the system can present them with the appropriate information. To gain this user profile data, the system must either watch users interact with the system [6], or ask the user for a variety of information either before the user proceeds into the site or regularly evaluate the user [13] during visitation. A combination of these methods can also be used. In each case, the adaptive system will select which links a user may have available, which content the user is best suited to see, and so on [1].

In 1994, Eco referred to the differences between Mac and DOS-based machines (and their users) as the differences between Catholic and Protestant approaches to salvation: in the one, everyone is saved by Grace; in the other it is only by one's own work that one reaches heaven.[1] In hypertext, the web itself in its current graceless state may be seen as the Protestant, Calvinist analog, which implies only by dint of one's own surfing and searching may one reach the Journey's End. The above-described user model-based, adaptive approach to customized documents of select and occluded links, however, sounds remarkably reminiscent of the medieval Catholic Church, with its list of banned books, which only elite liturgists were allowed to access, or its sacred texts, which were interpreted for the common folk in Mystery Plays rather than through direct reading of the sources.

If that is the case, then what we present in the following paper may be seen as perhaps an Anglican middle ground between the protectiveness of adaptive hypermedia and the cold indifference of the raw web. That middle ground, open to user models, domain-models, adaptable and raw web hypertext is IML-authored, Intensional Hypertext. In [12] we presented an overview of Intensional Hypertext as a class of adaptable hypertext which allows the user to determine at any time which version of a document they wish to see. The previous intensional hypertexts created by IML do not require that an explicit user model be managed by the system. Rather, adaptable hypertexts hold the user model implicit and present more explicitly a model of a domain which users refine in their own way (as touched on in [8]). The approach is based on the heuristic that users know what they don't know when they see it or read it, and can, if given the opportunity which adaptable hypertexts afford, refine a document themselves, possibly better than an automated user model can.

There are obviously cases, however, when either a user or a designer may wish to take advantage of the benefits of user-modeled support in a hypertext. Therefore, we have extended IML to act as a general tool which can support either user-selected, user-model selected, or a combination of both through its parameterized, intensional (perhaps agnostic) versioning model.

In the following paper, therefore, we present a more detailed look at the markup language for intensional hypertexts, IML. We will present a general model of an adaptive/adaptable architecture to situate the potential for the intensional approach in particular to adaptable/adaptive hypertext. We will therefore review intensional versioning which underlies our approach and we will describe key attributes available to authors for creating intensional hypertexts, which can function adaptably "as is," or be combined with user-model input, or natural language generated dialogs (as pre [7]) to support an adaptive approach.

[2] and [10] present compelling work on the creation architectures for managing the entire adaptive hypermedia authoring process, from UM to front end. We hope to contribute to this kind of approach by adding a tool for a specific aspect of this process: a kind of model-neutral front end authoring tool, IML, that can manage input from adaptive and adaptable sources, but that specifically embodies an intensional approach to managing these versionable sources. We hope by this presentation to investigate where, through use of tools such as these, we might (a) be able to define a

[1] From Umberto Eco's back-page column, "La bustina di Minerva," in the Italian newsweekly *Espresso* (September 30, 1994).

Paper contact: mc@cs.toronto.edu

heuristics for building complementary adaptable, adaptive texts for enhanced user experience and (b) consider how this approach may suit an Open Adapative/Adaptable Hypermedia System.

2 General Adaptive/Adaptable Architecture and the Role of Parameters

The goal of adaptive and adaptable hypertext[2] is of course to make reading easier for the readers; but at the same time it can greatly complicate the task of authors and publishers.

By the publisher we mean the hardware and software (usually, a web server and/or processes that pass information to that server), which produces the currently needed parts of the document, sends these to the reader, and handles reader requests. The publisher of an adaptive document in particular must, by definition [1], build and maintain a user model. Each time the publisher sends another part of a document to the reader, the publisher takes the current state of the reader model into account and modifies the document part accordingly.

In general there are many different possible states (configurations) of a reader model - many different possible reader profiles. In adaptable hypermedia, there are likewise multiple possible instances (or versions) of a domain object, such as the large color view of an image with English captions or the small, black and white one with French captions. The publisher is therefore responsible in each case for delivering a whole family of variants of a document (one for each possible profile or domain instance), rather than one single monolithic document.

Before the publisher can deal with this content, however, the content itself must be generated. Even if natural language processing is brought to bear to synthesize a variety of components for a specific context, such as the HealthDoc work [4], the multiple components to serve that synthesis must still be generated or retrieved from somewhere. Authors of adaptable and adaptive "documents" are responsible for writing a whole family of (admittedly related) documents. Because these documents themselves are hypertexts, this means that the link topology of the document should be subject to adaptation, and not just the contents of individual pages. While this concept of link versioning and multipointing links is familiar in adaptive hypermedia [3] it is not commonly represented in web-based hypertexts, and we are particularly interested in the Web case. We are also interested in making it easy for authors to incorporate combinations of adaptive and adaptable approaches as appropriate.

---

[2] To be clear, by adaptive hypermedia, we refer to hypermedia systems which rely on a user model to support the delivery of user-specific content. By adaptable, we mean that users select from a variety of parameters to adapt the hypertext to their needs. A stateless version of adaptable hypertext can be created in client-side JavaScript, for instance, in which users can change the color of the background of a page dynamically. IML as a server side process can maintain the state of such a change across any other page selections made on that site.

A (hardly original) way to support these varieties of inputs is to provide the publisher system with some kind of generic meta-document source. The publisher then generates actual parts from this source based on the current user profile, the current page, and the latest user input.

In the following sections of this paper, we describe a methodology and an authoring/publishing tool which can be understood as the result of using intensional logic to formalize these general observations about adaptive/adaptable architecture. We hope by this to provide a general front-end or author-based tool, which will facilitate the authoring of versionable, web-based hypermedia which can then be supported by either adaptive, adaptable or both kinds of inputs.

3 The Intensional Approach

Our approach is based on intensional logic [16], a natural choice for versioning, since it is the logic of assertions and expressions which vary over a collection of contexts or possible worlds. In our approach the contexts are sets of values for parameters which specify the user profile (if one exists), current page, and user input. It is important to note two aspects of this intensional approach which distinguish it from other forms of parameter selection. One is *version refinement* and the other is *best fit*. We describe these attributes and their significance in detail below.

3.1 Intensional Versioning

The intensional (possible-worlds) approach to versioning was originally developed by Wadge and Plaice [11] for use in configuring families of programs from families of components. Most software configuration tools work bottom-up, and allow the user to create a variant of the program by selecting different variants of the components. In [11], by contrast, each variant is described/determined by "version expression" assigning values to parameters. For example, the expression

`processor:ppc+OS:8+language:french`

might indicate the french language version for a PPC macintosh running OS 8.

A particular version of a program is configured, as usual, by assembling particular versions of components. In intensional versioning, however, the component choices are not arbitrary; instead, they are determined by the expession for the version requested. For this to work it is necessary that each variant of each component be labeled with a version expression. In the simplest case, when configuring version V of a program, we use version V of each component. However in general we do not require that each component have a version whose label is exactly V. If there is no such component, we are allowed to choose a component whose label is strictly more generic than V, in that it omits some of the parameter values prescribed by V.

For example, suppose that we wanted the `French PPC OS 8` version described above. It may be that there is no version of the (say) windowing component which has exactly that label; but that there is one with the label `processor:ppc+OS:8`.

Paper contact: mc@cs.toronto.edu

Then we can use this component, the justification being that the windowing component is language independent.

In general, although we do not require that there be a component labelled V, we insist that there among the more generic alternatives there is one which is best in that it is the most specific. If there is a most specific alternative component, that one must be taken. If there is no such best alternative, the configuration attempt is abandoned.

The net effect of the best-generic-alternative or *best-fit* rule is that a relatively small number of component variants can form the basis for a much larger family of program variants. The leverage comes from the fact that different versions of the program can share the same versions of particular components (for example, as the `French` and `English` programs `PPC OS 8` share the `PPC OS 8` windowing component). The more-generic relationship between versions is thus an inheritance relationship, so that more specific versions inherit (that is, use by default) more generic versions of components.

3.2 IHTML, ISE, and IML

The system presented here is the latest stage in an effort, as described in [12], begun in 1996, to produce an authoring/publishing system based on an intensional versioning scheme analogous to the one just described for programs.

In these systems authors define Web sites as linked collections of pages, each of which can exist in many different versions. A request issued by a browser consists of a conventional URL together with a version expression. The server software generates the requested version of the requested page by combining the particular version expression with the (usually generic) source.

The first such system was Intensional HTML [18], a minimal extension of ordinary HTML. Initially, the most important feature of IHTML was that it allowed multiple source files for the same page, each labeled (as above) with a version expression. When a request arrives for a particular version of a particular page, the IHTML server (an Apache server plug-in) uses the best-alternative rule described above to select the appropriate source file.

In generating the actual HTML all the copy and most of the mark up is duplicated verbatim. Links, however are specialized, so that they connect analogous versions of pages. For example, the IHTML source for page A might contain an ordinary-looking link to page B. When a request arrives for the HTML source of (say) version `language:french+level:expert` of page A, the link is changed to lead to the `language:french+level:expert` version of page B.

IHTML also allowed what we call transversion links: these are links which connect one version of a source page with a different version of the target page. For example, in the source for A, the author can include a link to B in which it is specified (in the tag attributes) that target version has the language parameter set to english. When the `language:french+level:expert` version of A is requested, the IHTML link is turned into an HTML link to version `language:french+level:expert` of B. With transversion links the author can allow the user to change parameter values (and therefore adapt the target page) simply by following a link.

IHTML worked well enough for small sites but turned out to be impractical for larger projects. Maintaining a large number of small files proved to be a real headache. We added a version-sensitive conditional tag (iselect) [15] that allowed us to refine components within documents themselves, rather than refer to external files. This case statement gave us two things: a way for authors to define within-page version selections easily, and, as a side effect of this, a way to create Nelson-like stretchtext sections with markup rather than, for instance, a great deal of browser-specific JavaScript code. That said, for large files with many degrees of stretchtext, we sometimes needed a proportional number of additional lines of markup to define the regions. This required no great expertise, but it was tedious to do. The real problem was that IHTML was not author extensible, so we could not create a simple wrapper for the stretch text case, for instance. Furthermore, adding new tags for genuinely new features involved extending the specialization software.

In 1998 Paul Swoboda, then an MSc student, decided to abandon markup as the basis for intensional Web versioning.[3] He designed and implemented ISE, which is to Perl as IHTML is to HTML [14]. Since ISE is a full featured algorithmic language, there is no *a priori* limitation on the kinds of features it can support. Also, since it has functions and procedures, sophisticated versioned Web sites do not necessarily translate into extremely large ISE programs.

ISE, however, is still a programming language, and therefore is inherently more complex than a markup system. We therefore added IML as a kind of front end to retrieve the obvious advantages of markup authoring. IML offers all of the power of IHTML but, since it is implemented using macros, it is easily extensible, even, to a limited extent, by authors without any programming skills or knowledge of the language ISE (we explore this in more detail in section 4.8 below). Adding new constructs involves adding definitions for new macros to a file. These definitions can be intricate, but only have to be written once, and then can be used by authors who have no knowledge of ISE or (for that matter) of any other programming language.

In the next section of the paper, we describe some of the most commonly used macros in IML. Some can be used simply to add stretch text easily to Web pages; others can be used as conduits for more sophisticated versioning applications.

4. Common IML authoring patterns

IML is designed to make the full power of ISE available to authors without obliging them to become programmers; so that they remain authors in the colloquial sense of the word. IML can be understood as an extension of HTML. An IML source file consists of text together with markup which is either ordinary HTML tags, interpreted in the usual way, or special IML tags (we call them intensional tags), which are context sensitive. The two kinds of markup can be mixed and nested.

[3] As described by Yannis Kassis, there is some debate within the Intensional Hypertext group about the IHTML-like markup approach vs the IML/ISE approach described here [5]. For our purposes, however, the effects both approaches wish to support are the same: efficient, effective authoring of parameterized, versionable documents.

Paper contact: mc@cs.toronto.edu

In this section we will use, for the sake of clarity, a (hypothetical) XML-style syntax for the IML markup. At present, however, IML is based not on XML but on the macro language of Groff. The (primarily pragmatic) reasons for using Groff instead of XML will be discussed in section 6, Future Directions, below.

4.0 Plain HTML

Since IML is an extension of HTML, the simplest IML documents are plain HTML documents, recast as IML documents. The recasting process is trivial: the `<html>` and `</html>` tags are replaced with `<iml>` and `</iml>` tags. Everything between the two remains unchanged, including JavaScript. The IML implementation by default treats markup as text and does not try to analyze non-IML tags.

One of the simplest ways to create an IML document is to modify an existing HTML document. The first step is to replace the `<html>` and `</html>` tags as described, and the result can be fed to the IML implementation, which will produce a corresponding ISE program (consisting mainly of a large statement). The page produced by running this program should be identical to the original.

4.1 Parameter Substitution

One of the simplest (but also most useful) features of IML is the ability to retrieve the value of a parameter, in almost any context. For example, suppose the page was a message and we want to insert the adressees surname in the salutation. Using the XML-style syntax, we can write

```
 <p> Dear Mr/Ms <impl-param name="surname">,
pleaseconsult … </p>
```

Then the generated ISE program is run in an environment in which the surname parameter has the value Jones, it will produce

```
<p> Dear Mr/Ms Jones please consult … </p>
```

The 'real' groff-based IML implementation allows a more concise notation: the parameter name preceded and followed by ##. This nonstandard notation has the advantage that it can be used even inside HTML tags. For example, <body bgcolor=##EGG##> will set the background color to the current value of the EGG parameter (whatever that may be).

4.2 Conditional Inclusion

Of course, it would be very strange to personalize a message by including the adressee's name, and then take no account of their gender. If the current context has a title parameter we can include it directly; but if necessary we can choose the title on the basis of the gender parameter (if there is one).

We can make the choice using IML's <iselect> construct, as follows:

```
<iselect>
<icase version="gender:F">Ms</icase>
<icase version="gender:M">Mr</icase>
</iselect>
```

In general, there can be any number of <icase...>...</icase> sections between the opening and closing `iselect` tags. Each tag specifies a version attribute. Only those icase sections whose attributes are consistent with (can refine too) the current context are considered. If there is exactly one such *eligible* section, the enclosed copy is included. If there is no such section, the iselect clause contributes nothing (and is essentially ignored). If there is more than one eligible section, the best fit rule is used to select the a single icase section.

Suppose, for example, that there is also a parameter mstatus with value Y to indicate the individual is married, N to indicate not married, and vanilla to indicate no information. The following iselct clause, together with the best-fit rule, specifies the traditional rules for choosing a title:

```
<iselect>
<icase version="gender:F">Ms</icase>
<icase version="gender:F+mstatus:Y">Mrs</icase>
<icase version="gender:F+mstatus:N">Miss</icase>
<icase version="gender:M">Mr</icase>
</iselect>
```

4.3 Levels

One common method for adapting documents is to allow one or more integer-valued parameters which specify some kind of depth of treatment. This parameter could measure the (estimated) level of the reader's expertise, or the degree of detail deemed to be appropriate.

The author indicates in the source document either at the document level, or within each section, a minimum level required for that section to appear. In publishing the document, the value the parameter has in the publishing context is used to producer a version which is longer or shorter, according to whether the value in question is greater or smaller.

4.3.1 Stretch Text. The simplest approach is to supply the author with tags, say `<iml-incdepth>` with the understanding that any text enclosed beteween `<iml-incdepth>` and `</iml-incdepth>` has an associated depth which is one greater than that associated with the enclosing text.

For example, suppose that at the topmost level the source contains

```
<p>At the southern end of Vancouver Island we find
Victoria<inc-depth parm="detail">, the provincial
capital</inc-depth>.</p>
```

The simplest way to produce this kind of telescoping text is to take an existing (mono-level) document and add `<iml-incdepth>` tags. Our experience is that well

Paper contact: mc@cs.toronto.edu

written text lends itself readily to this kind of factoring, even though the authors never intended to produce less than the full-length version.

One minor difficulty is that omitting sections may cause grammatical errors, for example, verb forms which have to change from plural to singular in the shorter versions. This problem is easily solved by allowing <iml-incdepth> to have an extra alternate attribute of text to be included in case the enclosed copy is to appear. For example, the source

```
<p>At the southern end of Vancouver island we find
Victoria<iml-incdepth>, Saanich, and Oak Bay</iml-
incdepth>, which <iml-incdepth alt="enjoys">enjoy</iml-
incdepth> a very mild climate.
```

4.3.ii Drop Text. By drop text we mean a single block of text that can be made to appear or disappear separately, without affecting any other part of the document. Drop text can be considered (and implemented) as a simple kind of telescoped text in which the dropped block has its own private two-valued depth dimension.

A very basic kind of drop text consists of an anchored heading and a body, with the heading always visible. When the text is absent, following the heading link causes the text appear, and when the text is present, following it makes it disappear (with nothing else changed). Sections like this can be specified by the iml-drop tag which encloses the disappearing text. The "heading" attribute determines the anchored heading. For example,

```
<iml-drop heading="Other cities to visit">
Nanaimo, Sooke, Esquimault, Duncan. </iml-drop>
```

4.4 Stereotypes

Depth parameters can be considered a special case of "stereotype parameters" [1] that have a discrete set of values (not necessarily ordered) each of which represent a kind of user profile. For example, we might have a "purpose" parameter which specifies the reason for a visit to an attraction, with values such as "tourism", "shopping" and "business."

We supply authors with a tag `iml-stereo` that includes enclosed text when the value of the "purpose" tag has the value specified. For example,

```
<p>The parliament buildings are located just east of
the Empress<iml-stereo param="purpose" type="tourist">,
and are open to visitors most mornings</iml-stereo>.
</p>
```

The parameter selection can be set up to be determined by the user selecting a link to establish the "tourist" version of the page, or by a user model, passing through the tourist value to this version of the page, or by both. The parameter values associated with the reader queries can be combined with those computed by a UM and passed on an equal footing to the ISE interpreter running the ISE version of the adaptive/adaptable document.

The same, of course, can be said of all the other constructs here: they do not take into account the origin of the parameter values they examine. That *should* mean more flexibility for authors and system developers: it becomes easier to determine where and under what conditions parameters are to be interpreted by user, whether selection or user model (see the diagram in section 5 below).

4.5 Transversion Links

As we explained earlier, IML supports adaptable hypertext, in which the users make explicit choices about the form and content of the document they are reading. These choices must be captured as parameter values but the reader (user) need not know anything directly about IML version expressions. Instead, the author can arrange that particular hyperlinks be associated with changes in parameter settings.

The simplest form is what we called in IHTML a transversion link: an ordinary a tag with an extra vmod attribute which specifies updates. For example, suppose the IML source has the following.

```
Si vous preferez, on peut vous presenter ces
renseignments <a href=info vmod="lang:french">en
francais</a>.
```

When the document is rendered, this becomes a link to the version of the "info" page that is identical to the current version except that the `lang` parameter is set to `French`. Note that the values of the other parameters are carried across the link unchanged: if the reader was looking at the `purpose:tourism+lang:english` version of the page on which the link appears, clicking it will take the reader to the `purpose:tourism+lang:french` version of the info page.

Similar tags allow authors to enable reader-adaptations using menus, forms, image maps and the like. Notice that transversion links may alter more than one parameter. For example, the author could offer a link

```
<a href=expl vmod="lang:french+level:newbie"> J'ai rien
compri</a>
```

that allows the user to self-identify as an inexperienced francophone.

It is also worth pointing out that the reader is not necessarily aware of the existence of some of the parameters being changed, and may not immediately see any effect of their change. In fact, the document might be completely insensitive to the values of these parameters. Instead, these values might be meant for the User Model, which monitors all the user requests.

4.6 Nesting

It goes without saying that the constructs described can be nested - but we'll say it anyway. For example, we can have telescoping text in which deeper level copy has both tourist and business variants; or variants of one page corresponding to different stereotypes can offer different sets of options (as transversion links).

Paper contact: mc@cs.toronto.edu

The advantage of nesting, of course, is that it gives us a component model: the author can construct very elaborate adaptive/adaptable documents using a relatively small set of primitives. This is a vital part of what makes IML a general-purpose tool.

4.7 Extensibilty

The other *sine qua non* of a component model is the ability to define new components. In IML this takes two forms, which we might call "author-extensibility" and "implementer-extensibility." The first is simply abstraction (sometimes called encapsulation), the ability to give a name to combination of already existing constructs and invoke the combination using this name, possibly with parameters.

In terms of XML-style syntax, this means the ability to define new tags by expressions involving existing ones. A very simple example would be to define the tag `</hello>` as appropriate to the word or phrase in the language specified by the value of the `lang` parameter. We have already seen how to do this, but the resulting source expression can be very long, especially if there are a number of alternate languages. Abstracting this expression as `</hello>` shortens the source and protects part of it from needing alteration in case we change our language options.

IML's author-extensibility comes directly from groff's macro definition facility. Unfortunately this feature is tied closely to groff's syntax and can't be described without going into details of groff, which we would rather avoid (those interested can consult [17]). At the moment it is not clear to us what a full XML-style syntax would look like, so we won't offer even hypothetical examples of parameterized definitions (we return to this point in section 6, below).

IML is also implementer-extensible. This means simply that people who understand ISE and its versioning system can write macros, which produce fragments of ISE programs, rather than just more IML. The IML-to-ISE macros tend to be short but dense but there is no other way to define the primitive constructs. Fortunately, with nesting and abstraction, a few complex implementer-extensions go a long way. These have been used currently to specifiy a variety of versioning effects, as per [12].

