Making Courseware Reusable

Khaldoun Ateyeh, Jutta A. Mülle
Department of Computer Science
Universität Karlsruhe, Germany
{muelle, ateyeh}@ ipd.uni-karlsruhe.de

Abstract
In this paper we discuss how the development of reusable courseware can be supported by using modularization and ontologies. We argue that those concepts will enhance the quality of courseware and its cooperative development as well as the degree of its reusability and exchange. An adequate development process is presented, and it is shortly sketched how to use the resulting courseware in a new manner. Our goal is to provide an approach to develop adaptable learning material that can be reused by different authors and educators and even students for many target groups in different contexts.

1. Motivation

The development of reusable generic components has been a very important and effective instrument in every engineering discipline. They make it possible to develop high quality products that meet market requirements such as cost effectiveness and time-to-market. For a long time, these well-known engineering principles were neglected in the development of software. This lasted until the software crisis in the seventies forced software developers to recognize the need and importance of applying engineering principles to software development. We believe that courseware is bound to become the next candidate for a crisis unless we are willing to turn courseware development into an engineering discipline.

In this paper we will discuss how concepts such as modularization and ontology can effectively be used in the development of courseware. We argue that applying modularization to courseware development and the use of ontologies will result in high-quality courseware that can be re-used beyond today's practice, which is mainly based on the motto "One Size fits all" and "Take it or leave it". What we need is courseware that is adaptable so that it can be re-used by many authors and educators for many target groups and in different contexts. That is, we require that courseware can be adapted or tailored to a broad range of needs. In turn this requires authors and educators to take a fresh look at how they produce and re-use courseware.

To illustrate the need for a fresh approach, imagine the following two scenarios. John, Samantha and Norman are three professors responsible for a course on "database systems" at three different institutions of higher learning with target groups that differ in their focus on theoretical and practical aspects in their respective curricula. The three institutions decide to share in the development of the accompanying courseware. However, their cooperation should extend beyond mere exchange of multimedia elements such as animation, simulation, video-clips, pictures and audios. Equally important, it should be possible to exchange self-contained learning units ("learning modules") that cover a specific topic, and to adapt these to the preferences of each institution. The second example deals with on-the-job learning or training in companies. In this scenario, employees have permanent access to a learning system that offers them problem-oriented learning material that helps them solving problems in their working process. We are using these two scenarios because they differ in their requirements and so their combination characterizes courseware in a comprehensive way.

The paper is organized as follows. Section 2 discusses a learning model and derives the requirements of the different user groups within the authoring environment and the learning system. Section 3 discusses our approach to enhance the cooperative development of courseware and courseware reuse. Finally, we conclude the paper with Section 4.

2. Requirements

In the first example, our focus is the authoring and teaching environments, the so-called author and educator views. The courses on "database systems" will be held in a lecture-style with less or more interactivity with the students. They will differ according to the teaching context as, e.g., how much examples are included, how much theory has to be mediated, which media types will be used, which basic learning theory does the professor pursues and so on.

In the second example we mainly concentrate on the learning environment, the so-called student view. To meet market requirements and to take advantage over their competitors, companies have to keep their employees up-to-date on new technologies or business processes. New working models that consider the learning process as an integral part of the working process have been suggested. Technically, employees should have permanent access to a (web-based) learning system that offers them problem-

oriented learning modules. The expectation is that in this way employees can quickly fill a specific knowledge gap during the working process. Consequently, for such a model to be successful, it is crucial to have a modular design of the courseware that respects the working process and combines learning and working in an effective way.

In this learning context, Constructivism[1] [1] is often the learning theory of choice. It imposes a number of conditions on the learning process:

- Representations of a complex reality instead of an abstract content.
- More authentic activities of the learner and fewer activities by the teacher.
- Presentation of multiple perspectives on the same problem.

Obviously, the theory requires treating student and teacher differently. In fact, all current learning models distinguish three user groups in the learning process, namely authors, educators and students. These users have different needs and requirements. In the following we will discuss the view of each of them, in order to derive their specific requirements on the learning system.

