Towards Collaborative Domain Module Authoring

Mikel Larrañaga, Ianire Niebla, Urko Rueda, Ana Arruarte, Jon A. Elorriaga

University of the Basque Country (UPV/EHU)
E-mail: mikel.larranaga@ehu.es

Abstract

Technology Supported Learning Systems have proved to be useful in many learning situations. However, the development of such systems is still a hard task. Ontologies and Learning Object technologies, as they improve the knowledge reuse, can lighten the content authoring process. This paper presents Elkar-DOM, a collaborative Domain Module authoring tool. The main goal of Elkar-DOM is to enhance the cooperation of human instructors and instructional designers in the Domain Module definition.

1. Introduction

Years of research have facilitated the development of Technology Supported Learning Systems such as Intelligent Tutoring Systems (ITSs), Collaborative Learning Systems, Adaptive and Intelligent Web-based Educational Systems and so on. However, the content authoring bottleneck has held the migration of these technology supported learning systems from labs to classrooms despite their proven usefulness. This bottleneck is mainly due to the lack of average teacher oriented authoring tools; i.e. tools that teachers with low expertise on computers can use with the aim of developing Technology Supported Learning Systems. Most existing authoring tools have been developed for computing aware users or knowledge engineers. These tools become too complicated for average teachers, and they usually give up the development of Technology Supported Learning Systems. Furthermore, Murray [1] claims that “the average teacher should not be expected to design ITSs any more than the average teacher should be expected to author a textbook in their field”. Most teachers choose a reference book and use it in their classroom. They usually enhance the information with their own explanations, graphics, examples and exercises. Brusilovsky et al. [2] propose a new role for teachers in the development of Technology Supported Learning Systems: “while the construction of the core systems has to be done by expert developers, the teachers add their favourite contents (e.g.: explanations, examples, exercises, ...)”. This approach has been applied in the development of an Adaptive and Intelligent Web-based Educational System on programming.

Content authoring is a time and effort-consuming task, and the idea of knowledge reusing should not be ignored. Even more, in order to lighten the work and the content authoring process, knowledge reuse must be the base. Learning Objects (LOs) [3], which are defined as “any entity, digital or non-digital, which can be used, re-used or referenced during technology supported learning”, provide a means to facilitate knowledge reuse as they are “reusable pieces of educational material intended to be strung together to form larger educational units such as activities, lessons or whole courses” [4]. In order to facilitate the reuse of LOs, the digital didactic resources must be described in a common format before storing them, so that any learning system can retrieve the appropriate didactic resource for the learning process. For storing and sharing LOs, Learning Object Repositories (LORs) are needed [5].

However, the above-mentioned approach may not be sufficient, not even considering knowledge reuse by means of LOs technology. Teachers usually cooperate while preparing their courses. They share their experience and inform other colleagues about the problems they found in the past and propose the way they think the course should be structured, the kind of resources they use, etc. This collaborative process should also be supported in authoring and, therefore, collaborative authoring tools must be developed. Besides, a formalism is needed to represent the course structure that has been agreed by consensus. The Ontology, which is defined as “an explicit specification of a conceptualization” [6], may be the means to formalise the domain structure. Ontologies provide a mechanism to represent a concrete domain that can be understood without any ambiguity either by different people or by a computer. Obviously, a broadly accepted representation of the domain is hardly reached immediately, so the process of building ontologies must be incremental and requires several refinements. Bourdeau and Mizoguchi [7] consider that in the Technology Supported Learning Systems there should be a process in which Knowledge Engineers and Instructional Designers cooperate on the development of the ontology the system is based on.

The work presented along this paper claims for the use of collaborative authoring tools that facilitate the development of broadly agreed courses by means of ontologies and LOs. The paper is structured as follows: first, an architecture for Domain Module Authoring Tools
is proposed; following, a collaborative Domain Module Authoring tool, named Elkar-DOM, is described; finally, some conclusions and future work are remarked.

2. A Generic Architecture for Domain Module Authoring Tools

In this section, an architecture for Domain Module Authoring Tools, which relies on the use of ontologies and Learning Objects technologies, is presented. In order to enhance knowledge reuse on the Domain Module development process not only building new LORs should be considered while developing a Domain Module Authoring Tool, interactions with existing LORs should also be considered. If the project starts from scratch, the system may profit from the existing resources and a lot of time and effort will be saved. Thus, the interaction with existing LORs has been taken into account while defining this architecture.

Figure 1 illustrates the proposed architecture. The instructional designer or the human instructor uses the Domain Module Authoring Tool (DMAT) to define the domain ontology that contains the main topics and the pedagogical relationships among those topics. For each topic s/he can look for didactical resources in the LORs. With the aim of facilitating knowledge reuse DMAT may look for candidate LOs in different LORs, either local or remote.

