Authoring Adaptive Learning Material on the Semantic Desktop

Maurice Hendrix[1,2], Alexandra Cristea[1], and Wolfgang Nejdl[2]

[1] Faculty of Mathematics and Computer Science, Eindhoven University of Technology
(TU/e), Postbus 513, 5600 MB Eindhoven, The Netherlands
m.hendrix@student.tue.nl, a.i.cristea@tue.nl
[2] L3S Research Center, University of Hannover,
30539 Hannover, Germany
nejdl@l3s.de

Abstract. Manual annotation has proven to be the bottleneck of authoring of adaptive hypermedia. Therefore, all means for supporting this authoring process by reusing automatically generated metadata are helpful. In this paper we discuss the integration of the generic AH authoring environment MOT into a semantic desktop environment. In this setup, the semantic desktop environment provides a rich source of automatically generated meta-data, while MOT provides a convenient way to enhance this meta-data manually, as needed for an adaptive course environment.

1 Introduction

Current desktop file structures are surprisingly poor from a semantic point of view. They merely provide fixed folder hierarchies and use simple names to denote entities. The *semantic desktop* [6, 21] promises much richer structures: resources can be categorized by *rich ontologies,* and *semantic links* express various kinds of *semantic relationships* between these *resources*. For a document, for example, the semantic desktop stores not only a filename, but also information about where this paper was published, when and by whom, which of my colleagues sent it to me, and how often and in what context I accessed it. All these metadata are generated automatically, by the appropriate applications, and stored in an application independent way as RDF metadata, in the user's personal data store. This rich set of metadata clearly makes it easier for the user to retrieve appropriate material for different contexts: for example, when he wants to select fitting materials for a course lecture. Of course, in this context, a course author still has to create basic lesson material, as well as add additional metadata information about course structures and other relevant attributes, like pedagogically relevant attributes. In this paper we will describe course authoring on the semantic desktop. Specifically, we describe the interaction and exchange of data between the Beagle++ environment [2, 5], which is an advanced search and indexing engine for the desktop, generating and utilizing metadata information, and the adaptive hypermedia authoring environment MOT [8,15], a sophisticated system for authoring personalized e-courses.

The remainder of this paper is organized as follows. Section 2 introduces a motivating scenario for our work. Section 3 presents the concept of enriching metadata, as well as the input and output schemas applied. Section 4 describes the transformation workflow from both a conceptual and an implementation point of view. Finally, in section 5 we discuss the results and draw conclusions.

2 Scenario

The following motivating scenario for adaptive authoring builds upon both automatically and manually generated metadata:

Dr. Van Bos prepares a new on-line course on Adaptive Hypermedia for undergraduate 4th year TU/e students. The university work distribution allocates a limited amount of time for this, equivalent to the creation of a static, linear course. However,

- due to the fact that Dr. Van Bos considers it useful to be able to extend the course in the future with more alternative paths guided by *adaptivity*, but also,
- because he wants to benefit from *automatic help* during the authoring process,

he uses a *concept-based adaptive educational hypermedia authoring environment* with *adaptive authoring support*, MOT [8,15]. This decision costs him slightly more time than the static course creation, as he has to *manually* divide his course into conceptual entities with explicit, independent semantics and semantic labeling.

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. Van Bos as the version for *beginner* students, which are only moderately interested in the subject, and don't aspire at higher grades or deep knowledge on the subject. For *advanced* students, wishing to pass the course with high honors, 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. Van Bos'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. Van Bos is a specialist in the subject taught, his interest is wider than that given by the limitations of the course; he therefore both publishes and reads papers of interest on the subject, which are likely to be stored on his 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 several years, Dr. Van Bos may not know exactly where on his computer each individual article relevant to the current course is;
- however, Dr. Van Bos has been using Beagle++ Semantic Desktop System [2, 5] 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. Van Bos's own papers on Adaptive Hypermedia, as well as some copies that he has of his colleagues on the topic, such as for instance the paper of Brusilovsky, called "Adaptive Educational Hypermedia: From generation to generation" [3], or the paper "Adaptive Authoring of Adaptive Hypermedia" [9]. He may have saved these papers by himself, or might have received them by e-mail, from a colleague working in the same field, or may have used his browser's bookmarks to mark their position on the Web.