Content Provider (Author) View: The authoring team, essentially the content provider, is responsible for developing and providing learning materials. The development of learning material is an interdisciplinary and cooperative process. The authoring team normally consists of domain experts, pedagogues, media specialists, graphic designers, computer scientists, and psychologists. So, from the point of view of content providers a platform supporting cooperative interdisciplinary work is crucial. Since re-use is essential to the engineering process, the platform should also include a sophisticated content management system that provides intelligent search and navigation functions on repositories of learning modules and multimedia elements.

Educational institution (Educator) View: Educators are responsible for developing, mediating and coaching a specific course. The development of a course should concentrate on:

- Determining the context, target group and learning form (such as life auditorium versus distance supervised versus distance self-learning versus lifelong-learning while working).
- Specifying the content to be mediated considering the overall course goals, a context, the target group, the learning form, and perhaps other contexts. The content specification results in a list of needed learning units.
- Searching the specified units in learning units' repositories. If not found a developing request is sent to the authoring team or content provider.
- Choosing a suitable didactical template.
- Adapting the modules to the context, e.g. by adding context specific materials.
- Composing or connecting the modules to courses by adding a navigation structure on the basis of the chosen didactical template and the learning form.

In our scenario John, Samantha and Norman have to conduct the above steps to develop their respective courses for the subject "database systems".

Student View: New results in learning theories, especially constructivism suggest a stronger focus on students and less focus on the educator in the learning process. Courseware should be authentic and should include interactive components in order to allow students to construct their knowledge. Students should be able to develop different views on the same subject so that they can compare different use scenarios in which they can apply the acquired knowledge. They should also have access not only to prefabricated course materials, which is important for a first insight on the subject, but should also be able to explore a module repository so that they can develop a more comprehensive view on the subject from different points of view. For example, if the participating universities place different emphases on the material, that is, operate in different contexts, a student may augment his knowledge by switching contexts. Finally, students should have the possibility to build their own views by connecting the modules in a way that matches their own knowledge structure and mind map.

3. Concepts

The three professors in our example agreed that for a successful cooperation a cooperative framework and a modularization concept are highly necessary. They also agreed that their cooperation should not be restricted to exchanging multimedia elements such as animation, simulation, video-clips, pictures, audios, etc. but rather should be based on exchanging adaptable self-contained learning units "learning modules" that cover a specific topic and encapsulate a semantic meaning.

In the Company scenario, we have the requirement to support the learners with a flexible learning environment that allows a personal learning process for each student. They should get the information specific to their individual state of knowledge. They also need intelligent search functions on "learning modules" (see above) and the relationships between them, e.g., students, often needs to know which modules are required, in order to be able to understand a specific module. Additionally, students often

[1] In the constructivist perspective, knowledge is constructed by the individual through his/her interactions with his/her environment [2]

want to learn in an explorative way so that they want to use some form of browsing space on the learning material.

In this section, we will present a courseware development platform that meets the requirements derived in Section 2. The platform is based on the modularization concept introduced in [3], ontologies, and XML. Figure 1 shows how these concepts are related to each other.

In the following we will give a brief description of these concepts.

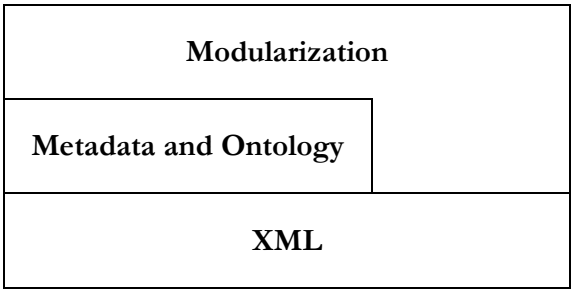

Figure 1: Basics of a courseware development platform

3.1. Modularization

Analyzing courseware reusability, exchange, and modularization, we believe that the break through in courseware reuse has been stopped mainly because systems [4, 5, 6] and concepts developed so far have been mainly considering prefabricated inadaptable[2] modules mixing and scattering aspects like content, didactic and presentation in a way that does not allow to adapt one of these aspects without the need to adapt the others. In most cases of courseware reuse, the ability to adapt these aspects is a crucial requirement. Without it, such modules are similar to puzzle pieces that fit together only in one way and thus can only be re-used in one context for one target group and in most cases only by their author.

