D. Dicheva, L. Aroyo & A. Cristea, Collaborative Courseware Authoring Support, CATE'02, ACTA Press, ISBN:0-88986-332-6, pp.52-57

COLLABORATIVE COURSEWARE AUTHORING SUPPORT

Darina Dicheva,
Winston-Salem State University
601 Martin Luther King, Jr. Dr, Winston-Salem, N.C. 27110
United States of America

Lora Aroyo and Alexandra Cristea
Technische Universiteit Eindhoven
P.O. Box 513, 5600 MB Eindhoven
The Netherlands

ABSTRACT

We refined our knowledge classification and indexing approach applied in our previously developed system AIMS (Agent-based Information Management System) by introducing ontology-oriented support for collaborative courseware authoring. In order to provide a basis for formal semantics and reasoning in performing generic authoring tasks, we add ontology-based layers in the courseware authoring architecture. Ontological structuring also opens the way for cooperative authoring, (e.g., re-usage), setting the basis for authoring collaboration.

KEYWORDS
Ontology, Web-based courseware, Collaboration, ODL, Adaptivity, Adaptability

1. INTRODUCTION

The event of the World Wide Web had impact on many domains. For education, this means information available from all around the world, and courses with global dimensions. A Web-based course doesnít only provide access to local documents, but has also the function of a *gateway*, allowing access to educational materials or full courses elsewhere in the world.

These changes have a direct impact on the courseware authoring process. Authoring web courses means authoring environments *implicitly* or *explicitly* open for sharing.

This requires on one hand efficient organization of Web resources, with a powerful indexing and knowledge classification approach, in other words, building the material from start in such a way as to make cooperation and collaboration easier. An example of this novel approach is AIMS [2], an intelligent system aimed at supporting students in retrieving, evaluating, and comprehending information when performing learning tasks in a Web-based learning environment.

On the other hand, such authoring requires facilitation or implementation of cooperation or even collaboration tools.

The solution we found to efficiently organize and maintain Web-based resources is based on the *conceptualisation* of the course material or course subject domain. The basis of this conceptualisation is the creation of a *subject domain ontology*, which represents the backbone for course structures definition and more efficient information search implementation [3]. This solution conforms to the guidelines set by the recent advances in the *Semantic Web* area (e.g., the layered architecture [4]) and in the *ontologies* area. These research areas provide the ground for applying semantic structures within the educational field as well.

However, authoring in such a concept-based Web courseware includes domain-, course-, and library authoring, thus is more complicated and labor intensive than the ëstandardí courseware authoring. Apparently, this calls for enhancing the authoring support for concept-based Web courseware.

Looking from another perspective, Web-based education cannot limit itself to a fixed scheme of one-author with one-audience anymore. Concerning the latter, one of the hot issues discussed by the research community is that learning has to be custom-designed [7,9,11], to take into consideration the cognitive (learning) styles, the psychological predilections, the motivational aspects, the cultural backgrounds of the targeted students, the environmental influences, etc. However, if we look at authoring under the above requirements, each course would have to be re-authored with, roughly speaking, a multiplicative factor combining all the considerations above. Such authoring is extremely difficult and time-consuming. Therefore, it needs specialized, modern authoring support. Moreover, one author might never manage to cover all aspects mentioned above. Therefore, such requirements naturally lead to the necessity of re-usage, cooperation and collaboration among authors.

2. AUTHORING GOALS

Following the outlined needs of further support for concept-based courseware authoring [8], we envisage such support to include: automatic or semi-automatic performance of some authoring activities; intelligent assistance to the author in the form of hints, recommendations, etc.; and support of the activities of different instructors for *collaborative* building and/or *cooperative* reuse of domain and course ontologies.

There are two key ideas in our approach.

