Towards more efficient generic semantic authoring for adaptive hypermedia

Mudit Saksena
Faculty of Mathematics and Computer Science
Eindhoven University of Technology,
Postbus 513, 5600 MB Eindhoven, the Netherlands
m.saksena@student.tue.nl

Alexandra Cristea
Faculty of Mathematics and Computer Science
Eindhoven University of Technology,
Postbus 513, 5600 MB Eindhoven, the Netherlands
a.i.cristea@tue.nl

ABSTRACT

Personalization of presentation has always been important in any consumer industry. With the expansion of the WWW into our everyday life, education has become increasingly a consumer commodity. Today's learners expect high quality, relevant educational materials, delivered to them in a timely and appropriate manner. Adaptive educational hypermedia (AEH) systems aim to personalise the delivery of educational materials to the needs of the user – both to their stated requirements as well as to their less obvious desires. This paper describes steps taken towards creating more efficient generic semantic authoring for adaptive hypermedia, starting from an existing framework, LAOS, an existing system, MOT, and evaluation results thereof. The improvements were implemented and a preliminary pilot usability study was performed.

Categories and Subject Descriptors

H.1 [**Information Systems**] Models and Principles; I.2.4 [**Artificial Intelligence**]: Knowledge Representation Formalisms and Methods; H.5.4 [**Information Interfaces and Presentation**]: Hypertext/Hypermedia - *architectures, navigation, user issues*; H.3.3 [**Data**]: Data Structures - *distributed data structures, graphs and networks*; K.3.1 [**Computers and Education**]: Computer Uses in Education - *distance learning*.

General Terms

Measurement, Documentation, Performance, Design, Reliability, Experimentation, Security, Human Factors, Standardization, Theory.

Keywords

Adaptive Hypermedia, Authoring of Adaptive Hypermedia, LAOS, LAG, MOT.

1. INTRODUCTION

Adaptive hypermedia system (AHS) are becoming nowadays more popular, due to their correlation with the recent strive of the W3C and the IEEE LTTF community towards (ontology-based) customization and the Semantic Web (SW).

Adaptive Hypermedia (AH) [6] is the solution to the problem of personalization on the Web, especially for educational systems. Specifically, *Adaptive Educational Hypermedia* (AEH) [7] caters to the needs of each individual student, adapts to their *goals*; *knowledge* level; *background*; *interests*; *preferences*; *stereotypes*; *cognitive preferences* and *learning styles*.

Personalization of presentation has always been important in any consumer industry. With the expansion of the WWW into our everyday life, education has become increasingly a consumer commodity. Today's learners expect high quality, relevant educational materials, delivered to them in a timely and appropriate manner. Adaptive educational hypermedia (AEH) systems aim to personalise the delivery of educational materials to the needs of the user – both to their stated requirements as well as to their less obvious desires. This has led to the development of many on-line educational delivery systems (e.g., Interbook [8], AHA! [32], TANGOW [10], WHURLE [39]). Many of these systems adapt their educational content to different *dimensions* of each learner, such as: current knowledge levels, computed user goals, immediate tasks, educational context (e.g., are they in school, university, or learning from home?), and more recently learning styles in adaptive hypermedia. Each of these AEH systems uses its own content model, coding methodology, and style. This leads to an unnecessary and undesired level of complexity for the content or lesson author. An author faces a multitude of problems when they decide to use one of these systems, trying to answer the question *"Which one of the many AEH systems available should they use?"*, such as:

- These systems are invariably very complex to use from an authoring perspective
- They often require a great degree of technical expertise to setup and use the AEH system itself.
- As each system is unique (there are as yet no AEH standards) materials created for a single system are non-transferable or re-usable.
- The time and effort an author spends on learning a system is also non-transferable to other AEHs. If an author wishes to use an improved but different system, they will have to learn how to author for it once again.
- There is often very little support for authors in AEH systems, in the form of tutorials or help files.

A methodology to combat these problems is that of "*write once, use many*" [27]. That is, an author should only ever have to write or create a piece of material once. This material should be easily re-usable in many other AEH systems. This means that an author only has to learn how to use one AEH authoring system; if they subsequently wish to deliver their lessons in another AEH then a simple and transparent conversion process will enable this. This allows for the cost-effective re-use of both materials and skills. This idea is very close to the goals of the Semantic Web. The only difference is, the core of the SW reuse has centered around static material reuse. In adaptive hypermedia, both static, as well as dynamic 'material' (e.g., behavior patterns, learning/teaching strategies, etc.) have to be exchanged.

2. AIMS AND PROBLEMS SETTING

With the above issues in mind, the current research is aimed at the creation of a generic, collaborative authoring environment for adaptive hypermedia, that allows the instantiation of the "*write once, use many*" paradigm and extensions to the Semantic Web. This environment should incorporate new findings about authoring support in adaptive environments, automatizing the process of authoring where possible, as well as adding intuitive visual tools for guiding the authors.