In addition, ISE itself is extensible - it allows procedures to be written in C but called in ISE. This allows the implementer, for example, to provide authors with tags, which access a database. The implementer writes the access primitives in C then defines the tags with macros which generate ISE source which calls these primitives.

4.9 Best Fit

One of IML's most useful features, and perhaps the trait which distinguishes it the most from other markup based systems for version support like [9] is its ability to combine a special kind of author-extensibility with the *best fit* protocol as introduced in section 3.1 above. An IML author is able to give a name to a block of IML code, invoke the code using the name, but define the abstraction incrementally in a case-by-case manner, with inherited defaults. The extra definitions are labeled with version expressions, much like the branches of an iselect. But they can be added one by one

and scattered throughout the source. This kind of abstraction is based on ISE procedures, which themselves can be defined incrementally and with defaults.

For example, we do not have to give a single monolithic definition for the tag `</hello>` mentioned above. We can start by giving default definitions, say "hello" in English, then use it wherever we want. If we do nothing else, `</hello>` always expands to "hello."

Next, we can add special case definitions of `</hello>` as "Bonjour" and "Ola" labelled with expressions `lang:french` and `lang:spanish` respectively. These definitions will now override the default, which nevertheless remains in force for languages other than French and Spanish, Later, we may decide that Hispanic business people might prefer something more formal and add fourth definition of `</hello>` as "Buenos Dias", labelled with `lang:spanish+purpose:business`. This overrides the "Ola" default for hispanics, but Spanish speaking tourists will still see the more informal greeting.

In this way authors can concentrate their efforts to produce documents that are more carefully adjusted for certain audiences. And they can improve the adaptability gradually, without large-scale revisions at each step. Indeed, authors can encode these versions in advance of a specific UM taking advantage of them, should that approach be appropriate. Until the UM is ready to take these into account, they may not appear to the user.

5. Publishing Modes

One great advantage of the approach described here is that it is possible to produce useful, version-based sites with only a simple UM, or none at all. All of the documents produced earlier using IHTML fall into this last (UM-free) category. This kind of site is referred to as "adaptable" rather than "adaptive" because the variations are the result of the user making explicit choices (mainly by choosing links). For example, in an adaptable site the user might identify herself as a tourist by following a link anchored to an image of sunglasses, request French language material by selecting that language from a menu choice, and open up the drop section on "Other Cities" by following the link anchored to that heading.

Recall that transversion links carry over, unchanged, all parameter values that are not altered by the link. The result is that the URLs of the pages the reader views normally accumulate the choices made (and hence act as a very simple implicit user model). One very simple case of this publishing mode results if the author uses only regular HTML tags. Then no parameter values at all are generated, the site is neither adaptive nor adaptable, but the author still has the advantage of IML's author/implementer extensibility.

On the other hand, even a very simple UM can greatly improve the effectiveness of the site. For example, the UM could simply guess at the reader's language preference by examining the IP address associated with the reader's request. Or it could supply the parameter values which specify the date and time (definitely something the reader should not have to provide). In either case, the author could take advantage of this

Paper contact: mc@cs.toronto.edu

extra UM-supplied information to, say, select the kind of visitor information displayed. A more sophisticated model could have access to a database - a regular DBMS or perhaps just a UNIX text file. This UM could extract the reader's identity (they might be asked to provide it), look up the individual in the database, then pass on (to the ISE interpreter) associated parameter values, for example the reader's degree program or midterm marks.

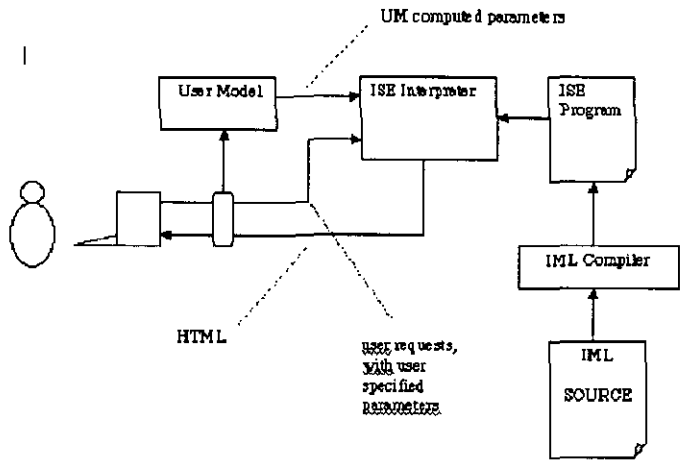


Figure 1: Model-neutral IML architecture for versioning

Of course, there is no limit to the sophistication of a user model. Our approach to publishing puts only one constraint on the nature of the model: that it be able to pass on the relevant parts of its current state as a set of values for parameters.

6. Conclusions and Research Directions

In this paper we have presented a mechanism for letting authors author versionable hypertext without being primarily constrained by how those versions will be delivered, whether by user-determined link choices or UM parameter determination. We anticipate that this approach offers ecumenical advantages for hypermedia researchers with either adaptable or adaptive biases since it frees adaptable/adaptive hypermedia researcher from designing their own author layer for their system. They can simply pass their parameters to the IML/ISE front end. The IML/ISE tools for creating the adaptable/adaptive author layer also foreground the intensional approach to versioning, which offers particular advantages not available in other version-markup systems, namely, version refinement and best fit. One of the chief advantages of version refinement is that it allows a large number of potential versions to be created from a relatively small set of primitives, or for that matter, from a small set of source files. Likewise, the abstraction/encapsulation facility collaborates with the version best fit protocol to allow incremental refinement of the version families, in which the author specifies general defaults first, then overrides where appropriate. We suggest that this intensional approach to handling versionable content, as embodied in

the IML methodology, can act as general front-end tool for preparing versionable content to be delivered to a user, whether the versioning is managed by an adaptive or adaptable approach. That said, there are certain attributes of the current IML implementation that need further refinement and evaluation for the mehod and tool to be successful in this open context. We describe these below.

Current IML Syntax. The most urgent task in IML is to abandon groff in favor of an XML-compatible syntax, like that used in our hypothetical examples. For example, the drop text example presented earlier is actually, currently, written as

```
.biml-drop "Other cities to Visit"
Nanaimo, Sooke, Esquimault, Duncan.
.eiml-drop
```

Groff's quaint line-oriented syntax is simple enough once you get use to it. Groff is distributed with every Linux implementation and its implementation is solid. Also, nesting and abstraction are completely straightforward. The current groff-based markup is, in a sense, very successful and has allowed a number of people (beyond the authors) to experiment with the ideas presented here.

Nevertheless, it is syntactically incompatible with XML and on those grounds alone hopelessly obsolete. And even if we ignore syntactic issues, there are also serious problems with groff's form of abstraction. The arguments of groff macros are listed in a fixed order, rather than assigned as values of parameters. This makes it practically impossible to arrange sets of default values or generate different output according to the presence or absence of particular arguments. Instead, one has to define whole families of related macros with separate complete definitions; a very tedious process and one, ironically, at odds with the IML philosophy of versioning.

The major challenge in replacing groff is designing (not to mention implementing) an XML-compatible abstraction notation. This effort is only just under way.

Evaluation. We have not yet implemented a large-scale intensional hypertext project and exposed it to rigorous user evaluation. We are in the process of doing this for two reasons in particular. The first is that we wish to understand better how we need to support authors in developing intensional hypertext documents, since version management and version visualization or tracking are not insignificant additional considerations for authors or teams developing content. The second is that we wish to see if we can, in participation with authors and site users, develop heuristics for a versionable site's evolution, from plan, to integration with adaptable or adaptive techniques, to maintenance and refinement. We are engaged in a project with Biomedical Communications at the University of Toronto to develop a multi-user site on breast cancer and the sentinel node biopsy procedure which will also be the test bed for our proposed evaluation.

Paper contact: mc@cs.toronto.edu

7 Acknowledgements

We wish to thank Michael Milton and Adele Newton of the Bell University Labs Research Program at BCE and the University of Toronto for their continued enthusiastic support of this research. Thanks also to Paul Swoboda and Neil Graham for their development efforts and insights throughout the lifecycle of IML.

8 References

1. Brusilovsky, Peter, Methods and techniques of Adaptive Hypermedia. Adaptive Hypertext and Hypermedia. Kluwer Academic Publishers, Amsterdam 1998, 1-43,

2. De Bra, Paul, Licia Calvi. Towards a Generic Adaptive Hypermedia System. Proceedings of the 2nd Workshop on Adaptive Hypertext and Hypermedia HYPERTEXT'98, Pittsburgh, USA, June 20-24, 1998, 5-12.

3. DeBra, Paul. Design Issues in Adaptive Hypermedia Application Development. 2nd Workshop on Adaptive Systems and User Modeling on the WWW. Toronto (1999) 29-39.

4. Hirst, Graeme, Chrysanne DiMarco, Eduard Hovy, and Kimberley Parsons, "Authoring and generating health-education documents that are tailored to the needs of the individual patient." In Anthony Jameson, Cécil Paris, and Carlo Tasso, (editors), User Modeling: Proceedings of the Sixth International Conference, UM97, (Chia Laguna, Sardinia, Italy), Springer, 1997, 107–118.

5. Kassis, Yannis. Noema as Alternative to ISE/IML. Unpublished Hypermedia Graduate Course Paper, U of Toronto, Canada, 2001.

6. Kushmerick, Nicholas James McKee, and Fergus Toolan. Towards Zero-Input Personalization: Referrer-Based Page Prediction. P. Brusilovsky, O. Stock, C. Strapparava (Eds.) Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, Trento, Italy, August 2000. Proceedings. Springer-Verlag, Lecture Notes in Computer Science, Vol. 1892, 133-143.

7. Moore, Johanna D. Participating in Explanatory Dialogues: Interpreting and Responding to Questions in Context. MIT Press, Cambridge, Mass. (1995).

8. Murray, Tom, Tina Shen, Janette Piemonte, Chris Condit, and Jason Thibedeau. Adaptivity for Conceptual and Narrative Flow in Hyperbooks: The MetaLinks P. Brusilovsky, O. Stock, C. Strapparava (Eds.) Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, Trento, Italy, August 2000. Proceedings. Springer-Verlag, Lecture Notes in Computer Science, Vol. 1892, 155 –166.

9. Paradis, Franois and Anne-Marie Vercoustre and Brendan Hills. A Virtual Document Interpreter for Reuse of Information, Proceedings of Electronic Publishing '98, Saint-Malo, France, 1-3 April, 1998. Springer-Verlag, Lecture Notes in Computer Science 1375, 487-498.

10. Petrelli, Daniela, Daniele Baggio, and Giovanni Pezzulo Adaptive Hypertext Design Environments: Putting Principles into Practice P. Brusilovsky, O. Stock, C. Strapparava (Eds.) Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, Trento, Italy, August 2000. Proceedings. Springer-Verlag, Lecture Notes in Computer Science, Vol. 1892, 202-213.

11. J. Plaice and W. W. Wadge, A New Approach to Version Control, IEEE Transactions on Software Engineering, March 1993, 268-276.

12. schraefel, m.c. "ConTexts: Adaptable Hypermedia." P. Brusilovsky, O. Stock, C. Strapparava (Eds.) Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, Trento, Italy, August 2000. Proceedings. Springer-Verlag, Lecture Notes in Computer Science, Vol. 1892, 369-375.

13. Stern, Mia K. and Beverly Park Woolf Adaptive Content in an Online Lecture System P. Brusilovsky, O. Stock, C. Strapparava (Eds.) Adaptive Hypermedia and Adaptive Web-Based Systems International Conference, AH 2000, Trento, Italy, August 2000. Proceedings. Springer-Verlag, Lecture Notes in Computer Science, Vol. 1892, 227-238.

14. Swoboda, P. Practical Languages for Intensional Programming, MSc Thesis , (Computer Science), University of Victoria, Canada (1999).

15. Wadge, W. Intensional Logic in Context. Intensional Programming II, World Scientific Publishing, Singapore, 2000, 1-13.

16. Wadge W., G. Brown, m. c. schraefel, T. Yildirim, Intensional HTML, Proc 4th Int. Workshop PODDP '98, Springer Verlag LNCS 1481 (1998) 128-139.

17. Wadge, W. Intensional Markup Language. Peter Kropf, Gilbert Babin, John Plaice, Herwig Unger (Eds.): Distributed Communities on the Web, Third International Workshop, DCW 2000, Quebec City, Canada, June 19-21, 2000, Proceedings. Lecture Notes in Computer Science, Vol. 1830, Springer, 2000, 82-89.

18. Yildirim, Taner. Intensional HTML Masters Thesis, U of Victoria, Canada. (1997).

Paper contact: mc@cs.toronto.edu

A Modelling Method for Designing Adaptive Hypermedia

Toni Alatalo
Toni.Alatalo@oulu.fi

Janne Peräaho
Janne.Peraaho@oulu.fi

Department of Information Processing Sciences, University of Oulu
OWLA research group, http://owla.oulu.fi/

Abstract

This position paper presents a novel method for modelling adaptivity in hypermedia design specifications. The method is based on an approach for designing adaptive hypermedia that integrates adaptivity in the structure of the system, recognizing some objects as adaptors and developing heuristics based on what the content and functionality may adapt. The heuristics are captured in so-called shadows that may hide the potentially complex underlying algorithms from the designer. The method is intended to complement existing methodologies, such as OOHDM, which currently lack means for visualizing adaptivity aspects. An user-modelling centric example is presented, accompanied by an implementation on-line for the interested to try. Current limitations and directions for future work on the method are recognized.

Introduction

Several methods for modelling hypermedia as objects have been proposed, such as OOHDM (Schwabe&Rossi 1998), OO-HMethod (Gómez et al 2000) and an UML extension (Baumeister et al 1999). Of the existing methods, only the adaptive hypermedia design method (AHDM) supports adaptive hypermedia by including the design of user models within the methodology (Koch 1998). But even the AHDM approach does not support *the modelling of adaptivity dependencies when specifying the content and functionality of the hypermedia application.* That is, the previous modelling methods do not have means for the designer to visualize e.g. how certain features of the system depend on certain properties of the user. Here we present a modelling method for adaptive hypermedia, proposing a solution to that problem. The proposal is preliminary, and has not been thoroughly analyzed or put to practice yet.

De Bra, Eklund et al (1999) define adaptive hypermedia as a collection of nodes and links, that is accompanied by user profiles, which are used to adapt the presentation. Also Brusilovsky (1996), in his authoritative definition, specifies having a user model as the criteria for a hypermedia system to be adaptive. In our research, we attempt to approach the issue more broadly, in a more general way. While adapting to user is central in many applications, in our view it is not all there is to adaptivity, and in some cases not even a requirement for adaptive hypermedia. For example, considering mobile technologies, there are several different kinds of devices with very different user interfaces and adapting the hypermedia based

on device profiles - but not necessarily regarding any user profiles at all - makes sense. On the other hand, for example security aspects - while clearly having to do with user profiles - are not necessarily tied to the (psychological?) human characteristics often emphasized in adaptive hypermedia, but e.g. classifications based on organizational structures.

Thus, here adaptive hypermedia is understood in a more general sense: as hypermedia, that is adaptive respect to something(s), which depend(s) on the application domain. This is the basis for our view on designing adaptivity. In this paper, however, the modelling method is presented with an user-modelling centric example, demonstrating how the notation can be used to illustrate dependencies between user specific properties and behavior of the system. In this respect our approach differs from e.g. the AHAM Adaptive Hypermedia Architecture Model (De Bra, Houben & Wu 1999) in that we integrate the adaptivity in the content/functionality specification.

Also, we are aware that there is another kind of adaptivity (in the broad sense of the word) that our approach does not currently address. In our preliminary sketches we have separated between *adapting to difference* (e.g. to different users, which is the focus here) and *adapting to change* (e.g. changes in the whole systems as time goes) but in this position paper, that discussion is left out to be dealt with later. An example of a method for dealing with adapting to change can be found in the Demeter method for adaptive programming (Lieberherr 1996).

The rest of the paper is structured as follows: first, an overview of the design approach is given in the following section, then the actual modelling method is illustrated with examples, and the research position and future direction summarized in the final section.

Designing adaptive hypermedia

The modelling method under development is based on an adaptation concept which we call structural adaptation. Structural adaptive system consist of three parts (or group of system components), namely Adaptors, Heuristics, and Transformants. They can be understood as specific roles some objects in the system have with respect to adaptivity.

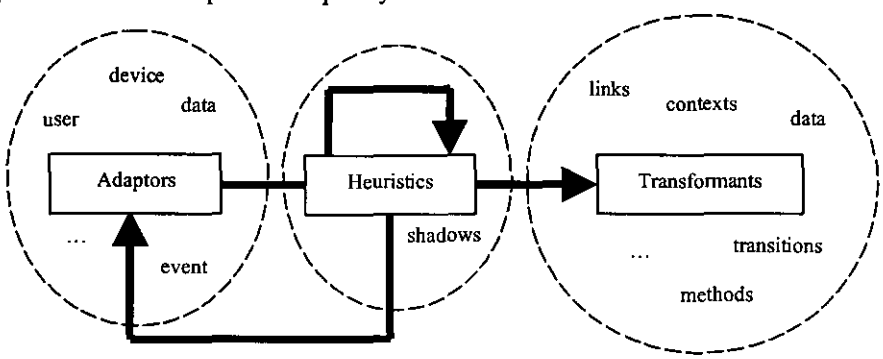


Figure 1. Adaptive system

Adaptors are system components to which the rest of the system adapts. Adaptors are defined by the application domain, system requirements and/or by software designers, so they can be anything: users, (client) devices or events, to mention just a few of them.

In order to be adaptive system's adaptors have to possess some information or attributes by which the system's adaptive components have a child-like relation. (Note that along with attributes adaptor can contain methods, too.) We call the information attached to the adaptor as adaptor's *properties*.

Transformants are system components which adapts to adaptors according to heuristics. Transformants consists of non-adaptive parts (skeleton) and adaptive parts (transformants), which are connected to non-adaptive parts through heuristic rules.

Because we are dealing with hypermedia bounded design method, one can identify and name transformants accurately. Transformants can be based upon data (related to application domain), links, contexts, transitions (links between contexts), methods, adaptors, and heuristic rules. Adaptors and heuristic rules acting as transformants is quite rare, but still possible and justified in special occasions.

Heuristics defines two-fold relations between adaptors and transformants: 1) in which adaptor's options transformant has relation to (variables and constants) and 2) what is the nature of the relation (mathematical and logical dependency).

Heuristics can be classified. Classification is based on adaptors which the heuristic rule is mainly involved. We call these heuristic classes as *shadows*. Thus one can speak of and name several shadows: user shadows, device shadows, and many more, depending on number of adaptors and shadows' objectives dictated by application domain, system requirements and/or system designer.

The shadow consists of one clause (heuristic rule), containing variables, constants, mathematical- and logical operands, which defines the relation of one or several adaptors and transformants. As a result of this each shadow has a binary output value, 0 (false) or 1 (true), which indicates adaptor's adaptive part's inclusion or disjunction. In other words, shadows acts as a logical glue which holds the transformant's skeleton and transformants together. Without these links adaptive system can not exists.

The design process of an adaptive system can be divided in six tasks based on concepts of hypermedia and adaptivity:

- design of adaptors
- design of shadows
- design of the information content
- design of contexts
- design of a navigation
- design of an user interface

Although the process issues are not the primary focus of this paper, but the modelling method, some considerations for the design process are presented here briefly.

We propose that designer approaches these tasks via design issues, which are put in form of questions. While carrying out the tasks, designer seeks answers to questions and draws related diagrams, which are utilized in software implementation phase, in base of the answers.

We suggest an incremental iterative design process, along the lines of the classic spiral model (Boehm 1988), i.e. not trying to get the design right at once but to improve diagrams, adaptors and shadows gradually. The more system requirements are taken into account in design diagrams, the need of change in adaptors' properties and shadows' rules is more apparent. To maintain the integrity of system design it is advisable to make the required adjustments right away to adaptors and shadows. Making it afterwards may break the design scheme due to shadows reusability.

Adaptive system is multidimensional and often a horrible mess. Getting a clear picture of a system being design is difficult and the risk of misinterpretation is high. The separation of adaptors and heuristics (shadows) from the rest of the system, and the ability to reuse shadows makes system design clearer, but despite of all this produced diagrams are complex. To our mind designer's mental burden should be ease by choosing a flexible design strategy and by using special design tools, which allows designer to examine the system in different viewpoints (in point of adaptors).

Modelling adaptivity

An easy way to understand the structural adaptation method, is to approach it through user adaptation. That is why the following presentation of adaptivity modelling is user centric, not because it is the only

adaptivity type supported. Furthermore, we believe that this approach deepens user modelling by forcing an user model more integral part of a system, instead of being attached on top of it.

In the following you will see partial design process of an adaptive system. The goal is to design system which contains scientific articles (publications) around our research project OWLA. These publications should be available, in different degrees, for system's users.

