A Layered Approach towards Domain Authoring Support

Lora Aroyo, Alexandra Cristea
Faculty of Mathematics and Computer Science
Technische Universiteit Eindhoven
The Netherlands

Darina Dicheva
Department of Computer Science
Winston-Salem State University
United States of America

This paper presents an approach to authoring support for Web courseware based on a layered ontological paradigm. The ontology-based layers in the courseware authoring architecture serve as a basis for formal semantics and reasoning support in performing generic authoring tasks. This approach represents an extension of our knowledge classification and indexing mechanism from a previously developed system, AIMS, aimed at supporting students while completing learning tasks in a Web-based learning/training environment. We propose the addition of two vertical layers in the system architecture, Author assisting layer and Operational layer, with the role of facilitating the creation of the ontological layers (Course ontology and Domain ontology) and of the educational metadata layer. Here we focus on the domain ontology creation process, together with the support that the additional layers can provide within this process. We exemplify our method by presenting a set of generic tasks related to concept-based domain authoring and their ontological support.

Keywords: Ontology, Concept Mapping, Web Courseware, Layered Processing

1 Introduction

Web-based education opens many new possibilities and is an exciting new field, that allows bringing together ëoldí AI techniques and even older teaching strategies, on a new, clearly not yet fully experimented testing bed, the Web. Although mainly praised for the advantage of bringing information to the disadvantaged users (in terms of time, space, etc.), a great potential of the Web, especially from a course designerís perspective, is the reach, i.e., the multitude of potential users. The latter means, for instance, that such issues as user modeling (which almost reached a research dead end) gain popularity again, because on large population of users the chance of spotting patterns is greater than on stand-alone computers. Moreover, this raises issues of re-usability ñ everything is out there, it is only up to us to get it. Finally, this leads to the question of how to make things more findable. Instead of blaming search engines, the semantic Web research group [4] advocates that the meaning and structure (or ontology [11,13]) should provide the necessary information.

How does this apply to courseware? This term, initially coined to name computer-supported presentation and instruction individualization, traditionally meant frames-based *teaching material* and a *courseware delivery engine*, with the goal to support authors in creating and linking frames. The term evolved to *multimedia courseware* [3,9], allowing multi-modality, *hypermedia courseware* [5,8], without constraints of predefined paths in the learning material and *adaptive educational hypermedia* [6,10].

Finally, nowadays Web-based courseware is a *gateway* to a variety of Web educational materials on specific topics or with specific educational goals, developed by the course author or represented by Web addresses and descriptions. All these changes obviously affect the courseware authoring process [2,7,16].

To address the new challenges we employ a powerful approach for *knowledge classification and indexing* in on-line learning environments, based on conceptualisation of the course subject domain. An important aspect of the proposed approach is the building of a *subject domain ontology* [2].

In this paper we present our view on ontological support of Web-based courseware authoring, which is an elaboration of our approach to knowledge classification and indexing, aimed at supporting students during learning tasks in a Web-based learning/training environment.

We start by shortly presenting our layered approach to support courseware authoring, especially aimed at ontology creation support. Then, we define a set of generic atomic tasks related to concept-based courseware domain authoring and their possible support, detailing one of these functions as an illustration. Finally, we draw some conclusions and consider future research perspectives.

2 Ontology Authoring Support

In order to understand the types of authoring support that are possible in such a system as the one we propose, we use the AIMS architecture as the basis on which to built the ontological support functionality. The AIMS authoring environment has three main modules: Domain Editor, Library Editor, and Course Editor [2]. These modules correspond to three horizontal layers in the systemís information layer: library metadata, domain ontology, and course information (Fig. 1).

As authoring of such Web courseware spreads in the three dimensions defined by the domain-, course- and library authoring, this process is obviously more complex and labor intensive than the process of linear courseware authoring. Such difficult and time-consuming authoring requires specialized, modern authoring support and re-usage facilitation [1,15].

