
Authoring adaptive educational hypermedia on the semantic desktop

Maurice Hendrix* and Alexandra Cristea

Department of Computer Science
The University of Warwick
Coventry CV4 7AL, UK
E-mail: maurice@dcs.warwick.ac.uk
E-mail: acristea@dcs.warwick.ac.uk
*Corresponding author

Wolfgang Nejdl

L3S Research Center
University of Hannover
30539 Hannover, Germany
E-mail: nejdl@l3s.de

Abstract: Personalisation of information is a desired feature, but authoring and creation of adaptive hypermedia is a difficult endeavour. Especially, manual annotation is *the* bottleneck of such authoring. Thus it is vital to support the authoring process by reusing automatically generated metadata. Here, we show the integration of the generic AH authoring environment MOT into a semantic desktop environment, as well as its first small-scale evaluation. In this set-up, the semantic desktop environment provides the rich source of automatically generated metadata, while MOT allows enhancing this metadata manually, as needed for the authoring of an adaptive environment – here, an adaptive course.

Keywords: authoring; Adaptive Hypermedia; AH; adaptive educational hypermedia; semantic desktop; metadata; RDF; Common Adaptation Format; CAF; semiautomatic adding; learning technology.

Reference to this paper should be made as follows: Hendrix, M., Cristea, A. and Nejdl, W. (2007) 'Authoring adaptive educational hypermedia on the semantic desktop', *Int. J. Learning Technology*, Vol. 3, No. 3, pp.230–251.

Biographical notes: Maurice Hendrix is a PhD student in the Department of Computer Science at the University of Warwick. His research interests are in automatising the design and authoring process for adaptation. He has previously finished his masteral studies at the Eindhoven University of Technology, the Netherlands, on the same subjects. His work has already led to three papers at international conferences: AH 2006, eLearn 2006 and ECTEL 2006.

Dr. Alexandra Cristea is an Associate Professor in the Department of Computer Science at the University of Warwick. She is the Head of the Intelligent and Adaptive Systems Group. Her research interests include adaptive educational systems, authoring of adaptive hypermedia, user modelling, intelligent tutoring systems, semantic web technologies, concept mapping, and artificial

intelligence. She has published more than 100 papers on these subjects. She is leading a new Minerva project entitled Adaptive Learning Spaces (ALS) started on October 2006. She was previously working at the Eindhoven University of Technology, the Netherlands, as Assistant Professor. She received her PhD at the University of Electro-Communications, Tokyo, Japan. She has been an organiser of workshops, co-organiser, panellist and programme committee member of various conferences in her research field. She is an executive peer reviewer of the *IEEE LITF Education Technology and Society Journal* and she is co-editor of the *Advanced Technologies and Learning Journal*.

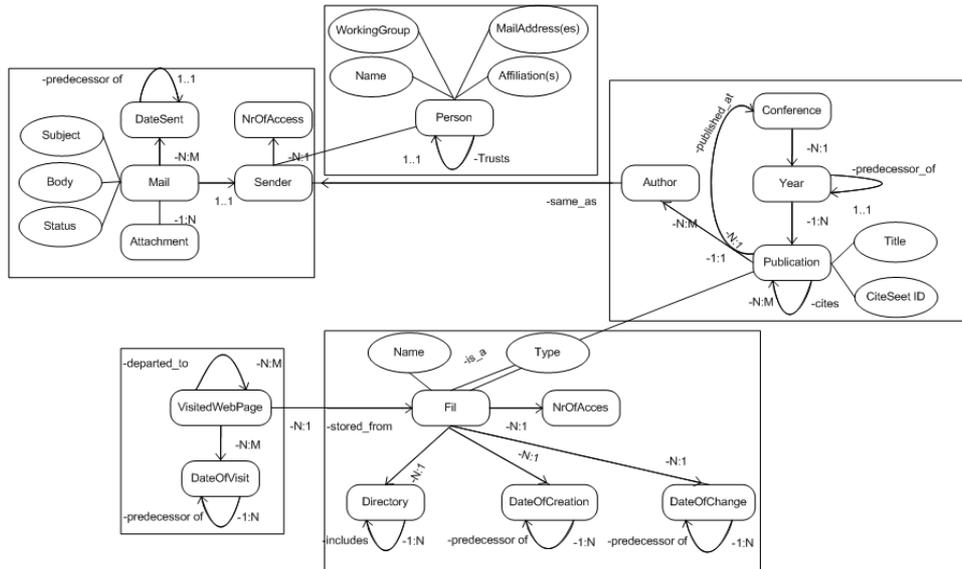
Prof. Dr. Wolfgang Nejdl (1960) has been a Professor of Computer Science at the University of Hannover since 1995. He received his MSc (1984) and PhD (1988) at the Technical University of Vienna; was an Assistant Professor in Vienna from 1988 to 1992, and an Associate Professor at the RWTH Aachen from 1992 to 1995. He was a Visiting Professor at Xerox PARC, Stanford University, University of Illinois at Urbana-Champaign and EPFL Lausanne. He heads the Knowledge Based Systems (<http://www.kbs.uni-hannover.de/>) and L3S Research Centers (<http://www.l3s.de/>), and does research in the areas of semantic web technologies, peer-to-peer information systems, adaptive hypermedia systems and artificial intelligence. In the L3S he coordinates the Network of Excellence PROLEARN, and participates as core partner in the KnowledgeWeb and REWERSE Networks of Excellence focusing on semantic web technologies. Three new projects have started in 2006 at L3S. He published over 200 articles and was programme committee and editorial board member of numerous conferences and journals (see <http://www.kbs.uni-hannover.de/~nejdl/>).

1 Introduction

Authoring of adaptive hypermedia is a difficult endeavour, and researchers are therefore looking to find new sources of automatic generation of at least part of the content alternatives required. In this context, the semantic desktop (Chirita *et al.*, 2005; Semantic Desktop, 2006) looks like an ideal candidate, as it promises rich structures: resources can be categorised by flexible, information-rich ontologies, with semantic links expressing various kinds of semantic relationships between resources. The semantic desktop stores not only a paper's filename, for example, but also information on where it was published, when and by whom, which of the colleagues sent it, and how often and in what context it was accessed by the current user (see Figure 1). All these metadata are generated automatically by other applications, and stored in an application-independent way as RDF metadata, in the user's personal data store. This rich set of metadata enables the user to easily retrieve material for various contexts: *e.g.*, for the current running example, when he wants to select fitting materials for a course lecture. Of course, a course author still has to create the basic lesson material, and might need to add more metadata information about course structures and other relevant attributes, such as any pedagogically relevant attributes. In this paper we focus on course authoring on the semantic desktop, as well as its first prototype evaluation. Specifically, we show the interaction and exchange of data between the Beagle++ environment (Beagle++ website, 2006; Chirita *et al.*, 2006), which is an advanced search and indexing engine for the desktop, generating and utilising metadata information, and the adaptive hypermedia

authoring environment MOT (Cristea and De Mooij, 2003a; MOT homepage, 2006), a sophisticated system for authoring personalised e-courses, and a small-scale evaluation of this whole process.