These general aims were concretized in the form of an extension and improvement to an existing authoring system for adaptive hypermedia, MOT [16], which was used in extensive testing [20,22,23]. An initial set of requirements for the new system is that it should be:

- based on the extant MOT authoring system;
- based on the many evaluations [20,22,23] performed with students and specialists of the MOT environment (in the roles of *Authors* as well as *Learners*)
- based on the new theoretical developments (based on theoretical papers about authoring models [8,11,17,18,24,25,26,27,29,34])
- the output to a special intermediate format, called CAF [21] should be preserved;
- output to a standard semantic web [50] format, RDF [42], should be added;
- the internal representation in MySQL [41] should be preserved;
- compatibility with other systems, such as WHURLE [38] and the SCORM compliant Blackboard [4], should be preserved;

The exact working method and steps for fulfilling these goals was as follows:

1. a large, unfiltered list of complaints about MOT system functionality (adding also comments about complaints about conversions from MOT to other systems, such as MOT2AHA [21], MOT2WHURLE [46], only when directly relevant) was obtained, containing qualitative and quantitative data coming from the tests performed with MOT. Many of the critics were focused on the same problem.
2. next, these comments were filtered and grouped to describe in a concise manner the type of problems that appeared. Also an estimation was made on the changes which could be performed on the MOT system, in a table with two headings: *problem*, *possible solution*. Below, an example entry is shown:

Table 1. Filtered list of problems and their possible solutions based on MOT evaluations

problem	*Possible solution(s)*
order of concepts in lessons is difficult to use	○ Better visualization of list order (e.g., only concepts, concepts and labels & weights, etc.)
Search functionality	○ make it possible to search for concepts, concept attributes, lessons (by name or partial name)

3. however, not all these problems were equally urgent and important to implement; moreover, the difficulty of the implementation varied. Therefore, following, an order of difficulty and then priorities for all the above problems-solutions pairs were computed; e.g.:

Table 2. Filtered list with difficulty and priority of solution based on MOT evaluation

difficulty	*priority*	*problem*	*Possible solution(s)*
65%	90%	order of concepts in lessons is difficult to use	○ Better visualization of list order (e.g., only concepts, concepts and labels & weights, etc.)
40%	50%	search functionali ty	○ make it possible to search for concepts, concept attributes, lessons (by name or partial name)

The 'difficulty' represents here the implementation difficulty, whilst 'priority' represents how many evaluators have spotted that problem, in other words, how important it is to remove it.

4. next, in a similar way, an unfiltered list of features that are not present in MOT, but exist in the theory (in the LAOS authoring framework [17], and in LAG [18]) was collected. Below, an example is given:

Table 3. Filtered list of problems and their possible solutions based on theoretical comparisons

feature	*Possible solution(s)*
Unlimited number of weights and labels per lesson	○ implementing weights & labels as lists or vectors
Naming is different: Conceptmaps are called DM and lessons are called GM	○ change labeling of DM, GM

5. following, an estimate of the implementation difficulty and importance was done:

Table 2. Filtered list with difficulty and priority of solution based on theoretical comparisons

difficulty	*priority*	*feature*	*Possible solution(s)*
20-25%	30%	Unlimited number of	○ implementing weights &

		weights and labels per lesson	labels as lists or vectors
5%	100%	Naming is different: Conceptmaps are called DM and lessons are called GM	○ change labeling of DM, GM

6. finally, an overall, merged list of priorities generated from the two lists based on evaluations and theory was obtained; this was transformed into a user requirement document (URD);
7. The last step is the implementation of the features according to their priority. This step has been divided into a few sub-steps, grouping implementation according to a measure reflecting the cumulated effect of difficulty and priority, called index, as follows:

   index = difficulty*weight + priority*(1-weight)

   where: weight = 0.5 (experimental value).

This paper reflects the implementation of steps 1-6 and items in step 7 with an index > 70. These implementations were tested within a pilot study. These results will be further used for the implementation of the next items, in an iterative process, similar to the ones used in industrial settings, when developers get feedback from users as early as possible in the development process. In the following, we shortly describe the system we are using and extending.

3. MOT

MOT (My Online Teacher) [16,20], an AHS adaptive authoring system built at the Eindhoven University of Technology for on-line adaptive material creation, instantiates the LAOS theoretical framework in the context of adaptive (educational) hypermedia. MOT is an AEH authoring system implementing an interpretation of the LAOS framework [17,34]. At the time of the writing, MOT implements the following from LAOS:

Domain model (DM), as a *conceptual domain map* for courses; these domain maps are based on domain models, representing the domain ontology.

Goal and constraints model (GM), as a *Lesson map*; these lesson maps are based on goal and constraints models, representing the goal ontology.

User model (UM), featuring, among others, *stereotypes* and *overlay user maps*; the user maps based on the user model represent a user ontology.

Presentation model (PM) is implemented in the form of a hybrid model, similar to the user model (extended currently with a quality of service layer). The presentation model represents the presentation ontology.