Specifically, such authoring assistance should imply:

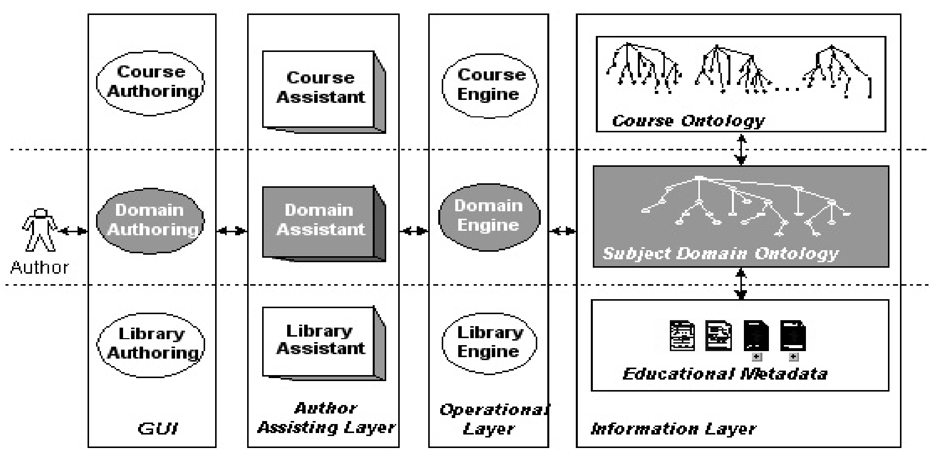

Fig. 1. 2D-layer approach towards courseware authoring support

- automatic or semi-automatic performance of some authoring activities,
- intelligent author assistance in the form of hints, recommendations, etc.,
- building and reuse activities support regarding domain and course ontologies.

Our central idea is to use the systemís ontology, which captures the semantics of the subject domain terminology, to provide enhanced support for authoring concept-based Web courseware. The power of this approach is in the reusability (i.e., cooperation) aspect: the same domain ontology can be used by several different courseware authors to perform searches and moreover, it can be used by the system for (semi-)automatic authoring activities. Therefore, the additional *ontology-based layers* allow *intelligent assistance* of courseware authors.

As in our work the ontologies are represented as concept maps (CM) [2,7], authoring involves manipulation of concept maps, i.e., creation and modification of CMs. The proposed semantic layers for intelligent authoring assistance include reasoning, consistency check, and introduce additional operations on CMs (such as comparing CMs, mapping and merging CMs, extracting subsets of CM, analyzing CMs).

We proposed in previous research [1] two extra vertical layers in a 2D-layer approach (Fig. 1) for concept-based courseware authoring support. This approach allows re-usage, in the sense of authoring cooperation. The *Y-axis* represented initially the main information objects in the information base of the courseware system (library objects, domains, courses). The *X-axis* is used to represent the systemís support for the authoring of information objects (*GUI*, *Assisting -*, *Operation -*, *Information layer*). The *Y-axis* was generalized to represent a layered architecture that implements also the system functionality. The *GUI layer* supports user-system communication. The *Information layer* contains the actual description and structuring of the information objects in the courseware system (*educational metadata*, *subject domain ontology*, *course ontology*), also organized in (internal) layers. The description of the data sources (which are local or distributed) is stored in the educational metadata layer. The two new layers in the extended architecture are the *Assisting* and the *Operation layer*. The *Operation layer* handles the operations related to data in each information layer thus providing means for modeling data into an ontology and creating alternative goal-oriented structures of courses. The Operation layer is also responsible for facilitating information manipulation, consistency and co-operation. It consists of three processing engines: (a) *course engine*, (b) *domain engine* and (c) *library engine*. All of them include two types of support operations: (a) *consistency check*, and (b) *co-operation support*. The consistency modules perform their activities over each sub-layer within the information layer. 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 [15]. They deal with tasks such as: handling notions of *semantic equivalence* [17] and conflict, conflict resolution rules, 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 (or simultaneous) course structure creation by a number of different authors. On the other hand, their role is to offer predefined functions (patterns and templates) to facilitate effective reusability of the available course structures built by different authors.