- The first one is that in order to provide enhanced support for authoring concept-based Web courseware we may well use the *systemís domain concept map* - the ontology, which captures the semantics of subject domain terminology used by the students when searching (asking questions) for relevant information necessary to perform course related tasks. The same ontology can be used by courseware authors to ask authoring-related questions or by the system to perform (semi-)automatically some authoring activities. Thus we propose to introduce additional *ontology-based layers* to the courseware authoring architecture, which allow for *intelligent assistance* to the courseware authors.
- The second goal is related to supporting cooperative authoring. *Collaborative authoring* occurs in project-like settings, where the project delegates authoring sub-tasks to a group of authors. This kind of authoring needs synchronization, dialogue support, and coordination of the whole project. In contrast, *cooperative authoring* mainly involves synchronous re-usage of authoring products, such as course materials, libraries, ontologies, etc. In our current work we focus on the latter, more precisely, on supporting primitive interaction activities in cooperative authoring.

In this paper we present an ontology-oriented support for collaborative courseware authoring. This comes as an elaboration of our approach to knowledge classification and indexing in the previously developed system AIMS (Agent-based Information Management System) aimed at supporting students in retrieving, evaluating, and comprehending information when performing learning tasks in a Web-based learning/training environment. We also focus on the cooperative authoring, which allows re-usage, in the sense of authoring cooperation, and sets the basis for authoring collaboration. In order to exemplify our method we define a set of generic tasks related to concept-based courseware authoring and present their ontological support by the newly added operational and assistant layers in the AIMS architecture.

3. COURSEWARE AUTHORING IN AIMS

From an information point of view, the general information model of a courseware system consists of *subject domain*, *library*, *course*, and *user profile objects* [2]. We envisage domain concepts as the main *linking* component for all the other objects. In other words, every document in the library is linked to one or many domain concepts and every course task is linked to several domain concepts [3]. A courseware system can accommodate different subject domains and multiple courses within them. For the purposes of this paper we will focus on one domain and the collection of courses (course structures) that different authors can create within it. Its information base includes definitions of subject domains and courses within them and a library, a collection of documents related to these courses and domains.

The domain model defines subject-domain ontology and is represented as a concept map (CM) of domain concepts with links between them. Domain concepts are linked to documents and each link is assigned a weight indicating document relevancy to the concept. The library model provides means for maintaining a collection of information items (course materials and domain-related documents) and their links to the different courses and domains. The course model defines the structure of a course and includes course topics and course tasks, pre-defined in a course task library. Each course task is represented in terms of domain concepts, which the student must learn to successfully complete the corresponding course assignment.

As a result, the process of authoring of such concept-based Web courseware should include domain-, course-, and library authoring. By supporting the authoring activities further we aim at increasing the efficiency with respect to information reuse and collaboration between the course authors.

There are three main modules in AIMS authoring environment: Domain editor, Library editor, and Course editor [3]. The *Domain editor* allows the author to perform functions, such as add, delete and update domain terms and links between them, in order to construct a domain concept map structure. The editor facilitates the full description (name, definition and classification in the concept mapping hierarchy of terms) of domain terms and the links between them. The editor also allows the author to create new types of links and to create links between a domain term and existing documents in the AIMS library.

The *Course editor* provides the author with a framework to define the structure of tasks and topics for a course. One course can consist of several topics and each topic can have several tasks. The author constructs this structure on the basis of domain terms and direct links from them to the library documents. This way s/he ensures a link between the course structure and the appropriate course material.

The *Library editor*, as most of the library systems, enables the maintenance of information collections. In this case the Library editor provides access to all the information and data related to different courses and domains. The novel feature here is the *task*- and *use*-oriented description of documents, by including instructional and presentation formats within the description of each document.

4. ONTOLOGY-BASED SUPPORT TO COURSEWARE AUTHORING

We propose a 2D-layer approach (Figure 1) towards courseware authoring support, for cooperative authoring. This approach allows re-usage, in the sense of authoring cooperation, and sets the basis for authoring collaboration. The Y-axis represents the main information objects in the information base of the courseware system (library objects, domains, courses). The X-axis targets systemís support for the information objects authoring (GUI, Assisting layer, Operation layer, and Information layer). It basically represents the layered architecture implementing system functionality.

The *GUI layer* supports user-system communication. The Information Base contains the layered description and structuring of the information objects in the courseware system (educational metadata, subject domain ontology, course ontology). The Educational metadata layer contains the description of the data sources in terms of metadata.