In our scenario John, Samantha and Norman have different didactics and presentation tastes. Thus, if Samantha for instance tries to reuse a piece of courseware form John, she will be forced to adapt it to her own didactic and presentation. If these aspects are strongly mixed together in the module, it will be very difficult for Samantha to adapt them and the costs for doing so will be very likely higher than a new development.

Instead, Our modularization concept is based on the separation of aspects. It separates between the content and other aspects that prevent and complicate its reuse such as presentation, didactic and pedagogic. It also considers the fact that the development process of courseware is an interdisciplinary process and thus is cooperative in nature. Dealing with these aspects separately by providing different modularization levels (see Figure 2) gives users the possibility and the ability to choose and to decide which aspect they want to change and which aspect they want to keep depending on their needs, capabilities and recourses (see Figure 3). Back to our scenario: In this case, Samantha will be able to choose besides reusing the content to reuse the structure and/or the presentation of John or she can easily plug in her own.

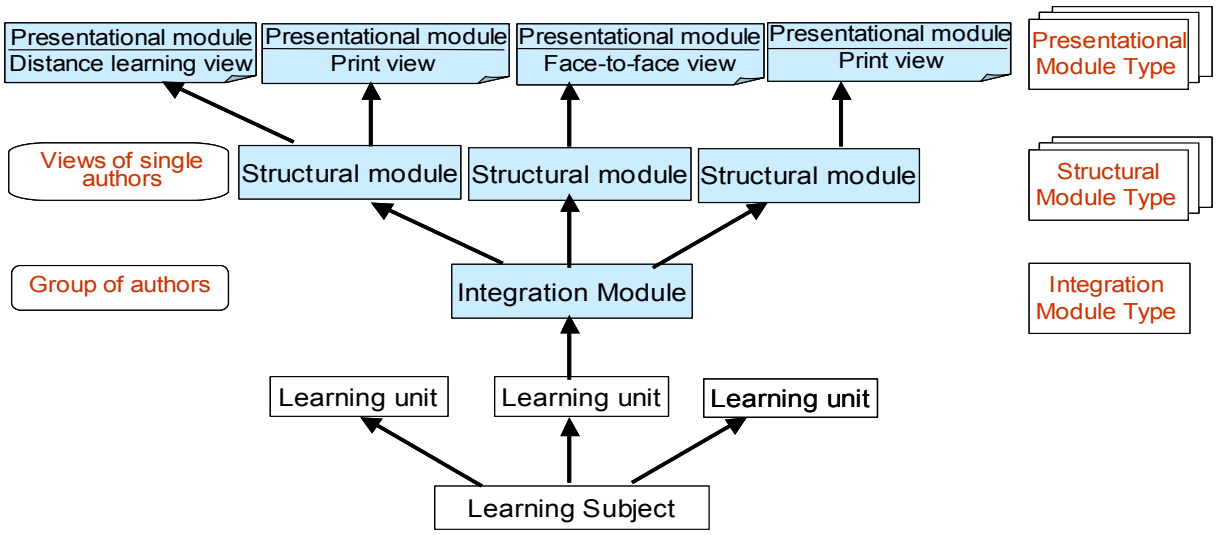

Figure 2: Levels of modularization

[2] with a reasonable effort

The modularization concept proposes to divide a learning subject or domain in semantically self-contained, adaptable and reusable learning modules[3]. In order to allow and support adaptability, the modularization concept also suggests a development process that handles different modularization levels (see Figure 2). Each modularization level deals with different aspects and produces modules that support a different degree of reusability. Figure 4 shows an UML class diagram describing the main components of the modularization concept and their relationships. We make a difference between the following components (see Figure 4):

- **Reusable learning unit**. Considering content aspects we can mainly distinguish between two kinds of reusable units:
  - **Learning atom**: this is the smallest reusable – or changeable - unit in a learning module. Semantically, it is not self-contained and it is not conceivable to be re-used without the context of the learning module to which it belongs. An example for a learning atom is the proof of a theorem.
  - **Learning module**: important characteristics of a learning module are that it is semantically self-contained and adaptable. A learning module has at least three aspects that are considered separately namely content, didactic and presentation. According to these aspects a module has three different variations: **integration module**, **structural module** and **presentational module**. In contrast to a learning atom a learning module can be used in different contexts.

