Embedding Information Retrieval in Adaptive Hypermedia: IR meets AHA!

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

ABSTRACT

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

Keywords

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

1. INTRODUCTION

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

Certain areas of (adaptive) hypermedia research identify the need to consider content that is outside the application, outside meaning that the data fragments are not under direct control of the application. Often, this is interpreted in the way that the application can have access to those data fragments to include them in the presentation, but there is less knowledge available to (the designer of) the application about which fragments to consider when designing the application (and its adaptation). Applications in the area of Web information systems often

take an approach like this, not knowing the actual fragments existing at the level of instances, only having information at the (schema) level of classes. Several research projects, e.g. Hera [14,15], focus on the control of adaptation in the context of data that is available only at the schema level and that is instantiated at run-time. While this approach allows for a limited notion of "outside" data, there is a need to go further. A natural next step is to include the concept of information retrieval (IR). Allowing the application to use data fragments that are obtained through IR mechanisms has an impact on the design in the sense that the designer of the application has a limited insight in the fragments and their properties. The major challenge with this step is to understand the consequence of including IR in AH applications.

This challenge leads to a number of relevant research questions.

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

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

2. ADAPTIVE HYPERMEDIA

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

2.1 AHAM

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

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

2.2 AHA!

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

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

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

3. RETRIEVING CONTENT FROM OUTSIDE

3.1 Content Inside

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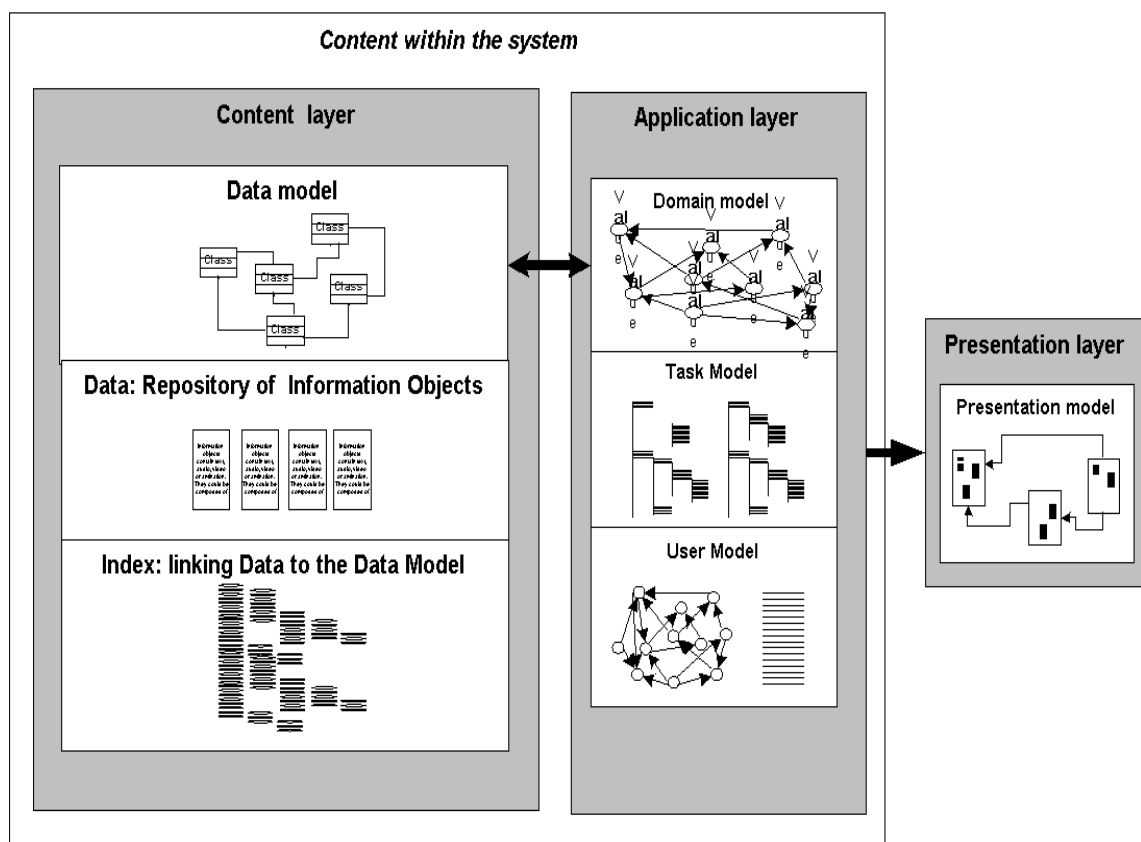

Fig. 1. Content layer inside the system.

3.2 Content Outside

Placing the content outside of the application is a move that is already visible in the context of a large class of information systems known as Web-based Information Systems (WIS) [18]. They are information systems, often database-driven, that exploit the Web paradigm and use Web technologies to retrieve data from sources reachable through the Web and deliver them to their users (see Figure 2 for Hera's perspective on WIS [14,15]). Typically, a WIS delivers the data to the users in a web or hypermedia presentation, requiring that in comparison to a handcrafted (adaptive) hypermedia application, a WIS needs to generate (automatically) a hypermedia presentation for the data to be delivered. We mention (general) methodologies or