Figure 1 RDF schema of semantic desktop metadata in Beagle++



Source: Beagle++ website (2006)

The remainder of this paper is organised as follows. Section 2 introduces the generic motivating scenario of our work. Section 3 presents the idea of enriching metadata, as well as the input and output schemas applied. Section 4 describes the conversion from both a conceptual and an implementation point of view. We are currently evaluating our process. A small-scale evaluation of the prototype was performed to gain some qualitative feedback. Also a larger-scale user study has been performed. In Section 5 we present both evaluations and the results of the small-scale evaluation, with some pointers from the larger-scale evaluation, which is still being processed. In Section 6 we discuss the results, draw conclusions, and we discuss possible directions for future research.

2 Motivating scenario

We use a scenario for adaptive authoring that builds upon a combination of automatically and manually generated metadata, as introduced in Cristea *et al.* (2006):

Dr. Brown prepares a new online course on Adaptive Hypermedia for 4th year masteral students at Warwick. The university work distribution allocates a limited amount of time for this, equivalent to the creation of a static, linear course. However:

- owing to the fact that Dr. Brown not only considers it useful to be able to extend the course in the future with more alternative paths guided by adaptivity, but also,
- because she wants to benefit from automatic help during the authoring process,

she uses a *concept-based adaptive educational hypermedia authoring environment with adaptive authoring support*, MOT (Cristea and De Mooij, 2003a; MOT homepage, 2006). This decision costs her slightly more time than the static course creation, as she has to *manually* divide her course into conceptual entities with explicit, independent semantics and semantic labelling.

The advantage is that the *adaptive authoring system* can afterwards *automatically* enrich the course based on pedagogical strategies. For instance, the system can consider the version of the Adaptive Hypermedia course created by Dr. Brown as the version for *beginner* students, who are only moderately interested in the subject, and do not aspire for higher grades or deep knowledge on the subject. For *advanced* students wishing to pass the course with high honours, or simply wanting to acquire more information for their future professional lives, the adaptive authoring system can use *semantic personal desktop search* to automatically find on Dr. Brown's desktop any existing scientific papers that are relevant to the current course. These scientific papers can be used as alternative or additional material to the main storyline of the static course. This mechanism builds upon the following assumptions:

- As Dr. Brown is a specialist in the subject taught, her interest is wider than that given by the limitations of the course; she therefore both publishes and reads papers of interest on the subject, which are likely to be stored on her computer.
- These papers can be considered as useful *extra* resources for the current course, and can therefore be reused in this context.
- As this storing process has taken place over many years, Dr. Brown may not know exactly where on her computer each article relevant to the current course is.
- However, Dr. Brown has been using Beagle++ Semantic Desktop System (Beagle++ website, 2006; Chirita *et al.*, 2006) to store both papers and all relevant metadata automatically, in RDF format.

This situation can be exploited by the authoring tool; a quick search will find some of Dr. Brown's own papers on Adaptive Hypermedia, as well as some papers written by her colleagues on the same topic, such as the paper by Brusilovsky, called 'Adaptive Educational Hypermedia: From generation to generation' (Brusilovsky, 2004), or the paper 'Adaptive Authoring of Adaptive Hypermedia' (Cristea and De Mooij, 2003a). She may have saved these papers by herself, or might have received them by e-mail, from a colleague working in the same field, or may have used her browser's bookmarks to mark their position on the web (thus they are stored as shown in Figure 1).

In order for these retrieved resources to be relevant to the overall Adaptive Hypermedia course, two conditions have to be fulfilled:

- 1 The domain concept in the course where each resource is most relevant has to be found (*the right information*).
- 2 The resource has to be bound afterwards to that particular domain concept (*in the right place*).

This means that the first paper can be added to the course at a higher level, somewhere next to the explanation of generic principles of adaptive hypermedia, whereas the second paper should only be placed somewhere in connection with the authoring process in adaptive hypermedia, otherwise its content might be too specific to follow.

How can Brown find the right resource and store it in the right place? The search can take place via the keywords labelling both the course pieces created by Brown, on one hand, and the matching keywords labelling the papers and resources on her desktop (as described in Section 4) on the other hand. The following sections will describe in more detail how Dr. Brown can enrich her course semiautomatically, without much extra work, as well as keep the overall control and overview at all times.

3 Enriching metadata and their schemas

As highlighted by the scenario, both *automatically* generated metadata as well as *manually* added metadata are necessary. *Automatically generated metadata* enable description and retrieval of appropriate articles. *Manual annotation* allows, besides a higher level of control, the addition of extra content, as well as of attributes such as pedagogical weights and labels, which are necessary to build the final adaptive course product. The next subsections detail these two kinds of metadata, from the point of view of the two worlds that come together: *semantic desktop* and *authoring of adaptive hypermedia*.

3.1 Metadata in the semantic desktop

Beagle++ (Beagle++ website, 2006; Chirita *et al.*, 2006) is an extension of Beagle (Beagle website, 2006) that generates and stores additional metadata describing the documents indexed by Beagle, as well as other resources and their relationships on a semantic desktop. Such additional metadata automatically annotates material the user has read, used, written or commented upon. Applications that generate such desktop behaviour metadata are: applications for file access or editing on the desktop; internet browsing and files stored from the internet; and mail exchanges with their respective files stored from mail attachments (Chirita *et al.*, 2006; Ghita *et al.*, 2005). Figure 1 depicts an instance of this ontology illustrating files annotated with their publication metadata, file system attributes, web history, and mailing context (*e.g.*, the fact that files are attached to specific e-mails). In the figure, the rectangle in the upper-left part describes e-mails (with subject, body, sender and status attributes) and their attachments; the rectangle in the upper-right part shows publications written by different authors at various conferences (with title, publication year, *etc.*); the rectangle in the lower-left part displays the web cache history (web pages visited and dates of visits). The rectangle in the lower-right part of the figure shows files that were saved on a person's desktop, with their names and specific directories. Files on the desktop may have been saved via any of the other three processes (from e-mails, from websites, or from conferences), so an attachment entity and a file entity may refer to the same object. There is currently, however, no version control in Beagle++, so no clear relationship found between versions of the same file.