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

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

This means that the first paper can be added in 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 Van Bos find the right resource and add it in the right place? The search can take place via the keywords labeling both the course pieces created by Van Bos, on the one hand, and the matching keywords labeling the papers and resources on his desktop (as described in section 4). The following sections will describe in more detail how Dr. Van Bos can enrich his course semi-automatically, without much extra work, as well as keep at all times the overall control and overview.

3 Enriching Metadata

As we have seen in our scenario, both *automatically* generated metadata as well as *manually* added metadata are important. The automatically generated metadata allow description and retrieval of the appropriate articles. The manual annotation step allows addition of additional content, as well as of attributes like pedagogical weights and labels, which are necessary to build the final adaptive course product. The following two sub-sections describe these two kinds of metadata in more detail.

3.1 Input Metadata Schema

Beagle [1] is a desktop search system implemented for Linux that indexes all documents on the user's desktop. *Beagle*++ [2, 5] is an extension of Beagle that generates and stores additional metadata describing these documents, other resources, as well as their relationships. Such additional metadata automatically annotate material the user has read, used, written or commented upon. Three such obvious sources of desktop behavior information are: files on the desktop, Internet browsing and files stored from the Internet, and mail exchanges and files stored from mail attachments [5, 13]. Figure

1 shows an instance of this ontology depicting files annotated with their publication metadata, file system attributes, web history, as well as the mail context (e.g., the fact that files are attached to specific e-mails).

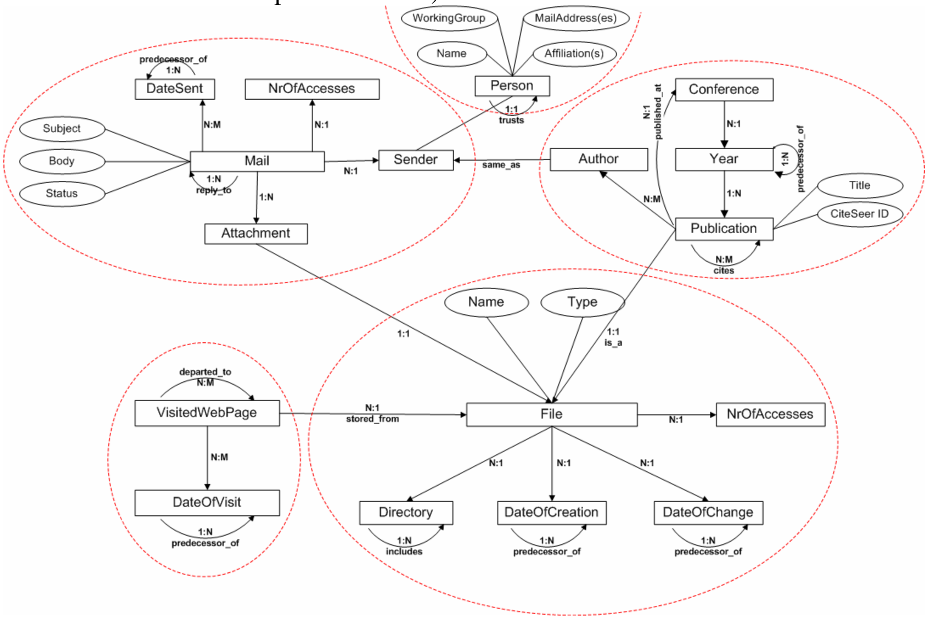


Fig. 1. RDF schema of metadata from Beagle++ [2]

In Figure 1, the ellipse in the upper left part describes emails (with subject, body, sender and status attributes) and their attachments. The ellipse in the upper right part describes publications written by several authors at different conferences (with title, publication year, etc.). The ellipse in the lower left part shows web cache history representation (web pages visited and dates of visits). Finally, the ellipse in the lower right part of Figure 1 describes files that are saved on a person's desktop, with their name and specific directory. Files on the desktop may have been saved via any of the other three processes (from emails, 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 Output Metadata Schema

These files and metadata are however not enough to generate a complete course. Specifically, we have to add information about the hierarchical structure and order of the material in the context of a lesson, as well as additional pedagogical annotations describing which students the material is best suited for (e.g., *beginner* versus *advanced*). Figure 2 shows this target schema, as defined in MOT (My Online Teacher

[8]), an adaptive hypermedia authoring system. The schema describes two models used in the adaptive hypermedia authoring paradigm: the *domain map* (left side of Figure 2) and the *lesson* (right side of the figure).