Adaptation model (AM) (MOT-adapt), in the form of an (*instructional) adaptive strategy* [18] creation tool; the interpretation of the above ontologies is done in the adaptation model. Therefore, the decision of how 'heavy' the ontologies are (from light-weight to fully fledged) is done here. This means that the same domain taxonomy can be converted into a lightweight or a fully fledged ontology by an appropriate respective adaptation strategy. This is a more flexible notion than usual in Semantic Web applications.

The part of the system considered and extended here represents a second, updated version of three of the LAOS models: domain model, lesson model (GM) and adaptation model [22,23,26].

MOT has been tested for authoring purposes in a classroom setting, using a first version which implemented the DM and GM [20]; and then as a version with three models (DM, GM, AM) [22,23,26]. Further evaluations were done whilst interfacing MOT with AHA! (Version 3.0) system [31].

MOT conforms to the LAOS principles, using a concept-oriented approach. This means that the information about a course, for instance, is stored in MOT in the form of linked domain concepts, expressed by their attributes. MOT features also some recommended, standard attributes, selected by the author from a set determined by the system manager; examples thereof are: *title*, *keywords*, *pattern*, *introduction*, *text*, *explanation*, *conclusion* and *exercise*.

4. PROBLEMS IN MOT

Depending upon the index as defined above, the complaints and improvement points were further on categorized into three sections, which are presented here. For readability, we are not listing all of the issues found, but just highlight the main categories.

The **generic problems** were:

Complexity: Students found that they were not that confident in using the system; they found the system somewhat complex; some of the reasons for this are explained below.

Installation: a highly quoted problem was that of installation: installation steps were too many and too time-consuming; Perl packages are very difficult to install; although this issue was only valid for the users that wanted to use MOT on their own server, as a few online trial versions are available, it was considered an important one[1];

Security: The system lacked security; it was possible to impersonate other authors by doing a little URL diving. The teacher could become another teacher just by changing a few pieces of information. Moreover, passwords were not encrypted; Such issues were especially stressed by computer science students and researchers.

Look and feel: The 'look & feel' of the system was perceived as outdated; e.g., the blue background color of MOT was criticized: many of the students felt that it wasn't appropriate; moreover, the way information was grouped was also considered inappropriate: too many clicks were necessary for some of the operations. These types of remarks came mainly from user-system interaction users and specialists.

Logout & Login: A logout option from MOT was not available: there was a login option but the system lacked logout.

Search: The system lacked *search* functionality; it was possible to list all the keywords that connect the various concepts, as well as their source; it was possible to find related concepts via a relatedness function search; but beside this, no real filtering or general search was available in the previous version;

[1] Later on, this importance will be quantified.

Removing option: A very important omission in the previous version was that no confirmation option for the *remove* function was present; this was, of course, very risky: if by accident the remove button was clicked, all the work was deleted in a single click, without any warning or redo option;

Frames reload: MOT frames reload was not functioning as expected: when a concept attribute was changed, the reload attached would display the beginning of the current list (of concepts, attributes, etc.). This meant that a user had to scroll back to the current concept s/he was editing after each change.

Some **Domain Model** problems were as follows:

Naming: The naming was confusing and somewhat inconsistent with the theory;

Scalability: domain maps and lessons were hard to find if a large number of them are available; this happened when testing the system exhaustively online with multiple classes and generations of students and researchers; another issue also related to scalability was the fact that the relatedness function (a function that finds semi-automatically related concepts) ceased to function for a large number of domain maps;

Roles: There are more roles necessary in the system, beside the author and the learner. A manager or administrator role is necessary (accessible via username and password), to be able to perform some generic functions, for example, to add a user guide that is available to all teachers, etc. These kind of issues were not focused upon in the beginning on MOT, where the focus was on representation of all aspects of adaptivity to the student. However, from actual use, it turns out that these features of lesser research value, present already in commercial learning management systems, need to be present.

Some **Goal Model** problems were as follows:

Display option: Similar to the Domain Model, the naming was confusing and somewhat inconsistent with the theory. Moreover, other display options were not helpful: for instance, changing order or adding semantic weights and labels (pedagogic meta-data) to the lesson was somewhat cumbersome, due to the identification of groups of sub-lessons: each of them was identified by the word 'Group' only, without making any reference to the actual name of the group.