3.2 Metadata in authoring environments for adaptive courses

For our running example, the files above and their metadata are not enough to generate a complete course. Concretely, we also have to add information about the hierarchical structure and default presentation order of the material in the context of a lesson, as well as supplementary pedagogical annotations describing which target students the material

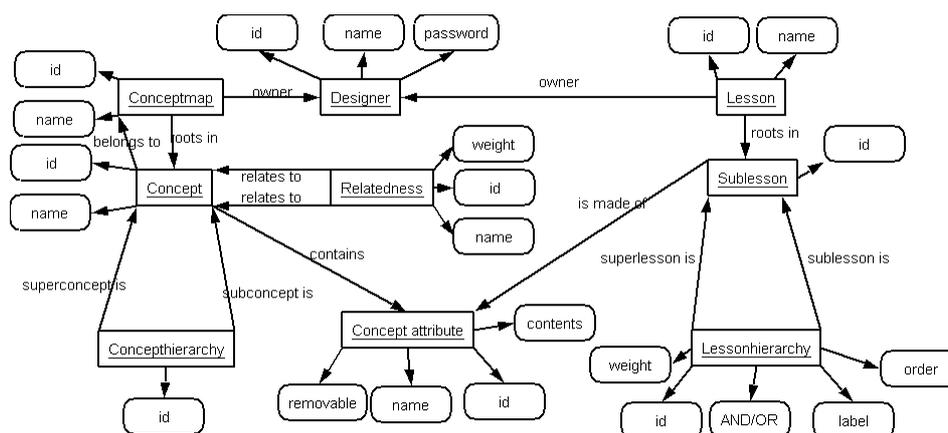
is best suited for (e.g., *beginner* versus *advanced*). Figure 2 shows this target schema for authoring of adaptive lessons, as defined in MOT (My Online Teacher; originally proposed in Cristea and De Mooij (2003a), and currently extended at the MOT homepage), an adaptive hypermedia authoring system. The schema describes two models used in the adaptive hypermedia authoring paradigm: the *domain map* (Figure 2, left side) and the *lesson* (Figure 2, right side).

A *domain map* is a hierarchical representation of domain concepts. Each domain concept is built from domain concept attributes, containing or linking to semantically independent (e-learning) content alternatives describing the same domain concept. In addition to being involved in hierarchical relations, domain concepts can also participate in relatedness connections, binding together related domain concepts.

A *lesson* is a hierarchical collection of sublessons. Each lesson represents a filtered, pedagogically labelled, weighted and ordered version of the domain concept attributes (see the relation between sublessons and domain concept attributes, Figure 2).

The figure also contains more details omitted here, as they are not relevant to the current paper.

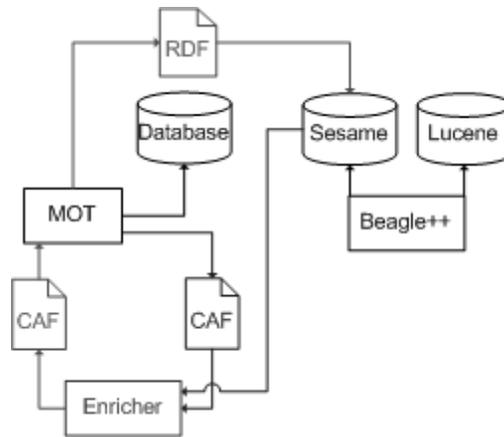
Figure 2 RDF schema of authoring metadata in MOT



Source: Cristea and De Mooij (2003b)

4 Conversion and enrichment process in steps

As can be seen in Figure 3, Beagle++ stores all its metadata in the Sesame RDF database (Schlieder and Naumann, 2006). All Beagle++ components, which *generate* metadata (for example, the e-mail, publication, web cache and file metadata generators) add metadata to this database. All Beagle++ components that *use* metadata (for example, the search and indexing module, the ranking module or the browsing modules) retrieve their data from this repository, and, in some cases, write back new data (such as the PageRank value for documents or other resources).

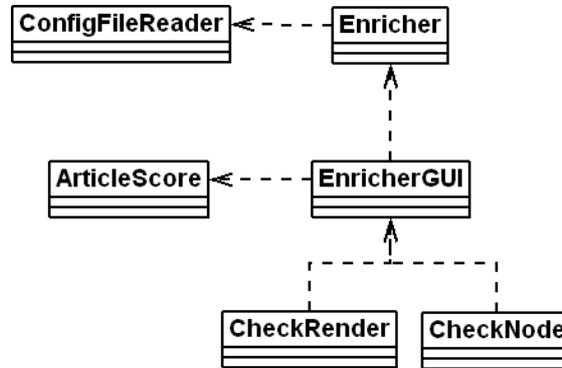
Figure 3 Dataflow and system overview

It is easy to accommodate additional modules in this environment by writing appropriate interface components, which read and write from this repository. This is what we have done for *MOT* (Cristea and De Mooij, 2003a; MOT homepage, 2006), as described in this paper. For our scenario, we have focused on the semiautomatic addition of articles stored on the user's desktop to a *MOT* lesson (Cristea and De Mooij, 2003a). In *MOT*, this addition is done to an existing lesson. Based on her pedagogic goals, the author can process the data by changing labels and weights and adding other information on the article. After this enrichment the lesson can be imported back into the RDF store. We use Common Adaptation Format (CAF) (Cristea *et al.*, 2005), a system-independent XML exchange format) in order to simplify the transformation process from RDF to the *MOT* MySQL storage format.¹

4.1 RDF to CAF transformation

In this step, we transform available articles into a CAF sublesson. This is a semiautomatic step, where the author selects a lesson in *MOT* and then the system searches (based on keywords and title) for related articles in the Sesame data store. This is done by a Java utility, which takes the metadata in the Sesame store and the lesson in CAF format as input and generates a new (updated) lesson as CAF file. As the articles are stored in files on the desktop, as an extra step, they have to be physically transported to the supporting server for *MOT*. This step, as well as the fact that the author can check first what is added before performing the addition, ensures that confidential files are not mistakenly publicised. A simplified UML model² is given in Figure 4.

Figure 4 Simplified UML model of the Enricher



As we can see, the Enricher is controlled by the EnricherGUI user interface class. To decide which resource to add, it is necessary to compare the scores of resources and the places in the hierarchy where they should be added. The Enricher first constructs a list of *{resource, score, place in the hierarchy}* triplets. These are stored using the ArticleScore class. These triplets are then compared to determine the ones to be added.

The Enricher and the EnricherGUI use the configFileReader to read the default XML configuration file (Figure 5). This could, in principle, be changed by an external application.

Figure 5 The XML configuration file

```

<Config>
  <baseURI>http://localhost</baseURI>
  <schemaName>articleSchema.RDF#</schemaName>
  <SesameUser>user</SesameUser>
  <SesamePassword>password</SesamePassword>
  <SesameDb><RDF-db></SesameDb>
  <SesameServer>http://<URI>/Sesame</SesameServer>
  <addedLessonWeight>50</addedLessonWeight>
  <addedLessonLabel>advanced</addedLessonLabel>
  <addedLessonAttrWeight>50</addedLessonAttrWeight>
  <addedLessonAttrLabel>advanced</addedLessonAttrLabel>
</Config>
    
```

The configuration file offers settings for the base URI, name of the main schema used, user and password with which to connect to the Sesame store, the database name used by Sesame, the URI pointing to the Sesame store, and the default weights and labels for the added lessons and attributes. These weights and labels (see also Figure 2) enable the adaptation. In our example in Figure 5, they determine the fact that added material is only considered appropriate for advanced learners.