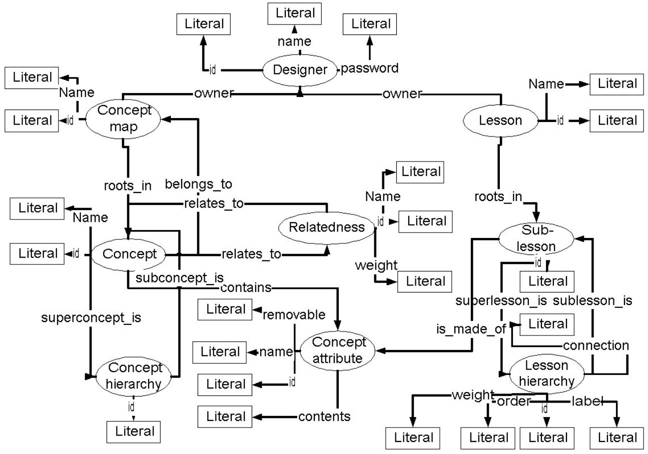


Fig. 2. RDF Schema of MOT [9].

A *domain map* is composed of a hierarchy of concepts. Each domain concept has concept attributes, containing or linking to e-learning content alternatives describing the same concept. In addition to the hierarchical model, domain concepts can also connect to related concepts.

A *lesson* is composed of a hierarchy of sub-lessons. The lesson represents a filtered, pedagogically labeled, weighted and ordered version of the concept attributes (see the relation between sub-lessons and concept attributes).

4 Transformation workflow

Beagle++ stores all metadata in the Sesame RDF database [18]. All Beagle++ components, which generate metadata (for example, the email, publication, web cache and file metadata generators) add the metadata to this database. All Beagle++ components which 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).

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* [8, 15], as described in this paper. For our scenario, we have focused on the semi-automatic addition of articles stored on the user's desktop to a MOT lesson [8]. In MOT, this addition is done to an existing lesson. Based on his 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 CAF (Common Adaptation Format [7], a system-independent XML exchange format) in order to simplify the transformation process from RDF to the MOT MySQL storage format.

4.1 RDF2CAF

In this step, we transform available articles into a CAF sub-lesson. This is a semi-automatic 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, ensure that confidential files are not mistakenly publicized.

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 of course 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 query looks as follows:

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

where x is an entity or attribute; p is a relationship and y is an entity or instance.

Next, to ensure that we add every resource only once, the place with highest 'relevance' is sought. Relevance is computed as follows.

$$rank(a,c) = \frac{card(k(c) \bigcap k(a))}{card(k(a))}$$

where:

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*;

If a resource is ranked equally for two domain concepts in the hierarchy, we add it to the topmost concept. The comparison takes place top-down. This means that the same resource may appear only in one of the siblings, if their rank is similar[1]. The number of articles to be added to any concept is limited to 3, as adding too many articles to one concept probably confuses the learner rather than help her.

[1] Further ranking functions and ranking algorithms are considered, as a choice for the author.

4.2 CAF2MOT

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 lookup table. In this table, every domain concept attribute created is stored as follows:

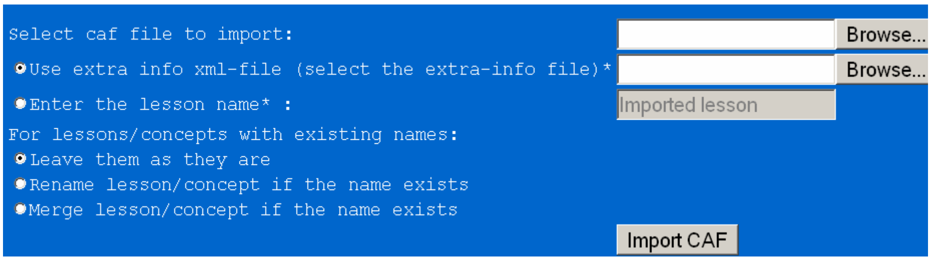
```
[name_of_domain_map\name_of_concept_1\...\
  name_of_concept_N\attribute_name] => attribute_id
```

This allows a faster lookup in the next step then 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:

- 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.
- 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.
- 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). 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.

As the CAF format, which is a portable format used in several adaptive hypermedia systems, does not include the lesson name, this can be exported as a separate XML file. Figure 3 shows a screenshot of the CAF import interface.