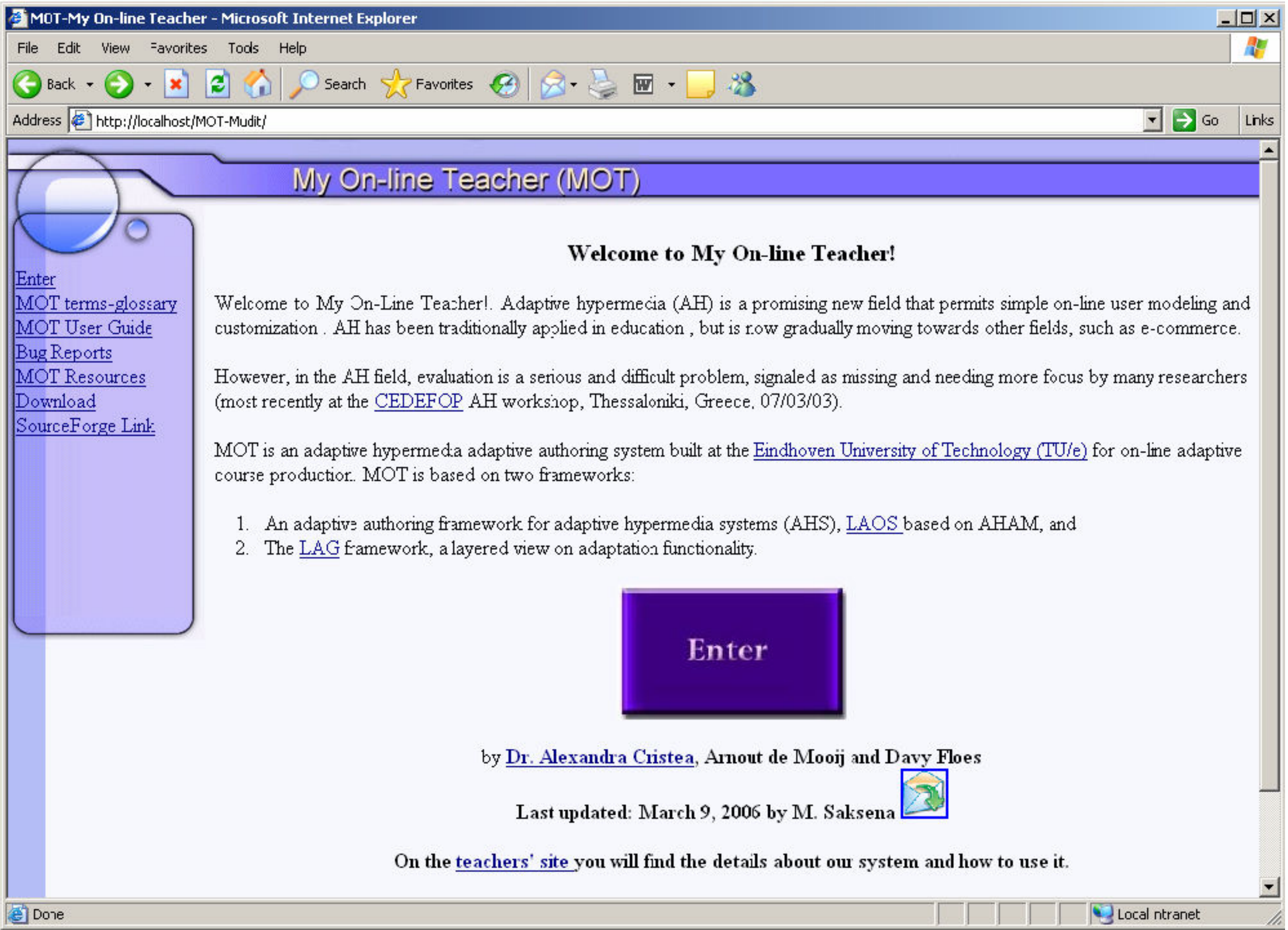

Figure 1. New MOT 'look and feel' based on frames.

Editing: One issue that users raised was that during conversion from the domain map to the lesson, the ordering of concepts was changed[2]. Another issues was that, due to special character display issues and to truncating of information display in the short version of attribute content display, the resulting concept tree looked 'messy', or even, more importantly, misses some of the information; this could happen after converting a domain concept map into a lesson, or when editing the lesson (adding information, deleting, etc.), if any specific characters were present. This bug had been identified early, and the solution for the tested version of MOT was to warn authors about it (in some early help documents) and instruct them about how to construct their concepts in order to avoid it. However, tests showed that authors would easily forget about the instructions and therefore have problems with the system. Recovering from such problems was much more difficult than avoiding them in the first place, so a different solution needed to be found.

5. SOLUTIONS OF PROBLEMS AND IMPLEMENTATIONS THEREOF IN MOT

According to their indexes, some of these problems, belonging to the first implementation sub-step, were solved in the new implementation. The improvement claims are based on some small-sized pilot testing that is described in section 6. The possible generic solutions[3] and their actual implementations, conform with the URD, are presented below:

Complexity, Installation: A proper step by step documentation of MOT Installation and MOT User Guide was created, as suggested by testers.

1. *Installation:* Providing every installation tool in a zipped folder was thought as an initial possible solution for this problem. The actual, more simple solution involves using the XAMPP application [49] for Windows and Linux (among others). With it, the installation becomes a simple, 3-4 steps process. Now, with proper guidance from the installation document, a novice can download and install MOT within minutes.
2. *Security:* A possible solution to this problem illustrated in section 4 is to replace 'GET' server information requests with 'POST', and make all the visible variables hidden; another possible (quite popular) solution was to use cookies for storing session information; finally, a third possible solution was to use sessions; in our implementation, we used the first solution, as being the quickest to implement; this solution proved during the pilot testing to be sufficient.
3. *Look and feel*: A possible solution was to have an option for the author at the time of registration about the color schemes available to choose from. The new first version of the implementation has opted instead for a fixed frame-based look, according to some prescribed, professional-looking, online page templates [46]; and after doing some extra cosmetic changes to these acquired frames, the pages look more appealing and user friendly. A glimpse of the new 'look and feel' can be seen in Figure 1.
4. *Logout & Login:* A logout option was added to the main page of the new version of the MOT system. The registration page has been updated, so that a minimum of work is required for registration.
5. *Search:* make it possible to search for names of concept maps and lessons can be achieved with the help of search functionality of the web browser at the moment; a better, yet still implementationally cheap solution is to create a dedicated java script.
6. *Removing option:* A Java script has been created which pops up a dialogue box warning a user before removal of any item. A sample dialogue box can be seen in Figure 2.

