Component-based Content Development for E-Learning Systems

Andreas Bollin, Roland T. Mittermeir, Roland Wohlfahrt
Institute for Informatics-Systems
Klagenfurt University
Klagenfurt, Austria

{andi, roland, rolze}@isys.uni-klu.ac.at

Abstract: Based on modularisation approaches for easy reuse and adaptation we developed linkage structures between e-learning modules that allow for the dynamic adaptation of presentations to interest- and knowledge profiles of individual learners.
The realization of these ideas is possible by extending LOM-metadata by "standardized" descriptions of the respective LOM's content and the preconditions presumed from learners. These descriptions can play the role of dynamic link attributes.

1. Motivation

Following the definitions of Rosenberg (Rosenberg, 2001), e-learning refers to the use of internet technologies to deliver a broad array of solutions to support self-directed, open and distance education. But e-learning is more than delivering available content by technical media, it also encompasses new didactical concepts. After the first hype in the mid 1990s it became clear, that the requirements of e-learning systems (exchangeable and reusable learning units, system independent courses) cannot be achieved by just using hyperlink-technology. Various companies and organisations engage in the development of standards and concepts to assure that e-learning resources will be identified easily, are reusable and portable among different platforms. While component reuse is state of the art in software development, it is still a dream of course-authors to write content only once and to reuse it wherever and whenever needed. To support this kind of modularisation, standards (cf. SCORM or LOM, see http://www.adlnet.org and http://ltsc.ieee.org/wg12) are evolving. The general goal is to generate basic learning items (also called e-learning objects, shareable content units, or simply components) that are interoperable, reusable, durable, extendable and, manageable.

The next chapter introduces the concept of hard- and soft-links for supporting content reuse. Chapter 3 describes the basic steps necessary to implement hard- and soft-links and to integrate them in existing platforms. It concludes with an example of how to use the concept for generating a course and how the learner can benefit from the flexibility of the connected chunks.

2. Hard- and soft-links

The basic idea behind this concept is to support the reuse of existing learning components. To do so, we modularise the content of courses into components that can be recombined by future course-authors. The elementary components, referred to as content chunks, are kept with their metadata in a repository. They can be combined to higher level entities, modules, which form eventually the content of the whole course.

This goal is achieved by embedding content kept in a repository (content chunks) in metadata as shown in Fig. 1. The *general metadata* follows the LOM standard and positions the content in a general classification framework. *Pre-conditions* and *post-conditions* describe the learners assumed knowledge before and after studying this component. Finally, a *link section* relates it to other assets in the repository. This link section contains also links relating assets to form higher level entities (modules). The distinction between these link types is seen as a response to the criticism raised in (Seeberg et. al, 2000).

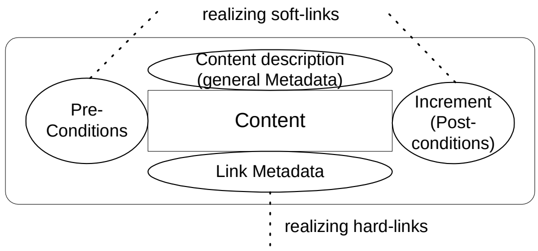

Figure 1: Metadata is attached to the content in order to enable hard-, and soft-link facilities.

Aggregations of chunks or modules leading to fixed sequences of presented material are realized by a type of hard-links we call *directed-link*. This type of link is to be provided by the author when creating a course (or when creating a germane set of learning components). It forces the learning system to provide chunk A (the source of the directed link) before chunk B (its destination). The second type of hard-link is called *r-link* (for relational link) and it describes the following types of relationships: "optional", specifying, that another chunk contains optional information: "special", stating that another chunk provides information in more detail; "general", leading to a chunk that contains a much broader information about the same topic; and "learner type", leading to a chunk containing the same information that is provided for a different type of user. The primary purpose or r-links is to support authors and to structure the repository.

To allow dynamic adaptation of e-learning content during the execution of a course, a user profile has to be taken into consideration. The key-idea is to supply the course in general with more material than the average learner wants to consume or with material presented on different levels. The dynamically updated interest- and knowledge profile of the learner is then used to select the appropriate chunk out of a pool of chunks based on the chunks' pre- and post-conditions. The knowledge necessary to understand a specific chunk has to be described in its pre-condition. The newly gained knowledge is described in a section called increment (or post-condition). Pre- and post-condition tags express implicit connections between matching chunks. Therefore, we call them *soft-links*.