Figure 1: Architecture of a Domain Module Authoring Tool

The LOR Communication Module (LCM) allows the communication with every LOR the system uses, making this communication transparent for the user, even if those systems use different communication interfaces or protocols. When the instructional designer or the human instructor wants to relate some LOs to a particular topic of the domain ontology, s/he starts a searching process. In this process LCM looks for the corresponding LOs in each LOR. The LCM will return all the found LOs to the DMAT, which will present them to the user.

Besides, the creation of new LOs that will be shared and reused must be supported. Therefore, authoring tools, such as text-based LO editing tool, video-based LO editing tool, audio-based LO editing tool, open-answer exercises editing tool, closed-answer exercises editing tool or procedural exercises editing tool, should be provided. Nevertheless, many instructional designers and human instructors are reluctant to learn how to use new applications. So, they can also use their favourite authoring tools provided that they allow the LO annotation in order to facilitate their reuse.

All the domain ontologies are stored in the local Domain Module Repository (DMR). This repository facilitates the reuse of domain modules and domain ontologies in order to build new ones or facilitate their incremental development.

3. Elkar-DOM: a Collaborative Domain Module Authoring Tool

Teachers usually cooperate while building the material for their courses, so tools that facilitate and even enhance collaboration should be provided. For designing a collaborative DMAT it is necessary to know what kind of tools or facilities it should provide. In order to answer that question the collaborative knowledge building process has to be analysed, the activities categorised, and, then, the set of necessary tools identified. Stahl [8] describes a model of collaborative knowledge building process and the set of necessary computer supported tools. Based on this model, next Elkar-DOM, a collaborative DMAT, is presented.

Elkar-DOM is part of a more ambitious project, which main goal is to facilitate the domain module authoring by
acquiring knowledge from existing electronic documents in a semi automatic way [9, 10]. Elkar-DOM, which has been developed with the aim of facilitating the collaborative development of Domain Modules, is an evolution of CM-DOM [11], a concept map based tool for supervising the Domain Module Acquisition.

A Concept Map (CM) is a graphical way of representing and organising knowledge that is comprised of nodes and links [12]. Different authors consider that concept mapping is a suitable knowledge representation formalism for collaborative knowledge building. Sut hers [13] analysed how students interacted in collaborative knowledge construction and observed that many students interact via graphical representations instead of using other communication tools such as chat systems. He concluded that graphical representations, such as concept maps, are suitable external representations for knowledge building. Khamesan & Hammond [14] argue that one of the most promising uses of CMs is its integration into collaborative learning activities.

Elkar-DOM has been developed with the aim of enhancing collaboration in the domain knowledge building process. It allows synchronous collaboration based on token-passing. Several users can be working at the same time seeing the current state of the domain ontology but only one can perform operations on it at a time. When an instructional designer or a human instructor wants to modify the domain ontology, s/he must request the token. Once the instructional designer or the human instructor gets the token, s/he has a limited time to work on the domain ontology.

Elkar-DOM addresses two kinds of users: the system administrator and the domain ontology authors. The authors may have two roles depending on their responsibility in the authoring process: supervisor or contributor. Each domain ontology has at least one supervisor and a set of contributors that collaborate on its development.

Next, the design of Elkar-DOM and the facilities it provides to human instructors and instructional designers are presented.

3.1 Design of Elkar-DOM

Elkar-DOM relies on a client-server architecture that entails two kinds of clients: the application for building domain ontologies collaboratively and the server management application.

The server of Elkar-DOM is the module responsible for knowledge sharing in the community of users. It supports the communication among the users as well as the management of the collaboration.

The Management client allows the remote control of the server. The administrator can use this application to manage the users of the system, manage the group of authors of a domain ontology, configure the server (e.g., the limit time for the turns can be specified), and stop the server. Just one instance of this application can be running at a time, although it is not necessary to be running the management client for Elkar-DOM to work.

The Domain Module Authoring client allows the collaborative edition of the domain ontology. Figure 2 illustrates a snapshot of the Elkar-DOM client application in which a collaborative authoring session can be observed. In the figure, a human instructor is defining the Domain Module while s/he interacts with other users through a chat session.