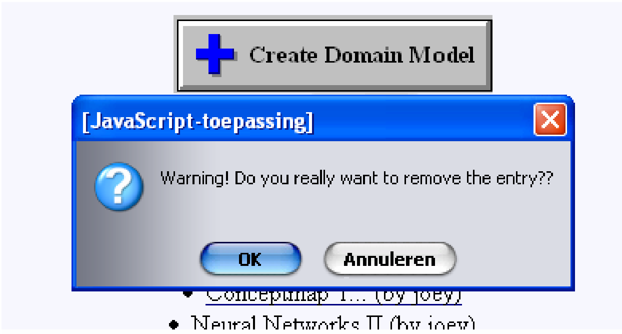

Figure 2. Removing dialogue preventing accidental removal.

7. *Frames reload:* The initial possible solution was to reload frames in such a way that they return to the current place being edited. The quicker way of dealing with the long lists of information and scrolling, in the current implementation, was to divide it into two semantically different tabs: one for the domain concept map and one for the goal map (lesson), respectively.

The **possible Domain model solutions** and their **actual implementations** are:

1. *Naming:* The naming of concept map to domain models and lessons to goal models were changed first, to conform to the theoretical model.
2. *Scalability* The possible solution for scalability in terms of quick, intuitive access to information was to divide the information first in semantically relevant categories, thus minimizing the corpus of information on display at one time. This was done, partially, by dividing domain maps and lessons on different information tabs. Another related improvement was to implement buttons with clear visual semantics for the creation of domain maps and lessons. After some cosmetic changes, it is very easy to locate them (see Figure 3 for an example). For issues such as relatedness relation, a possible solution is to optimize the database queries, in order to eliminate such elements as query processing order on the size of the partial results, etc. These solutions are currently being implemented.

[2] This was no bug, as in domain maps order of concepts was irrelevant. However, authors used to linear behavior were surprised and sometimes frustrated by this seeming inconsistency.

[3] as described in section 4;

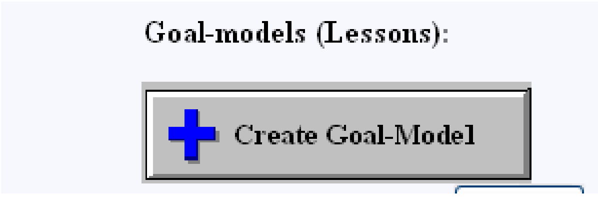

Figure 3. Example of Semantic Buttons.

3. *Role*: A separate manager login is not yet created as suggested. However, links to user guide and other useful information on the web pages of each author were added; these type of additions represent needed information that corresponds to the management role. To make these changes automatically in the future, the separate role still needs to be added.

The **possible Goal model solutions** and their **actual implementations** are:

1. *Display options:* A possible solution to display problems of groups of sub-lessons, for instance, was to display, along with the word "Group", the actual name of that group, to identify it. The extended implementation opted for a simpler version of just informing the user that a sub-group is being changed; this represents a slight improvement, but there is definitely room for more.
2. *Editing:* The ordering problem is due to the fact that domain map ordering isn't important, only lesson ordering is (as the lesson already contains the information about how the course can be presented to the user); in the lesson, an author can perform re-ordering according to necessity; however, this process can be very confusing for authors, which expect the same initial ordering after a conversion from DM to GM; therefore, preferably, this order should be kept.

6. RELATED RESEARCH

This research belongs to the larger category of authoring of adaptive hypermedia. Although this research direction has been somewhat neglected in the past, where focus was mainly on the adaptation delivery tools and engines, and their capabilities [6],[7], recently, interest in authoring of adaptive hypermedia has been revived, among others, with a series of workshops [11],[12],[13]. This new research direction supports the idea that, if regular authors of adaptive hypermedia, or even, adaptive educational hypermedia [7] are supported via different techniques (clear models [34, 44], help functionality [29], patterns- [35] or standards-based processing [14], export functionality, automatic generation of content [46] or exercises [28], construction of better feedback mechanisms [43], better visualization techniques for the complex processes present [4], etc.), the creation process of adaptive hypermedia will become possible for the 'laymen' author. Our research embraces these ideals and ideas, however, it goes one step further, in the sense that it is based on intensive testing with multiple groups of different users, from different countries and backgrounds.