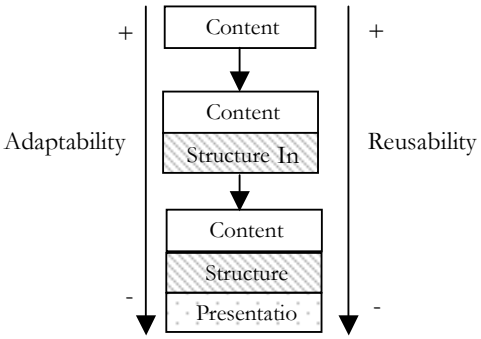

Figure 3: Learning unit variations

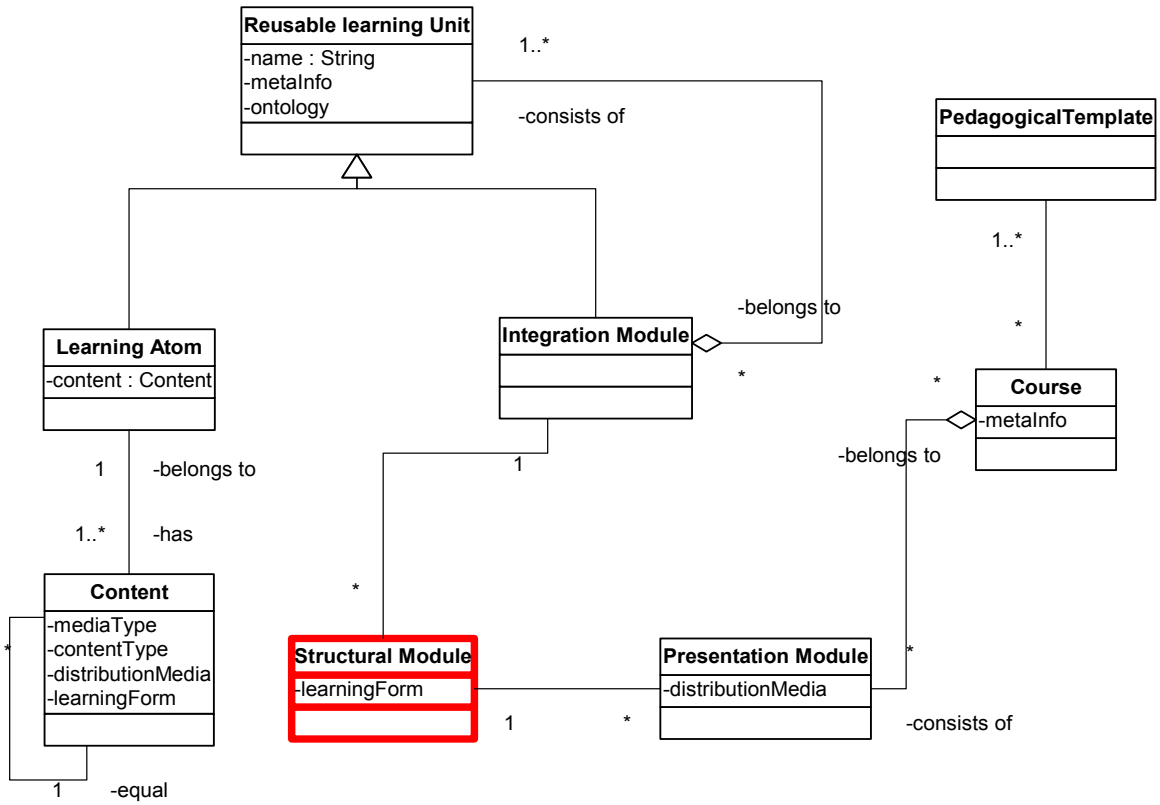

Figure 4: Levels of modularization

[3] Choosing a good and reasonable granularity is very crucial for reusability. We believe that developing standardized ontologies for the different subject domains can facilitate this.

- **Course:** A course covers in general a whole learning subject in relation to a specific context. Therefore, a course represents a view of an educator or an educational institution on a learning subject. A course consists of learning modules. In our scenario, we have three courses for the same learning subject but with different target groups and with different focus.