Select caf file to import:

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

○ Enter the lesson name* : Imported lesson

For lessons/concepts with existing names:

● Leave them as they are

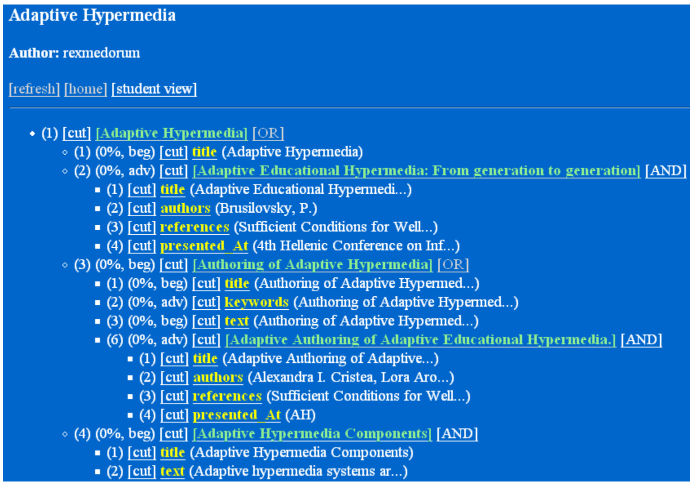
○ Rename lesson/concept if the name exists

○ Merge lesson/concept if the name exists

Fig. 3. Importing from CAF to MOT (My On-line Teacher)

4.3 Working in MOT

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



Adaptive Hypermedia

Author: rexmedorum

[refresh] [home] [student view]

- (1) [cut] [Adaptive Hypermedia] [OR]
    - (1) (0%, beg) [cut] title (Adaptive Hypermedia)
    - (2) (0%, adv) [cut] [Adaptive Educational Hypermedia: From generation to generation] [AND]
        - (1) [cut] title (Adaptive Educational Hypermedi...)
        - (2) [cut] authors (Brusilovsky, P.)
        - (3) [cut] references (Sufficient Conditions for Well...)
        - (4) [cut] presented_At (4th Hellenic Conference on Inf...)
    - (3) (0%, beg) [cut] [Authoring of Adaptive Hypermedia] [OR]
        - (1) (0%, beg) [cut] title (Authoring of Adaptive Hypermed...)
        - (2) (0%, adv) [cut] keywords (Authoring of Adaptive Hypermed...)
        - (3) (0%, beg) [cut] text (Authoring of Adaptive Hypermed...)
        - (6) (0%, adv) [cut] [Adaptive Authoring of Adaptive Educational Hypermedia.] [AND]
            - (1) [cut] title (Adaptive Authoring of Adaptive...)
            - (2) [cut] authors (Alexandra I. Cristea, Lora Aro...)
            - (3) [cut] references (Sufficient Conditions for Well...)
            - (4) [cut] presented_At (AH)
    - (4) (0%, beg) [cut] [Adaptive Hypermedia Components] [AND]
        - (1) [cut] title (Adaptive Hypermedia Components)
        - (2) [cut] text (Adaptive hypermedia systems ar...)

Fig. 4. Adding manual metadata in a MOT lesson

Figure 4 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, labeling the material 'adv' for advanced learners.

4.3 MOT2RDF

MOT2CAF

Importing MOT into CAF is also implemented as a PHP script. First, the lesson is constructed, based on the 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 sub lessons.

CAF2RDF

For the sake of flexibility, the MOT lessons should also be exported as RDF; we are currently implementing this step. These RDF data can then be used again by Beagle++ [2]. The schema of the RDF metadata is the one as described in [9]. Since we already export the CAF file and the extra information file, and they are both in an XML format, we can use a XSLT style sheet and do an XSLT transformation to translate these two files into one RDF file describing the lesson. 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 [8] 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 which are only valid in a particular context (e.g., for the given domain 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:

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

7 Conclusion

The semantic desktop is a rich source of metadata about resources as well as relationships between these resources. While many of these metadata can be generated automatically, some additional metadata necessary 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.

Future issues we are currently investigating focus 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 [11] to flexibly describe our metadata, and on fuzzy querying and matching, based on these schemas, to flexibly retrieve metadata.