4.2 Enrichment of the lesson and domain model

As MOT is mainly a tool for authoring educational (adaptive) material, the internal information structures are based on strict hierarchies (see Figure 2). When enriching the domain model and lesson, we obviously want to get the right information in the right place in this hierarchy.

To achieve this, the program first queries the Sesame database, using as search terms title and keywords of each domain concept found in the current lesson. This procedure may result in the same file being relevant in many places within the hierarchy. The RDF query in the SeRQL (Broekstra and Kampman, 2004) language looks as follows:

```
select x from x {p} y where y like "*keyword" ignore case
```

Next, for each resource, the place with highest *relevance* is sought.³ For computing relevance, we have developed two slightly different theoretical alternatives, as follows.

4.2.1 Concept-oriented, relevance-ranking method

The first relevance-ranking method is computing relevance according to Equation 1:

Equation 1 Concept-oriented ranking

$$\text{rank}(a,c) = \frac{|k(c) \cap k(a)|}{|k(a)|},$$

where:

$\text{rank}(a,c)$ is the rank of article a with respect to the current domain concept c ;

$k(c)$ is the set of keywords belonging to the current domain concept c ;

$k(a)$ is the set of keywords belonging to the current article a ;

$|S|$ = the cardinality of the set S , for a given set S .

We have chosen to implement two different versions, one with a set, and the intersection operation on sets, and one with multisets or bags and the intersection operation on bags.

Within the set theory, a *bag* can be formally defined as a pair (A, m) where A is some set and $m: A \rightarrow N$ is a function from A to the set $N = \{1,2,3,\dots\}$ of (positive) natural numbers. Set A is called the underlying set of elements. For each $a \in A$, the multiplicity (*i.e.*, number of occurrences) of a is the number $m(a)$.

The intersection can then be defined as $(A \cap B, f(A \cap B))$ where $f(x) = \min\{m(x \in A), m(x \in B)\}$.

For example:

$$\{(a,2),(b,1)\} \cap \{(a,3),(x,1)\} = \{(a,2)\}.$$

The reason for using bags instead of sets is that, for our system, the number of times keywords appear in certain texts can be relevant in itself (not just *which* keywords). A text containing a greater number of occurrences of a specific keyword could be a better match for that keyword than a text with only one occurrence of the respective keyword.

Equation 1 shows how many of the keywords (shared by the article and the concept) are present in the article; *e.g.*, if an article has less keywords than another one, but the keywords shared between articles and concept are the same, the former article will have a higher rank and ‘win’ the place in the concept. Therefore, this formula is concept-oriented, in the sense that articles ‘battle’ for the same concept.

The actual implementation of the concept-oriented formula is not trivial for the RDF database that we have, as there is no keyword property available for retrieved objects (articles). Therefore, these keywords had to be extracted from the title property of the object. This means that the computation of the cardinality of the keywords, $|k(a)|$, can be interpreted in many ways. The easiest way is to set it to the number of words in the title property. Another option is to compute the number of relevant keywords (*e.g.*, eliminating words such as ‘the’, ‘a’, *etc.*) by connecting them to a given ontology or dictionary, such as Wordnet. Both options were implemented, but the current Wordnet connection is very slow and thus inefficient.

4.2.2 Article-oriented, relevance-ranking method

A second implementation of the rank is given by Equation 2.

Equation 2 Article-oriented ranking

$$rank(a,c) = \frac{|k(c) \cap k(a)|}{|k(c)|}.$$

The equation shows how many of the keywords (shared by the article and the concept) are present in the concept; *e.g.*, if a concept has less keywords than another one, but the keywords shared between article and concept are the same, the former concept will have a higher rank and ‘win’ the article. This formula thus shows to which of the concepts the article should be attributed. Therefore, this formula is article-oriented, in the sense that concepts ‘battle’ for the same article.

4.2.3 Comparison of relevance-ranking methods with information retrieval success measurements

Information retrieval is a well-known area of computer science, which deals with ranking of retrieved results (Ghosh and Lee, 2006). Information can be retrieved from various sources, such as databases and XML files. A widely accepted method to measure the success of an information retrieval system is the use of *precision* and *recall*. Precision shows how close a given retrieved object is related to what the user asked, whilst recall tells us how many of the relevant objects were retrieved. Precision and recall can be measured using the following formulae:

Equation 3 Recall

$$\frac{|A \cap B|}{|A|}.$$

Equation 4 Precision

$$\frac{|A \cap B|}{|B|}$$

where:

$A \subseteq U$ = the set of relevant objects

$B \subseteq U$ = the set of retrieved objects

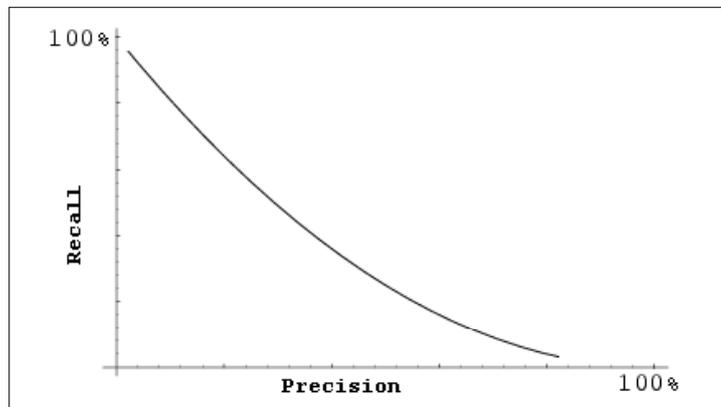
U = the total set of documents used in the experiment

$|S|$ = the cardinality of the set S .

In information retrieval, usually, the standard Boolean model is used, which means that relevance is determined in a binary way: either a document is relevant, or it is irrelevant. The precision and recall are computed on a set of training data, with a set of test queries. A user would have to manually assess the relevance of the resources in the training set with regard to the query.

Ideally, one would want to achieve a high level of recall, without sacrificing precision. However, as the figure below shows, as the recall rate increases, the precision usually deteriorates very rapidly. Moreover, this trend is not linear. Depending on the type and quality of the information and the retrieval algorithm, precision will change. The ideal balance between precision and recall will be found at a payoff point that represents an acceptable balance of the precision and recall lines in a graph. Traditionally, this point is defined by either fixing the recall (thus the number of allowed articles – see Google), or by fixing the minimum precision threshold.