With regard to reusability support, we pay special attention to issues associated with the merging of ontologies [14], such as: extracting portions of an ontology to be merged with another [17], deciding which frames to extract from the source ontology, identifying semantic overlaps or conflicts between extracted information and target ontology, assisting in merging ontologies, recording the sources of inserted sub-ontologies for later reference and update, selecting patterns and templates in an educational ontology for presenting them as predefined objects to other authors.

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 on 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.

We will explain in more details the functioning of the Assisting layer and Operation layer with the help of some example functions for the domain ontology creation support.

According to the computational semantics of an ontology, we can integrate and situate our research within the larger research framework presented in [13], by noting that the ontologies we consider here can be situated at *level 1* (term collection, as shown previously (Fig. 1,2) and *level 3* (executable task ontologies). We still lack the connection given by *level 2* (formal definitions, constrains and axioms).

As highlighted in Fig. 1, in this paper we focus mainly on the authoring role and activities related to domain authoring. By supporting these activities we aim at increasing the efficiency of information reuse and creation of the subject domain ontology.

3 Domain Author Task Support

In this section we focus on authoring tasks supported by the operation sets [13] of the Operation layer and the presentation options provided by the Assisting layer for Domain Authoring.

We are defining a complete set of generic authoring tasks at all three information layers (course [1], subject domain and library) that are supported by the course, domain and library engines. For this complete set, we define some atomic operations, used (alone or in combinations) at all information layers (see Table 1).

In this paper however we present only a sample of the domain engine supported authoring tasks (see Table 2). In our framework, domain authoring means, as already said, domain ontology authoring or, actually, domain concept map authoring. The domain ontology creation is vital for the cooperation and collaboration aspect and, generally speaking, for re-usage.

Table 1. Atomic operation definitions

Atomic operation	**Range**	**Description**
ëAddí	performed over sets of objects *{T_o, T_a, C_o, L_i, D_{oc}}*, where: $T_o \in$ *{course topics}*, $T_a \in$ *{course tasks}*, $C_o \in$ *{domain concepts}*, $L_i \in$ *{domain links}*, $D_{oc} \in$ *{library documents}*.	adds each object to either course structure, domain ontology or metadata library
ëDelí,	as above	deletes an object from the corresponding structure
ëEdití	as above	edits the object settings
ëUí	set *{CM, CS, EML}*, where: *CM*=Concept Map, *CS*=Course structure, *EML*=Educational Metadata Library.	ensures current state update of the corresponding information structure of set
ëLí	sets *{DirLC, RelC, $RelC_o$, $RelT_a$, $RelD_{oc}$}*, where: $DirLC_o$ = Directly linked concepts, *RelC* = Related courses, $RelC_o$ = Related concepts, $RelT_a$ = Related tasks, $RelD_{oc}$ = Related documents.	lists the objects of the set(s)
ëVí	set *{Graph, Text}*, where ë*Graph*í is a graphical and ë*Text*í gives a textual results view.	gives alternative views of the *engine* results to the author
ëChkí	set *{T_a, T_o, C_o, L_i, D_{oc}, $RelC_o$, $RelT_a$, $RelD_{oc}$, DirLC}*	checks the existence of objects within the set(s)

The Operation layer provides the *Domain Engine*, and the Assisting layer provides the *Domain Assistant* for the Domain Authoring task (see Table 2). We further illustrate the interaction between the domain engine and the domain assistant in supporting the author by presenting an activity diagram of the system support for the atomic authoring task ë*add concept*í (Fig. 2: *Add (C,CM)*). Table 1 above presents the abbreviations and definitions of atomic operations used throughout this section. In Table 2 we present an excerpt of the domain authoring ontology with a selection of basic atomic tasks and the interaction between the domain engine and assistant.

Due to space restrictions we are not presenting the ontology for the course and educational metadata authoring. The interaction stream between the domain assistant and engine is triggered by common authoring tasks, e.g., ëcreate/edit/copy-domainí, ëmerge-domainsí, etc. These tasks involve basic concept-maintenance such as ëadd/delete/edit conceptí, ëcreate/delete/edit link/typeí between concepts. At a higher level, authoring tasks include: ëremove-all-direct-links-to-conceptí,ëremove-all-segments-of-a-path-between-two-conceptsí, ëedit/create-the-domain-mapí (ontological domain structure) or ëmake-links-between-domain-structure-and-libraryí. These tasks trigger operations performed by the domain engine over the ontological domain structure.