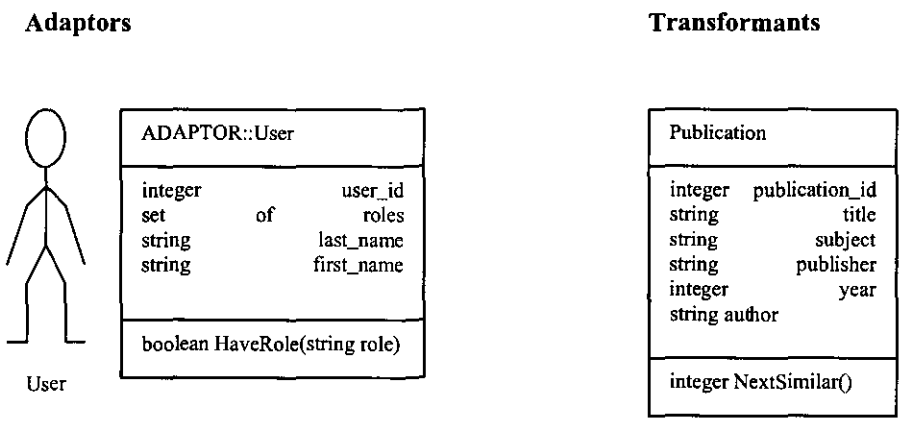


Figure 2. An adaptor and a transformant skeleton.

Modelling of an adaptive system should begin based on requirements gathering results – the overall view. The first design task is to identify and define preliminary adaptors and transformants, or adjust class definitions if such already exists.

Two objects were identified in requirements gathering: Publication and User. We start by creating corresponding classes "User" and "Publication". One of the requirements was that system adapts different users. Because of this "User" is defined as an adaptor class and "Publication" is left as a normal class (transformant). See figure 2.

In the requirements gathering we found out that system has three different users: Financiers who provides research capital, Researchers who are interested in research results (publications), and Press which is interested in new brilliant ideas. Each of these groups has own interests and demands.

Financiers require condensed and focused information. They need to know what they are paying for and how they can gain advantage of research results.

Researches should have full access to publications but because financiers are involved, we will have to limit the access rights.

The press need to know about what is being done (in general) and about new findings, if this does not conflict with interests of financiers.

We can now see that user's role (Financier, Researcher, or Press) is the dominant factor in adaptation - according to user's role, system produces different output.

This means that each publication should have own summary for each of the role: one summary for Financiers, one summary for Researchers, and one summary for Press. This also means that these summaries are Publication's class attributes - transformant attributes.

In base of this knowledge we can define shadows (see figure 3) and sketch Publication class with new attributes.

Shadows

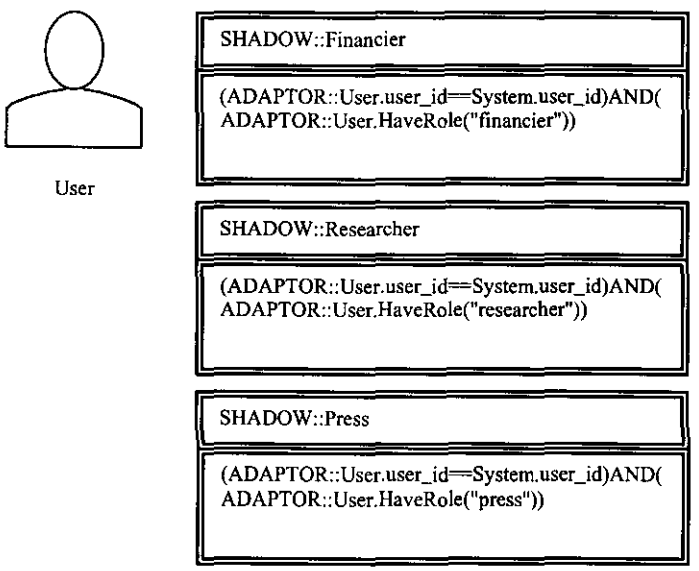


Figure 3. Shadows.

Limiting Researchers' access rights is still an open question. The policy is that researchers who are involved with the OWLA research project have right to view the related publication, other researchers may only view the summary.

We decided to secure publications by encoding them and passing the decoding key to researchers' who are qualified viewing whole publications.

We use User's class attribute "securityclass" to indicate qualification. The decoding key is also User's class attribute but it is a transformant because it is possessed only by certain group of users.

To accomplish the access restrictions we add the two new attributes to User class and define a new shadow. After this we can draw new User class where security issues have been taken account. See figures 4 and 5.

Shadows

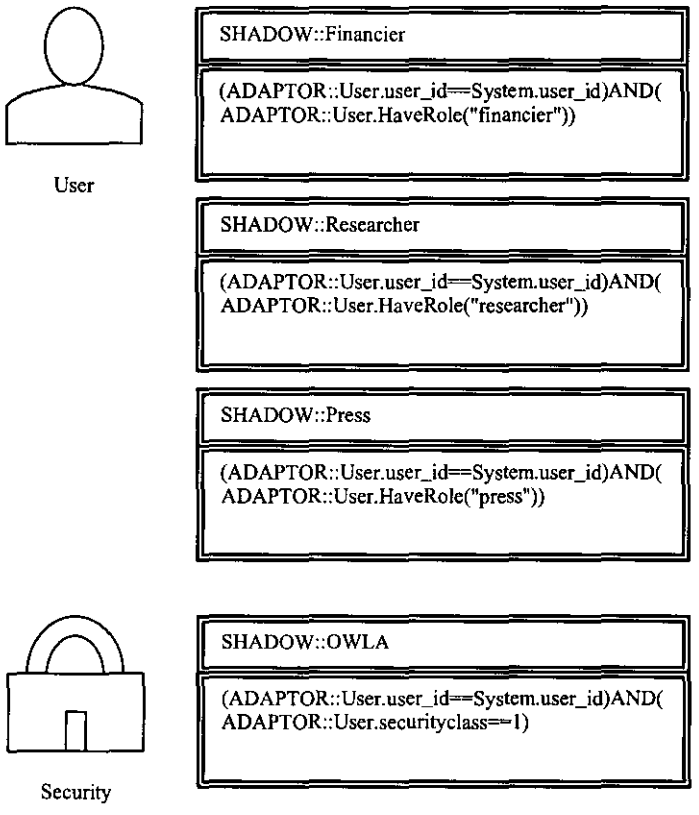


Figure 4. Completed shadows.

In the figure above you can see completed shadows, three user shadows and one security shadow, and in figure 5 is completed adaptors, one user adaptor with transformant.

Adaptors

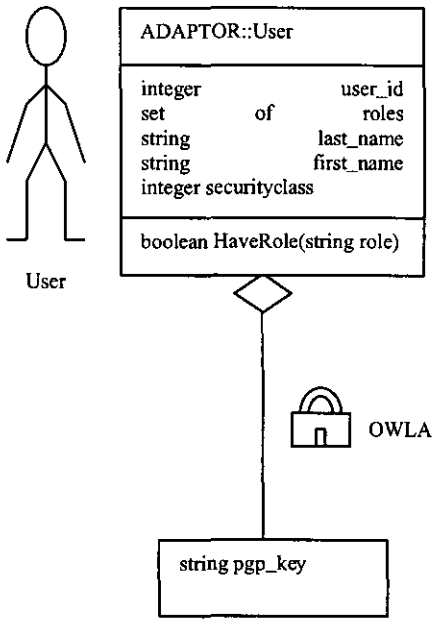


Figure 5. Completed adaptor.

We have now all the material we need to draw an Adapted Information Content Diagram (AIC) which is used for describing application's data structure and related functions. The previously defined Publication class and shadows are used. See figure 6.

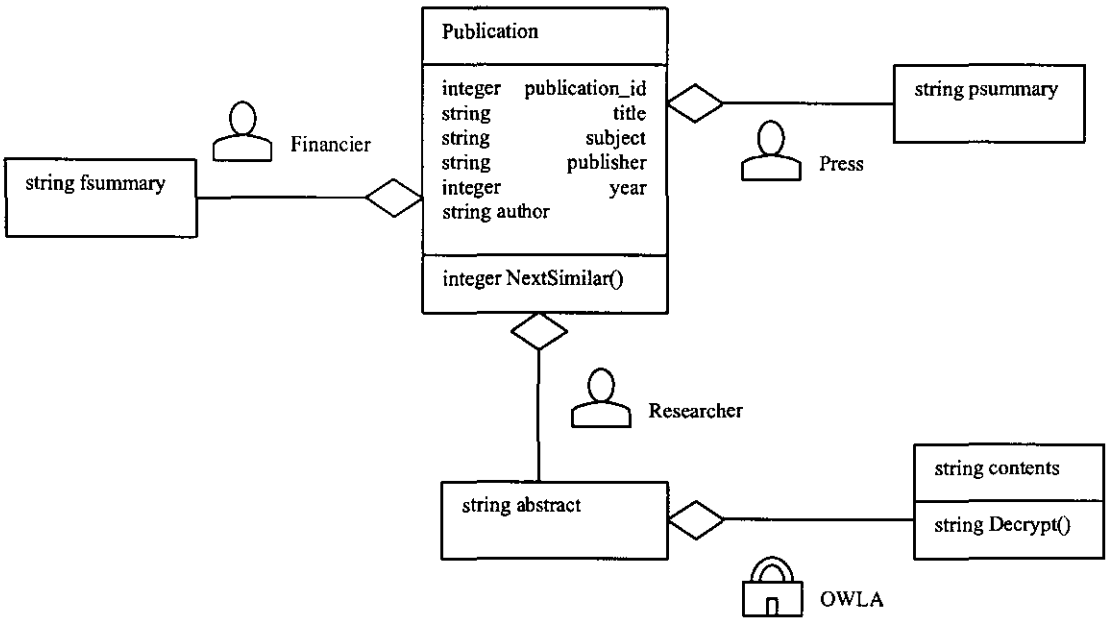


Figure 6. Adapted Information Content (AIC) -diagram.

Diagram's skeleton consists of one class, Publication, which has six attributes and one method. Completing the diagram there are four transformants connected to skeleton through four shadows. Each transformant has one attribute and one of them has also one method.

When you start reading adapted diagram please keep in mind that diagram's skeleton is formed by classes and transformants are their conditional extensions containing attributes and/or methods.

Reading the AIC-diagram is quite straightforward operation: you examine the from the adaptors' point of view, adaptor by adaptor if necessary. Skeleton is the part which is always there despite of adaptation. Transformants are adaptive parts which extend skeleton if adaptor has certain property values (what these values should be is determined by shadows).

Let us examine the AIC-diagram in point of User's view. First we assume that user has role of financier (User class attribute "roles" has value "financier").

Publication has six attributes (publication_id, title, subject, publisher, year, and author) and one method (NextSimilar) as you can see from the figure 6. In addition to this, because user has role financier, Publication class has also en extra attribute "fsummary". The reason why this attribute is included in Publication class and nothing else lies inside the shadows. Only the condition inside the Financier shadow is true, all other conditions inside the other shadows are false. See figure 4.

If user's role is changed to press, publication has the same six attributes and one method as in financier's case. The only difference is that instead of attribute "fsummary" there is an attribute "psummary". The

reason is shadow Press, which is the only shadow who condition is true when User class attribute "roles" has value "press".

Examining the rest of the diagram requires giving values to two attributes in User class. First attribute "roles" receive value "researcher", stating that user is a researcher, and then attribute "securityclass" is set to 1 indicating that this researcher is part of the OWLA team.

When we look at the figure 5 we can see that user is entitled to decryption key (attribute "pgp_key" belongs to User class) if OWLA shadow allows it, which it does if User class attribute "securityclass" has value 1.

Thus the Publication object appears to OWLA researcher as follows: Publication has the same six attributes (publication_id, title, subject, publisher, year, and author) and one method (NextSimilar) as in previous cases. Because User class attribute "roles" has a value "researcher", publication object has an extra attribute "abstract". Now when User class attribute "securityclass" was set to indicate that researcher is part of the OWLA team, publication has also attribute "contents" and method "Decrypt". In total OWLA researcher's Publication object has eight attributes and two methods.

This example is implemented on-line for the interested to try at http://owla.oulu.fi/demo/publication.py

Summary and open questions

We have developed an approach for designing adaptive hypermedia, and based on that a method for visualizing dependencies between the *adaptors* (i.e. objects respect to what the system adapts) and *transformants* (i.e. the parts that adapt) using so-called *shadows*, which hide the underlying algorithms (heuristics) and are meant to be an easy and intuitive tool for designers.

The method is intended to complement existing hypermedia design methodologies, such as OOHDM and AHDM, which currently lack means for showing the dependencies between e.g. content and properties in the user model in the design specifications for an adaptive hypermedia system. We have not yet, however, evaluated the proposed method in practice nor fully analyzed the questions related to fitting it in the existing design methodologies. For example, the use of shadows (illustrating adaptivity dependencies) in the various different design diagrams remains an open question (e.g. in OOHDM, could shadows be used in every diagram, or only in the navigational class diagram? How does adaptivity on lower levels affect the higher?).

There are also other types of adaptivity that this method does not currently address – in the words of our preliminary classification, it now deals with *adapting to difference* but ignores *adapting to change*. Addressing the questions related to this are left for future work.

We have recognized the need for tool support for the method – especially the possibility for the designer to examine the system with different views is an interesting function. Otherwise the concept of association class in UML (Fowler 1997, p. 93) can be used to model the shadows (as stereotypes) with standard tools, as we will be experimenting in real-world projects in near future.

References

Baumeister H., Koch N. and Mandel L. (1999). Towards a UML Extension for Hypermedia Design. In UML'99 The Unified Modeling Language - Beyond the Standard, LNCS 1723. Springer Verlag.

Boehm, B. (1988). A spiral model for software development and enhancement, *Computer* May, 61-72.

De Bra, P., Eklund, J., Kobsa, A., Brusilovsky, P., Hall, W. (1999). Adaptive Hypermedia: Purpose, Methods, and Techniques. The 10th ACM Conference on Hypertext and Hypermedia, Hypertext '99.

De Bra, P., Houben, G. J., Wu, H. (1999). AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. The 10th ACM Conference on Hypertext and Hypermedia, Hypertext '99.

Brusilovksy, P. (1996). Methods and Techniques of Adaptive Hypermedia. User Modeling and User-Adapted Interaction, Vol. 6, pp. 87-129, Kluwer academic publishers.

Fowler (1997). UML Distilled – Applying the standard object modeling language. Addison-Wesley Object Technology Series.

Koch, N. (1998). Towards a Methodology for Adaptive Hypermedia Systems Development. In *Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, Proceedings of the* 6th Workkshop, ABIS-*98,* U. Timm and M. Roessel (Eds.)

Gómez J., Cachero C. and Pastor O. (2000). Extending a Conceptual Modelling Approach to Web Application Design. B. Wangler, L. Bergman (Eds.): CAiSE 2000, LNCS 1789, pp. 79-93. Springer-Verlag Berlin Heidelberg.

Schwabe, D. & Rossi, G. (1998). Developing Hypermedia Applications using OOHDM. Workshop on Hypermedia Development Processes: Methods and Models, Hypertext'98. http://www.inf.puc-rio.br/~schwabe/papers/ExOOHDM.pdf.gz

Lieberherr, K. (1996) Adaptive Object-Oriented Software: The Demeter Method. PWS Publishing Company

Designing mobile-aware adaptive hypermedia

Toni Alatalo
Toni.Alatalo@oulu.fi

Janne Peräaho
Janne.Peraaho@oulu.fi

A position paper for the 2nd session of the adaptive hypermedia workshop 2001, to be held at the ACM Hypertext conference, August 2001.

Department of Information Processing Sciences, University of Oulu
OWLA project, http://owla.oulu.fi/

Abstract

Mobile use presents new challenges for hypermedia design. One of the key elements in mobile-aware hypermedia is adapting to users' situations, but somewhat differently than in the traditional user-modelling centric view of adaptive hypermedia. This paper presents an example of designing new navigational structures for an existing Web service, with the mobile use in mind. A previously presented modelling method for designing adaptive hypermedia is used in the specifications. The system's adaptation to location and time information - but not to any personal user specific properties - is modelled and implemented as a proof of concept demo, as a new layer on top of the existing service. The usefulness of the previously introduced modelling method and other approaches to designing adaptive hypermedia in this context are discussed, and lessons learned from the test and questions for future work pointed out in conclusion.

Introduction

In recent years, much attention has been paid to mobile computing within IT industry. Successful mobile IT has been developed including mobile phones, PDAs (Personal Digital Assistants), communicators, and wearable computers. Following the Internet revolution has even been said to be the wireless revolution (Hjelm 2000).

A fundamental attempt to conceptualize mobility in a larger sense has been carried out in Swedish Viktoria Institute, where a reference model has been developed to provide designers with a framework of concepts to understand how people use IT in mobile settings (Dahlbom & Ljungberg 1998). The concerns in the model are environment, modality and application of mobile IT use, each of them having further details. Here the focus is on the first entity - environment - and specifically the physical environment as the use context. Design of mobile applications often aims to *context-awareness*, the ability of application to extract, interpret and use situational information and adapt functionality to the current context of use. Examples of context information are (Korkea-aho 2000): identity, spatial information (location, orientation, speed, acceleration), temporal information (time of the day, date, season of the year), environmental information

(temperature, air quality, light or noise level), social situation (who you are with, people that are nearby), resources that are nearby (accessible device and hosts), availability of resources (battery, display, network, bandwidth), physiological measurements (blood pressure, hart rate, respiration rate, muscle activity, tone of voice), activity (talking, reading, walking, running) and schedules and agendas. A concise overall definition: *"context is everything but the explicit input and output"* (e.g. the states that effect the application's behaviour) is given in (Lieberman & Selker 2000).

The current design methods - hypermedia or others - do not currently support the design of adaptivity in general, nor especially contextuality. An exception is the adaptive hypermedia design method (AHDM) which includes the design of user models within the methodology (Koch 1998), but does not concern context-awareness - nor does it support the actual modelling of adaptivity in design specifications. An attempt to address these issues was presented in (Alatalo & Peräaho 2001), but with a user modelling based adaptation example only. Here we experiment with that modelling method further, addressing the issues of mobile use and contextuality from a hypermedia design perspective.

It has been suggested that the *design of context* - i.e. considering context as the object of a design activity - would be "a natural extension of current approaches that treat context as information for the design of the artifacts (design with context) or as the scenario for enacting the design activity (contextual design or design in context)" (Roque & Almeida, 2000). We share this interest, and therefore focus on the design phase that occurs after early analysis and requirements elication, trying to come up with ways to model specifications for actual implementations with the richness of information that can be found surrounding the use. The example this position paper presents is yet a simple one, but we consider it as a proof of concept to begin with.

In the following part, the previously presented design approach and modelling method is recalled. Then the actual design problem - supporting the mobile use of an existing food place information service - is presented as the starting point. The design of the solution is presented phase-by-phase with the according diagrams resulting in the adaptive navigation structure. The work done is discussed regarding the limitations of the study, related efforts and ideas for the future - then concluded in the end.

Approach

Adaptation can be understood as a process where information, contexts, navigational structures, and/or user interface changes according to (user) profiles and adaptation rules. In these kind of systems profiles and adaptation rules can be static (set once, never change), updatable (change by request), or dynamic (in constant change). Examples of this in adaptive hypermedia are e.g. AHM (da Silva et al 1998) and AHAM (De Bra, Houben & Wu 1999), IMPS SRM (Bordegoni, Bulterman et al 1999) and for an early example in the field of human-computer interaction see (Gargan, Sulliwan & Tyler 1988). We call this *the engine view of adaptation, see* figure 1.

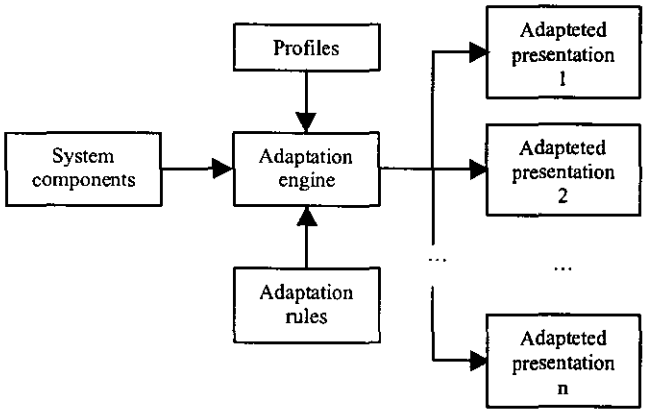


Figure 1. The engine view of adaptation.

The Structural Adaptation (Alatalo & Peräaho 2001) approach to adaptation is quite different. Structural adaptation suggests that adaptation can be hypermedia system's property, too, and not merely a process. In practice, the architectures may well be similar, but as the focus here is on conceptual modelling in design specifications, the different view on adaptation matters when defining what to model.

In the structural adaptation adaptation is integrated in the system's information structures, functions, contexts, navigational structures and user interface. The basic idea is that system can be divided into three parts which together forms an adaptive hypermedia system. The parts are as follows: Adaptors, Heuristics, and Transformants. See Figure 2.

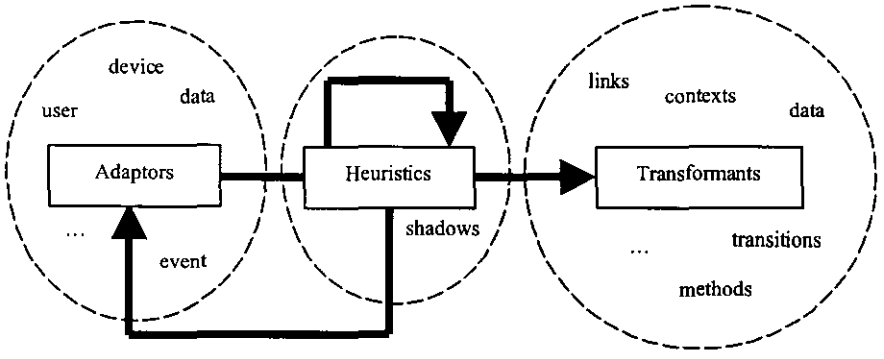


Figure 2. Structural adaptation.