Another related research direction is that of conversions between adaptive hypermedia systems. The aim of conversions is to bring adaptive hypermedia researchers and researches together, by finding commonalities between the systems [48], [27]. The one-to-one conversions are only the first step towards multi-to-multi conversions, based on common terminology, languages [18], patterns [35] and standards [14]. Our system will ultimately be aiming at passing this one-to-one barrier, due to the generic nature of the way in which adaptive hypermedia components are described in the system. Previous work [21], [48] has proven that this goal is not that far away. One-to-many conversion has already been achieved.

Finally, this research is related to Semantic Web [50] research. The extended version of the MOT system is aimed, ultimately, at complying with (Web) Standards (such as LOM, RDF). In fact, the system can export already to an light weight XML format [21], which is used as intermediary to the actual system conversions. Moreover, the modular, meta-data-rich expression type is advocated by the LAOS authoring framework [17] on which MOT is based. The main reasons for desiring this connection to Semantic Web techniques and technologies are:

- Authoring systems of adaptive hypermedia need to have reusable components; this is due to the fact that the authoring process is cumbersome, needs a lot of different components, which can be repetitive, and which would have to be designed by authors with different area of expertise; the semantic web's main aim are reusable components;
- Authoring systems need to be system-independent; until now, authoring systems for adaptive hypermedia are oriented towards one system only; this was the reason why standardization was considered unnecessary; however, reusing materials needs to break the within-system barrier, and move cross-system; in this case, it is desirable to turn to standards, such as provided by the semantic web;

Drawbacks that hinder this complete connection process are:

- Current semantic web technologies and standards don't offer all the features necessary to define adaptive hypermedia environments; especially, expressing adaptive strategies is difficult with the current standards (although SCORM [2] does some steps towards adaptivity; the new OWL [51] web standard is also promising towards accomplishing the necessary representation power).

7. PILOT STUDY

The new version of MOT was tested informally with five students, who participated in the Adaptive Hypermedia 2005 course [1] and therefore had already used the previous version of the MOT system. The testing method used was informal, based on the *think-aloud* method: the students were asked to perform some simple actions that they already performed in the previous version of the system, and comment freely on the changes. The first reactions to the speed of installation and to the new look and feel, as well as to security issues are positive. This small-sized usability test confirmed that our first set of changes make the page user friendly and provide more information.

According to Tom Landauer and Jakob Nielsen [42], the number of usability problems found in a usability test with n users is *N**

(1-(1-*L*) *n*); where *N* is the total number of usability problems in the design, and *L* the proportion of usability problems discovered while testing a single user. This formula shows that a test with five users can already find about 80% of the problems. This is a cheap way of spotting problems early, especially when using the iterative testing and implementing process. This method is only valid if the users are appropriate and representative. Due to the fact that these users had already used the previous system, they can be considered experts with it. Their opinions are relevant for the reflection of the change in quality of the system. The method in [42] thus justifies our pilot testing of our new system on five users.

8. CONCLUSIONS AND DISCUSSIONS

This paper presents the new extended version of the MOT authoring system, representing yet another step towards the "*write once, use many*" [27] philosophy of adaptive hypermedia authoring.

The first positive results with the pilot study are encouraging us to proceed with the indexed list as initially planned, and respond to the other problems with appropriate small fixes, or with larger extensions of the system. The next version of the system will be evaluated with students, as before, but also with a small group of experts in learning, to reflect not only the functional, but also the pedagogic aspects of the system.

Moreover, there is the need to more clearly and explicitly identify parallels between this work on reusability in authoring applications of adaptive hypermedia and Semantic Web. The technology, such as XML and RDF, etc., is already there and being used. Current work also dives into user model exchange based on Web Services, for instance. The modeling and data versus meta-data aspect is present in the frameworks (such as LAOS) and has been previously compared to SW approaches [11]. An important comparison should be made in more detail between the notion of SW ontologies versus layered model representations. Current work looks into the amount of flexibility that is generated by removing the interpretation from the actual static data structuring and connecting, as compared to SW ontological representation.

