Abstract

Traditional Computer-Aided Teaching and Learning (CAT/CAL) environments in multimedia-based tele-education do not empower knowledge consumers (trainees, students etc.) to practice content transfer and knowledge acquisition in a self-managed and context-sensitive way. Since such kind of transfer process has to involve both, teachers and learners, it requires the support of different perspectives on material as well as interaction features for self-directed learning and context-sensitive interaction. In order to implement a proper solution, we have developed a didactic scheme for self-managed content transfer. Subsequently, we have embedded that framework into meta-models of course material. In doing so, elements of the didactic framework have been mapped to courseware features. Finally, we designed features for collaboration. As result, a generic architecture for self-managed and context-driven CAT/CAL can be defined. We exemplify our solution through a corresponding development project targeting to support academic education in the field of Business Information Systems.

1. Introduction

The use of traditional CAT/CAL (Computer-Aided Teaching / Computer-Aided Learning) environments per se in the course of distance education does not necessarily lead to an increase in effectiveness and efficiency of transferring and acquiring knowledge actively. One way to actually improve the quality of learning through tele-technology is to couple tele-education with the objectives of self-managed learning. Besides Galileo (“You cannot teach man, you can only help mankind to discover knowledge in and by itself”), modern psychologists, such as Greif [13], notice that self-managed learning processes are more efficient than traditional ones, since they facilitate the start of learning processes and enable the continuous enhancement of existing work processes.

Learning this way leads to sustainable effects, since it leads to conceptual structures that remain more persistent than the ones acquired through traditional learning techniques. In contrast to other learning techniques, in self-managed learning already individually established and well-understood structures lay ground for novel ones.

In our contribution we suggest to migrate the principles and features of self-managed learning with those existing in CAT/CAL environments. So far, these environments have been extended with features stemming from collaborative work, e.g., [3][4], rather than focusing on work spaces for self-organized learning. We address the migration at a fundamental level: We propose to identify a set of generic elements that have to be part of any course material that might be used along self-managed processes. This course material cannot only be navigated in a variety of ways and contain several codalities of information (audio, video, text etc.), but can also become integral part of existing individual knowledge structures.

These developments should help to avoid currently experienced, substantial deficiencies with technologies for tele-education, as, e.g., demonstrated by the National School Network Testbed (http://nsn.bbn.com):

(i) Use of tele-media: Mentoring and supervision via tele-media have not been accepted by learners.

(ii) Knowledge-transfer support: Existing technologies, such as Internet- or Web-based tools, do neither support knowledge transfer for individuals nor groups effectively or efficiently.

(iii) Material preparation: Due to the lack of proper support features the production of material is a time-consuming procedure.

(iv) Added value: Overall, participants in tele-education so far experienced little added value when using electronic material and related features compared to conventional teaching / learning settings.
In the paper we first discuss related work (section 2). Then we detail the concepts of self-management according to our experiences in academic education and develop a framework for knowledge acquisition – focussing on a learner-centered perspective in tele-education (section 3). The developed framework allows to specify requirements for providing online support capturing both, different roles in the process of knowledge transfer, and different views on knowledge items (section 4). Meeting these requirements in a generic way requires to develop a corresponding architecture for CAT/CAL environments as well as XML specifications for course material (section 5). This solution demonstrates that certain generic elements have to become part of course material, as soon as learners may have self-control over the learning process and are to be directed towards the intended goal(s) of the knowledge-transfer process.

2. Related Work

Learning "on demand", self-managed learning and self-determined control of the learning process, such as claimed by Schoop et al., can naturally be met by hyper-text courseware [26]. "But it is still not clear how to design the hypertext documents for sustainable, informational benefits... It is very difficult to find didactical founded concepts for integrative, media-comprehensive offering of networked electronic documents." [26] Although Schoop et al. themselves have developed a XML-structure for building courseware, they only took advantage of XML with respect to a hyperlink structure, rather than developing an enabling structure for the support of didactically grounded integrative knowledge transfer processes.