frameworks for WIS engineering, like Araneus [20], WebML [8], UWE [19], and Hera [14,15], of which only a few explicitly include the adaptation aspect. Most of these approaches consider the outside content to be known at the schema (class) level, such that the design is based on that level, leaving the actual run-time retrieving and rendering of the data at instance-level a separate and less prominent issue.
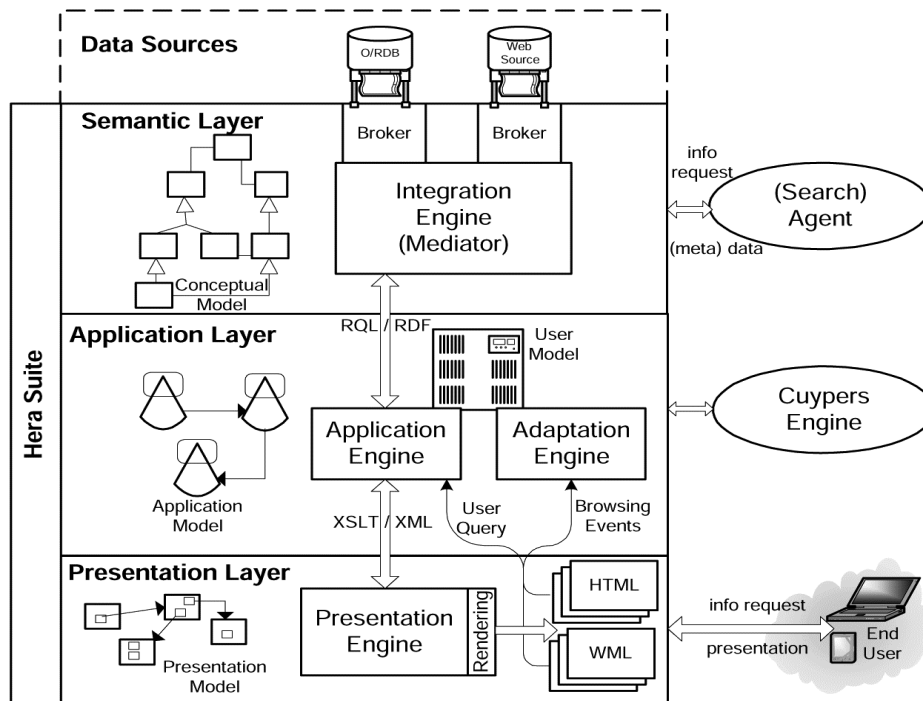

Fig. 2. WIS architecture (Hera view).

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

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

3.3 IR Extension to AH Model and Application

In the case of retrieving content from outside the application, we may want to distinguish between different situations. It is possible to just have an "outside" fragment, while it is also

possible to have a search (retrieval) request that results in an entire set of fragments that have to be made accessible (we address these issues further in the next section). In each of the situations, we are dealing with outside contents that have to be connected to the given AH application.

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

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

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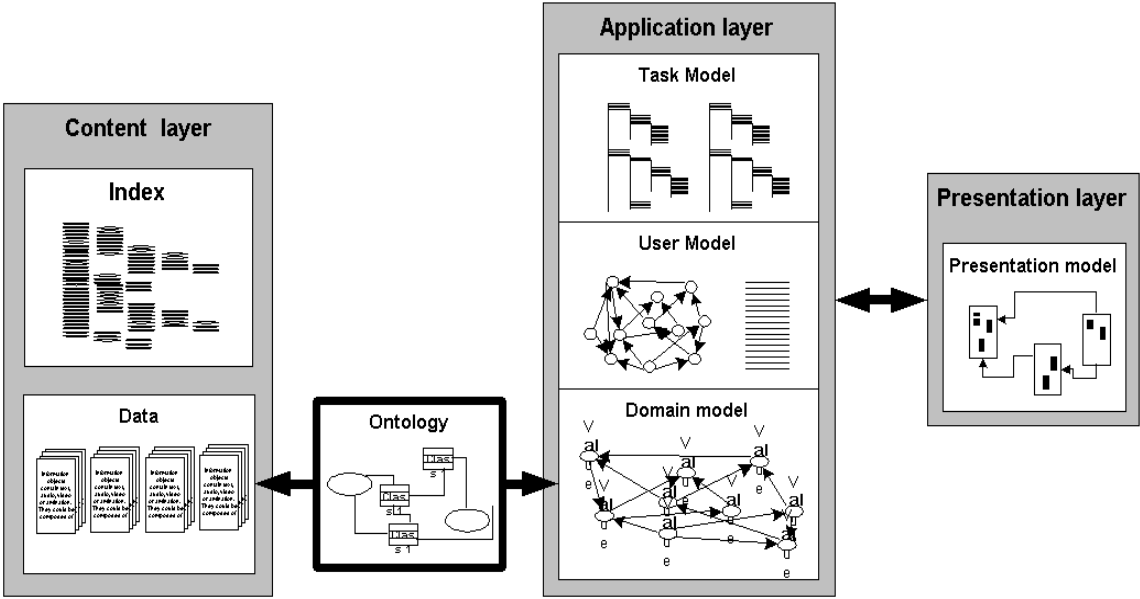

Fig. 3. Content available through ontology.

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

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

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

4. IR EXTENSION ALTERNATIVES

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

4.1 Outside Fragment

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

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

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

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

4.2 Retrieval through Search

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

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

5. PRESENTING RETRIEVAL RESULTS

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

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

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

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

6. RETRIEVAL AND USER MODEL

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

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

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

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

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

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

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

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

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

7. RELATED WORK

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

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

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

8. CONCLUSION

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

application. Concentrating on these aspects of information retrieval, the next step is to translate the matured conceptual ideas to concrete implementations.

REFERENCES

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

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