These parts can be understood as specific roles some objects in the system have with respect to adaptivity.

Adaptors are defined as a system components to which the rest of the system adapts. In harmony with the object oriented design each adaptor consist of attributes, known as adaptor's *properties*, and methods. Adaptor's properties play an important role in the adaptation, since they determine relations between adaptors and system's adaptive components, as defined by heuristics.

Transformants are the name for system's adaptive components. They adapt to adaptors according to heuristics. Transformants are class, attribute and method constructs containing conditional and non-conditional assosiations. According to transformant structure it can consist of non-adaptive parts (uses non-conditional assosiations), known as the skeleton, and adaptive parts (uses conditional assosiations).

In (Alatalo & Peräaho 2001) seven different transformant categories were presented. We add "Nodes" category to complete the classification. The complete eight category list for hypermedia system's transformants is as follows:

- Data – transformants based on application domain data
- Nodes - transformants based on hypermedia nodes
- Links – transformants based on hypermedia links
- Contexts – transformants based on context
- Transitions – transformants based on links between contexts
- Methods – transformants based on methods
- Adaptors – transformants based on adaptors
- Heuristic rules – transformants based on heuristic rules

Adaptors and heuristic rules acting as transformants is quite rare, but still possible and justified.

Heuristics defines two-fold relations between adaptors and transformants: a) to which adaptor's properties transformant has relation to (disclosed with variables) and b) what is the nature of the relation (mathematical and logical dependency).

Heuristics, which are modeled with *shadows*, can be classified based on adaptors to which the heuristic rule is mainly involved. The classification cannot be strict, only suggestive, due to cross references between different adaptors. Nevertheless, giving names, even unaccurate ones, to the shadows does seem support the design work.

The shadow consists of one clause (heuristic rule), containing variables, constants, mathematical- and logical operands, which defines the relation of one or several adaptors and transformants. As a result of this each shadow has a binary output value, 0 (false) or 1 (true), which indicates adaptor's adaptive part's inclusion or disjunction. In other words, shadows acts as a logical glue which holds the transformant's skeleton and adaptive parts together. Without these links adaptive system can not exists.

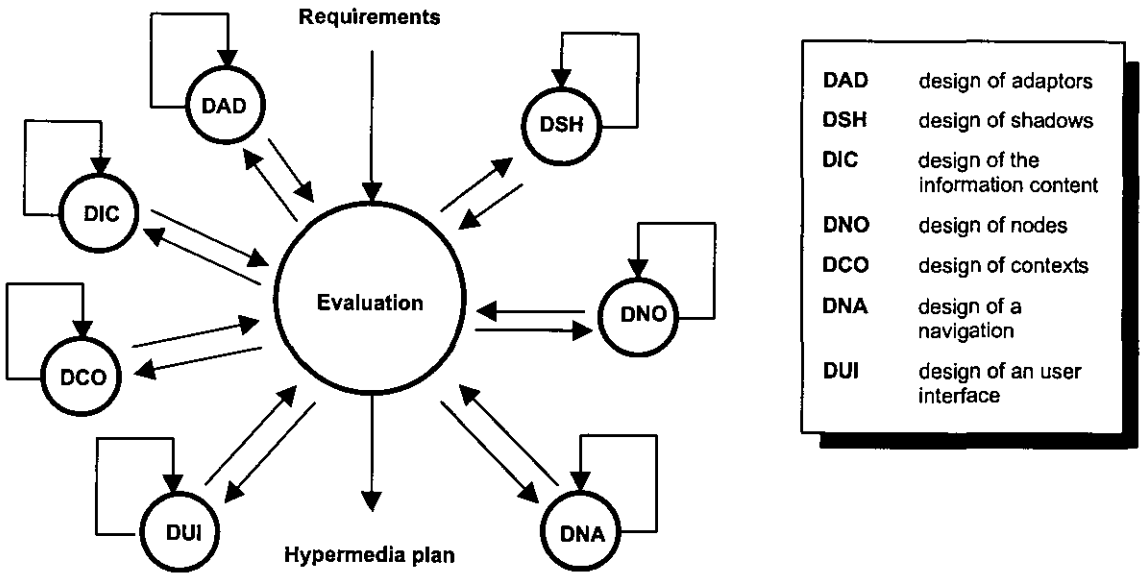


Figure 3. Design tasks of an adaptive hypermedia system.

The design of an adaptive hypermedia system was divided into six tasks and design issues and an incremental iterative design process along with special design tools was recommended for accomplishing the tasks.

The design of hypermedia nodes was part of the navigation design in the previous publication (Alatalo & Peräaho 2001). We now extract the node design into a separate design task: design of nodes. See figure 3. As will be shown later, although presented here as separate concerns, nodes and links can be specified in the same diagram, as in the navigational class diagrams in OOHDM (Schwabe&Rossi 1998) and in the navigational structure diagrams in the UML based method presented in (Hennicker & Koch 2000).

Design problem

Background

The particular design problem at hand is based on an existing service, which provides basic information, detailed descriptions and ratings of food places in the city of Oulu, Finland, on the web. The website is quite a peculiar one, and became an instant success soon after its launch early 2001. The service is run by individual private people, who set it up for fun, without thinking of it as a business nor consulting the actual food place owners. It should be noted, that the scope of the service is not just any food places, and not at all any high-class restaurants, but initially the certain pizzerias that had become somewhat a phenomenon in the city. The slang word for the food in these lower-class nightlife-oriented food places in Oulu is "känkky", hence the URL of the service: http://kaenkky.u--3.com/ (in Finnish only) . Later, the service has grown to cover other similarly used food services such as grills, and finally quite nice restaurants too (although the food there is not considered "känkky").

The content is entirely created by the users, who can submit reviews, and edited by (a team of) administrator(s). As the word spread (mostly on the Internet), the totally unorganized people quickly contributed insightful and greatly humorous stories soon covering the whole range of food places in the region - some enthusiasts had digital cameras with them, so the site has photos of the places and even of the actual meals. In addition, there's a rating function (points from 1 to 5) and a top-five and low-five list derived from the ratings.

Initially, the service was (implicitly) designed for desktop-based use only. Being busy with reseach related to mobile services, we tried using the service with mobile devices too - first with a PDA (Psion5mx running Opera on Epoc) and with a WAP-phone (after tweaking a wrapper for WML on the site). With the PDA, the service was barely usable, but for the WAP-phone with an even smaller display it was just too much. Also with the PDA it helped to know the service from before. To ease the download times and browsing, we created a lite-version of the html-based web service with pictures and other unnesessary information excluded, which was already better for the PDA. But still it was evident that to make the service really usable for the mobile user it would need to be thought over.

Requirements for mobile use

To specify the requirements for mobile use of the food place service, we identified two simple use cases (or user stories): 1. A mobile user is hungry, and does not know where to go to eat, so needs to find a place that can satisfy the hunger soon enough but with an adequate quality too. 2. A mobile user is standing by a foodplace, wondering whether it's a nice one - not necessarily being hungry right now, but perhaps trying to decide about going there some time later - and would like to have more information about the place. These cases are depicted in figure 4.

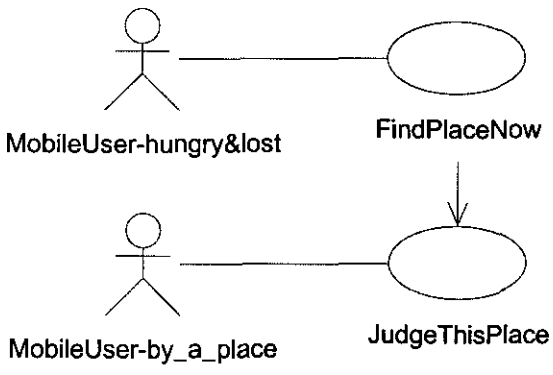


Figure 4. The two use cases, setting requirements for the mobile use

The actual information about the food places can be derived from the existing service via an interface (see figure 5).

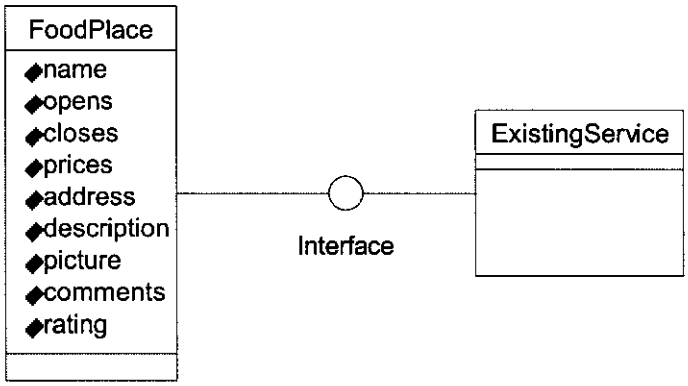


Figure 5. Initial conceptual model, with information derived from the existing service

Following the design process presented before, the next tasks are the design of the information content, design of nodes, design of context, design of navigation, and design of an user interface. We decided to do this using the OOHDM-like (Schwabe&Rossi 1998) method with UML-based notation as presented in (Hennicker & Koch 2000). Their methodology covers the same design objectivities and provides ready-to-use syntax for diagrams. At the same time it allows us to show how an existing hypermedia design model, which was not originally meant for designing adaptive hypermedia, can be complement to meet the challenges set by the adaptive hypermedia systems by extending it with the concepts adaptor and shadow.

Solution

We have gathered requirements for food place service, constructed use cases, and drawn a preliminary conceptual model on the basis of the requirements. So far the design process has evolved like in any other hypermedia system design, but now it is time to think system's adaptive qualities more closely.

Design of Adaptors

Design of food place service's adaptive qualities starts by identifying adaptors. From the use cases we can derive that the system needs to be adaptive with respect to time and place. Time refers to the relationship of current time and food place's opening and closing times, and place refers to the relationship of food place's location and user's location.

Opening and closing times and food place's location are clearly food place's properties and they are definitely needed for adaptation. In the domain model (figure x) we defined the class *FoodPlace*. Now this class can be used as a basis for an adaptor called *FoodPlace*. See figure 6.

Food Place

FoodPlace
name: String location: Location address: String contact: String opens: Time closes: Time prices: Money description: String picture: Image comments: Set(String) rating: Float
TravellingTime(): Time Open(): Boolean

Figure 6. Adaptors.

We supplement the class with *location* attribute in order to make location adaptation possible, and add two methods to fulfill the time and location adaptation requirements. Method *Open* determines if food place is open in the basis of food place's opening and closing times and current (system) time. Method *TravellingTime* uses food place's location and user's current location while evaluating user's travelling time and/or distance to the food place.

User's location is needed only for evaluating travelling time between user's current location and the food place, and the user's location is required only when user is requesting recommendation for food place or acquiring food place information which is close by. There is really no need to store any user information, not even the frequently changing location, and we will not therefore need a user adaptor. As there is no information tied to an individual user, there is no "user" in the class diagram, and no user model at all. Instead, in order to adapt, the system requires information about the *use location*. Technically this position information can be e.g. included within the http-request (coming from the client via the network to the server).

Design of Shadows

Designing the adaptation, the shadows, is about designing filters, which hide transformant qualities depending on values of properties of adaptors. However, in this case it is not possible to make a clear difference between adaptors and transformants, and for this reason defining the adaptation is a bit tricky and maybe confusing.

It was required that the system recommends food places based on their availability (time and location) and supports quality judgements with food places' ratings and reviews. The second requirement was that system should provide information about food place where user is now or is about to enter.

In the first scenario system should recommend food places in basis of food places' reviews and availability. Earlier we recoqnized the adaptor *FoodPlace* which holds all the information associated to food place, including review, opening and closing times, and location which are needed for making recommendation. Because the result of recommendation is a prioritized group of food places, or more precisely a list of some of the food places' properties, adaptor *FoodPlace* is not only an adaptor but also a transformant. As noted when introducing the design method, this is not contradictary (an object can well have several roles).

To satisfy the needs set by the first use case, one should determine if a food place is currently open and if it is reasonably near. In addition food places need to be ordered by their reviews.

Ordering the food places is not the matter of shadows but the two previous tasks are. (Observe that shadows can be used as search and ordering criteria but not as a method, which produces ordered list of items.) We create two shadows: *Open*, which determines if food place is currently open, and *Near*, which determines if food place is near the user requesting recommendation. See figure 7.

In the second use case, the system should display information about the food place whereby the requesting user is now. For this purpose we create a shadow *AtDoor* depicted in figure x. Note that this functionality requires quite accurate positioning (current GPS is adequate, and emerging networking technologies such as Bluetooth offer this kind of possibilities), and has to be omitted when there is not such (this detail is not implemented here, but could be easily programmed within the shadow).

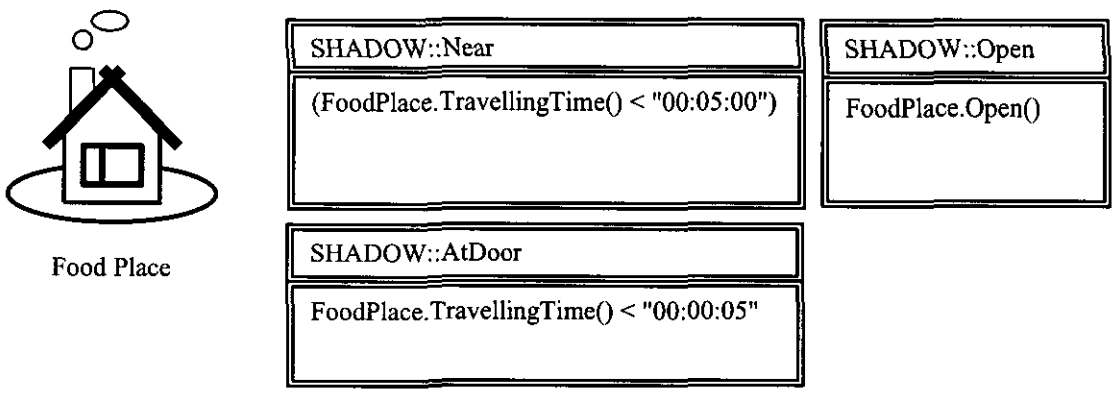


Figure 7. Shadows.

As the focus here is on conceptual modelling, the implementation of shadows is not tied to any specific techniques. One solution is to convert shadows to the existing class's methods. In this case shadows *Near*,

AtDoor and *Open* can be added to the *FoodPlace* class as methods, to be used for determining if food place is near or open, or if user is standing next to the food place.

We have now enough information to polish the conceptual model presented earlier.

Design of Information Content

The concept of location is more complicated than it may look at the first sight. When evaluating travelling times, user and food place location should be expressed with reasonable high accuracy to avoid coarse misjudgements. The accuracy by which the location is expressed varies depending technology in use. For full functionality, we require location information which error marginal is within one to three meters. The most axiomatic way to indicate location is to use longitude and latitude coordinates, which are used here. The users of advanced mobile devices, with accurate positioning and graphical interfaces, may then be e.g. provided with a map where the relevant locations are highlighted - or even a compass-like arrow guiding them to the recommended places. For inaccurate positioning, such as the the nearest GSM base station, the abstraction *Area* is introduced. An area can indicate e.g. a part of the city, when traffic signs and regional information combined with food place's address guide the user the food place's location unequivocally in terms of community planning. Even if there's no GUI (as with many WAP-clients and e.g. speech interfaces), the name of the area and the street address are usable.

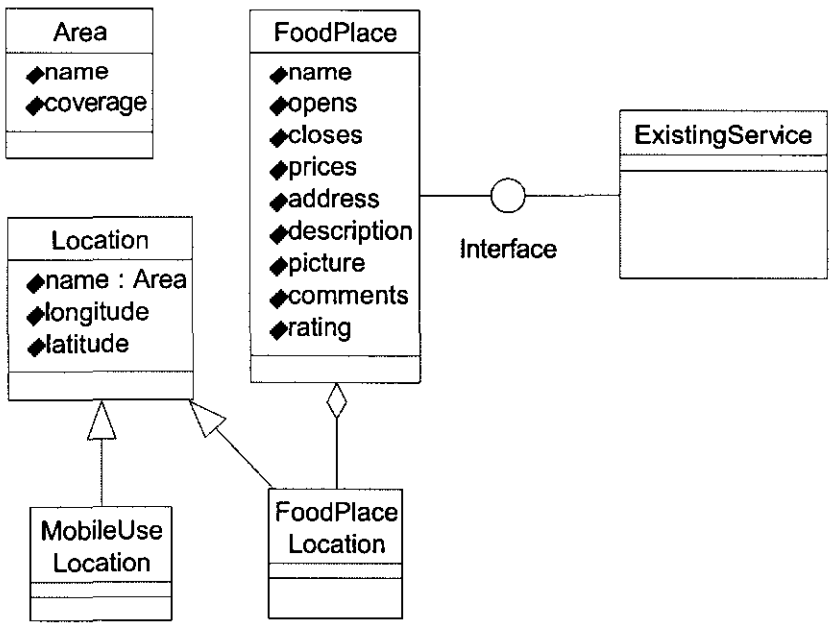


Figure 8. Final Conceptual Model.

The polished conceptual model is depicted in figure 8. Food place has a geographical location for evaluating user's travelling time to the food place and showing the exact position when possible, and regional location so that it can be found without a special equipment.

Design of nodes, contexts and navigation

In terms of our own adaptive hypermedia system design process model (see figure 3), we have designed adaptors, shadows and the system's information content. Design tasks still uncompleted are design of nodes, design of contexts, and design of navigation. (Note that there is an user interface design task, too, but we decided to exclude it from this paper.) These tasks will be completed next in the form *of Adaptive Navigational Structure Model and Adaptive Navigational Class Model* using the UML-based method presented in (Hennicker & Koch 2000). In OOHDM, the respective design phase is navigational design, and the corresponding diagrams are called the navigational class diagram and the context diagram (Schwabe&Rossi 1998). Besides nodes and links, also access structures (such as indices) are designed in this phase, and here they are our first concern.

Two kinds of queries are needed: the one which lists maximum of ten priorised food places and the one which fetches information attached to a single food place, to a food place which is next to the user (if one is really close, as defined in the shadow AtDoor).

The first query satisfies the needs of an user who is requesting food place recommendation, where as the second query services user who is by a certain place. The navigational structure model depicted in figure 9 specifies this design. In the diagram, we use the stereotypes for query (the questionmark), index (with the little box with lines) and navigational class (with the empty box) as defined in (Hennicker & Koch 2000).

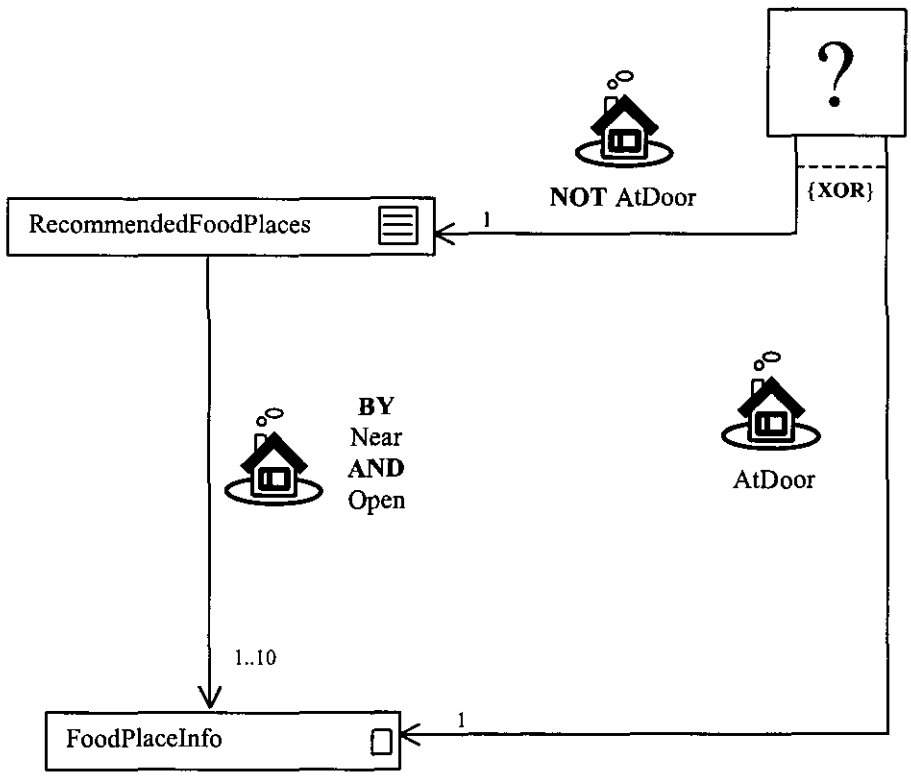


Figure 9. Adaptive Navigational Structure Model (or Adaptive Context Diagram).

Choice of a query depends on user's location. If shadow *AtDoor* is true, system displays information about food place which is next to the requesting user. If shadow *AtDoor* is false (there is no food places nearby), a list of food places is constructed. This list is ordered by by shadows *Near* and *Open* so that food places which are near the user and which are currently open appear on the top of the list.

Index *RecommendedFoodPlaces* and node *FoodPlaceInfo* need to be elaborated. We have done this in a way that can be seen in figure 10., again following (Hennicker & Koch 2000).