Figure 6 Precision and recall



Source: See the Measuring Search Effectiveness web page

If we compare the precision and recall method in information retrieval to our first method of concept-oriented results ranking, they look surprisingly similar. Indeed, we can apply substitution to Equation 4 and obtain Equation 1 (see Equation 5).

Equation 5 Substitution in the precision formula

$$\left[\frac{|A \cap B|}{|B|} \right] (A := k(c); B := k(a)) = \frac{|k(c) \cap k(a)|}{|k(a)|}.$$

The substitution is interesting, as, indeed, the set of keywords from the original concept in MOT could be considered to represent the set of (potentially) relevant objects, whilst the set of article keywords represents the retrieved objects. Therefore, our formula represents some sort of precision measurement: how precisely the chosen article expresses the chosen concept.

Similarly, by comparing the formula for the article-oriented retrieval with the information retrieval success measurements, one can see that this is comparable to a computation of the recall. Unlike in information retrieval, however, when designing a course, precision is probably preferable to recall, because adding too many articles can be confusing instead of helpful. This is the reason why we have limited our recall to a threshold of three⁴ articles maximum per concept. However, our two formulas respond to two different goals:

- 1 finding the correct article for a given concept (concept-oriented)
- 2 finding the correct place in the hierarchy for a given article (article-oriented).

Just as in recall versus precision in information retrieval, these two goals can be leading to different responses and error structures.

The first might extract only relevant articles, but might attach them in too many places (the same article might be relevant for more than one concept, and trying too hard to find an article for each concept might, per formula alone, generate too many copies of it⁵).

The second might distinguish between merit concepts only, letting concepts compete, but may try to add too many articles per concept. For finding the optimal retrieval for a certain dataset, these two formulas should be balanced against each other, finding the optimal pay-off point, in the same way it is done in information retrieval with recall and precision.

4.3 CAF to MOT transformation

The import of CAF files into MOT is implemented as a PHP script and done in two steps. First, the domain map is processed. During this process, an associative array is created as a look-up table. In this table, every domain concept attribute created is stored as follows:

```
[name_of_domain_map\name_of_concept_1..\
name_of_concept_Nattribute_name] => attribute_id
```

This allows a faster look-up in the next step than would be possible by doing a direct query on the database.

In the second step, the lesson and its parts are generated. While importing, conflicts between lesson names or names of domain maps may occur. There are currently three ways of handling this:

- 1 The author can choose to not care about it, and as a result get multiple domain maps and lessons with the same name in MOT.
- 2 The second option is to have domain maps and lessons with conflicting names renamed. In this case, after the import, the user will see a list with the conflicting names and be asked for new names.

- 3 The third option is to merge lessons and domain maps with the same name, taking into consideration that they might be extensions of each other (see Figure 7). For domain maps, merging means that two domain maps with the same name are compared, concept by concept. Domain concepts and attributes that were not present before are added. This merge option is essential to our application. For our purposes a lesson is exported, then articles are added to the CAF file and then the CAF file is imported back into MOT with the merge option. Using this merge option means that possible additions made to the lesson between the export and the import will not be deleted, and IDs of already existing attributes will not change, so other lessons using one of the adapted domain maps will not be affected. The merge option is especially necessary to be able to update goal maps, which are already in use either by people or by other goal maps. First, the domain maps and goal map are imported as a duplicate with the same name and then merged into the old domain maps and goal map. The merge completely rewrites the domain and goal maps with similar name as the imported one, leaving alone only concepts and attributes that were existent in the old maps and do not appear in the import maps.

As the CAF format, which is a portable format used in several adaptive hypermedia systems, does not include the lesson name, this can be changed or imported from a separate XML file. Figure 7 shows a screenshot of the CAF import interface.

Figure 7 Importing from CAF to MOT

My On-line Teacher: Import CAF file

Username: finland [Close This Window](#)

Select CAF- or RDF-file to import:

Use extra info xml-file (select the extra-info file)*

Enter the lesson name* :

For lessons/concepts with existing names:

Leave them as they are

Rename lesson/concept if the name exists

Merge lesson/concept if the name exists

Source: My On-line Teacher

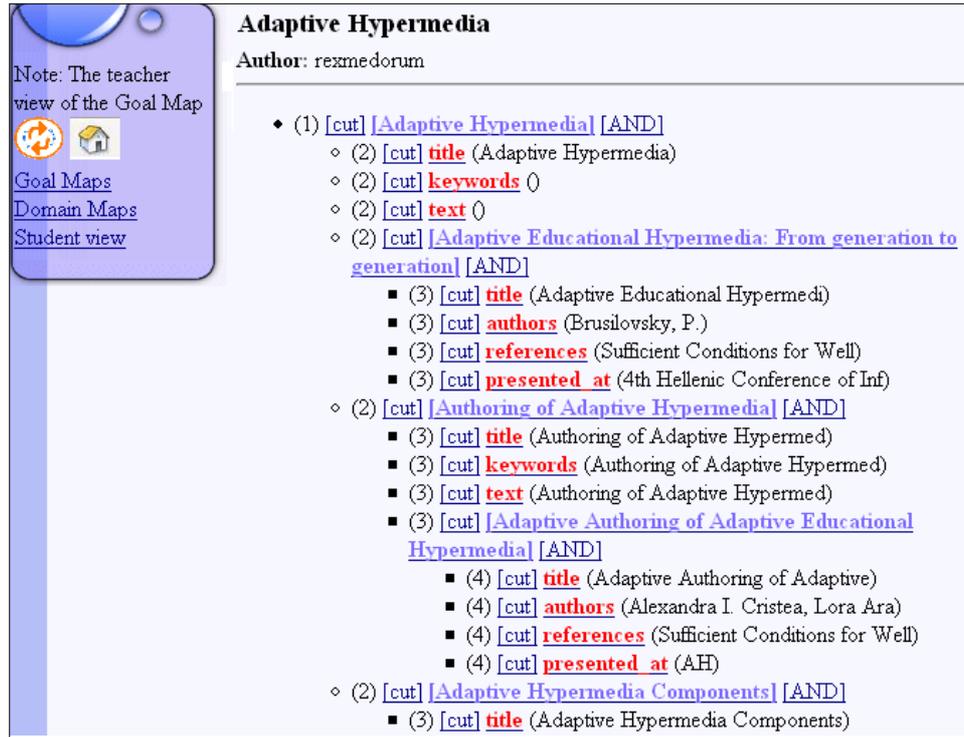
4.4 Working in MOT

In MOT, the data and metadata imported from Sesame can be reordered and edited.

Figure 8 shows an extract of editing in the lesson environment (the import from Beagle++ generates extra information both for the domain map as well as for the lesson). The paper called ‘Adaptive Educational Hypermedia: From generation to generation’ and all its relevant metadata (title, authors, references, conference where presented) have been added automatically during extraction to the ‘Adaptive Hypermedia’ domain concept, at the highest level in the hierarchy. The paper ‘Adaptive Authoring of Adaptive

Educational Hypermedia’ has been added to the domain concept ‘Authoring of Adaptive Hypermedia’, in a lower position in the hierarchy, as it is a more specific paper. The author can now manually add pedagogical labels to this new material, labelling the material ‘adv’ for advanced learners.