9. REFERENCES

1. Adaptive Hypermedia course 2005 questionnaires: http://wwwis.win.tue.nl/~acristea/AH/Questionnaires/
2. ADL SCORM, http://www.adlnet.gov/index.cfm
3. Aroyo, L., Dicheva, D., and Cristea, A. Ontological Support for Web Courseware Authoring. ITS 2002, Intelligent Tutoring Systems, LNCS 2363, Springer, 270-280, 2002
4. Armani, J., Visualizing Adaptivity for Teachers: an Authoring Tool for Designing Educational Adaptive Websites, 2nd Workshop on Adaptive and Adaptable Educational Hypermedia at the AH'04 conference, http://wwwis.win.tue.nl/~acristea/AH04/papers/AAAEH-proceedings_1-11newest.pdf
5. Blackboard, http://www.blackboard.com/
6. Brusilovsky, P. (2001) Adaptive hypermedia. *User Modeling and User Adapted Interaction*, Ten Year Anniversary Issue (Alfred Kobsa, ed.) **11** (1/2), 87-110.
7. Brusilovsky, P. (2001). Adaptive Educational Hypermedia (Invited talk). 10th International PEG conference, Tampere, Finland, June 23-26, 8-12.
8. Brusilovsky, P., Eklund, J., and Schwarz, E. (1998). Web-based education for all: A tool for developing adaptive courseware. Computer Networks and ISDN Systems (Proceedings of Seventh International World Wide Web Conference, 14-18 April) 30 (1-7), 291-300.
9. Calvi, L., and Cristea, A. I., Towards Generic Adaptive Systems: Analysis of a Case Study, AH 2002, Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 2347, Springer, 79-89, 2002
10. Carro, R.M., Pulido, E. & Rodríguez, P. (2001). TANGOW: a Model for Internet Based Learning. International Journal of Continuing Engineering Education and Life-Long Learning, IJCEELL, Pub. UNESCO. Special Issue on "Internet based learning and the future of education", http://www.inderscience.com/ejournal/c/ijceell/ijceell2001/ijceell2001v11n12.html
11. Cristea, A. I. (2005), Proceedings of the First Third Workshop on Adaptive and Adaptable Authoring, http://wwwis.win.tue.nl/~acristea/AAAEH05/AIED2005.htm
12. Cristea, A. I. (2004), Proceedings of the Second International Workshop on Adaptive and Adaptable Authoring, http://wwwis.win.tue.nl/~acristea/AH04/workshopAH.htm
13. Cristea, A. I. (2004), Proceedings of the First International Workshop on Adaptive and Adaptable Authoring, http://wwwis.win.tue.nl/~acristea/WBE/
14. Cristea, A. I., What can the Semantic Web do for Adaptive Educational Hypermedia? International Peer-Reviewed On-line Journal "Educational Technology and Society" 7(4), Special Issue on Ontologies and the Semantic Web for E-learning, IEEE, LTSC, pp 40-58, 2004.
15. Cristea, A. I., and Aroyo, L. Adaptive Authoring of Adaptive Educational Hypermedia, AH 2002, Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 2347, Springer, 122-132, 2002
16. Cristea A.I. and de Mooij, A., Designer Adaptation in Adaptive Hypermedia Authoring " ITCC'03 (International Conference on Information Technology: Coding and Computing), Las Vegas, US, IEEE Computer Science, 444-448, 2003
17. Cristea A.I. and de Mooij, A., LAOS: Layered WWW AHS Authoring Model and Their Corresponding Algebraic Operators. Int. World Wide Web Conference (WWW'03), Alternate Track on Education, Budapest, Hungary, 2003
18. Cristea, A.I. and Calvi, L., "The Three Layers of Adaptation Granularity", Int. Conf. on User Modeling (UM'2003), Pittsburgh, US, 2003.
19. Cristea A.I. and de Mooij, A., Adaptive Course Authoring: My Online Teacher " ICT'03 (International Conference on Telecommunications), Papeete, French Polynesia, IEEE,IEE, ISBN: 0-7803-7662-5, pp. 1762-1769; authors: A.I. Cristea and A. de Mooij, 2003
20. Cristea, A.I., and de Mooij, A., Evaluation of MOT, an AHS Authoring Tool: URD Checklist and a special evaluation class. CATE'03 (International Conference on Computers and Advanced Technology in Education) Rhodos, Greece, IASTED, ACTA Press, ISBN 0-88986-361-X, 241-246.
21. 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 July, 2005, Amsterdam, The Netherlands.
22. Cristea, A. I., and Cristea, P., Evaluation of Adaptive Hypermedia Authoring Patterns during a Socrates Programme Class, International Peer-Reviewed On-line & Print Journal "Advanced Technology For Learning" 1(2), 2004.
23. Cristea, A. I. Evaluating Adaptive Hypermedia Authoring while Teaching Adaptive Systems. SAC'04 (ACM Symposium on Applied Computing) March 14-17, 2004, Nicosia, Cyprus, 2004
24. Cristea, A. I. 2004, What can the Semantic Web do for Adaptive Educational Hypermedia? International Peer-Reviewed On-line Journal "Educational Technology and Society" 7
25. Cristea, A., 2004, Is Semi-Automatic Authoring of Adaptive Educational Hypermedia possible? International Peer-Reviewed On-line & Print Journal Journal "Advanced Technology For Learning 1(4), Special Issue on Authoring of Adaptive Educational Hypermedia, ACTA Press, 227-236