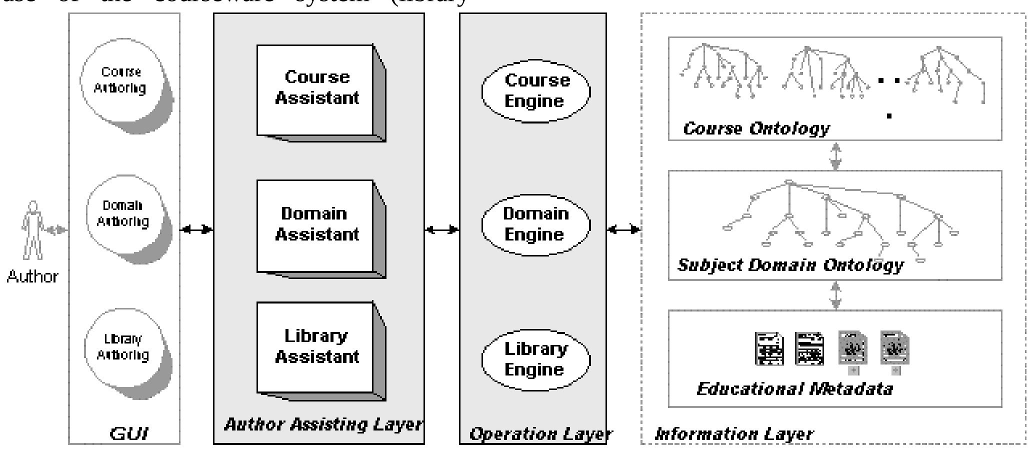

Figure 1. 2D-layer approach towards courseware authoring support

The two new components (layers) in the extended architecture are the Assisting layer and the Operation layer. The *Operation layer* enables the process of data modelling into an ontology by managing operations related to data in each information layer. This way it also contributes to the process of creation of alternative goal-oriented structures of courses. In this layer all functions related to information manipulation, consistency and co-operation are also handled. As a processing layer, it consists of three modules: *course engine*, *domain engine* and *library engine*. All of them include two sets of support operations: *consistency check* and *co-operation support*.

The consistency modules perform their activities over each of the information base layers. They provide functions to facilitate the process of authoring the domain ontology, course ontology and educational metadata in a semi-automatic way. These modules also guarantee the consistency of the educational sources. They deal with tasks such as:

- handling notions of semantic equivalence and conflict,
- handling conflict resolution rules,
- handling equivalence comparison rules,
- enhancing the resulting ontology and defining additional constraints if necessary.

The co-operation support modules offer on one hand a set of operations to check the consistency in alternative (simultaneous) course structure building by different authors and on another - predefined functions (patterns and templates) to facilitate effective reusability of the available course structures developed by different authors. In relation to the reusability support, we pay special attention to the issues associated with *merging ontologies*, such as:

- extracting portions of an ontology to be merged with another,
- identifying which frames are to be extracted from the source ontology,
- determining if the extracted information has semantic overlaps or conflicts with the target ontology,
- assisting in merging ontologies, recording the sources of inserted sub-ontologies for later reference and update,
- selecting patterns, templates in an educational ontology, to present them as predefined objects for other authors.

Among the issues of importance when merging two ontologies are those related to the following types of *semantic overlaps and conflicts*:

- semantically equivalent concepts but with different names,
- semantically different concepts but with the same name,
- semantically equivalent concepts with the same name but different definitions,

- semantically equivalent concepts linked to different (sometimes conflicting) concepts, etc.

While the Operation layer actually implements the authoring operations, the *Assisting layer*, which is based on the ontological mapping of the domain, is responsible for helping the author in the process of courseware authoring. For example, it gives hints to the author how to create a course structure, or how to link a document to the ontology, or how to link a course item to the ontology, etc. In the following section we discuss the support for some generic tasks in concept-based courseware authoring.

5. AUTHORING TASKS

In this section we discuss generic authoring tasks supported by the operation sets of the *Operation layer* and the presentation options for the author provided by the *Assisting layer*.