Figure 8 Adding manual metadata in a MOT lesson



4.5 MOT to RDF transformation

The MOT to RDF transformation has two steps: MOT to CAF (an intermediate format for data exchange) and CAF to RDF. In the following, these two steps are explained.

4.5.1 MOT to CAF transformation

Exporting MOT to the CAF intermediate format is implemented as a PHP script. First, the lesson is constructed, based on the target lesson hierarchy of the CAF XML file. During this process, a list of the mappings between lesson objects and domain objects is kept. After producing the lesson, the corresponding domain map is generated. Every domain map used in the lesson is added to the domain model, with all its sublessons. For flexibility reasons, an RDF file can also be imported, via an XSLT transformation. This transforms the RDF file into a CAF file and then imports it. The import will detect whether it is dealing with a CAF file or an RDF file. Below we will describe the important parts of the XSLT style sheet. Writing out XML headers is skipped here.⁶

First the XML header and the CAF open and close tags are written to the output and then, between them the template for the CAF item is applied. This template applies templates for the domain model and goal model as can be seen below:

```
<xsl:template match="lesson:domainmodel">
  <domainmodel>
    <xsl:apply-templates select="lesson:concept"/>
  </domainmodel>
</xsl:template>

<xsl:template match="lesson:goalmodel">
  <goalmodel>
    <xsl:apply-templates select="lesson:lesson"/>
  </goalmodel>
</xsl:template>
```

The domain model in turn applies templates for concepts and attributes. These templates select the required contents and write it in the way it has to be represented in CAF:

```
<xsl:template match="lesson:concept">
  <concept>
    <name>
      <xsl:value-of select="lesson:name"/>
    </name>
    <xsl:apply-templates select="lesson:attribute"/>
    <xsl:apply-templates select="lesson:concept"/>
  </concept>
</xsl:template>

<xsl:template match="lesson:attribute">
  <attribute>
    <name>
      <xsl:value-of select="lesson:name"/>
    </name>
    <contents>
      <xsl:value-of select="lesson:contents"/>
    </contents>
  </attribute>
</xsl:template>
```

The goal model applies templates for lessons and links writing the required contents in RDF representation:

```
<xsl:template match="lesson:lesson">
  <lesson weight="{lesson:weight}" label="{lesson:label}">
    <xsl:apply-templates select="lesson:lesson"/>
    <xsl:apply-templates select="lesson:link"/>
  </lesson>
</xsl:template>

<xsl:template match="lesson:link">
```

```
<link weight="{lesson:weight}" label="{lesson:label}">
  <xsl:value-of select="lesson:linkPath"/>
</link>
</xsl:template>
```

Importing a ZIP file containing the CAF file and the resources to be added in a resources subdirectory is also possible. If a ZIP is detected, it will be extracted, the resources placed on the server and the CAF file imported.

4.5.2 CAF to RDF transformation

For the sake of flexibility, the MOT lessons should also be exported as RDF. These RDF data can then be used again by Beagle++ (Beagle++ website, 2006). The schema of the RDF metadata is the one as described in Cristea and De Mooij (2003b). Since we already export the CAF file and the extra information file, and they are both in an XML format, we can use an XSLT style sheet and do an XSLT transformation to translate these two files into one RDF file describing the lesson.

The export to RDF is handled much in the same way as is the import of RDF. First, the goal model and the domain models are exported to CAF, and then this CAF is transformed into RDF using an XSLT style sheet (XSLT page at W3C, 2006) in a way symmetrical to the sheet used in the CAR2RDF section. The RDF MOT output then looks as shown in Figure 2. If we export the changed information to RDF so that it can again be used by Beagle++, we might introduce duplicates. If the duplicates are exact duplicates, and overwrite existing information, this is obviously not a problem. If, however, the information contradicts existing information, the effects may be less desirable. A solution to this problem would be to keep track of which objects were already present, and not export those. However, in this case, any changes made to data with MOT (Cristea and De Mooij, 2003a) would not be reflected. Therefore, if an author manually corrects some error, this will not appear in the target RDF. On the other hand, an author may make changes that are only valid in a particular context (*e.g.*, for the given domain maps), and do not apply to the resource in general. If changes would overwrite the original, this could have undesired effects. There are two possible solutions to this problem:

- 1 We search for the existence of the resources to be exported and ask the user whether to overwrite the current resources or not.
- 2 We save the information together with its context as a new information item.

5 Evaluation

A small-scale evaluation of the prototype was preformed in cooperation with the University of Joensuu, Finland, to gain some qualitative feedback. In this chapter we shall describe the experiment and its results. We shall also describe a larger-scale user evaluation that took place at the University Politehnica of Bucharest in January 2007.

The *Joensuu experiment* was conducted in November and December 2006 during a course on Adaptive Personalised Learning for PhD students enrolled in the IMPDET programme. The course had five participants, a small number, appropriate for our first usability studies according to Nielsen (2000). The experiment consisted of a series of steps, taking the participants through the process of using additional resources for advanced learners to a given linear course and visualising the results; then, they were asked to write a short critique about their experience with the system. The steps they were asked to perform, are the steps described in Chapter 4.⁷ From the feedback provided it is clear that they understood the different goals of the experiment:

- connecting the semantic desktop with an adaptive hypermedia authoring system
- converting a linear web-course to an adaptive web-course via a semiautomatic resource retrieval procedure based on metadata
- the resources retrieved for the course extension are articles from the desktop of the course author; the author can decide ultimately to allow the semiautomatic addition of his resources or not (thus respecting privacy).

We also asked for feedback regarding the suitability of our approach for helping authors create adaptive content for advanced learners by adding articles on the topic to the course. This was evaluated from two points of view:

1 The appropriateness of the idea

There was no clear response on the appropriateness of the idea in general, mostly because of the open nature of the assignment. A more closed assignment will give a better idea as to students' position on this matter. From the point of view of the generated adaptive courseware, the remark was made that the added articles could disturb the reading flow of the lessons, as they are not tuned to the particular narrative. A simple solution in our case would be to add the articles only at the end of the given domain concept, with a narrative smoothening element introducing them as follows: "Find more information in the following article(s)". In this case, only one such narrative smoothening element should appear per given domain concept, to avoid unnecessary repetitions.

2 The appropriateness of the formulas for adding resources

Furthermore, we asked respondents to indicate whether they noticed any difference between different settings for the ranking method, set computation and adding method (hierarchy preserving versus non-hierarchy preserving). Although the participants stated that they noticed a difference, they were unable to tell whether it complied with their expectations. This shows that more specifically targeted questions should be designed for the following experiment.