Besides dynamic linking, pre- and post-conditions support course authors. During course-creation, the connections (directed hard-links) between chunks can be checked for feasibility. Pre- and post-condition tags of succeeding components must be compatible and authors could be warned, if the "knowledge distance" between learning components is too large. During repository maintenance, pre-/post-conditions can be used to support consistent definition of r-links.

3. Profiles and pre-conditions for dynamic generation of e-learning content

The previous section explained the concepts needed for dynamically generating user-specific e-learning courses. Here, we show how these concepts can be realized from both, the learner's and the system's perspective. In order to do so, we have to consider first a usage scenario: Let's assume a learner is interested in how to cope with e-learning content management. In a heterogeneous group of learners, some might want a general introduction and progress in a systematic manner. Others have specific questions and want just an answer to them. Fig. 2 shows a map through the landscape of e-learning content available for any of these learners.

It is obvious that the content exhibited in Fig. 2 can be perused via different paths. However, it cannot be meaningfully studied in random order. Hence, the course author is challenged to provide sequences of chunks (learning strings) that are to be studied linearly, but can be combined in multiple ways. Students can use different entry points to this content. To satisfy these constraints dynamically, learners cannot be confronted with the raw chunks of e-learning content. These chunks or modules have to be rather embedded in an environment, that allows to assess a user profile and to update it along the users learning path. This embedding is sketched in Fig. 2 by the entry "Course-Portal". It serves to tell the system the users learning aspirations as well as the knowledge, the s/he has already acquired beforehand. In the case of our example, this information can be obtained by offering the student in addition to the main entry ("Introduction to Content Management") also two side entries

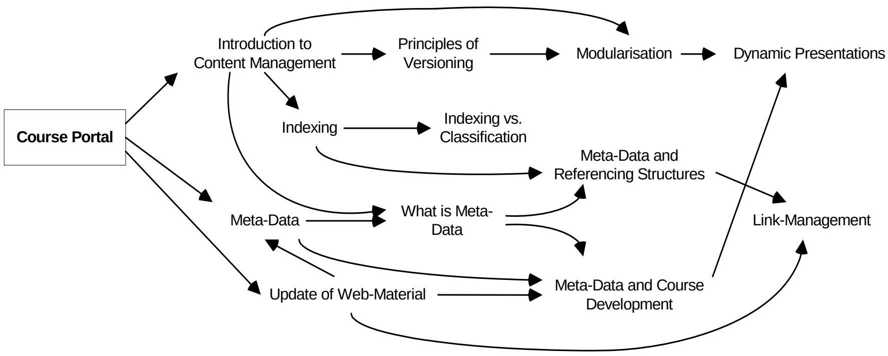

Figure 2: Content structure and relationships between content chunks in a course on Content Management

("Meta-Data", "Updates of Web-Material"). The individual topics (modules) consist of chunks connected by the course-author using directed-links. Hence, the internal modularisation of these entries remains hidden from the learner. The arrows shown in Fig. 2, represent soft-links. I.e. they exhibit dependencies seen by the author of the course and thus they are dependencies implicitly expressed in the pre-/post-condition embedding of the individual modules.

As pre- and post-conditions specify the learners knowledge in relation to the given module, the knowledge increment provided by the module can be computed as the difference between pre- and post-condition. Basically, for each chunk, this increment can be expressed by a single attribute. Hence, for a complete module or learning-string, it will be a list of attributes. Since many chunks will not start from scratch but rather expand on an issue, the progression of knowledge in a certain domain can be indicated on an n-point scale. (The current implementation uses n = 10). Consequently, the user-profile and the pre-/post-condition structure have to be expressed in the same language. Propositional calculus (with fuzzy logic extensions) serves this purpose. Under these assumptions we have the following rules: *PostCond := PreCond   Inc* (with   representing conjunction of the pre-condition with the module's increment such that higher numeric values in the increment override lower numeric attributes in the pre-condition). Likewise, the choice of candidate modules made available to the learner at a given point is initially the portal only. After the questions of the portal have been satisfied, all modules with pre-conditions weaker than the positive user-profile obtained from the interaction with the portal are furnished in the candidate set of modules offered for study at this point. Preference is given to those modules, where the profile contains a negated term that figures positively in the modules increment or where a positive increment in the degree of knowledge can be identified. Thus, one has basically: *Profile   PreCond*.