Table 1   Atomic operation definitions

Atomic operation	Range
ëAddí	$\{T_o, T_a, C_o, L_i, D_{oc}\}$, $T_o \in$ {course topics}, $T_a \in$ {course tasks}, $C_o \in$ {domain concepts}, $L_i \in$ {domain links}, $D_{oc} \in$ {library documents}.
ëDelí,	as above
ëEdití	as above
ëUí	*{CM, CS, EML}*, *CM*=Concept Map, *CS*=Course structure, *EML*=Educational Metadata Library.
ëLí	$\{DirLC, RelC, RelC_o, RelT_a, RelD_{oc}\}$, *DirLC* = Directly linked concepts, *Rel* =Related courses, $RelC_o$=Related concepts, $RelT_a$= Related tasks, $RelD_{oc}$ = Related documents.
ëVí	*{Graph, Text}*, ë*Graphí* gives a graphical view over the results and ë*Textí* gives a textual list of the results.
ëChkí	$\{T_a, T_o, C_o, L_i, D_{oc}, RelC_o, RelT_a, RelD_{oc}, DirLC\}$

We are defining a complete set of generic authoring tasks at all three information layers (library, subject domain and course) that are supported correspondingly by the library engine, domain engine and course engine. In this paper however due to lack of space we cannot present the developed so far ontology of courseware authoring tasks. Table 1 presents abbreviations and definitions of atomic operations used in this section [1].

There are number of composite actions such as ëdelete all topics of a courseí, ëdelete all concepts of a topicí, ëdelete all tasks of a topicí, ëdelete all concepts of a taskí or ëgive value ëaí to all the concept weights of a taskí, which can be implemented with a repetitive call to the atomic operation called ëdelete topicí and ëlist all topicsí and the corresponding operations for task and concepts.

5.1 COURSE

The interaction process between the course assistant and course engine is triggered by a set of common authoring tasks, such as ëcreate-new-course-structureí, ëedit-existing-course-structureí, ëdelete-existing-course-structureí and ëcopy-existing-course-structureí. Each of them involve set of basic course-maintenance related tasks such as ëadd/edit/delete topicí, ëadd/edit/delete taskí in a existing course structure, ëadd/edit/delete conceptí in an existing topic or task, ëlink/delete documentí to a topic or task. They, on the other hand, trigger a set of operations performed by the course engine over the existing course structures. The operations ensure data consistency by performing domain specific checks for conflicts. For instance, when the authoring task *Add* (T_o, CS) is performed by the author the course engine performs keyword search (both in the domain and in the course ontologies) on the entered topic expression. Then, the course assistant provides the option of manual editing options over those results. Next, the course assistant presents alternative views on the course engine results: (1) textual list of results with ranking according to their relevance to the search query, (2) graphical representation of the course trees with the matched concepts highlighted, and (3) graphical representation of the domain ontology with the matched concepts (ëyou are hereí indication). Within the same step the course engine also ensures the storage of the results for further reuse. Other possible course authoring tasks relate to document library and education metadata. They comprise: (a) Link a document to a topic, (b) Link a document to a task, (c) Delete a document from a task and (d) Delete a document from a topic.

5.2 DOMAIN

The interaction stream between the *domain assistant* and *domain engine* is triggered by a set of common authoring tasks, such as ëcreate-new-domainí, ëedit-existing-domainí, ëcopy-existing-domainí to a new one, and ëmerge-domainsí. These authoring tasks involve a set of basic concept-maintenance related tasks such as ëadd/delete/edit conceptí, ëcreate/delete/edit linkí between concepts, ëcreate/delete/edit link typeí. At a finer granularity level, authoring tasks include: ëremove-all-direct-links-to-a conceptí, ëremove-all-segments-of-a-path-between-two-conceptsí, ëedit/create-the-domain-mapí (the ontological structure of the domain) and ëmake links between the domain structure and the libraryí. These authoring tasks trigger a set of operations performed by the *domain engine* over the ontological structure of the domain. The operations ensure data consistency by performing domain specific checks for conflicts. For instance, when the authoring task *Add* (C_o, *CM*) is performed by the author the *domain engine* performs *Chk* (C_o, *CM*, *exist*), whether the concept C_o is already in the map, updates the CM with *U* (C_o, *CM*), and performs *Add* (C_o, *weight*) and finally provides the results to the *domain assistant* to be analysed and presented to the author. In this case the *domain engine* returns several options if the concept is in the CM: (a) *L* (C_o, *synonyms*) (b) *L* ($DirLC_o$) and (c) notification to the user that the new concept C_o has been added to the CM. These results are used as an input by the *domain assistant* to present them in an appropriate format and views to the author in order to customize and support his/her task most efficiently. In this case the *domain assistant* performs the alternative operations for the author to choose from *V* (*Text*, $DirLC_o$), *V* (*Graph*, $DirLC_o$) and another set of alternative views for the synonyms *V* (*Text*, C_o, *synonyms*), *V* (*Graph*, C_o, *synonyms*).