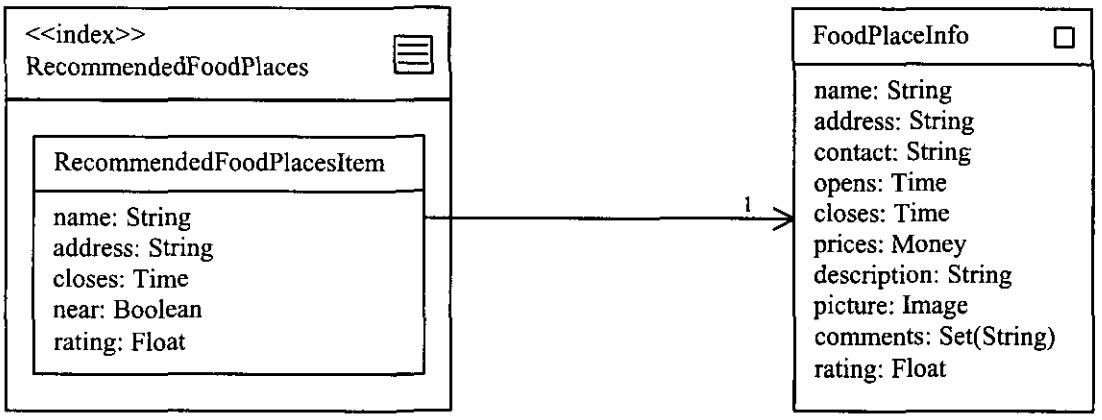


Figure 10. Adaptive Navigational Class Model.

Index *RecommendedFoodPlaces* consists of *RecommendedFoodPlacesItem* items. Each item contains food place's name, postal address, closing time, rating, and a piece of information indicating if food place is somewhere near. Link connecting item to a node (*FoodPlaceInfo*), containing detailed information about food place, is anchored to the food place's name (which is the default behaviour defined in (Hennicker & Koch 2000)).

The very same node (i.e. navigational class *FoodPlaceInfo,* with the content from the existing service) is pointed to by the query which displays information about food place whereby the user is. This completes the location- and time-adaptive design of the navigation for mobile use of the existing food place service.

User Interface considerations

The specification of the actual user interfaces for the system is outside the scope of this paper, but here are some remarks concerning especially mobile use and the assumptions made in the work. The mobile devices are expected to be of various kinds, and used in many different ways. The examples already mentioned in the description of the work done: PDAs with Web-browsers, phone-embedded WAP-clients and voice interfaces (e.g. over a phone line while driving a car), already have very different capabilities. The diversity may grow even greater with the advent wearable computing (possibly augmented reality goggles) or, on the other hand, the disappearing interfaces of ubiqutous computing. The hypermedia design models used here separate the content, navigational structures (e.g. nodes as views to databases) and the presentation for the

user, enabling user-interface metaphora independent design. E.g. the system at hand here uses existing content, adds what's needed to support mobile use, and allows for different user interfaces.

Discussion

Here we have presented a case where there is adaptivity in a hypermedia application without modelling the user, using the location and time information instead. Admittedly, however, user oriented adaptation would be interesting in this case, too. For example users could specify their individual tastes regarding the food, or preferred areas/locations and perhaps eating times - or the system could be set up to learn them (semi)automatically based on the behaviour of the users. Also social (collaborative) filtering could be used to e.g. acquire recommendation based on ratings by users with similar tastes etc. This point of not forgetting the user, but modelling them as well as the context, has been elaborated elsewhere (Jameson 2001). On the other hand, it can be argued that the location information as used in our example is user information - it is the current location of the user, after all. But the emphasis here is that the system keeps no record of the usage, and none of the adaptation is done with regard to any personal characteristics - just the use location. This makes a point when concerning privacy, which is often a hot topic when location information is used. In this example, the location information is never connected to the individual.

Thinking of the design approach, and specifically the modelling method, there are many questions. What is the actual scope of these models - do these diagrams capture the essence of the design problem at hand? In the end, after trying several different ways (even an activity diagram), we would argue that the solution here is an elegant one and our previously presented addition to support modelling adaptivity - the shadows - integrate well, complementing the UML-based notation for the OOHDM-like modelling presented in (Hennicker & Koch 2000). Compared with the original example in (Alatalo&Peräaho 2001) a notable point is that there we modelled adaptive content (a single node) whereas here there's adaptive navigation, too (thanks to the flexibility of UML class diagrams that can be used for both purposes when modelling hypermedia).

The results are limited as the case is still a simple one, if not trivial, but we are already working on a considerably more complex application. Another remaining issue is the relation of our approach to previous ones, such as the Adaptive Hypermedia Architecture Model (AHAM, De Bra, Houben & Wu 1999).

In the future, there are additional challenges still in achieving more complex ("deeper") adaptivity, perhaps along the lines explored in an attempt to create open-ended semiosis for adaptive knowledge management (Rocha 2000). A potential approach for designing complex adaptivity may also lie in multi-agent systems, and from a modelling perspective the proposed Agent-UML extension is an interesting one (Odell, Van Dyke and Bauer 2000). There even the concept of emergence has been noted, but no way to actually design it proposed. The work on open-ended semiosis mentioned above, perhaps combined with the recent

development in understanding the underlying monumental works of von Neumann on evolution of complexity (McMullin 2000), may offer solutions. In the meanwhile, the more basic kind adaptivity - such as the simple example here - combined with good design, is certainly enough to keep us busy.

Conclusion

In this paper, we have used a previously presented approach for designing adaptive hypermedia to address the requirements set by mobile use. It was shown, that the method could be used to model specifications for a time and location adaptive system too, although it was originally illustrated with a user adaptive (e.g. personalization) example only. This also makes a case concerning the definition of adaptive hypermedia, which currently includes user modelling only. Although the adaptivity in this case is also user-centered, in the sense that the hypermedia navigation changes according to the location of the user, it may be argued that it is not user adaptivity, as there in fact is no user model and none of the behaviour depends on any characteristics of individual users (that are not even recognized). This emphasis on *context awareness,* (being in this case specifically *mobile aware*), is a strong trend in information systems and human-computer interaction design, with hopefully fruitful connections with the hypermedia community in the future - not forgetting the user modelling aspects already elaborated in adaptive hypermedia, either.

References

Alatalo, T. & Peräaho, J. (2001). A Modelling Method for Designing Hypermedia. A position paper for the Third Workshop on Adaptive Hypertext and Hypermedia, the UM2001 session. <http://owla.oulu.fi/publications/modelling_method_for_AH-position1_AHWS2001.pdf>

De Bra, P., Houben, G. J., Wu, H. (1999). AHAM: A Dexter-based Reference Model for Adaptive Hypermedia. The 10th ACM Conference on Hypertext and Hypermedia, Hypertext '99.

Bulterman, D. C. A., Rutledge, L., Hardman, L.,van Ossenbruggen, J. (1999). Supporting Adaptive and Adaptable Hypermedia Presentation Sematics. The 8th IFIP 2.6 Working Conference on Database Semantics (DS-8): Semantic Issues in Multimedia Systems. January 5-8, 1999.

Dahlbom, B. & Ljungberg, F. (1998). Mobile Informatics. Scandinavian Journal of Information Systems. Vol. 10, no. 1&2.

Gargan Jr, R. A., Sullivan, J. W., Tyler, S. W. (1988). Multimodal Response Planning: An Adaptive Rule Based Approach. CHI '88 Conference Proceedings. May 15-19, 1988.

Hennicker, R. & Koch, N. (2000). A UML-based Methodology for Hypermedia Design. In A. Evans,

S. Stuart, and B. Selic, editors, UML'2000 - The Unified Modeling Language - Advancing the Standard, volume 1939 of Lecture Notes in Computer Science, York, England, October 2000. Springer Verlag.

Hjelm. J. (2000). Designing Wireless Information Services. New York: John Wiley & Sons.

Jameson, A. (2001). Modelling Both the Context and the User. Personal Technologies, 5, 1.

Koch, N. (1998). Towards a Methodology for Adaptive Hypermedia Systems Development. In *Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, Proceedings of the* 6th Workkshop, ABIS-98, U. Timm and M. Roessel (Eds.)

Korkea-aho, M. (2000). Context-Aware Applications Survey [online]. Mari Korkea-aho. <http://www.hut.fi/~mkorkeaa/doc/context-aware.html>.

Lieberman, H. & Selker, T. (2000) Out of Context: Computer Systems that Adapt To, and Learn From, Context. CHI 2000 Workshop Research Directions in Situated Computing <http://www.daimi.au.dk/~mbl/chi2000-sitcomp/pdf/Cheverst.pdf>

McMullin, B. (2000) John von Neumann and the Evolutionary Growth of Complexity: Looking Backwards, Looking Forwards... Presented at Artificial Life VII August 1-6, 2000, Portland, Oregon. <http://www.eeng.dcu.ie/~alife/bmcm-2000-01/>

Odell J., Van Dyke Parunak H. and Bauer B.(2000) "Extending UML for Agents". AOIS Worshop at AAAI 2000, Austin, TX, 2000.

Rocha, Luis M. (2000). "Adaptive recommendation and open-ended semiosis". International Journal of Human-Computer Studies. (In Press) <http://www.c3.lanl.gov/~rocha/ijhms_pask.html>

Roque, L. & Almeida, A. (2000) Towards the Design of Context. CHI 2000 Workshop Research Directions in Situated Computing. <http://www.daimi.au.dk/~mbl/chi2000-sitcomp/pdf/Roque.pdf>

Schwabe, D. & Rossi, G. (1998). Developing Hypermedia Applications using OOHDM. Workshop on Hypermedia Development Processes: Methods and Models, Hypertext'98. <http://www.inf.puc-rio.br/~schwabe/papers/ExOOHDM.pdf.gz>

da Silva, D.P., Van Durm, R., Duval, E., Olivie, H. (1998). Adaptive Navigational Facilities in Educational Hypermedia. The Ninth ACM Conference on Hypertext and Hypermedia. Hypertext 98. June 20-24, 1998.

History-based Link Annotation for Self-Exploratory Learning in Web-based Hypermedia

Muan H Ng, Wendy Hall, Patricia J Maier
University of Southampton, UK
E-mail: mhn99r, wh, pjm00r@ecs.soton.ac.uk

Ray Armstrong
Southampton General Hospital, UK
E-mail: rayarmstrong@btinternet.com

ABSTRACT
In this paper, we examine the notion of history-based link annotation for self-directed exploratory learning environment. We first explain the motivation for exploratory learning in web-based hypermedia and investigate the effectiveness of adaptation to reduce cognitive overload. We then present a simple user model that considers individual reading speed, comprehension rate and prior knowledge to determine effort spent on a page. An initial pilot study is conducted to support the work described in this paper.

KEYWORDS: Adaptive navigational support, history-based link annotation, reading speed.

INTRODUCTION

Browsing is a form of informal learning in hypermedia [1] and it is intrinsic to the nature of hypermedia – a tool to obtain information. As opposed to formal learning where a set of rigid paths and tasks are exposed, users are given the freedom to explore information stored in the hypermedia. The nature of browsing is driven more by the curiosity and presence of anchors rather than a definite learning goal. As users navigate from one node to another, pieces of information bits are accumulated to construct knowledge blocks. A *history-based link annotation* is proposed to aid cognitive overload in this style of learning. Links are annotated based on the users' browsing history to inform them which links they have already visited. Ideally, users can decide to pay more attention to pages that have been 'less attended to' from previous browsing.

PREVIOUS WORK

The system first needs to determine whether a user has indeed visited a page. Previous work in Interbook[2] recorded the display of a page to determine if the page has been read. But this method lacks accuracy as a student might only be surfing through the pages without paying much attention to the content. In MANIC[3], the system attempts to determine if the students has in fact read a page by considering time spent. An optimal reading time based on the length of a page is used to compare with the actual time spent. It assumes a high studied rating for a page if user spends an optimal amount of time on it. However, the optimal time that is generated based on the content displayed is static for every user irrespective of their individual reading speed. As far as we know, none of the existing adaptive hypermedia application considers individual differences in determining whether a page is read.

USER MODEL: CAPTURING EFFORT SPENT ON CONTENTS

Literature in psychology has proved that there are "astonishing differences in the rate of reading speed" among individuals [4]. This applies even if they are reading at normal speed. It has also been found that two individuals reading at the same speed can have different comprehension rates [5]. Similarly, if a subject reads comparatively slowly, they do not necessarily comprehend more than a fast reader does. This work has proved that both the **speed** and **power of assimilation** can affect reading efficiency and there is no direct relationship between them.

On top of these, individual reading style can also affect the reading speed. A person who knows the domain well may tend to read at a rapid rate to review familiar materials. By contrast, a person who is a novice may read at a normal rate to grasp relations to general ideas. Hence we consider another factor of **'prior knowledge'** to make a more accurate prediction of the user's reading speed. Users have to perform a self-rating on the domain knowledge for the system to obtain a value for prior knowledge.

We have built our user model based on the three individual factors formulating the notion of effective reading speed. This notion is greatly influenced by standard ways of measuring reading speed in commercial software solutions to enhance reading [6]. In our case, a factor of prior knowledge is added to refine the existing standard.

Hence, we defined the effective reading speed as the actual reading speed of a user weighted by his/her comprehension rate and prior knowledge.

Effective reading speed = reading speed x *(comprehension rate + prior knowledge) / 2*

For example, assuming a reader who can read at the speed of 200 words per minute. If he/she can comprehend 60% of the materials read (derived from the percentage of correct answers in a post-reading test), and his/her self-assessment of prior knowledge is 70%, the *effective* reading speed becomes 200 x (0.6+0.7) / 2, which is equivalent to 130 effective words per minute (ewpm).

HISTORY-BASED LINK ANNOTATION

In the process of browsing and navigating, the individual effective reading speed is used as a benchmark to build a history of browsing for each individual. An *optimal* time spent for each page in the domain is generated dynamically based on the length of the page and the individual effective reading speed:

Optimal time for a page, T
= number of words / effective reading speed

When a user spends an amount of time, t, on a page, we can determine the effort spent on that page by comparing t with optimal time, T. A page is considered well studied if the actual time spent approaches the optimal time ($t \approx T$). If the user spends too much or too little time on the page we assume that he/she is skipping the page or has left the application unattended. In that case, a zero effort is assumed (see Figure 1).

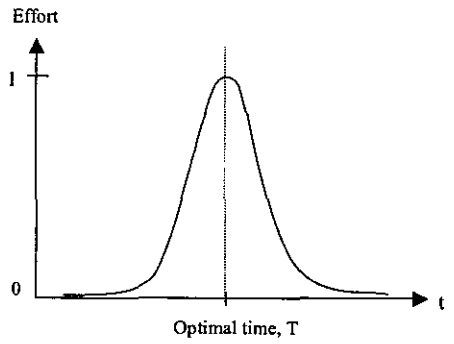


Figure 1: A normal distribution of effort based on time spent

The history of browsing is then used to annotate links as in Figure 2. A zero percentage means the user has never or has hardly looked at the information associated with Link8 and a 100% means the user has spent enough "effort" for the system to assume that the information associated with Link3 is well read.

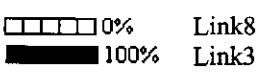


Figure 2: Links annotated by the percentage of effort

PILOT STUDY

A pilot study has been conducted to compare an application with and without the history-based (HB) link annotation. The subjects are tested on a web-based application for the study of Rheumatology. Initial results show that 63% out of 19 subjects found the HB link annotation useful. The results also show that the average time spent on the case with the HB link annotation is reduced for subjects who are novice to the domain. However the effect of the HB link annotation is not significant for subjects who are expert in the domain. In observing the number of links clicked, comparatively fewer links are followed in the case with the HB link annotation, provided if the list of the links are long (>10 links).

CONCLUSIONS

Our conclusion so far is that the HB link annotation might help to reduce cognitive overload in self-exploratory learning. We have demonstrated the use of individual reading speeds to model users' browsing history. However, even though the system only approximates the 'effort spent' on a page, to a certain extent, the 'students' understanding' of the information read is estimated since we take into consideration their comprehension rate and prior knowledge. Further work and evaluation will be done to find any significant learning effects, given the integration of the history-based link annotation into the hypermedia environment.

ACKNOWLEDGMENTS

We would like to thank the Arthritis Research Campaign for funding the project.

REFERENCES

1. Duchastel P. Formal and Informal learning with hypermedia, in Designing Hypermedia for Learning, Springer-Verlag, New York, 1989.

2. Brusilovsky P. and Schwarz E. User as Student: Towards an Adaptive Interface for Advanced Web-Based Application. In Proc of UM97. Springer-Verlag, New York (1997), pp 177-188.

3. Stern M. and Woolf B.P. Adaptive Content in an Online Lecture System, in Proc of the AH 2000, Trento, Italy (2000), pp 227-238

4. Romanes G. J. Mental Evolution in Animals, Appleton & Co, New York, 1885.

5. Huey E. B. The psychology and pedagogy of reading, The Macmillan Company, 1908.ACM Press, pp. 381-386.

6. ReadingSoft.com found at http://www.readingsoft.com

A co-operative distributed environment for adaptive Web-based education

Rosa M. Carro, Estrella Pulido, Pilar Rodríguez

Universidad Autónoma de Madrid, Ctra. Colmenar, km. 15, Spain

E-mail: {rosa.carro, estrella.pulido, pilar.rodriguez}@ii.uam.es

ABSTRACT

This paper presents a distributed environment that facilitates the cooperation among educational resources and systems that are located in different machines. The main purpose of the work presented is to use this kind of environment for adaptive Web-based education.

KEYWORDS: Distributed educational environments, adaptive systems, educational multimedia

INTRODUCTION

Since the Internet was born numerous educational Web-sites have been developed. When creating such systems, most of the developers build a set of HTML pages with explanations that the students should learn, enriched with other specific materials that can facilitate their comprehension, such as images, videos, animations or simulations.

All these multimedia materials and specific programs are usually available for a particular educational course. However, these resources could be very useful for other courses related to the same subject and located at different computers. On the other hand, the course designers may not know how to share their own specific powerful educational resources so as to be used by different educators and included in several environments.

One of the possibilities that facilitate the co-operation between teaching systems and resources is the development of distributed educational environments that allow different systems to access instructional resources located in different machines. In these environments, educators can place their resources in machines accessible from remote locations. At the same time, they can make use of remote resources that are relevant for a specific course. This approach avoids the generation of different specialized systems whose goals are similar, with the corresponding effort saving. Moreover, it widens the fan of possibilities that arise when many educational materials and resources are available. There exist several approaches in this direction based on the definition of learning objects such as the Learning Object Metadata (LOM) IEEE standard [1].

DISTRIBUTED EDUCATIONAL ENVIRONMENTS

The ideas that each author has in mind when thinking of an hypermedia educational environment can range from direct Web-mapped courses to more sophisticated curricula content generators. At present, most of the on-line educational courses are based on the electronic book paradigm. Designers can use the standard http Web protocol, and easily develop and maintain isolated content specialized machines [2]. The whole can be seen as a co-operative educational environment. In addition, it is quite simple to link any kind of the above-mentioned specialized sites at any point of the course. However, when user-modeling techniques are required for adaptation purposes, the previous approach needs to be extended.

The use of specialized educational resources across the Internet resembles the idea of database federation, which applies a weak coupling among the huge amount of rather different databases that should be used at the same time. In educational systems, adaptivity implies that some feedback related to the user features is needed for adapting the system behavior to the user, and the educational resource federation should allow such an information exchange [3].

A solution that would allow each database management system to keep its own features, while being accessible by a wide range of client applications, implies the development of a common application program interface (API) and a set of drivers capable of translating the messages in both directions: from clients to servers and the other way around.

As in the database case, there are several approaches to the common API that vary with respect to the site in which the drivers are placed. One of them is shown in figure 1. It balances the development effort in both sides.

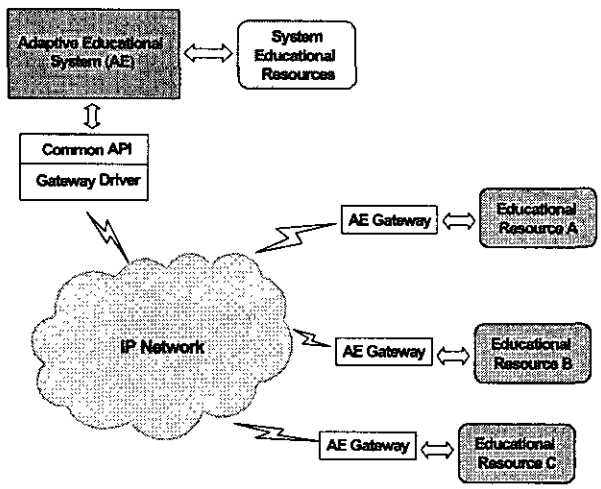


Figure 1: Scheme of a distributed co-operative network

In the adaptive educational system side, it is necessary to develop a common API as well as a gateway that understands messages received from the educational

resource side. On the other hand, in the educational resource side, the required module is a gateway that checks the correctness of the parameters and packages them into the appropriate format. The gateway mentioned in the figure should not be confused with the standard common gateway interfaces (CGI), being related to each specific educational system protocol.

A CASE STUDY

In the area of educational adaptive Web-based systems we have developed TANGOW (Task-based Adaptive learNer Guidance On the Web), a system that provides a personalized guidance to the students during their learning process [4].

In TANGOW, the course structure is separated from the course contents and both of them are stored in databases. The course structure is defined by means of a set of teaching tasks (TTs) and rules that are used to select, at every moment, the most suitable topics to be taught to the students depending on their personal features and on their actions while taking the course. TTs can be atomic or composed and they represent tasks that a learner has to perform in order to acquire certain knowledge. Atomic TTs may correspond to i) a concept or procedure that must be learned, ii) an example about a concept or procedure, or iii) practical exercises to be solved. A rule indicates how a composed task is decomposed into subtasks for a specific user. The educational resources are included as content fragments associated to TTs. Different versions of each content fragment exist depending on the student profile and they can be stored on different computers. The HTML pages presented to the students are dynamically generated by selecting the most appropriate version for each student.