It should be noted that the modules profile is mainly determined by the pre-conditions of the individual head-chunks of learning strings. As the learner decides for one of these candidate strings, the learners profile is updated in a piecemeal manner by the knowledge increments of the individual chunks studied by the learner. Within a module, the individual content chunks are hard-linked in a linear manner. When a given topic has been covered in such a way, the learner is offered a new chance to decide. Thus, after perusing the "Introduction" s/he might keep on going to "Principles of Versioning", or directly to "Modularisation", as well as to "Indexing" or to "What is Meta-Data". However, if the learner uses rather the problem-oriented entry of "Updates of Web-Material" s/he can next learn about "Meta-Data" or, if s/he indicated in the profile (or at this point) that s/he is conversant about metadata, go directly to "Meta-Data and Course-Development".

How can this be achieved technically? Even while the course has semantically several independent entry points, formally it has a unique entry point with the single purpose of asking the user what s/he expects from the course and what s/he knows already. This entry point offers the user the front level modules, in case of Fig. 2 "Introduction", "Meta-Data", and "Updates of Web-Material". It might also ask some further questions in order to see whether the user is already conversant about other topics dealt with in this course.

Thus an initial dialog might be: "*This course offers you to learn about A, B, and C… . If you know already   you might also go directly to A1 or A2, if you are familiar with   , you might as well study B1, B2 or B3. However, if* *you are only moderately aware of , we recommend that you look at E before going to E1 or E2. …*". It is up to the designer of the portal-page to decide whether these questions are raised as boldly as indicated, or whether a neat graphical interface or an elaborate system of an entry examination leads to the information needed. In any case, a set of concepts indicating positive knowledge results. This set can be considered as a conjunction of concepts known by the learner. Linguistically, these concepts are expressed by words either directly drawn from the vocabulary of terms used for the classification of the module or representing refinements of such terms.

If the initial dialog yields an answer of the form: *"I know A, B, and C. But I want to know X, Y, and Z."* the respective propositional term would be: *A B C (X Y Z).* Assume, the course contains 6 modules with the following specification:

M1:	*pre: A B*	*inc C*	*post: A B C*
M2:	*pre: A C*	*inc E*	*post: A C E*
M3:	*pre: A E*	*inc X*	*post: A E X*
M4:	*pre: B C*	*inc Y*	*post: B C Y*
M5:	*pre: X Y (D,2)*	*inc (D,5) Z*	*post: X Y (D,5) Z*
M6:	*pre: E*	*inc D*	*post: E D*

The user has the choice of immediately studying modules *M4*, *M1*, and *M2* with a certain prioritisation for *M4*, since this module leads directly to a leaning goal specified by the user. *M1* can be perused at any time, but it will be always offered as a low key option, since the user claims to know the concepts (i.e. *C*) contained in *M1* already. Hence, *M2* would be a viable alternative, since it teaches *E*, thus preparing the user to study *M3* thus reaching learning objective *Y*. Knowing *X* and *Y,* however, is insufficient to comprehend *Z*. Thus, before venturing into *M5*, the user has to look at *M6*, and using the concepts acquired in *M2* try to grasp *D* sufficiently to study carefully *M5*, a module that does not only teach *Z* but also raise the students understanding about *D* from an introductory to an intermediate level.

One can see from this example that the precondition structure defines an implicit network that needs no global perspective. The global perspective is reduced to the aspect that the individual modules (or chunks) contribute to the overall learning objective of the course. Thus, the terms used in their pre-/post-conditions (resp. increment) have to be drawn from a relatively narrow knowledge domain. Usually it is a sub-domain of some domain referred to in the general metadata classification section. Adding an additional learning concept to the course thus does not require complex updates. It is just necessary to check, whether the pre-conditions of the new module can be met at all by either initial profile specifications or by profiles updated by knowledge increments taught by other modules. It is to be noted, that the pre-condition structure is not only local to the individual modules (learning strings) but repeats itself at the chunk level. Thus, the concept of pre- and post-conditions can also be used by course builders in finding adequate chunks to be combined to learning strings and thus eventually to presented modules of the course.

4. Conclusion

The paper presents an approach to modularise and dynamically re-link e-learning content. It rests on an extension to the metadata proposed by the LOM standard. The concepts presented are implemented on top of the content management system from HyperWave (Maurer, 1996). The power of the approach rests on a stratification of connection types as well as on a flexible notion of reusable learning modules.

References

Rosenberg, M. J. (2001). *E-learning – Strategies for Delivering Knowledge in the Digital Age*. McGraw-Hill.

Maurer, H. (1996). *HyperWave – The Next Generation Web Solution*. Reading, MA ua. Addison Wesley.

Seeberg, C. et al (2000). *Coherence in Modularly Composed Adaptive Learning Documents*. In Peter Brusilovsky P. et al (eds.): Adaptive Hypermedia and Adaptive Web-Based Systems, LNCS 1892, pp. 375–379. Springer-Verlag.