As stated in Chapter 4, the concept-oriented ranking method should theoretically be better than the article-oriented one. We are keen to see whether that is indeed the case with the next practical experiment.

Moreover, the experiment also highlighted a number of issues regarding the user interface and helped us find and patch some system bugs, as follows:

- The MOT pages have general titles that can be confusing. This comes from the fact that the system uses frames, and accesses the database to retrieve the courses only within the child frames, and not the parent frame. This could be rectified by reading from the database in the parent frame as well; however, the traffic between client and server would thereby increase.
- After opening the CAF file in the Enricher, it appears as if the Enricher is not responding, whilst in fact it is retrieving relevant resources in the background. A progress bar or a notification will have to be added.
- When retrieving resources from the RDF store to MOT (via the CAF format), there is an intermediate step allowing a user to select which attributes of the retrieved resources should be included in the final adaptive course (*e.g.*, author of article, conference name, conference date, *etc.*). The pop-up window allowing the selection called these attributes 'standard attributes' (as a refined attribute selection per actual article included is possible afterwards). This allowed confusion with the term 'standard attributes' used in the MOT authoring system. Therefore, the naming in the pop-up window was changed to 'common attributes'.
- There is no way to go back to the selection of these common attributes, when a mistake has been made – the whole process should be repeated from scratch. There is no quick fix for an undo button, but this will have to be treated in the future.
- A bug was detected, where the selection of common attributes was not functioning as expected. This bug was fixed in the new version.

Respondents also agreed with the idea of making a web application out of the Enricher or perhaps a web start application.

The *Bucharest experiment* was already conducted in January of 2007. It took place within the Intelligent Systems course, attended by 61 students. The students were selected from fourth year Engineering students and second year masteral students in Computer Science, both from the English-language stream. We have constructed more precise questions to answer both generic and specific issues regarding the acceptance of automatic generation of adaptivity and personalisation via the methods described here. Moreover, we have already found with the help of the previous small-scale experiment some problems and bugs regarding the user interface and solved some of them, and will try to identify possible others, as well as prioritise problems. The assignment made sure that the students will have to try every option in both the import and export of CAF/RDF and of the Enricher at least once, providing at the same time a scalability test for the system, with larger number of users. The results of this experiment have been processed partially and will be published elsewhere. In short we can add here that most of our hypotheses on the appropriateness of our automatic authoring approach, on the similarity between theory and praxis (*i.e.*, implementation), on the different options of the conversion process have been validated by the test results. We are still processing some of the numerical data, as well as the qualitative feedback, in the form of filled-in free forms. The full results of this experiment will be published elsewhere.

6 Conclusion and future research

The semantic desktop is a rich source of metadata on, and the relationships of, resources. While many of these metadata can be generated automatically, some additional metadata, necessary, *e.g.*, for adaptive courses, still have to be added manually. In this paper, we have described how the semantic desktop, building on semantic web techniques, can be interfaced with a sophisticated adaptive hypermedia authoring environment. The semantic desktop system provides automatically generated metadata describing publications and other materials relevant for an advanced course. The MOT authoring system adds context and pedagogical information. Moreover, we have evaluated this interfacing and conversion process with the help of two experiments: (1) one small-scale experiment conducted with a small number of PhD students, which gave us some relatively quick, qualitative feedback – that had some immediate impact, resulting in a series of bug-fixes, and also some long-term impact, resulting in new considerations; and (2) another larger-scale experiment conducted with an entire regular class of undergraduate students, whose first batch of results were remarkably positive, and from which we expect a greater future impact.

The concept-oriented ranking method aims at adding the most relevant resources to every concept. An advantage is that only the most relevant articles get added, but the number of added articles may vary, and duplicates are, in principle, allowed. On the other hand, the article-oriented method tries to find the concepts that match the articles. In this way more articles may get added, but perhaps not with the highest relevance. It is interesting therefore to further research on these issues, and set, for instance, different thresholds for acceptance of articles – *i.e.*, not just the once with the highest relevance, but also some with relevance above a given threshold, should be added. Moreover, it is interesting to also analyse the impact of duplicates with different nature: duplicate articles with identical relevance versus duplicate articles with different relevance, *etc.*

These issues are closely related to the issues with precision and recall, mentioned before in Section 4. As the Measuring Search Effectiveness web page (2006) illustrates, there is an inverse relationship between precision and recall, and the best performance will be obtained by a good trade-off between precision and recall. Therefore we think it may be a good idea to combine both methods to determine where and which articles should be added. A possible way is applying the concept-oriented method after the article-oriented method. We expect that this will improve the overall relevance of the selected set of articles, as well as how relevant the articles are to the individual concept they get added to. Another option would be to take some normalised mean of both measures.

It is interesting also to consider how the extracted articles can influence possible interlinking between concepts that connect to the same article, *e.g.*, by adding extra relatedness relations.

Furthermore, the maximum number per concept could be an author tuneable parameter, allowing an author to directly choose the density of articles per concept.

Yet another direction we are considering is that of recommender systems. Based upon techniques such as collaborative filtering, or other clustering techniques, these systems typically recommend, *e.g.*, movies to people. However, it may be possible to use some of these techniques to ‘*recommend*’ articles to concepts, for instance.

We are also currently working on flexible ways to describe metadata in this environment. Conventional schemas are not suited for that purpose, as they cannot cope with changing or evolving schemas, which are very important for the semantic desktop (new tasks make new metadata attributes necessary, additional data sources with additional schemas will be added over time, *etc.*) The solution we are currently working on relies on malleable schemas (Dong and Halevy, 2005) to flexibly describe our metadata, and on fuzzy querying and matching, based on these schemas, to flexibly retrieve metadata. See also Cristea *et al.* (2006).