At a first stage, the TTs and rules descriptions, along with the content fragments that composed the courses, were stored at a unique machine. Students connected to the TANGOW system to access the courses developed.

On the other hand, at the Computer Engineering School of the Universidad Autónoma de Madrid, some resources related to a course on compilers were developed to provide educational aids through the Web to third year students. The subject complexity, that requires the understanding of several techniques for morphological, syntactical and semantic analysis, suggested the development of some kind of support that could help students to understand and practice with these concepts and procedures.

With this purpose, some specialized programs that permitted the creation of simulations or exercises related to the above-mentioned subject were created. These resources consist basically of two kinds of specialized exercises: i) exercises whose statement is automatically generated by a program and ii) exercises proposed by the student that are solved by the system. These Web-based resources were developed independently from TANGOW and accessed by students through Internet. They are examples of the educational resources shown in figure 1.

Some time later, a complete Web-based course on "Language Processors" was developed by using the TANGOW-Design tool which included theoretical explanations, illustrative examples and exercises to be solved by students. It was decided to include the previously created specialized exercises so that the students could also develop some practical skills. These programs could have been modified in order to be inserted into the course. Instead, an interface was developed that allows the communication between them and the TANGOW system so that new programs could be included at any moment. Moreover, the programs were installed in a different machine to that in which the TANGOW system was located and they were kept there because: i) students could either run the exercises directly or access them through the TANGOW system and ii) the evaluation of the described exercises was a complex task so it was convenient to have an independent machine in charge of this task.

These specialized exercises are included as components of a TANGOW-based course structure as practical teaching tasks and the fragments containing the exercise statements are specified in the task definitions. The specialized program sends the problem statement along with the feedback about the student actions to the CGI that composes the HTML page with the comments about the problem resolution. This information is also included in the page presented to the student so that it can be received by the TANGOW system after the student reads and checks the corrections made by the specialized program.

This is the way specialized programs were introduced in the TANGOW system and co-operation among different educational systems was firstly implemented. Other TANGOW-based courses have been developed following the same approach, also including access to simulations and code checking, and distribution is proving to be a feasible way of facilitating the co-operation among different educational sources and environments for adaptive purposes.

ACKNOWLEDGMENTS

This paper has been sponsored by the Spanish Interdepartmental Commission of Science and Technology (CICYT), project number TEL1999-0181.

REFERENCES

1. IEEE Learning Technology Standards Committee LOM Working Draft 6.1 available at http://ltsc.ieee.org/wg12/index.html

2. Gilbert J.E. and Han C.Y. Adapting instruction in search of 'a significant difference'. Journal of Network and Computer Applications, 22, 1999.

3. N. Henze and W. Nejdl. Extendible Adaptive Hypermedia Courseware: Integrating Different Courses and Web Material. Adaptive Hypermedia and Adaptive Web-Based Systems. Lecture Notes in Computer Science LNCS 1892 pp 109-120.

4. Carro, R.M., Pulido, E. and Rodríguez, P. Dynamic generation of adaptive Internet-based courses. Journal of Network and Computer Applications, 22 (Nov. 1999), 249-257.

Software Engineering for Adaptive Hypermedia Applications?

Nora Koch[1,2], Martin Wirsing [2]

[1] F.A.S.T.
Applied Software Technology GmbH
Arabellastr. 17, D-81925 München, Germany
koch@fast.de

[2] Institute of Computer Science
Ludwig-Maximilians University of Munich
Oettingenstr. 67, D-80538 München, Germany
{kochn,wirsing}@informatik.uni-muenchen.de

www.pst.informatik.uni-muenchen.de/{~kochn,~wirsing}

Development of Adaptive Hypermedia Applications

The Programming and Software Engineering Research Group of the Institute of Computer Science of the Ludwig-Maximilians University of Munich is focusing on software engineering for hypermedia and Web applications in general and, particularly, for adaptive applications. One main goal of the software engineering discipline is to find techniques that support the development process of software applications. Our goal is to find, between others, appropriate analysis and design techniques that support development and authoring of adaptive hypermedia and Web applications.

General object-oriented software engineering approaches, such as the Unified Process (Jacobson, Booch & Rumbaugh, 1999) or specific methodologies for hypermedia like RMM (Isakowitz, Stohr & Balasubramanian, 1995), OOHDM (Schwabe & Rossi, 1998), and HFPM (Olsina, 1998) are not sufficient. They do not cover aspects relevant to personalization, i.e. user modeling and adaptation issues. A significant contribution in this field is AHAM (De Bra, Houben & Wu, 1999). AHAM is an application model for adaptive hypermedia that describe such applications from the authors' point of view.

We propose the UML-based Web Engineering approach (UWE) (Koch, 2000 & Koch et. al, 2001). UWE includes a design method for adaptive hypermedia applications and a development process for such applications. UWE is a systematic and object-oriented – in this way they differ from AHAM – design and development approach. We propose an integrated methodology for object-oriented development of adaptive hypermedia (Web) applications by presenting an extension to the Unified Modeling Language (UML). As basis for the software engineering approach we have developed the Munich Reference Model, i.e. a Dexter-based reference model which is formally specified using UML and OCL (Koch, 2000).

Third Workshop on Adaptive Hypertext and Hypermedia at the 8th International Conference on User Modeling, Sonthofen, Germany, July 13-17, 2001

Our Approach

By way of an analogy to hypermedia engineering (Lowe & Hall, 1999), *engineering for adaptive hypermedia applications* can be defined as a systematic, disciplined and measurable approach that supports the entire life cycle of adaptive hypermedia systems. This life cycle goes from conception through the elaboration, construction, delivery and maintenance to the cessation of the application.

The goal of an engineering approach is to support developers during these different phases in organizing mental activities, working at various levels of detail and abstraction, generating visual representations adapted to the designers level of experience, presenting the solution's constraints, building representations of the application and finally outlining plan structures and strategies.

The main motivation of our work was to define an engineering approach to adaptive hypermedia systems based on the object-oriented techniques that are state of the art in the software development (as schematically shown in Figure 1). The techniques used are UP (Unified Process, Jacobson, 1999), UML (www.omg.org/uml/) – a standard for object-oriented software design since 1997 that was chosen for all models and notations – and OCL (Object Constraint Language, Kruchten, 1998).

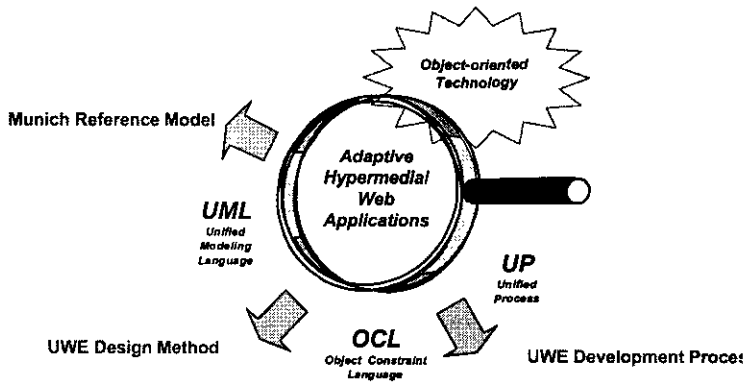
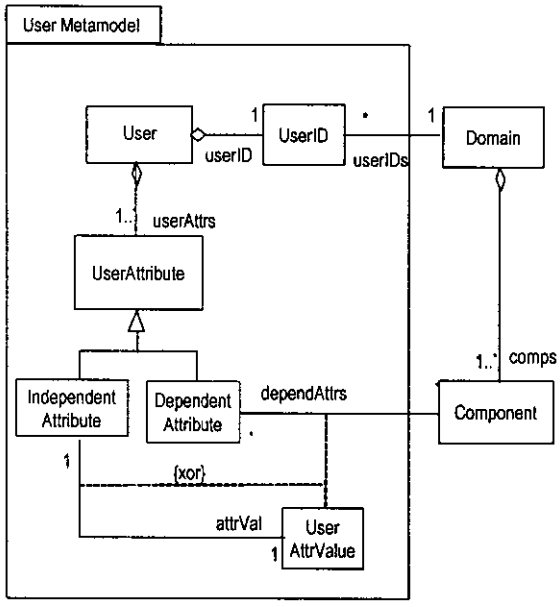


Fig. 1: UWE Software Engineering Approach

Reference Model

The aim of the Munich Reference Model is the formal specification of the features that constitute an adaptive hypermedia system and distinguish a high quality application. It is defined as an extension of the Dexter Hypertext Reference Model including a user metamodel and an adaptation metamodel. UML provides the notation and the object-oriented modeling techniques for the visual representation of the model. The Object Constraint Language (OCL), which is also part of UML, is used to supplement the semi-formal visual representation with semantic information. The formal description specifies invariants on the model elements and attributes as well as pre-conditions and post-conditions on the functions of the model. This formal specification is equivalent to a specification in languages, such as Z. A simplified user metamodel of the Munich Reference Model is shown in Figure 2.



User At ibu sID1{ xox}xu l x1id pxrt1*smansoxp1c .x }1ix mdpx n

Design Method

Visual modeling techniques are introduced in this work to define methodical analysis and design of adaptive hypermedia applications. The methodology of the UWE approach consists of modeling elements, notation and a method. The method is proposed to support the systematic construction of adaptive hypermedia applications and to identify as many steps as possible to be performed in an automatic way.

The notation and semantics of these elements define a "lightweight" UML extension, a so-called UML profile. It is defined as a set of stereotypes and properties for adaptive hypermedia that are used in some of the models proposed by UWE. These models are: a conceptual model to define the content, a user model, a

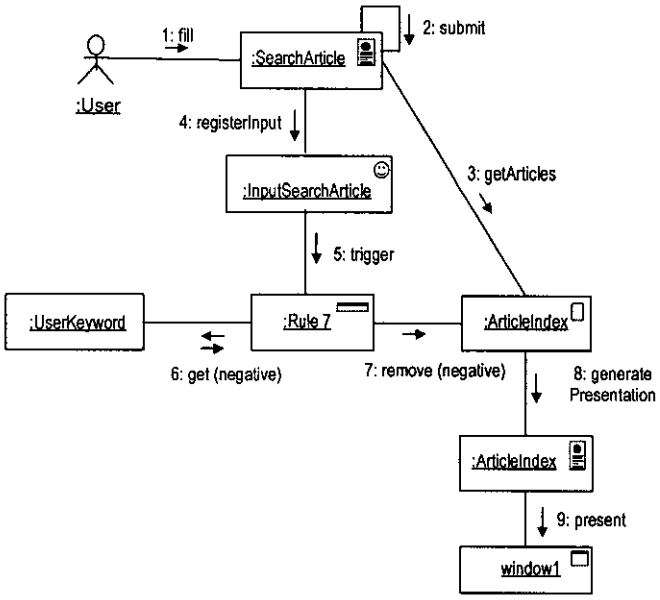


Fig. 3: UWE Design Method: Scenario of an Adaptation Model

navigation model, a presentational model to define static and dynamic aspects of the presentation and an adaptation model. Figure 3 shows – as an example -a scenario of an adaptation model of an Online Library application.

UWE is characterized by the separate treatment of hypermedia issues, such as content, navigation and presentation and from user modeling and adaptation issues.

Development Process

The development process of our approach, is based on the Unified Process, i.e. UWE adapts the Unified Process to support the development of hypermedia (Web) applications in general and to include special activities needed in the development of adaptive systems. Figure 4 shows an overview of the UWE development process and the supporting workflows.

UWE describes an object-oriented, workflow-based, user-centric, systematic, iterative and incremental process. Each workflow is textually described and graphically represented by activity diagrams. Both include workers, activities and artifacts. UWE specializes the Unified Process for the development of adaptive hypermedia applications describing which "experts" (workers) are required, which activities they perform and which specific artifacts they produce. UWE extends the coverage of the Unified Process development cycle adding a maintenance phase and a quality management workflow. It changes the idea of quality control management incorporating workflows for requirements validation and design verification in addition to testing.

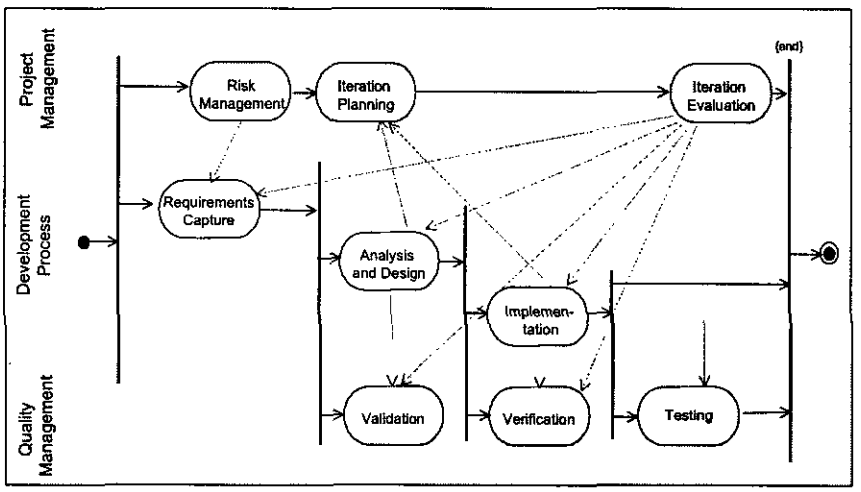


Fig. 4: Overview of the UWE Development Process

The design techniques are embedded in the UWE development process, which aims to cover the entire lifecycle form inception to maintenance of adaptive hypermedia applications.

Case Studies

The engineering approach was validated using several case studies, e.g. SmexWeb, an adaptive exercising system for students of a computer science introduction course

(pst1.pst.informatik.uni-muenchen.de:8000/indexE.html). SmexWeb is a framework developed at the Institute of Computer Science of the LMU that permits the development of teaching applications through instantiation. It supports adaptive content and adaptive navigation in all its variants (Brusilovsky, 1996). In addition it allows the system to take control over the process of navigation under special conditions, such as a period of user inactivity. This adaptive navigation technique is called passive navigation (Albrecht, Koch & Tiller, 1999). Figure 5 shows a screen shot of the EBNF application that has been developed using the SmexWeb framework. It is an EBNF (Enhanced Backus-Nauer Form) exercising session for an introductory course in computer science.

Currently we use the UWE approach in the LAMP project for designing and implementing an adaptive tutoring system for several Bavarian Universities (www.pst.informatik.uni-muenchen.de/projekte/lamp/index_e.html).

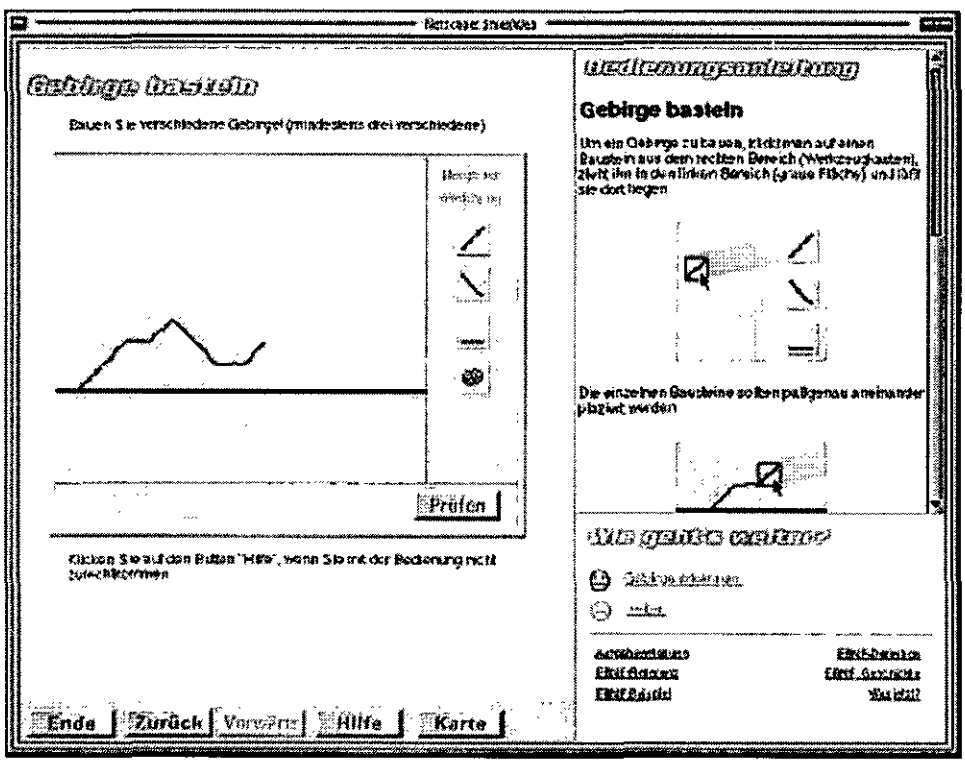


Fig. 5: SmexWeb: Task of the EBNF Application

Open Issues

The proposed engineering approach focuses on modeling and processes, but there are many open issues from the software engineering point of view, which still need to

be addressed and integrated, such as implementation techniques for adaptive hypermedia or personalized Web applications.

The development process requires case tool support. Tools for UML are developing fast, but they need to include special features for Web development since Web applications are becoming in a near future the most frequently developed software applications.

Our objective is to implement the stereotypes defined for (adaptive) hypermedia and Web applications (Koch,2000) as plug-in features of different case tools, such as for the open-source tool ArgoUML (argouml.tigris.org).

References

Albrecht F., Koch N. & Tiller T. (1999). Making Web-based training more effective. *Proceedings of the Seventh Workshop ABIS-99*: Adaptivität und Benutzermodellierung in interaktiven Softwaresystemen, T. Joerding (Ed.).

Brusilovsky P. (1996). Adaptive Hypermedia: An attempt to analyze and generalize. Proceedings of First Conference on Multimedia, Hypermedia and Virtual Reality 1994. Brusilovsky P. & Streitz N. (Eds.) LNCS 1077, Springer Verlag, 288-304.

De Bra P., Houben G.-J. & Wu H. (1999). AHAM: A Dexter-based Reference Model for adaptive Hypermedia. *Proceedings of the ACM Conference on Hypertext and Hypermedia*, 147-156.

Hennicker R. & Koch N. (2000). A UML-based methodology for hypermedia design. *Proceedings of the Unified Modeling Language Conference, UML'2000*, Evans A. and Kent S.(Eds.), 410-424.

Isakowitz T., Stohr E. & Balasubramanian P. (1995). A methodology for the design of structured hypermedia applications. *Communications of the ACM*, 8(38), 34-44.

Jacobson I., Booch G., & Rumbaugh J. (1999). *The Unified Software Development Process.* Addison Wesley.

Koch N. (2000). Software Engineering for Adaptive Hypermedia Systems: Reference Model, Modeling Techniques and Development Process. PhD. Thesis, LMU Munich.

Koch N., Kraus A. & Hennicker R. (2001). The Authoring Process of the UML-based Web Engineering Approach (Case study). Proceedings of the 1st International Workshop on Web-oriented Software Technology, 2001, to appear.

Lowe D. & Hall W. (1999). *Hypermedia & the Web: An engineering approach.* John Wiley & Sons.

Olsina L. (1998). Building a Web-based information system applying the Hypermedia Flexible Process Modeling Strategy. *1st International Workshop on Hypermedia Development, Hypertext'98.*)

Schwabe D. & Rossi G. (1998). Developing hypermedia applications using OOHDM. In Proceedings of *Workshop on Hypermedia development Process, Methods and Models, Hypertext'98.*

Wirsing M. & Knapp A. (2001). A Formal Approach to Object-Oriented Software Engineering. Theoretical Computer Science, 35 pages, to appear.

Adaptation as summarization: the role of rhetorical and narrative structures

Mettina Veenstra and Martin Alberink
Telematica Instituut
P.O. Box 589, 7500 AN Enschede, The Netherlands
veenstra@telin.nl, alberink@telin.nl
tel. +31 (0)53-4850485

Abstract: Research in the field of adaptive hypermedia aims at increasing the effectivity and comprehension of hypermedia. Important criteria for the effectivity of information are coherence and informativity. A hypermedia presentation is informative for a specific user if it does not contain information already known to that user and if it contains the required information. Precisely this combination of coherence and informativity may cause comprehension problems. Namely, by omitting the parts that are not informative for a specific user, for instance based on keywords, essential components for the coherence of the argumentation or narrative may disappear. This hampers the comprehension of the final tailored hypermedia presentation. In this paper we show how rhetorical and narrative structures between (groups of) nodes and objects within nodes contribute to adaptivity of hypermedia.

Keywords: Adaptive hypermedia, rhetorical and narrative structures

1 Introduction

The work presented here aims at developing an authoring framework that helps authors building informative adaptive hypermedia documents from which tailored presentations can be generated. For instance, an educational adaptive hypermedia document, consisting of several interconnected nodes (or pages), may be adapted to the knowledge or preferences of a specific pupil. Adaptation may imply leaving out certain nodes or parts of nodes (i.e. objects, e.g. a paragraph, an image, a video fragment) from the original document. The selection of nodes and objects may be based on the knowledge of the pupil, e.g. topics already known to the pupil are omitted. The selection may also be based on the available time. For instance, if the pupil can only spend one hour studying the information, a summary of the original document, containing only the most relevant information, is created.