![Figure 2. A snapshot of the Elkar-DOM Domain Module Authoring client](image-url)
Above the set of applications that the current prototype of Elkar-DOM provides to facilitate the collaborative domain authoring process are briefly described:

- **Articulation Editor**: Elkar-DOM provides the human instructor and the instructional designer with a graphical environment in which the beliefs are represented by means of concept maps. Nodes identify domain topics while arcs specify the relationships among those topics. The category of the topic is symbolised by the *node shape* (e.g., an oval node refers to a concept while a square node corresponds to a procedure) and the category of the pedagogical relationship (Is-A, Part-Of…) is shown in the label of the arc. Besides the domain topics and the pedagogical relationships, other information such as the topic relevance in the domain, the difficulty that the topic may entail to the learners, is also presented. However, the information should be shown in a way that facilitates instructional designers’ and human instructors’ work and does not produce any additional cognitive overload. Therefore, the following graphical resources have been used. *Flags* are employed to visualise graphically information about the topic relevance, difficulty and even the quantity of didactic resources related to the particular topic. The *arc thickness* is used to express the strength of the relationship between topics, e.g., the closeness between two topics. The information about a topic or a relationship is shown to the human instructor or the instructional designer when s/he double-clicks on it. The system presents a window where the user can set a different value for any of the characteristics of the topic or relationship by selecting it in the corresponding *Combobox*. When the user closes the window by clicking on the OK button, the domain ontology reflects all the changes s/he has done either with different *node shapes*, *relationship labels*, *line thickness* or *flags*. A usual concern when representing graphically a domain ontology is the scalability of the approach. In order to face this important issue, the tool implements two mechanisms. On the one hand, the user can contract and expand parts of the domain ontology represented by the concept map. On the other hand, the tool provides a *filter mechanism* that allows the user to work with parts of the domain ontology. For example, setting the corresponding view, the user may supervise only one kind of relationships. An argumentation editor should provide a means to encourage and help participants to formalise their beliefs [8]. An example of such facility is a brainstorming area. Elkar-DOM provides a chat tool, i.e., a synchronous communication tool, with the aim of enhancing the interaction among participants and facilitating a brainstorming process. The chat messages are logged, because the cooperation process of the human instructors and the instructional designers may be even as relevant as the final result. Elkar-DOM considers that each operation on the domain ontology is the consequence of the discussion the participants have held, so the message interchange is related to the operation on the domain ontology log.

- **Discussion forum**: The collaborative knowledge building process is not always a synchronous process. In many occasions, the participants are geographically distributed and therefore may have difficulties to cooperate at a time. Thus, a synchronous tool is not enough for this kind of situations, and a tool that supports asynchronous cooperation should be provided. While modifications can be directly performed on the ontology, the justification and discussion must be carried out in other kind of applications, such as a discussion forum. The discussion forum allows the different participants to state their beliefs, justify them and argue on others beliefs. The goal is to perform modifications in the domain ontology when they are agreed. The discussion forum allows the collaborative knowledge building when synchronous cooperation is not feasible.

- **Argumentation graph**: The development of the domain ontology is an incremental process in which the ontology is continuously refined and modified as a result of the argumentations that the participants state. Therefore, it is very important to be able to represent, in an intuitive and inspectable way, these argumentations. Elkar-DOM is able to record the stages during the domain ontology development and provide a dynamic reproduction that shows the evolution of the domain ontology. This feature provides the user a means to replay the different snapshots of the domain ontology using video-like buttons. Besides, the set of chat messages corresponding to each operation can be reviewed.

The main features that have been described above can be observed in Figure 2, which illustrates a Domain Module authoring session on *digital electronics*. The topics of the domain are represented by nodes and the relationships by arcs. The Domain Module has been semi-automatically developed [9, 10]
and must therefore be reviewed. The dashed lines show the contents or relationships that have not been inspected yet. The flags describe respectively the difficulty, relevance and the amount of LOS related to each topic of the domain. The Actions Window allows the user to request or release the turn, and chatting with other users.

4. Conclusion

In this paper, first a generic architecture for a Domain Module Authoring Tool that profits from ontologies and Learning Object technologies to enhance Technology Supported Learning Systems development has been proposed. Next, Elkar-DOM, a collaborative Domain Module Authoring tool which main goal is to enhance the cooperation of human instructors and instructional designers, has been presented. Elkar-DOM has been developed under the hypothesis that building the domain ontology that describes the domain must be performed by consensus. Elkar-DOM has been developed under the hypothesis that building the domain ontology that describes the domain must be performed by consensus. Elkar-DOM uses concept maps to represent the domain ontology and provides a set of utilities to improve the cooperation. Currently, Elkar-DOM is being evaluated with different domains.

Future work includes the implementation of other support applications that have been referred above and the interaction with existing Learning Object Repositories. Elkar-DOM is part of a more ambitious project in which the domain ontology is gathered in a semi automatic analysis of existing electronic document based on heuristic reasoning and Natural Language Processing techniques [9, 10]. The integration of the semi-automatic domain ontology generation will considerably lighten the human instructor’s and instructional designer’s workload and enhance knowledge reuse at different levels in Technology Supported Learning System authoring.

Acknowledgments

This work is supported by the Univ. of the Basque Country (UE06/19), the CICYT (TIN2006-14968-C02-01) and the Gipuzkoan Council in an EU Programme.

References