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 (Fig. 2, Table 2, task 1) the domain engine performs *Chk(C_o, CM, exist),* i.e. checks whether the concept Co is already in the map, updates the CM with *U(CM, C_o)*, performs *Add(C_o, weight)* and finally provides the results to the domain assistant for analysis and presentation to the author-user. Depending on whether the concept has been found in the CM, the domain engine returns: (a) *L(C_o, synonyms)* (b) *L(DirLC_o)* and (c) notification that the new concept C_o has been added to the CM. These results are input to the domain assistant, which is responsible for the customization and presentation in the appropriate format to the author so as to support his/her task most efficiently. In this case the domain assistant performs the alternative operations allowing 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)*. Note that some checks are superfluous due to the architecture implementation, such as no check is necessary to verify if a link to be deleted exists, because a non-existing link cannot be deleted (*Del (L, C_1, C_2, CM)* task). There are 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.

Table 2. Domain authoring tasks

Task	**Domain Assistant**	**Domain Engine**	**Result**
Add (C, CM)	▪ suggest options for the author: - add or delete *domain engine* results ▪ give alternative presentation: – V (Text, RelC, Relevance %) – V (Graph, Domain Ontology, Matched concepts) ñ ëyou are hereí ▪ notify other authors of adding C to CM	▪ Chk (C, CM, ∃) = true ▪ perform keyword search on C within DO (domain ontology) ▪ U (CM, C), notify the other authors of adding C to CM	▪ L (synonyms, C, CM) ▪ L (DirRelC, C, CM) ▪ L (NonDirLinks, C, CM)
Add (L, C_1, C_2, CM)	▪ suggest options for the author: - add or delete *domain engine* results ▪ notify if one of the concepts is missing from the CM ▪ notify if the link exists and offer editing option ▪ notify about an indirect linking path between the C_1 and C_2 and give altern. Present. of DE results: – V (Text, RelC, Relevance %) – V (Graph, Domain Ontology, Highlighted paths)	▪ Chk (C_1, CM, ∃) = true, ▪ Chk (C_2 CM, ∃) = true ▪ Chk (L, CM, ∃) = true ▪ Chk (C_1-*x*-C_2, CM, ∃) = true (indirectly linked with path ë*x*í) ▪ U (CM, L), notify the other authors of adding L to CM	▪ L (x, C_1-C_2, CM)

	▪ notify other authors of adding L to CM		
Del (L, C_1, C_2, CM)	▪ give alternative presentation of all NonDirLinks: – V (Text, L, Relevance %) – V (Graph, Domain Ontology, Highlighted paths) • Give user the option to either graphically or textually cut all segments or sub-sets of them • Notify other authors of removing L from CM	▪ Chk (C_1-*x*-C_2, ∃ for L in CM) = true ▪ Del (L) and U (CM)	▪ L (x, C_1-C_2, CM)
Del (C, CM)	• Notify the author if C has DirLinks • Give alternative views with the option to delete them or cancel the delete action: – V (Text, DirLinks, Relevance %) – V (Graph, Domain Ontology, Highlighted paths) • notify other authors of removing the C from CM	▪ Chk (L, ∃ for C in CM) = true ▪ Del (C, CM) and U (CM)	▪ L (DirRelC, C, CM) ▪ L (DirLinks, C, CM)