At this moment a first version of the framework is available (cf. [Veenstra, to appear]). This framework offers guidance with respect to several aspects of creating adaptive hypermedia, for instance:

1. Combining hypermedia objects into nodes (e.g. How does the use of stretchtext in a node influence the surrounding objects in that node?);
2. Indicating how the objects must be related in presentations (i.e. which objects must be used together, which ones can possibly be left out, given the purpose of the presentation);
3. User interface aspects (e.g. Add links to prerequisite knowledge to each node);

4. Applying different types of adaptivity, adaptation methods and techniques (e.g. navigation adaptation, link hiding [Brusilovsky 1996];
5. Using standard Web technology to develop and to make available adaptable content (e.g. XML, XSL-T, Web browsers)

The framework is based on ideas from the field of adaptive hypermedia and ideas from the fields of rhetorical and narrative analysis. Furthermore it was inspired by experiences with building a prototype of an adaptive hypermedia service, using XML, XSL-T, JavaScript and HTML. This service allows users to automatically generate a tailored version of a document by answering some questions about their preferences and goals. After they answer the questions they get access to a tailored version of the content (See Figure 1).

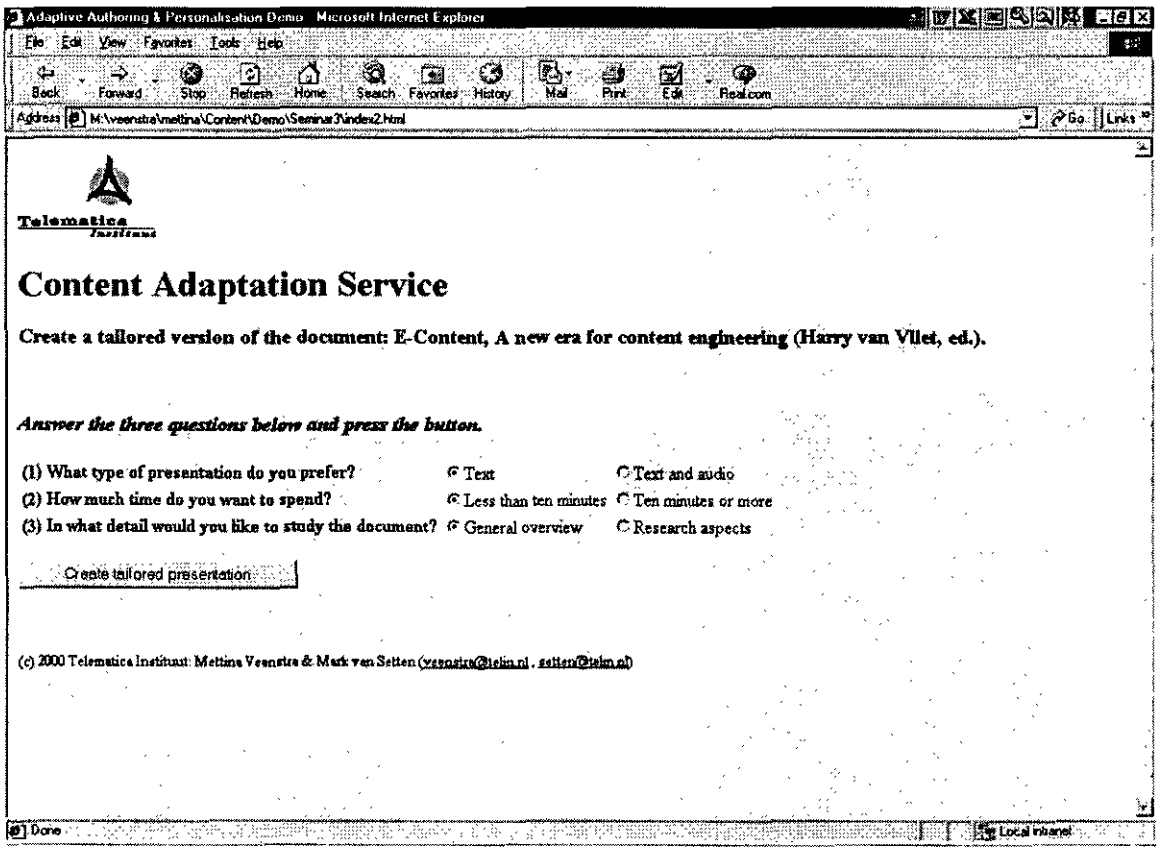


Figure 1: Page that makes it possible to create a tailored version of a document

In this paper the emphasis is on aspects of the framework described in point 2: Indicating how objects must be related in presentations. This point is relevant for adaptive hypermedia since one of the ways to increase the effectivity and comprehension of a hypermedia document is by making it adaptive in the sense that irrelevant objects or nodes for a specific user are omitted or made less accessible (e.g. by link hiding). However, omitting objects or nodes solely on the basis of e.g. keywords or information categories may harm the cohesion of the final presentation. Namely, keywords or information categories describing objects do not carry information about the relevance of

the object in the argumentation within a presentation. Nevertheless, a smooth argumentation and narrative is essential for the cohesion of a presentation. In this paper we identify ideas from narrative and rhetorical analysis that could be deployed in order to generate fluent tailored presentations.

2 Terminology

We briefly illustrate the terminology used in this paper with a short description of the adaptation process (see also Figure 2; the italic terms below refer to terms used in this figure). Adaptable hypermedia *documents* (i.e. sets of *multimedia objects* with *metadata*) are adapted to the user by matching a description of the user (*knowledge about the user*) with the *design knowledge* the system has and *metadata* of the available multimedia objects. The adaptation of the document performed by a *hypermedia adaptation service* results in a *presentation* that is adapted to the *user*.

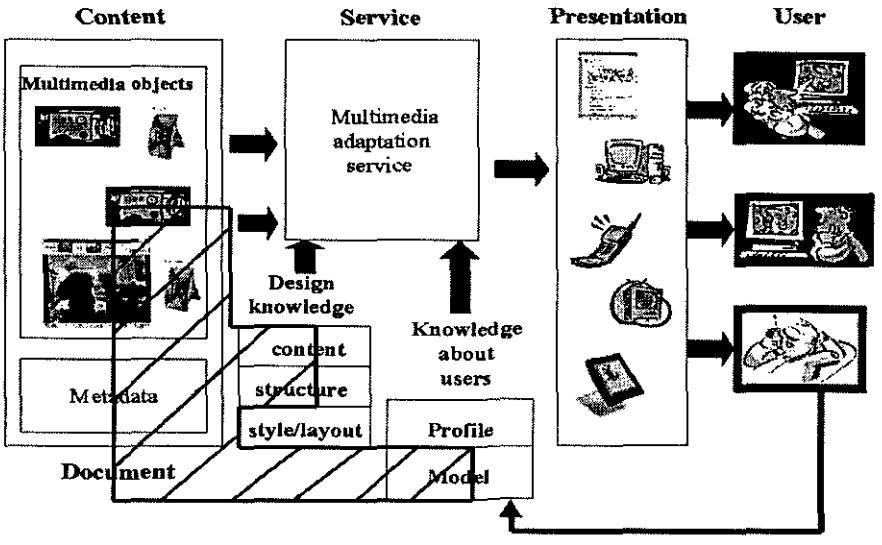


Figure 2: A simple representation of the hypermedia adaptation process.[1]

3 Adaptation as summarization

The focus of the framework is on accommodation of hypermedia information objects in a document, in which also the relationships between objects are made explicit, such as dependency and interchangeability between objects. The types of relationships give information about what can be left out or not. Our main research question is: "How to make adaptive hypermedia documents which allow for the generation of effective and comprehensible tailored presentations?" We think that comprehensibility is a natural point of departure since hypermedia may cause problems with respect to comprehension (e.g. being "lost in hyperspace", problems with lacking prerequisite knowledge).

Our approach to adaptive content can be described as follows. One relatively broad and elaborate hypermedia document is created, which is potentially useful for a broad variety

[1] The distinction between document and presentation is based on [Hardman 1998].

of users in a broad variety of situations. We assume that the tailored presentations that are created resemble summaries in the sense that they form a condensed version of the original, broad document. Essential properties for summaries are: 1. Conciseness (They contain only necessary information), 2. Coverage (They cover all key information), 3. Context (Sequencing should be done in a way so that critical terms are defined before being used) and 4. Coherence (The information flow should be natural and fluid).[2] It is obvious that conciseness (see point 1) is an essential property for both summaries and tailored presentations. A tailored presentation should contain only necessary information for a specific user, otherwise the presentation will become too broad. Unnecessary or less relevant information can either be removed completely, or be made less accessible. Point 2 through 4 with respect to tailored presentations are discussed in Section 4.

4 Structural and organizational aspects

The central question in this section is: What kind of structural and organizational aspects are essential for achieving optimal tailored hypermedia presentations for education purposes? Optimal means here: effective and comprehensible.

Background knowledge seems to play a prominent role in the comprehension of texts. Davidson [Davidson 1984] criticizes readability formulas strictly based on lexical aspects and syntactic structures. According to Davidson these formulas do not adequately measure text difficulty since they do not measure the prerequisite knowledge a text requires in order to be understood. This indicates that besides knowledge of the user, knowledge of the organization of the information itself is essential. In order to adapt content to a particular user resulting in a comprehensible hypermedia presentation, knowledge of the organization and possibilities of the basic (broad) document is as essential as knowledge about the needs and preferences of the user. For instance, when the prerequisite knowledge for understanding a particular object or node is known, then it is possible to automatically include links to nodes providing this knowledge. Another example that indicates the importance of structural or organizational knowledge of documents for adaptability is the following. When it is known which fragments of a video contain information or events that are essential for the understanding of that video, these fragments can be used for generating summaries. The knowledge can also be deployed in order to avoid that essential parts of the video are left out in tailored presentations. This type of organization knowledge can be captured in the metadata of the information objects and information nodes.

There are several disciplines that are studying the organization and structure of texts and other forms of expression such as film or speech. Relevant disciplines for the framework described in this document are narrative, discourse and rhetorical analysis. Narrative analyses studies a text or other type of expression (e.g. film) in what is depicted in it, for instance the events or characters. Discourse analysis studies a text or other type of expressions in how it is expressed. It studies the selection of language in which it is embodied, for instance pointers in the text to items in the text itself or to the context (e.g. <u>The boy</u> is tired. <u>He</u> goes to bed.). It also studies the rhetorical strategy that is applied. Thus rhetorical analysis is a sub-discipline of discourse analysis. Rhetorical analysis

[2] Cf. [He et al. 1999].

studies the roles of parts of a text in the argumentation. In order to create coherent texts or other types of expressions narrative and/or rhetorical structure may play an essential role. The framework contains some suggestions for implementation of such structures in metadata standards (under development) such as MPEG-7 and LOM (Learning Objects and Metadata). Below the most essential aspects of narrative and rhetorical structure for adaptive hypermedia identified in the framework are briefly described.

Rhetorical structure: nucleus-satellite and multinuclear relations
Rhetorical structures are essential for the coherence of presentations [Carter 1999] (point 4 in Section 3). An important level of rhetorical structure is the level dealing with nuclearity and relations. At this level the relations between parts of texts, either written or spoken, are described. A theory that formulates about 25 such relations is the Rhetorical Structure Theory (RST). [Mann 1999]. In a relation there always is a nucleus and there may be one ore more satellites. An example of such a relation is the background relation. In a background relation there is a text whose understanding is being facilitated. This text is the nucleus. Furthermore there is a text for facilitating the understanding. This text is the satellite. Another type of relation is the multinuclear relation. In a multinuclear relation there is more than one single nucleus. An example of a multinuclear relation is the list-relation. In a list-relation there is one span of text forming an item and another span of text forming the next item. The author must specify whether entities within relations can be omitted without damaging the coherence of the presentation. For instance, in some background relations the background information may be essential (i.e. prerequisite, see point 3 in Section 3), while in others it may be omitted.

Rhetorical structure: rhetorical devices
In rhetorical analysis several rhetorical devices are distinguished such as analogies and examples. Since people often have a preference for certain rhetorical devices, it is useful to label objects so that occurrences of desired rhetorical devices can be retrieved and used.

Narrative structure: key information
A story in a book or film always has key events which are essential for the course, and therefore for the understanding of the story [Chatman 1978]. These are typically the events that appear as a part of a summary of the story. Also informative (educational) hypermedia documents have such key points. These key points are often not key events but rather key facts or key information (point 2 in Section 3). Besides a function in a summary, key information may be given a central function in the hypermedia document itself (which is considered to be a kind of summary in our framework) in the sense that it is linked to a lot or in the sense that it gets a central position, e.g. close to the root of the document.

5 Discussion

This paper points out some ideas from narrative and rhetorical analysis that are useful for adaptive hypermedia. It does not offer any ready-made solutions. We only argue why

these ideas are relevant if increasing the effectivity and comprehension of hypermedia presentations is seen as an important goal of adaptive hypermedia.

The ideas from rhetorical and narrative analysis presented here are not only relevant for the reduction of broad and elaborate adaptive documents to concise tailored presentations. They are also relevant for the creation of "guided tours" through hypermedia documents, since "guided tours" are more coherent if they follow rhetorical and narrative structures. Furthermore, argumentation structures and narrative structures may influence the layout and presentation of hypermedia presentations (see [Rutledge et al. 2000]). For instance, some rhetorical relations are best represented at one page (e.g. the list relation), while the elements in another relationship may best be divided over two pages (e.g. the background relation: nucleus on one page, containing a link to the satellite on a second page).

References

[Brusilovsky 1996] Brusilovsky, P., Methods and techniques of adaptive hypermedia. In *User Modelling and User-Adapted Interaction 6*: 87-129, 1996.

[Carter 1998] Carter, L., Arguments in Hypertext: A Rhetorical Approach. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*. San Antonio, Texas, 2000.

[Chatman 1978] Chatman, S., *Story and Discourse: Narrative structure in Fiction and Film.* Cornell University Press, Ithaca, London, 1978.

[Davidson 1984] Davidson, A., Readability Formulas and Comprehension, In: L. Roehler, G. Duffy and J. Mason, eds. *Comprehension Instruction; Perspectives and Suggestions.* Longman, New York, London, 1984.

[Hardman 1998] Hardman, L., *Modelling and Authoring Hypermedia Documents*. Ph. D. Thesis, University of Amsterdam, 1998.

[He et al. 1999] He, L., E. Sanocki, A. Gupta and J. Grudin, Auto-Summarization of Audio-Video Presentations. In: *Proceedings of the ACM conference on Multimedia 1999*, 1999.

[Mann 1999] Mann, B., *An Introduction to Rhetorical Structure Theory (RST)*. 1999. http://www.sil.org/linguistics/rst/rintro99.htm

[Rutledge et al. 2000] Rutledge, L., B. Bailey, J. van Ossenbruggen, L. Hardman and J. Geurts, Generating Presentation Constraints from Rhetorical Structure. In *Proceedings of the ACM Conference on Hypertext and Hypermedia*. San Antonio, Texas, USA, 2000.

[Veenstra, to appear] Veenstra, M., *Framework for Authoring Adaptive Hypermedia: A Structure-Centered Approach*. Technical Report, Telematica Instituut, July 2001, to appear.

Adaptive Hypermedia in Augmented Reality

Patrick Sinclair and Kirk Martinez
Intelligence, Agents and Multimedia Group
Department of Electronics and Computer Science
University of Southampton

ABSTRACT
Applying Adaptive Hypermedia techniques to Augmented Reality museum tour guide applications promises great advances in presenting material on museum exhibits. Information selection and presentation can be adapted according to the visitor's goals, preferences, knowledge, and interests and this information can be overlaid over the real object and its features.

Existing mobile AR tour guides tend to present information using hand held devices or display hard coded, predefined labels over the object's features. This paper describes initial work on a system that can provide dynamic, adaptable information overlaid on objects.

KEYWORDS: Adaptive Hypermedia, Augmented Reality

Introduction

The use of Augmented Reality (AR) in museums promises great advances in natural interaction with museum artifacts and their data. AR systems combine real world scenes and virtual scenes, augmenting the real world with additional information. This can be achieved by using tracked see-through head mounted displays (HMD) and earphones. Rather than looking at a desktop or hand-held screen, visual information is overlaid on objects in the real world.

As museum visitors wear their own private HMDs, information being presented about a museum artefact can be adapted personally to each individual. This information could be about specific details of an object or the system could point out features of interest that they might have not noticed. Related objects could be projected next to the artifact for comparison. Different views could be presented, such as an x-ray view or a reconstruction of how the artefact originally looked.

Existing AR museum tour guides do not provide enough information on specific artefact details. Merging the real object with augmented information clearly presents the relationship between the data and the object. We propose a technique that dynamically adds adaptive labels to artefacts' 3D models that suits the user's needs.

Annotating Museum Artefacts

AR environments augment the real world with information or virtual imagery; virtual objects are usually stored as highly texture-mapped 3D models to appear realistic. An approach to presenting information about an artefact's features is to annotate it with dynamic labels, resulting in a 3D version of a labelled diagram.

Labels must be generated dynamically, depending on the visitor's preferences or the system's state. The user information is obtained from the user model, which could contain features such as their goals, preferences, knowledge, interests, previous interactions with the system and so on.

Taking an open hypermedia approach to the problem, all object information is referenced from or kept in a linkbase; it is separate from the artefacts' 3D models. This raises the issue of how to dynamically attach this information to the relevant features.

System Overview

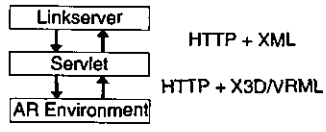


Figure 1: System Architecture

There are three main components in the system: the AR environment, a Java Servlet and a linkserver.

The AR environment loads an artefact model by requesting the model's ID through a Java servlet on a webserver. This servlet performs various tasks, such as querying the linkbase for information and placing description labels around the artefact object. The servlet will start by retrieving the artefact metadata and the model's location, which are both stored in the linkbase.

The servlet then loads the artefact model, which is stored as an X3D file; X3D is a next generation, extensible 3D graphics specification based on XML and is being developed by

the Web3D Consortium [1] and the World Wide Web Consortium [2]. The model is split into its various components, and each component is given a name or unique identifier. When the servlet loads the model, it looks for components with valid identifiers and uses these to query the linkserver.

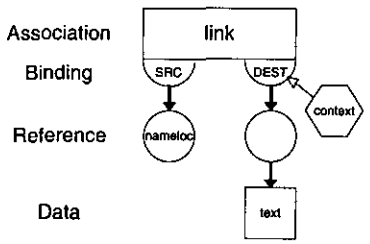


Figure 2: FOHM structure for a feature description

The linkserver looks for any relevant description and returns these to the servlet. The system uses the Auld Leaky linkserver, a context based link server implemented around the Fundamental Open Hypermedia Model (FOHM) [3]. In the FOHM linkbase implemented, an artefact's feature description is represented by an *Association* object. Associations have a source *Binding* object, containing a *Nameloc* object, and several destination bindings. Namelocs identify selections within any object or file by name, making Namelocs generic as a named feature can occur in various different objects. Each destination binding contains or links to a description for the feature and is bound to a *Context* object. The context object defines the description type and is used by Auld Leaky to distinguish between which bindings should be returned. This is shown in Figure 2.

When the linkserver receives a query, it returns an appropriate description. Similar objects will receive the same descriptions if they both have identically named features; this is useful as generic descriptions can be applied to objects without having to implicitly author descriptions for each object. New artefacts can be added to the system and be labelled without any material having been specifically written about them.

However, there will be cases when specific descriptions are needed. These can be resolved by adding FOHM context objects to bindings containing such descriptions. When the linkserver receives a query, the query will be bound to a context that is used to filter out any irrelevant descriptions.

An artefact's context is defined by it's metadata, which is stored as a *Data* object in the linkbase. This metadata can include various pieces of information, depending on the artefact; examples could include the name, type of object, reference number and so on.

The user model can be used to create context objects to tailor descriptions to the user's preferences. For example, each description binding can have a context stating the type of user: children or adults. When the query is made, only descriptions suitable for either ones (or both) will be returned. An example of this is shown in Figure 3. The label on the left is adapted for children, while the label on the right is adapted to adults.

When the servlet receives a description, it places it in a label next to the relevant feature. Labels are placed in a way so that they do not collide with each other or the artefact model.

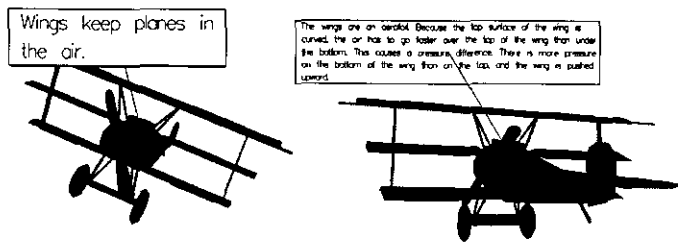


Figure 3: Adaptive labelling

Conclusions

The technique presented provides dynamic, adaptive labelling of artefacts that can be used in existing AR systems. New models can be added to the system and be annotated with existing descriptions. Different sets of descriptions can be applied by adding additional linkbases. As artefact information is kept separate from the artefact models, it is easier to author and maintain; descriptions can be changed in a link editor without loading a 3D model. Non-authored or generic links provide more information that is normally manually authored or maintained. The use of context is important in order to customise the information for the individual as well as prevent information overload. The simple example illustrated in Figure 3 shows how it can be used to create descriptions for different types of users. Future work will look at more complex ways of generating adaptive descriptions. Research done at Southampton on context and user modelling [4] could also be applied to the Auld Leaky linkserver to produce more dynamic and adaptive material.