Figure 2 gives an overview of the modularization concept. In the development process, a learning module evolves in three different processes on three different modularization levels. In the following, we will give a brief discussion of the modularization concept by describing the different levels of modularization:

- Level 0: deals with the learning subject as a whole. This level is mainly concerned with:
  - Dividing the learning subject or domain in reusable self-contained **learning units** considering existing ontologies (see next section) on the subject. Therefore the three professors divide the subject "database systems" in adequate learning units, e.g. normalization, or relational calculus.
  - Specifying every learning unit by listing its **content-based components**[4] covering the needs of all participants. In our scenario, this means that the specification should cover the needs of the three institutions[5], by including components covering both theoretical and practical aspects.

After identifying and specifying the needed learning units, every learning unit goes through a development process consisting of the three different modularization levels:

- Level 1: This phase develops modules that are concerned with content aspects and provides a cooperative framework for different authors, educators or universities. The result of this level is so-called **integration modules**. An integration module represents a general view on a learning unit and can be regarded as a content container for all potential users. It contains little or no information about the later purpose of the content or the context within which it has to be used. It also does not contain any information about didactic or presentation. At this phase an implementation of the specified learning unit is handled. The implementation follows the following schema:
  - Providing one or more implementations for every specified component in the learning unit. In our scenario this means that Samantha, John and Norman can add their learning atoms to the learning module.
  - Providing meta-data: this is done by generating an instance of the integration module type (see Figures 2 und 4) and binding the learning unit to the ontology used from the community (our three universities). To give you a feeling about the sort of meta information covered in the integration module type let us discuss briefly some of them. As Figure 4 illustrates, a learning module "the implementation of a learning unit" may be composed of other learning modules and learning atoms. A learning atom can have one or more realizations and is classified according to the following attributes:
    - Content type: this attribute describes the content type of one implementation for a learning atom. This attribute can have values like motivation, description, example, exercise, theorem, proof, summary, etc.. One learning atom may have different content implementations covering concepts such as description, example and so on or even different points of view.
    - Media type: this attribute describes the format in which the content of a learning atom is implemented. Possible values are: audio, video, text, graphic, animation, simulation, etc.. This also allows having many implementations for a learning atom based on format criteria.

  Other attributes could describe the distribution media and suitable learning forms for the content.
- Level 2: this phase represents the view of one author or educator[6] on the learning unit and corresponds to the **structural modules** at the modularization level 2. In our scenario this means that each of the three professors will construct his/her own view on the learning unit and prepare it in order to fit in her/his own context. An educator constructs his view on a learning unit by developing a structural module out of an integration module. This is done by selecting the suitable learning atoms of an integration module depending on the target group, learning form, and didactic methods of the educator. It is conceivable that this step can be automated by a software agent using a specific pedagogical template, a specific target group and user profile and by analyzing the meta-information (see above). Having a repository of integration modules, we can fulfill the needs of educators specified in the requirements section. Educators are only concerned with specifying their courses and choosing the adequate learning or teaching strategy.
- Level 3: the corresponding level of modularization deals with the appearance of learning units and handles

[4] according to the UML class diagram in figure 2 a learning module consist of reusable units which in its tern can be either a learning module or a learning atom.

[5] this process is iterative and the list of the content can be updated by adding new content at any phase in the development process.

[6] Or educational institutions or even single students

presentation modules. It defines precisely how the learning modules should be presented and how users interact with them. Psychological and pedagogical as well as didactic aspects have to be considered.

3.2. Metadata and Ontologies

Our three professors have so far modularized the subject domain "Database systems" and have developed integration modules and for some of those even structural and presentation modules. Samantha decides to build her own course. After determining the context, the target group, and the learning form she specifies conceptually[7] the needed learning modules. At this stage, a possibility to browse a general schema of the subject "database systems" is desired, in order to find out how concepts are semantically related to each other. Second, she needs to find out whether her specified (or similar) modules are already existing or not. To conduct this task she needs a sophisticated search possibility where she can precisely configure a search query depending on her specification.

As we can see, to enable courseware sharing and reuse, it is crucial beside the conceptual considerations in the development process to mark the learning modules with standardized metadata that describes the various aspects such as content, didactic as well as technical and legal aspects. This will enhance creating an open environment where courseware can be easily discovered, exchanged and reused between users, education communities and information systems.