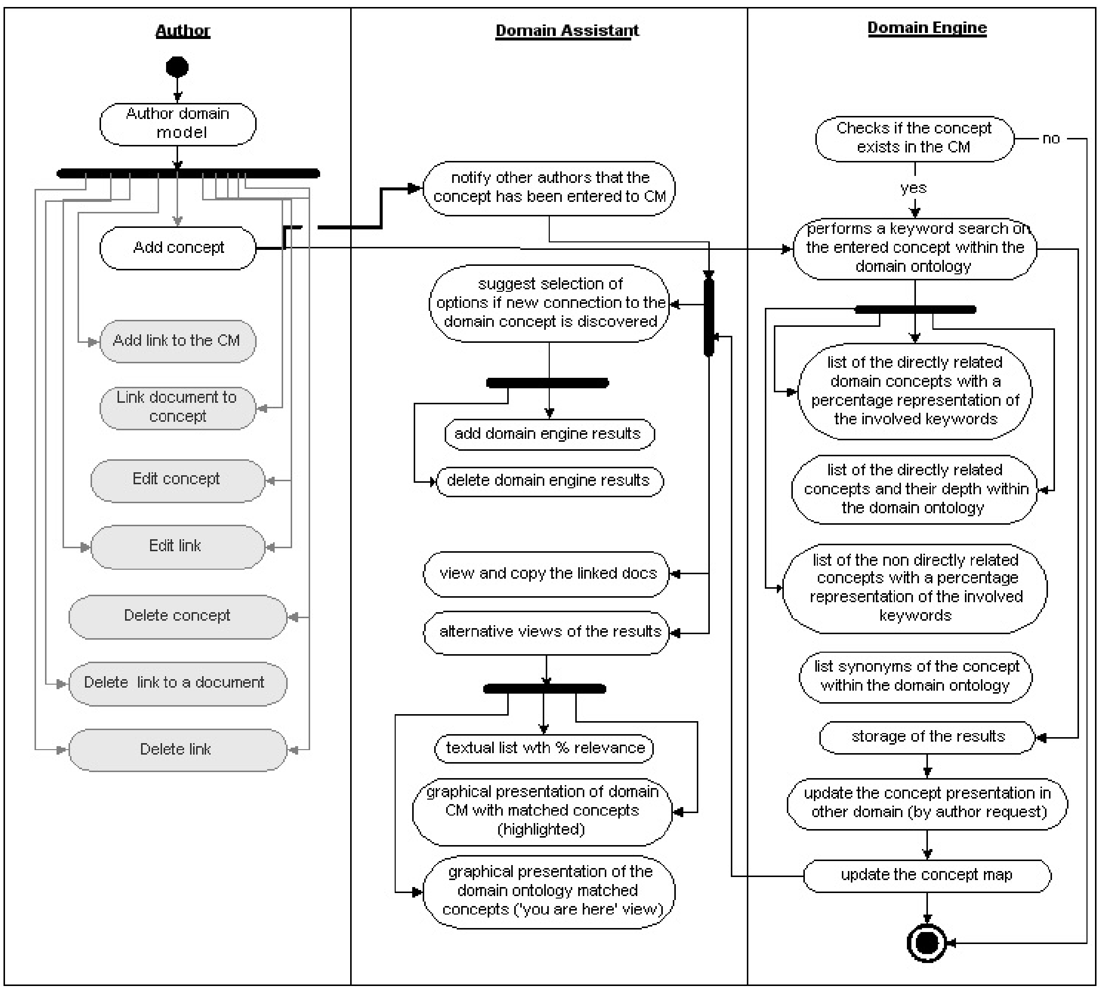

Fig. 2. UML activity diagram for the atomic domain authoring task ëadd conceptí: *Add (C,CM)*

4 Conclusions

In this paper we have given some insight into our 2D-layer approach to support Web-based courseware authoring, with special focus on the domain authoring support functionality. Our previous experience with AIMS [2] and with systems such as MyET [7] has convinced us that, although ontologies are powerful tools for Web courseware, their authoring process is difficult and needs template - and automatic support. Therefore, we have extended our support system with two additional vertical layers, the author assisting layer and the operational layer (as in Fig. 1), which have the role to support the authoring process for every horizontal layer. We claim that, besides giving a better overview on the different authoring tasks and material organization, such an extension allows intelligent authoring assistance. We are especially interested in the ontological design support, such as needed in the creation of the course ontology and the subject domain ontology. In this paper we give some explanatory details on some of the domain authoring support functions. Our research borders with another important direction, towards merging ontologies. Ontological processing is powerful when comparisons are possible, moreover, when different authors can cooperate within the same environment towards creating various courses by relying on existent courseware or courseware parts. Therefore, we will rely heavily on the developments and research in this important field [17].