There are also a number of composite actions such as ëdelete all direct links of a given conceptí or ëdelete all segments of a path between two conceptsí, which can be implemented with a repetitive call to the atomic operation called ëremove a link in the CMí.

5.3 LIBRARY

A set of common authoring tasks trigger the interaction stream between the *library assistant* and *library engine*. Example of such tasks are ëcreate/edit-existing-libraryí, ëadd/delete-documentí, ëlink/unlink-document-to-domain conceptsí, ëadd/delete-keywords-to-documentí, ëedit-weights-of-keywordsí, ëlink/unlink-document-to-course topicí and ëlink/unlink-document-to-course-taskí. Due to the lack of space we are not going to go into details of all the possible library-authoring tasks further on.

6. COOPERATIVE AUTHOR SUPPORT

In pure *collaborative authoring*, each author takes over an authoring sub-task(s). When each author accomplishes the sub-task(s), the group goal is reached and collaborative mutual interdependent authoring is achieved.

In the wider spread *cooperative authoring*, authors just re-use each otherís materials, style, learning goal settings, dictionaries, linking and sequencing, etc. The primitive interaction activities among participants during both cooperative and collaborative authoring, from a macro-granulation perspective, are as follows (listed in their order of priorities):

- Planning/Execution/Creation
- Coordination/Control
- Initiative/Supervision
- Observation/Suggesting
- Data/Idea sharing
- Dialogue (with Interaction)

Surprisingly, compared to collaborative learning [13], only turn taking doesnít appear as a primitive activity, and supplementary to pure execution, creation, as a higher-level execution form, is added. From a technological point

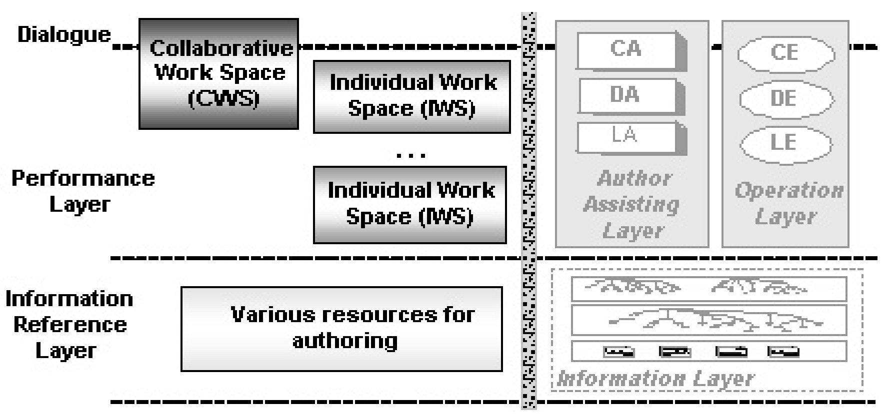

Figure 2. Mapping of the Cooperative/Collaborative Authoring on 2D-layers

of view, support systems for cooperative authoring should provide all appropriate tools for these activities. Thus, the following resources are required (see figure 2 left side):

- Individual (personal) workplace
- Collaborative (shared) workplace (shared object space)
- Dialogue channel (implicit or explicit)
- Technologically mediated remote communication (audio, visual) ñ optional.