More work will be done on the navigational hypertext aspect of the system so that link following causes a new model to be loaded. Labels need to be extended to use other types of media, such as images, video, audio and other 3D models.

REFERENCES

1. Web3d consortium, http://www.web3d.org.
2. World wide web consortium, http://www.w3c.org.
3. David E. Millard, Luc Moreau, Hugh C. Davis, and Siegfried Reich. FOHM: a fundamental open hypertext model for investigating interoperability between hypertext domains. In *Proceedings of the Eleventh ACM Conference on Hypertext and Hypermedia HT'00*, pages 93–102, 2000.
4. Samhaa El-Beltagy, Wendy Hall, David De Roure, and Leslie Carr. Linking in context. In *Proceedings of Hypertext 2001 (to be published)*, 2001.

The Future of AH

Licia Calvi

Department of Computing Science
Eindhoven University of Technology
The Netherlands
l.calvi@tue.nl

So far, adaptive hypermedia (AH) have mainly been used as a didactic tool, i.e., as a tool to develop on-line educational systems, despite the potentially wider spectrum of systems they could be applied to (see in [1]). A restriction adaptive educational systems soon showed was their application domain: most systems were used to teach computer science related or anyway scientific disciplines. Attempts at extending this limitation have been made, for instance by using adaptive hypermedia to teach foreign languages [2]. But the results obtained were not always so promising for the intrinsic difficulty at modelling such a domain: at a linguistic level (syntax and semantics), at a domain-related level (in the perspective of a situated learning approach to language learning), and at a level that was focusing on the intersection between the previous two.

What has been neglected so far is however the possibility of using AH for art. Art is a big word, indeed. And although, recently, a research trend has emerged which has pointed out the potential advantages of exploiting adaptive methodologies to the delivery of cultural information, which have a direct influence on the way in which cultural heritage information is approached, accessed and fruited, not as much as been devoted to promote the production of art by means of AH.

At a closer look, the contamination of art and technology is historically older than what the last years of Web explosion may seem to suggest: at the end of the nineteenth century, for instance, Rimbaud's fascination for photography was influencing his poetical style in determining which words to select, how to construct sentences, and how to juxtapose them in a visually effective way (see, for instance, in [3]). This is what I mean when I refer to using AH to produce artistic artifacts: AH should be considered as the canvas whose characteristics can influence the final artistic result in a peculiar way.

A first step in this direction has however already been made: it is the system developed by Kendall and Réty [4], the Connection System, which they presented last year at Hypertext'00. This system allows to write literature, both poetry and fiction, adaptively. I would like to extend the existing AHA architecture similarly. This is what I consider to be the future of AH

1. Brusilovsky, P. "Methods and Techniques of Adaptive Hypermedia", *User Modeling and User-Adapted Interaction*, 6, 87—129, 1996.

2. Calvi, L. "On the Formative Evaluation of CALLware", in P. Brusilovsky, O. Stock and C. Strapparava (eds.), *Adaptive Hypermedia and Adaptive Web-Based Systems, Proceedings of the AH'00 Conference, Lecture Notes in Computer Science*, 276—279, Spriger-Verlag, 2000.

3. Costa, M. *L'estetica dei media. Avanguardie e tecnologia.* Castelvecchi, 1999

4. Kendall, R. and Réty, JH. "Toward an Organic Hypertext", *Proceedings of Hypertext '00,* 161—170, 2000.

A Reference Architecture for Adaptive Hypermedia Systems

Hongjing Wu
Department of Computing Science
Eindhoven University of Technology
PO Box 513, 5600 MB Eindhoven
the Netherlands
phone: +31 40 2472733
fax: +31 40 2463992
email: hongjing@win.tue.nl

Web-based hypermedia systems are becoming increasingly popular as tools for user-driven access to information. They typically offer users a lot of freedom to navigate through a large hyperspace. Unfortunately, this rich link structure of the hypermedia applications causes some serious usability problems:

- A typical hypermedia system presents the same links on a page to all users. To eliminate *navigation problems* the system should offer each user (some) personalized links or navigation tools (such as a table of contents or a map). The system should thereby take into account what the user read before, and possibly what the user's interests are.
- Navigation in ways the author did not anticipate also causes *comprehension problems* for the user: for every page the author makes an assumption about what foreknowledge the user has when accessing that page. However, this is an impossible authoring task because there are more ways to reach a page than any (human) author can foresee. A page is always presented in the same way. This may result in users visiting pages containing redundant information and pages that they cannot fully understand because they lack some expected foreknowledge.

Adaptive hypermedia systems (or AHS for short) aim at overcoming these problems by providing *adaptive navigation support* and *adaptive content*. The adaptation (or personalization) is based on a *user model* that represents relevant aspects of the user such as preferences, knowledge and interests. The system gathers information about the user by observing the use of the application, and in particular by observing the *browsing* behavior of the user.

Many adaptive hypermedia systems exist to date. The majority of them are used in educational applications, but some are used, for example, for on-line information systems or information retrieval systems. An overview of systems, methods and techniques for adaptive hypermedia can be found in [B96]. Adaptive websites are also becoming popular. They typically have a name that starts with "My" (My Yahoo, My Excite, etc.) Some systems are only *adaptable*, meaning that the user enters a user profile through a registration form, and the system doesn't change that profile unless the user explicitly updates the profile through a form. An *adaptive* system performs updates to the user profile automatically by observing the user's browsing behavior. A primitive form of adaptation is found in systems that log which pages a user accesses, in order to be able to mark pages as "new" or "old" and in order to be able to generate "what's new" pages.

We have developed a reference model for the architecture of adaptive hypermedia applications: AHAM (for **A**daptive **H**ypermedia **A**pplication **M**odel) [DHW99], which is an extension of the Dexter hypermedia reference model [HS90, HS94]. AHAM acknowledges that doing "useful" and "usable" adaptation in a given application depends on three factors:

- The application must be based on a *domain model*, describing how the information content of the application or "hyper-document" is structured (using concepts).
- The system must construct and maintain a fine-grained *user model* that represents a user's preferences, knowledge, goals, navigation history and other relevant aspects.
- The system must be able to adapt the presentation (of both content and link structure) to the reading and navigation style the user prefers and to the user's knowledge level. In order to do so the author must provide an *adaptation model* consisting of *adaptation rules*. The rules define both the process of generating the adaptive presentation and that of updating the user model. An AHS may offer some built-in rules for common adaptation aspects and user model updates. This reduces the author's task of providing such rules.

The division into a *domain model* (DM), *user model* (UM) and *adaptation model* (AM) provides a clear separation of concerns when developing an adaptive hypermedia application. Unfortunately, a common shortcoming in many current AHS is that these three factors or components are not clearly separated [WHD00]. The AHAM model advocates the separation of these components in future AHS. This separation makes design of each part become clearer and make the system more flexible when each part can be changed. AHAM reference model make it easy to understand what adaptive hypermedia systems and provides a reference to compare different adaptive hypermedia system by translating the adaptive hypermedia systems to AHAM. AHAM aims to light the authoring burden on author side to make writing an adaptive hypermedia application more practical, of course authoring tools are needed to provide user-friendly interface [WHD99]

To understand how exactly the adaptive hypermedia system work we studied the behavior of *adaptation engine* (AE) which consists of *rule definition* and the *rule execution*. In our paper [WDAH00] we argued that *adaptation rules* should exist at the author level and the system level. System-defined rules simplify the task that remains for the author. However, for the analysis of the complete rule system this distinction is irrelevant and therefore not considered. In our new paper to the 12th ACM Conference on Hypertext and Hypermedia [WDD01], we describe *design issues for general-purpose adaptive hypermedia systems*. We define the rule language associated with AHAM as a subset of CA rules [BW00], and explain how the triggering works in AHAM. Because our AE is more powerful than that of most AHS, some of the design problems we present may not be present in many AHS with a simpler rule system. We focus on two design issues for the rule system in that paper:

- The designer wants to verify conditions that guarantee that the rule execution always terminates.
- The execution of the adaptation rules by the AE should be confluent. This means that under the same conditions (same domain model and same user model instance) the same action should always result in the same presentation and the same user model update.

We use research results for active database [BW00] to perform termination and confluence analysis, and improve on its results by applying domain knowledge. The analysis of is undecidable in general, but useful in many practical applications.

Aside from termination and confluence there is also the issue of efficiency (or "fast" termination and confluence). The detecting algorithm in [WDD01] has an exponential time complexity in some cases. In [WD01] we proposed sufficient conditions to guarantee termination and confluence for simple adaptive hypermedia applications. The complexity of

detecting algorithm is order of $N^2 \times M^2$ where N is the number of rule instances and M is the number of attributes in the user model.

Description of AHAM and AE are main contributions of my research on adaptive hypermedia, my future work is to describe two existing adaptive hypermedia system AHA [DC98] (designed by our group) and InterBook [B98] (designed by Brusilovsky et al.) in AHAM, and collect all my research results in my Ph.D. thesis.

References

[BW00] Baralis, E., Widom, J., "Better Static Rule Analysis for Active Database Systems", ACM Transactions on Database Systems, Vol. 25, nr. 3, 2000. (page numbers pending}

[B96] Brusilovsky, P., "Methods and Techniques of Adaptive Hypermedia". User Modeling and User-Adapted Interaction, 6, pp. 87-129, 1996. (Reprinted in Adaptive Hypertext and Hypermedia, Kluwer Academic Publishers, pp. 1-43, 1998.}

[B98] Brusilovsky, P., Eklund, J., Schwarz, E., "Web-based Education for All: A Tool for Developing Adaptive Courseware", Computing Networks and ISDN Systems (Seventh International World Wide Web Conference) 30, 1-7, 291-300.

[DC98] De Bra, P., Calvi, L., "AHA! An open Adaptive Hypermedia Architecture". The New Review of Hypermedia and Multimedia, pp. 115-139, 1998.

[DHW99] De Bra, P., Houben, G.J., Wu, H., "AHAM: A Dexter-based Reference Model for Adaptive Hypermedia". Proceedings of ACM Hypertext'99, Darmstadt, pp. 147-156, 1999.

[HS90] Halasz, F., Schwartz, M., "The Dexter Reference Model". Proceedings of the NIST Hypertext Standardization Workshop, pp. 95-133, 1990.

[HS94] Halasz, F., Schwartz, M., "The Dexter Hypertext Reference Model". Communications of the ACM, Vol. 37, nr. 2, pp. 30-39, 1994.

[WHD99] Wu, H., Houben, G.J., De Bra, P., "Authoring Support for Adaptive Hypermedia", Proceedings ED-MEDIA'99, Seattle, pp. 364-369, 1999.

[WHD00] Wu, H., Houben, G.J., De Bra, P., "Supporting User Adaptation in Adaptive Hypermedia Applications", Proceedings InfWet2000. Rotterdam, the Netherlands.

[WDAH00] Wu, H., De Bra, P., Aerts, A., Houben, G.J., "Adaptation Control in Adaptive Hypermedia Systems", Proceedings of the International Conference on Adaptive Hypermedia and Adaptive Web-Based Systems, Trento, Italy, LNCS 1892, pp. 250-259, Springer Verlag, 2000.

[WDD01] Wu, H., De Kort, Erik, De Bra, P., "Design Issues for General-Purpose Adptive Hypermedia Systems", Proceedings of the 12th ACM conference on Hypertext and Hypermedia (to appear)

[WD01] Wu, H., De Bra, P., "Sufficient Conditions for Well-behaved Adaptive Hypermedia Systems" (to appear)

Computer Science Reports **Department of Mathematics and Computer Science**
Technische Universiteit Eindhoven

If you want to receive reports, send an email to: m.m.j.l.philips@tue.nl (we cannot guarantee the availability of the requested reports)

In this series appeared:

97/02	J. Hooman and O. v. Roosmalen	A Programming-Language Extension for Distributed Real-Time Systems, p. 50.
97/03	J. Blanco and A. v. Deursen	Basic Conditional Process Algebra, p. 20.
97/04	J.C.M. Baeten and J.A. Bergstra	Discrete Time Process Algebra: Absolute Time, Relative Time and Parametric Time, p. 26.
97/05	J.C.M. Baeten and J.J. Vereijken	Discrete-Time Process Algebra with Empty Process, p. 51.
97/06	M. Franssen	Tools for the Construction of Correct Programs: an Overview, p. 33.
97/07	J.C.M. Baeten and J.A. Bergstra	Bounded Stacks, Bags and Queues, p. 15.
97/08	P. Hoogendijk and R.C. Backhouse	When do datatypes commute? p. 35.
97/09	Proceedings of the Second International Workshop on Communication Modeling, Veldhoven, The Netherlands, 9-10 June, 1997.	Communication Modeling- The Language/Action Perspective, p. 147.
97/10	P.C.N. v. Gorp, E.J. Luit, D.K. Hammer E.H.L. Aarts	Distributed real-time systems: a survey of applications and a general design model, p. 31.
97/11	A. Engels, S. Mauw and M.A. Reniers	A Hierarchy of Communication Models for Message Sequence Charts, p. 30.
97/12	D. Hauschildt, E. Verbeek and W. van der Aalst	WOFLAN: A Petri-net-based Workflow Analyzer, p. 30.
97/13	W.M.P. van der Aalst	Exploring the Process Dimension of Workflow Management, p. 56.
97/14	J.F. Groote, F. Monin and J. Springintveld	A computer checked algebraic verification of a distributed summation algorithm, p. 28
97/15	M. Franssen	λP-: A Pure Type System for First Order Loginc with Automated Theorem Proving, p.35.
97/16	W.M.P. van der Aalst	On the verification of Inter-organizational workflows, p. 23
97/17	M. Vaccari and R.C. Backhouse	Calculating a Round-Robin Scheduler, p. 23.
97/18	Werkgemeenschap Informatiewetenschap redactie: P.M.E. De Bra	Informatiewetenschap 1997 Wetenschappelijke bijdragen aan de Vijfde Interdisciplinaire Conferentie Informatiewetenschap, p. 60.
98/01	W. Van der Aalst	Formalization and Verification of Event-driven Process Chains, p. 26.
98/02	M. Voorhoeve	State / Event Net Equivalence, p. 25
98/03	J.C.M. Baeten and J.A. Bergstra	Deadlock Behaviour in Split and ST Bisimulation Semantics, p. 15.
98/04	R.C. Backhouse	Pair Algebras and Galois Connections, p. 14
98/05	D. Dams	Flat Fragments of CTL and CTL*: Separating the Expressive and Distinguishing Powers. P. 22.
98/06	G. v.d. Bergen, A. Kaldewaij V.J. Dielissen	Maintenance of the Union of Intervals on a Line Revisited, p. 10.
98/07	Proceedings of the workhop on Workflow Management: Net-based Concepts, Models, Techniques and Tools (WFM'98) June 22, 1998 Lisbon, Portugal	edited by W. v.d. Aalst, p. 209
98/08	Informal proceedings of the Workshop on User Interfaces for Theorem Provers. Eindhoven University of Technology ,13-15 July 1998	

		edited by R.C. Backhouse, p. 180
98/09	K.M. van Hee and H.A. Reijers	An analytical method for assessing business processes, p. 29.
98/10	T. Basten and J. Hooman	Process Algebra in PVS
98/11	J. Zwanenburg	The Proof-assistemt Yarrow, p. 15
98/12	Ninth ACM Conference on Hypertext and Hypermedia Hypertext '98 Pittsburgh, USA, June 20-24, 1998 Proceedings of the second workshop on Adaptive Hypertext and Hypermedia.	Edited by P. Brusilovsky and P. De Bra, p. 95.
98/13	J.F. Groote, F. Monin and J. v.d. Pol	Checking verifications of protocols and distributed systems by computer. Extended version of a tutorial at CONCUR'98, p. 27.
99/01	V. Bos and J.J.T. Kleijn	Structured Operational Semantics of χ , p. 27
99/02	H.M.W. Verbeek, T. Basten and W.M.P. van der Aalst	Diagnosing Workflow Processes using Woflan, p. 44
99/03	R.C. Backhouse and P. Hoogendijk	Final Dialgebras: From Categories to Allegories, p. 26
99/04	S. Andova	Process Algebra with Interleaving Probabilistic Parallel Composition, p. 81
99/05	M. Franssen, R.C. Veltkamp and W. Wesselink	Efficient Evaluation of Triangular B-splines, p. 13
99/06	T. Basten and W. v.d. Aalst	Inheritance of Workflows: An Approach to tackling problems related to change, p. 66
99/07	P. Brusilovsky and P. De Bra	Second Workshop on Adaptive Systems and User Modeling on the World Wide Web, p. 119.
99/08	D. Bosnacki, S. Mauw, and T. Willemse	Proceedings of the first international syposium on Visual Formal Methods - VFM'99
99/09	J. v.d. Pol, J. Hooman and E. de Jong	Requirements Specification and Analysis of Command and Control Systems
99/10	T.A.C. Willemse	The Analysis of a Conveyor Belt System, a case study in Hybrid Systems and timed μ CRL, p. 44.
99/11	J.C.M. Baeten and C.A. Middelburg	Process Algebra with Timing: Real Time and Discrete Time, p. 50.
99/12	S. Andova	Process Algebra with Probabilistic Choice, p. 38.
99/13	K.M. van Hee, R.A. van der Toorn, J. van der Woude and P.A.C. Verkoulen	A Framework for Component Based Software Architectures, p. 19
99/14	A. Engels and S. Mauw	Why men (and octopuses) cannot juggle a four ball cascade, p. 10
99/15	J.F. Groote, W.H. Hesselink, S. Mauw, R. Vermeulen	An algorithm for the asynchronous *Write-All* problem based on process collision*, p. 11.
99/16	G.J. Houben, P. Lemmens	A Software Architecture for Generating Hypermedia Applications for Ad-Hoc Database Output, p. 13.
99/17	T. Basten, W.M.P. v.d. Aalst	Inheritance of Behavior, p.83
99/18	J.C.M. Baeten and T. Basten	Partial-Order Process Algebra (and its Relation to Petri Nets), p. 79
99/19	J.C.M. Baeten and C.A. Middelburg	Real Time Process Algebra with Time-dependent Conditions, p.33.
99/20	Proceedings Conferentie Informatiewetenschap 1999 Centrum voor Wiskunde en Informatica 12 november 1999, p.98	edited by P. de Bra and L. Hardman
00/01	J.C.M. Baeten and J.A. Bergstra	Mode Transfer in process Algebra, p. 14
00/02	J.C.M. Baeten	Process Algebra with Explicit Termination, p. 17.
00/03	S. Mauw and M.A. Reniers	A process algebra for interworkings, p. 63.
00/04	R. Bloo, J. Hooman and E. de Jong	Semantical Aspects of an Architecture for Distributed Embedded Systems*, p. 47.

00/05	J.F. Groote and M.A. Reniers	Algebraic Process Verification, p. 65.
00/06	J.F. Groote and J. v. Wamel	The Parallel Composition of Uniform Processes wit Data, p. 19
00/07	C.A. Middelburg	Variable Binding Operators in Transition System Specifications, p. 27.
00/08	I.D. van den Ende	Grammars Compared: A study on determining a suitable grammar for parsing and generating natural language sentences in order to facilitate the translation of natural language and MSC use cases, p. 33.
00/09	R.R. Hoogerwoord	A Formal Development of Distributed Summation, p. 35
00/10	T. Willemse, J. Tretmans and A. Klomp	A Case Study in Formal Methods: Specification and Validation on the OM/RR Protocol, p. 14.
00/11	T. Basten and D. Bošnački	Enhancing Partial-Order Reduction via Process Clustering, p. 14
00/12	S. Mauw, M.A. Reniers and T.A.C. Willemse	Message Sequence Charts in the Software Engineering Process, p. 26
00/13	J.C.M. Baeten, M.A. Reniers	Termination in Timed Process Algebra, p. 36
00/14	M. Voorhoeve, S. Mauw	Impossible Futures and Determinism, p. 14
00/15	M. Oostdijk	An Interactive Viewer for Mathematical Content based on Type Theory, p. 24.
00/16	F. Kamareddine, R. Bloo, R. Nederpelt	Characterizing λ-terms with equal reduction behavior, p. 12
00/17	T. Borghuis, R. Nederpelt	Belief Revision with Explicit Justifications: an Exploration in Type Theory, p. 30.
00/18	T. Laan, R. Bloo, F. Kamareddine, R. Nederpelt	Parameters in Pure Type Systems, p. 41.
00/19	J. Baeten, H. van Beek, S. Mauw	Specifying Internet applications with *DiCons*, p. 9
00/20	Editors: P. v.d. Vet and P. de Bra	Proceedings: Conferentie Informatiewetenschap 2000, De Doelen, Utrecht, 5 april 2000, p. 98
01/01	H. Zantema and J. v.d. Pol	A Rewriting Approach to Binary Decision Diagrams, p. 27
01/02	T.A.C. Willemse	Interpretations of Automata, p. 41
01/03	G.I. Joigov	Systems for Open Terms: An Overview, p. 39
01/04	P.J.L. Cuijpers, M.A. Reniers, A.G. Engels	Beyond Zeno-behaviour, p. 15
01/05	D.K. Hammer, J. Hooman, M.A. Reniers, O. van Roosmalen, A. Sintotski	Design of the mine pump control system, p. 69
01/10	7 e Nederlandse Testdag editors L.M.G. Feijs, N. Goga, S. Mauw, T.A.C. Willemse	
01/11	Editors: P. de Bra, P. Brusilovsky and A. Kobsa	Proceedings: Third Workshop on Adaptive Hypertext and Hypermedia Sonthofen, Germany, July 14, 2001 Århus, Denmark, August 15, 2001