This task can be perfectly over taken by metadata standards and ontologies. Ontologies[8] can be seen as a special kind of metadata that deals with providing a shared and a common understanding of a domain that can be communicated between people and heterogeneous application systems [7]. Marking learning objects with standardized metadata and binding them to a domain specific ontology[9] provides:

- Intelligent search and query answering instead of keyword matching and pure information retrieval. A metadata standard provides a searching service with all semantic and structural information needed to find a specific learning module. An ontology that semantically describes the various learning modules provides a notion browsing space and can be used to navigate and search for the required learning modules.
- Learning objects[10] exchange via a translation language such as XSL.
- Definition of views on learning modules.
- Automatic searching and composing the required courseware by software agents.

To grantee the biggest possible distribution of the developed learning modules we put on standardized metadata schemas such as LOM [9], IMS [10] and ARIADNE [4]. The modularization concept supporting an authoring group also proposes to define a specification of shared vocabulary and to capture it in so-called logical dependency graph.

3.3. XML, XSL

XML [11] is a standard that enable us to define the structure and semantic of data and information. Another advantage is portability, what Java did for portability of code, XML claims to do for data [12]. Thus, using XML as a standard format for defining the various module-types, metadata and ontologies will strongly enhance the exchange and reusability of courseware among people and heterogeneous application systems.

On the other hand, XML-based tools such as XSL [13], XSLT[14] and XMLQuery [15] offers a good platform for the realization of our approach. XSLT for instance can be used as a transformation processor between the different levels of modularization and to define the appearance of modules and courses. With XMLQuery XML offers a query language that enables the execution of very complex search queries on the defined metadata and ontologies.

4. Conclusion

We propose a modularization concept for courseware development based on the separation of aspects that complicate the cooperative development and reuse of courseware such as presentation and didactic. Furthermore, the resulting development process and some usage issues are briefly presented. The influence of metadata, ontologies, and XML in this area have been identified as important for creating an open environment where courseware can be easily discovered, exchanged and reused between users, education communities and information systems.

Our approach profits from metadata standardization efforts such as IMS [5], ARIADNE [4] and LOM [6]. It also makes use of the software design pattern "Model View Controller" (MVC) as reflected in Smalltalk [16] or Java [17].

Currently, we are implementing a prototype that supports the modularization concept with the adequate author and educator environment. In the validation process our focus concentrates on the aspect of reusability of learning modules in different contexts and how easily modules may then be reused in different teaching and learning scenarios.

---

[7] This means these modules have not to be existing in the module repository

[8] An ontology is an explicit specification of a conceptualization [8]

[9] in our case study this means that an ontology for "database systems" is needed

[10] learning objects can be exchanged at the different levels of granularity: atoms, modules and courses.

References:

[1] http://carbon.cudenver.edu/~mryder/ itc_data/constructivism.html

[2] http://www.stemnet.nf.ca/~elmurphy/emurphy/cle.html

[3] K. Ateyeh, J.A. Mülle, P. C. Lockemann: Modular Development of Multimedia Courseware. WISE 2000, the first international conference on web information systems engineering Volume 2, June 2000, Hong Kong, China.

[4] ARIADNE, http://ariadne.unil.ch/

[5] http://www.macromedia.com/software/authorware/

[6] http://www.macromedia.com/software/director

[7] Dieter Fensel: Ontologies: A Silver Bullet for Knowledge Management and Electronic Commerce, 2001 Springer.

[8] Thomas R. Gruber: A translation Approach to Portable Ontology Specification. Knowledge System Laboratory, Stanford University, Technical Report KSL 92-71, http://www-ksl.stanford.edu/kst/what-is-an-ontology.html, September 1992

[9] http://ltsc.ieee.org/wg12/

[10] http://www.imsproject.org/

[11] http://www.w3.org/XML/

[12] Brett McLaughlin: Java and XML; O'Reilly, June 2000

[13] http://www.w3.org/Style/XSL/

[14] http://www.w3.org/TR/xslt

[15] http://www.w3.org/XML/Query

[16] S. Lewis: The Art and Science of SmallTalk; Prentice Hall, 1995.

[17] D. Geary: Graphic Java 2, Mastering the JFC; Prentice Hall