Furthermore, more refined cognitive tools would be desirable for facilitating group collaborative authoring,

corresponding to the activities enumerated above, such as, for example, concept mapping tools [8], etc.
Figure 2 presents the suggested global model of cooperative and collaborative authoring (left side of figure), as well as the mapping of the presented 2D model over it (right side of figure). In this paper we focused on the performance layer and information reference layer, with their respective ontological representations. The activities presented in Section 3 refer to activities at the level of the individual workspace, and their results are visible from the collaborative workspace. The other layers and aspects thereof (component objects, their functionality and interactions) are not further detailed in this paper, but would be important elements of a fully collaborative environment [13].

7. CONCLUSIONS

In this paper we have introduced a layered approach for Web-based courseware authoring, which was transformed into a collaborative ontology engineering process. We have started by integrating our research into the main research steps for courseware authoring, and stressing the need for cooperation and collaboration among authors especially in Web-authoring, where the information is plentiful. As an example for a course-authoring tool based on cooperation, we based our analysis on the AIMS system. Cooperation implies dividing the course contents into re-usable parts. In AIMS, these parts are concepts, tasks and topics. We have shown the present authoring tools present in AIMS and discussed that although the structure allows cooperation and is adequate for this purpose, the actual authoring task needs improving. The paper proposes an additional semantic layer for intelligent authoring assistance and details the various types of support that such a layer should provide for various types of authoring actions possible in an ontology-based courseware authoring environment (such as AIMS, but not restricted to it).

The issues of re-usage and cooperative information sharing are in our focus, as well as some steps towards collaborative authoring, in the sense of simultaneous performance of authoring activities. The processing presented in this paper is self-contained. However, many more aspects can be analyzed, and further research direction pursued. A direction already pointed to in section 6 is towards pure collaborative authoring environments. This would mean a merge between re-usage based cooperative environments, such as the one presented in this paper, and collaborative means of working extracted from previous researches on collaborative learning environments [13].

REFERENCES

1. L. Aroyo, D. Dicheva & A. Cristea, Ontological Support for Web Courseware Authoring, *ITSí02*.
2. L. Aroyo, D. Dicheva, AIMS: Learning and Teaching Support for WWW-based Education, *IJCEELL, 11*(1/2), 2001, 152-164.
3. Aroyo L., Dicheva D., Velev I. A Concept-Based Approach to Support Learning in a Web-based Course Environment. J. Moore, C.L. Redfield, W.L. Johnson (Eds.) *AIED*, IOS Press, *68,* 2001, 1-10.
4. D.Benyon, D. Stone, & M. Woodroffe. Experience with developing multimedia courseware for the World Wide Web: the need for better tools and clear pedagogy, *Int. J. H.-C. Studies*, *47*, 1997, 197ñ218.
5. T. Berners-Lee, Semantic Web road map. Internal note, World Wide Web Consortium, 1998, http://www.w3.org/DesignIssues/Semantic.html.
6. J.D. Bolter, M. Joyce & J.B. Smith, Storyspace: Hypertext Writing Environment for the Macintosh, *Computer Software*, Cambridge, MA: Eastgate Systems, 1990.
7. P. Brusilovsky, Adaptive Educational Hypermedia, *Proceedings of PEG conference*, Finland, 2001, 8-12.
8. A. Cristea & T. Okamoto, OO Collaborative Course Authoring Environment supported by Concept Mapping in MyEnglishTeacher, *ET*, *4* (2), 2001.
9. P. De Bra, & L. Calvi, AHA! An open Adaptive Hypermedia Architecture, *The New Review of Hypermedia and Multimedia*, *4*, 1998, 115-139.
10. E. De Vries, Educational multimedia for learning and problem solving, *EuroAIED*, Lisbon, 1996, 157-163.
11. J. Kay, & B. Kummerfeld, Adaptive Hypertext for Individualised Instruction, Workshop on Adaptive Hypertext and Hypermedia, *UM'94*, Cape Cod, 1994.
12. R. Hubscher. & S. Puntambekar, Navigation Support for Learners in Hypertext Systems: Is More Indeed Better? *AIED*, IOS Press, 68, 2001, 13-20.
13. T. Okamoto, M. Kayama, and A. Cristea, Consideration of building a Common Platform of Collaborative Learning Environment, *ICCEí01*, Korea, AACE, Ed.: C.-H. Lee, *2*, 2001, 800-807.