26. Cristea, A. I. Authoring of Adaptive Hypermedia; Adaptive Hypermedia and Learning Environments, book chapter in "Advances in Web-based Education: Personalized Learning Environments", Sherry Y. Chen and Dr. George D. Magoulas. IDEA Publishing group, 2005
27. Cristea A.I., and Stewart, C. Automatic Authoring of Adaptive Educational Hypermedia. book chapter in "Web-Based Intelligent e-Learning Systems: Technologies and Applications", Zongmin Ma. IDEA publishing group, 2005
28. Cristea, P., and Tuduce, R., Automatic Generation of Exercises for Self-testing in Adaptive E-learning Systems: Exercises on AC Circuits, 3rd Workshop on Adaptive and Adaptable Educational Hypermedia at the AIED’05 conference, (A3EH) http://wwwis.win.tue.nl/~acristea/AAAEH05/papers/4-AIED2K5_W9_PC_RT_1.pdf
29. Dagger, D., Wade, V. P., and Colan, O., Developing Active Learning Experiences for Adaptive Personalised eLearning, 2nd Workshop on Adaptive and Adaptable Educational Hypermedia at the AH’04 conference, http://wwwis.win.tue.nl/~acristea/AH04/papers/AAAEH-proceedings_1-11newest.pdf
30. De Bra, P., Aroyo, L., and Cristea, A., 2004, Adaptive Web-based Educational Hypermedia, Book chapter in Web Dynamics, Adaptive to Change in Content, Size, Topology and Use (Eds. Mark Levene, Alexandra Poulovassilis), Springer, 387-410
31. De Bra, P., Aerts, A., Smits, D., Stash, N., AHA! Version 2.0, More Adaptation Flexibility for Authors. Proceedings of the AACE ELearn'2002 conference, October 2002, pp. 240-246.
32. De Bra, P. & Calvi, L. (1998). AHA! An open Adaptive Hypermedia Architecture. The New Review of Hypermedia and Multimedia, 4, Taylor Graham Publishers, 115-139.
33. Dicheva, D., Aroyo, L., and Cristea, A. I., Collaborative Courseware Authoring Support, WBE-CATE 2002, Computers and Advanced Technology in Education, ISBN: 0-88986-332-6, 52-57, 2002
34. Freire, M., and Rodriguez, P. (2005) Comparing Graphs and Trees for Adaptive Hypermedia Authoring, , 3rd Workshop on Adaptive and Adaptable Educational Hypermedia at the AIED’05 conference, http://wwwis.win.tue.nl/~acristea/AAAEH05/papers/1-freire-aied2005-final-v2.pdf
35. Garzotto, F., Retalis, S., Cantoni, I., Papasalouros, A. (2004) Patterns for designing adaptive/adaptable educational hypermedia, 1st Workshop on Adaptive and Adaptable Educational Hypermedia at the AIED’05 conference,
36. Kravcik, M. (2004) Specification of Adaptation Strategy by FOSP Method, 2nd Workshop on Adaptive and Adaptable Educational Hypermedia at the WBE’04 conference, http://wwwis.win.tue.nl~acristea/WBE/416-802-Garzotto-Retalis%20Paper%20WBE%20patterns-formatted.doc http://wwwis.win.tue.nl/~acristea/AH04/papers/AAAEH-proceedings_1-11newest.pdf
37. LAOS definitions, http://wwwis.win.tue.nl/~acristea/MOT/help/DefinitionsLAOS.doc
38. Moore, A., Brailsford, T. J., and Stewart, C. D. Personally tailored teaching in WHURLE using conditional transclusion. Proceedings of the twelfth ACM conference on Hypertext and Hypermedia, Denmark., 2001
39. Moore, A., Brailsford T. J. & Stewart, C. D. (2001) Personally tailored teaching in WHURLE using conditional transclusion. Proceedings of the twelfth ACM conference on Hypertext and Hypermedia, Denmark.
40. MOT (My Online Teacher), http://wwwis.win.tue.nl/~acristea/mot.html
41. MySQL, http://www.mysql.com/
42. Nielsen, J. Why You Only Need to Test with 5 Users. http://www.useit.com/alertbox/20000319.html
43. Passier, H. and Jeuring, J. (2005) Using Schema Analysis for Feedback in Authoring Tools for Learning Environments, 3rd Workshop on Adaptive and Adaptable Educational Hypermedia at the AIED’05 conference, http://wwwis.win.tue.nl/~acristea/AAAEH05/papers/2-Passier-Jeuring_2.pdf
44. Retalis, S., and Papasalouros, A. (2004) Yet Another Approach to Support the Design Process of Educational Adaptive Hypermedia Applications, 2nd Workshop on Adaptive and Adaptable Educational Hypermedia at the AH’04 conference, http://wwwis.win.tue.nl/~acristea/AH04/papers/AAAEH-proceedings_1-11newest.pdf
45. RDF, http://www.w3.org/RDF/
46. Sosnovsky, S., Brusilovsky, P., and Yudelson, M., (2004) Supporting Adaptive Hypermedia Authords with Automated Content Indexing, 2nd Workshop on Adaptive and Adaptable Educational Hypermedia at the AH’04 conference, http://wwwis.win.tue.nl/~acristea/AH04/papers/AAAEH-proceedings_1-11newest.pdf
47. Steve’s templates, http://www.steves-templates.com/
48. Stewart, C., Cristea, A. I., and Ashman, H. 'Authoring once, Delivering many': Creating reusable Adaptive Courseware. WBE'05 February, Grindelwald, Switzerland, 2005
49. XAMPP, http://sourceforge.net/projects/xampp/
50. W3C OWL, http://www.w3.org/TR/owl-features/
51. W3C Semantic Web, http://www.w3.org/2001/sw/