References

1. Beagle, http://beaglewiki.org/Main_Page
2. Beagle++, http://www.kbs.uni-hannover.de/beagle++/wiki/pmwiki.php
3. Brusilovsky, P. (2004) Adaptive Educational Hypermedia: From generation to generation. Proceedings of 4th Hellenic Conference on Information and Communication Technologies in Education, Athens, Greece, September 29 - October 3, 2004, pp.19-33.
4. Buneman P., Davidson, S., and Kosky, A. Theoretical Aspects of Schema Merging, Proc. 3rd Int. Conf. Extending Database Technology, Vienna, Austria (March 1992) 152-167.

5. Chirita, P.-A., Costache, S., Nejdl, W., and Paiu, R. Beagle++: Semantically Enhanced Searching and Ranking on the Desktop. Proceedings of the 3rd European Semantic Web Conference, Budva, Montenegro (to appear, June 2006).
6. Chirita, P.-A., Gavriloaie, R., Ghita, S., Nejdl, W., and Paiu, R. Activity-Based Metadata for Semantic Desktop Search. Proceedings of the 2nd European Semantic Web Conference, Heraklion, Crete, May (2005).
7. Cristea, A.I., Smits, D., and De Bra, P. Writing MOT, Reading AHA! - converting between an authoring and a delivery system for adaptive educational hypermedia -, A3EH Workshop, AIED'05, Amsterdam, The Netherlands (July, 2005)
8. Cristea, A. I. & De Mooij, A. Adaptive Course Authoring: My Online Teacher. Proceedings of ICT'03, Papeete, French Polynesia (2003).
9. Cristea, A., De Mooij, A. LAOS: Layered WWW AHS Authoring Model and its corresponding Algebraic Operators. In Proceedings of WWW'03, Alternate Education track. (Budapest, Hungary 20-24 May 2003). ACM.
10. Cristea, A. I. and Aroyo, L. Adaptive Authoring of Adaptive Educational Hypermedia. AH 2002, Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 2347, Springer, (2002) 122-132.
11. Dong, X., and Halevy, A., Malleable Schemas, WebDB, Baltimore Maryland (June 16-17, 2005), ACM, webdb2005.uhasselt.be/papers/P-8.pdf.
12. Gardarin, G., and Dang-Ngoc, T-T. Mediating the Semantic Web. 4th journal on extraction and management of knowledge, Clermont Ferrand, Université Blaise Pascal (20-23 January 2004). http://georges.gardarin.free.fr/Articles/MediatingtheSemanticWeb.pdf
13. Ghita, S., Nejdl, W., and Paiu, R., Semantically Rich Recommendations in Social Networks for Sharing, Exchanging and Ranking Semantic Context, International Semantic Web Conference, Galway, Ireland (6 November 2005) 6 – 10.
14. He, B., and Chang, C.-C., Statistical schema integration across the deep web. In Proc. of SIGMOD (2003).
15. MOT homepage, http://wwwis.win.tue.nl/~acristea/mot.html
16. Pottinger, R. A., Processing Queries and Merging Schemas in Support of Data Integration, PhD thesis, University of Washington, US (2004), www.cs.ubc.ca/~rap/publications/thesis.pdf.
17. RDF, http://www.w3.org/RDF/
18. Schlieder, T., Naumann, F., Approximate tree embedding for quering cml data, ACM SIGIR Workshop on CML and Information Retrieval.
19. Sesame, http://www.openrdf.org/
20. Shasha, D., Wang, J., Shan, H., Zhang, K., Atreegrep: Approximate searching in unordered trees, in Proc. of International Conference on Scientific and Statistical DB Management (2002) 89-98.
21. Semantic Desktop, http://www.semanticdesktop.org/
22. Simon, B., Dolog, P., Miklós, Z., Olmedilla, D. and Sintek, M. Conceptualising Smart Spaces for Learning. *Journal of Interactive Media in Education*, 2004 (9). Special Issue on the Educational Semantic Web. ISSN: 1365-893X (2004). http://www-jime.open.ac.uk/2004/9
23. Velegrakis, Y., Miller, R.J., and Popa, L. Mapping Adaptation under Evolving Schemas, with Yannis Velegrakis and Renee J.. VLDB'03, Berlin, Germany, September (2003) 584-595. (full version in VLDB Journal) http://www.almaden.ibm.com/cs/people/lucian/MappingAdaptation_VLDB03.pdf
24. Wordnet. http://www.cogsci.princeton.edu/wn/
25. Zhang, K., Wang, J., Shasha, D., On the editing distance between undirected acyclic graphs and related problems, in Proc. of International Symposium on Combinatorial pattern matching (1998).