References

- Beagle website (2006) http://beagle-project.org/Main_Page (viewed 6 December 2006).
- Beagle++ website (2006) <http://beagle.kbs.uni-hannover.de/> (viewed 6 December 2006).
- Broekstra, J. and Kampman, A. (2004) 'SeRQL: an RDF query and transformation', <http://www.wis.win.tue.nl/~jbroekst/papers/SeRQL.pdf> (viewed 15 December 2006).
- Brusilovsky, P. (2004) 'Adaptive educational hypermedia: from generation to generation', *Proceedings of 4th Hellenic Conference on Information and Communication Technologies in Education*, Athens, Greece, pp.19–33.
- Chirita, P-A., Costache, S., Nejdl, W. and Paiu R. (2006) 'Beagle++: semantically enhanced searching and ranking on the desktop', *Proceedings of the 3rd European Semantic Web Conference*, Budva, Montenegro.
- Chirita, P-A., Gavrioloaie, R., Ghita, S., Nejdl, W. and Paiu, R. (2005) 'Activity-based metadata for semantic desktop search', *Proceedings of the 2nd European Semantic Web Conference*, Heraklion, Crete.
- Cristea, A. and De Mooij, A. (2003a) 'LAOS: layered WWW AHS authoring model and its corresponding algebraic operators', *Proceedings of WWW'03, Alternate Education Track*, Budapest, Hungary, ACM.
- Cristea, A.I. and De Mooij, A. (2003b) 'Adaptive course authoring: My Online Teacher', *Proceedings of ICT'03*, Papeete, French Polynesia.
- Cristea, A.I., Hendrix, M. and Nejdl, W. (2006) 'Automatic and manual annotation using flexible schemas for adaptation on the semantic desktop', *First European Conference on Technology Enhanced Learning, EC-TEL*, Crete, Greece.
- Cristea, A.I., Smits, D. and De Bra, P. (2005) 'Writing MOT, reading AHA! – converting between an authoring and a delivery system for adaptive educational hypermedia', *A3EH Workshop, AIED'05*, Amsterdam, the Netherlands.
- Dong, X. and Halevy, A. (2005) 'Malleable schemas', *WebDB*, Baltimore, Maryland: ACM, webdb2005.uhasselt.be/papers/P-8.pdf.
- Ghita, S., Nejdl, W. and Paiu, R. (2005) 'Semantically rich recommendations in social networks for sharing, exchanging and ranking semantic context', *International Semantic Web Conference*, Galway, Ireland.
- Ghosh, J. and Lee, D. (2006) 'Performance evaluation of information retrieval systems', <http://www.cs.utexas.edu/~mooney/ir-course/slides/Evaluation.ppt> (viewed 6 December 2006).
- JavaDoc homepage (2006) <http://java.sun.com/j2se/javadoc/> (viewed 6 December 2006).
- Measuring Search Effectiveness Webpage (2006) <http://www.hsl.creighton.edu/hsl/Searching/Recall-Precision.html>, Creighton University, Health Sciences Library (viewed 6 December 2006).
- MOT homepage (2006) <http://prolearn.dcs.warwick.ac.uk/mot.html> (viewed 6 December 2006).
- Nielsen, J. (2000) 'Why you only need to test with 5 users', *Alertbox*, 19 March, <http://www.useit.com/alertbox/20000319.html>.

- PhpDoc homepage (2006) <http://www.phpdoc.org/> (viewed 6 December 2006).
- Schlieder, T. and Naumann, F. (2006) 'Approximate tree embedding for querying cml data', *ACM SIGIR Workshop on CML and Information Retrieval*.
- Semantic Desktop (2006) <http://www.semanticdesktop.org/> (viewed 6 December 2006).
- XSLT page at W3C (2006) <http://www.w3.org/TR/xslt> (viewed 6 December 2006).

Bibliography

- Buneman, P., Davidson, S. and Kosky, A. (1992) 'Theoretical aspects of schema merging', *Proc. 3rd Int. Conf. Extending Database Technology*, Vienna, Austria, pp.152–167.
- Chernov, S. (2005) 'Playground for XML-retrieval', *Talk at the L3S Research Centre*, http://www.l3s.de/~chernov/InfoLunch_21_12_2005_2.ppt (viewed 6 December 2006).
- Cristea, A.I. and Aroyo, L. (2002) 'Adaptive authoring of adaptive educational hypermedia, AH', *Adaptive Hypermedia and Adaptive Web-Based Systems*, Springer, LNCS 2347, pp.122–132.
- Gardarin, G. and Dang-Ngoc, T-T. (2005) 'Mediating the semantic web. 4th journal on extraction and management of knowledge', Clermont Ferrand, Université Blaise Pascal, <http://georges.gardarin.free.fr/Articles/MediatingtheSemanticWeb.pdf> (viewed 6 December 2006).
- He, B. and Chang, C-C. (2003) 'Statistical schema integration across the deep web', *Proc. of SIGMOD*.
- Pottinger, R.A. (2004) 'Processing queries and merging schemas in support of data integration', PhD thesis, University of Washington, US, www.cs.ubc.ca/~rap/publications/thesis.pdf.
- RDF page at W3C (2006) <http://www.w3.org/RDF/> (viewed 6 December 2006).
- Sesame homepage (2006) <http://www.openRDF.org/> (viewed 6 December 2006).
- Shasha, D., Wang, J., Shan, H. and Zhang, K. (2002) 'Atreegrep: approximate searching in unordered trees', *Proc. of International Conference on Scientific and Statistical DB Management*, pp.89–98.
- Simon, B., Dolog, P., Miklós, Z., Olmedilla, D. and Sintek, M. (2004) 'Conceptualising smart spaces for learning', Special Issue on the Educational Semantic Web, *Journal of Interactive Media in Education*, No. 9, ISSN: 1365-893X, <http://www-jime.open.ac.uk/2004/9> (viewed 6 December 2006).
- Velegarakis, Y., Miller, R.J. and Popa, L. (2003) 'Mapping adaptation under evolving schemas', *Vldb'03* (full version in *Vldb Journal*), Berlin, Germany, September, pp.584–595, http://www.almaden.ibm.com/cs/people/lucian/papers/MappingAdaptation_VLDB03.pdf (viewed 6 December 2006).
- Wordnet website (2006) <http://wordnet.princeton.edu/> (viewed 6 December 2006).
- Zhang, K., Wang, J. and Shasha, D. (1998) 'On the editing distance between undirected acyclic graphs and related problems', *Proc. of International Symposium on Combinatorial Pattern Matching*.

Notes

- 1 The newest version of the MOT system, as well as the Enricher, can be found at <http://prolearn.dcs.warwick.ac.uk>. Their detailed implementation documentation can also be found at this address. This consists of more detailed UML models, changes logs and JavaDoc (JavaDoc homepage, 2006) and PhpDoc (Nielsen, 2000).
- 2 PhpDoc homepage (2006) documentation for the parts in Java and PHP, respectively.
- 3 An article can be relevant to the same degree in more places than one. It is up to the authors to decide if such multiple bindings between concepts and articles should be allowed or not. This option has not been further described in the current paper.

- 4 This value has been established experimentally; other values could, in principle, be used.
- 5 This could be acceptable in some cases, depending on the author's purpose. A very good article, for instance, might be relevant for, *e.g.*, ten concepts. We have already implemented a version allowing duplicates, but more variety might be needed, with options to establish exactly which type of duplicates are allowed, as well as different thresholds for acceptance of articles found.
- 6 The exact style sheet can be found in the CAF subfolder of the MOT implementation, see <http://prolearn.dcs.warwick.ac.uk>.
- 7 For more information, see also <http://prolearn.dcs.warwick.ac.uk/sesame2mot/>. Participants were given a version of the Enricher, which included settings for accessing a Sesame store at the University of Warwick. We were able to provide temporary access for trials.