This paper contributes, on one hand, towards collaborative and cooperative courseware authoring by both structuring, and adding semantics to the courseware subject domain, in particular, and to the Web courseware, in general, in the sense of the standardization efforts of the semantic Web community. This semantics is added with the help of two ontological layers. On the other hand, however, the paper demonstrates with the help of the domain ontology creation functions and possible support functions, that concept-based ontologies not only allow semantic labeling of the separate course building blocks, but also provide a structure that can be useful for reasoning and checking mechanisms.

References

[1] Aroyo L., Dicheva D. & Cristea, A. (2002). Ontological Support for Web Course Authoring. *ITS 2002*.

[2] Aroyo L., Dicheva D. (2001). AIMS: Learning and Teaching Support for WWW-based Education. *International Journal for Continuing Engineering Education and Life-long Learning*, 11(1/2), 152-164.

[3] Benyon, D., Stone, D., & Woodroffe, M. (1997). Experience with developing multimedia courseware for the World Wide Web: the need for better tools and clear pedagogy, *International Journal of H.-C. Studies*, 47, 197ñ218.

[4] Berners-Lee, T. (1998). Semantic Web road map. *Internal note, World Wide Web Consortium.* http://www.w3.org/DesignIssues/Semantic.html.

[5] Bolter, J.D., Joyce, M., Smith, J.B. (1990) Storyspace: Hypertext Writing Environment for the Macintosh. *Computer Software*. Cambridge, MA: Eastgate Systems.

[6] Brusilovsky, P. (2001). Adaptive Educational Hypermedia. In: *Proceedings of Tenth International PEG conference*, Finland, pp. 8-12.

[7] Cristea, A. and Okamoto, T., (2001). Object-oriented Collaborative Course Authoring Environment supported by Concept Mapping in MyEnglishTeacher, *Educational Technology and Society*, 4 (2), http://ifets.ieee.org/periodical/vol_2_2001 /v_2_2001.html

[8] De Bra, P. & Calvi, L. (1998). AHA! An open Adaptive Hypermedia Architecture, *The New Review of Hypermedia and Multimedia*, 4, 115-139.

[9] De Vries, E. (1996). Educational multimedia for learning and problem solving, Eu-

roAIED: *European conference on artificial Intelligence in Education*, Lisbon, 157-163.

[10] Kay, J. & Kummerfeld,B (1994). Adaptive Hypertext for Individualised Instruction, *Workshop on Adaptive Hypertext and Hypermedia, User Modelling '94*, Cape Cod.

[11] Gruninger, M. & Lee, J. (2002) Ontology: Applications and Design. *Communications of the ACM*, 45(2), 39-41.

[12] Lambiotte, J.G., Dansereau, D.F., Cross, D.R. and Reynolds, S.B. (1989). Multirelational semantic maps. *Educational Psychology Review* 1(4), 331-367.

[13] Mizoguchi, R., Bourdeau, J. (2000). Using Ontological Engineering to Overcome Common AI-ED Problems, *Int. Journal of AI in Education*, 11 (2), 107-121.

[14] McGuinness, D.L. et al. (2000). An Environment for Merging and Testing Large Ontologies, *KR00*, Breckenridge, Colorado.

15] Murray, T. (1999). Authoring Intelligent Tutoring Systems: An analysis of the state of the art. *International Journal of AI in Education*, 10, 98-129.

[16] Murray, T., Shen, T., Piemonte, J., Condit, C., Thibedeau, J. (2000). Adaptivity in the MetaLinks Hyper-Book Authoring Framework, *Workshop Proceedings of Adaptive and Intelligent Web-Based Education Systems workshop at ITS 2000*, Montreal.

[17] Sowa, J. F. (2001). Building, Sharing, and Merging Ontologies. http://www.jfsowa. com/ ontology/ontoshar.htm.