Süß et al. [27][28] follow a meta-modeling approach to support the development of adaptive hypermedia-based electronic teachware. This approach focuses on document structures and navigational services. The architecture comprises several distinct levels: The bottom layer, termed real world layer, consists of elements concerning the subject to be taught. The superior layer, termed hypermedia layer, consists of hypermedia documents, including access and navigation aspects, describing the given domain. Domain-specific models describe the content and navigation structure of course materials or, in general terms, of hypermedia, of a given domain of application. Finally, the common abstract meta-model is located at the top level [27].

These models are specified using the LMML (Learning Material Markup Language) which is based on XML [27]. The didactic background, described through so-called conceptual units [28], is limited to the use of hypermedia. Other concepts, such as collaboration along the knowledge transfer process, individualization of material, or mutual feedback are not captured.

The Educational Markup Language (EML) [21][22], developed at the Open University of the Netherlands, is based on architectural and pedagogical meta-models. The pedagogical meta-model contains 4 packages [21]:

1. The learning model which describes how learners acquire knowledge and has been constructed based on several commonalities in learning theories.

2. The unit-of-study model that describes how knowledge items (as they are applicable in practice) look like, given a learning model and a model for instruction.

3. The domain model capturing the type of content and the organization of that content, addressing domains like economies, law, or biology.

4. Theories of learning and instruction comprising principles and models of instruction as they are described in literature or as they are conceived by teachers.

The architectural reference model explained in [21] describes the phases of developing technological solutions, the publishing and delivery of material, as well as the embodiment of roles and states into the phases of development. For instance, "developers work with the development environment. The development environment may consist of design tools, authoring tools and a content management system." Publishing might occur at a static and a dynamic layer. Hence, the EML code can be interpreted both ways.

It has to be noted that the "unit of study" concept is central in this work. It is the smallest unit providing learning events for learners, satisfying one or more interrelated learning objectives. The authors define requirements for units-of-study, which are kept at the level of general objectives to be met compared to the requirements derived from our SMAC® learning framework (see also section 3). Some of the listed requirements are: (1) Formalization, (2) pedagogical flexibility, (3) explicitly typed learning objects, (4) completeness, (5) reproducibility, (6) personalization, (7) medium neutrality, (8) interoperability and sustainability, (9) compatibility, (10) reusability and (11) life cycle. Most of the requirements can be met by an adequate use of XML (1, 2, 3, 4, 5, 7, 9, 10, 11).

Finally, the delivery platform for units-of-study can consist of one or more media. This property acknowledges multimedia solutions in the domain of tele-education.

In comparison to our work the requirements for the Educational Markup Language solution do not refer to an educational learning framework the same way as the SMAC® framework given in section 3. The learning framework of the EML is located at a different level of abstraction than SMAC®. Consequently, the EML requirements and proposed features are also provided at another level of abstraction. Since the EML requirements are located at a more general level, the SMAC® (derived) elements can be checked against the EML elements for
completeness. It turned out that the EML requirements can be met by the notational system presented in section 5.

The pedagogical learning framework represents a metamodel for pedagogical models. As such, it does not provide any concrete instructional procedures. In contrast to EML we provide elements for instructional procedures through the Scholion Markup Language SchoML as well as through the Profile Markup Language ProML.

SchoML and ProML (see section 5), in comparison to other approaches using XML, allow to specify the basic didactics for a domain, e.g., for teaching foundations of Business Information Systems. This specification has enabled through deriving the didactical framework as given in section 3 by abstracting from a certain domain of education (in our case Business Information Systems).

In addition to the underlying framework, the production, authoring, or development process of the courseware is of central importance, since it requires didactical grounding as well as a reference to learning theories. With respect to the latter, according to C. Argyris [1][2], both, single- and double-loop learning should be supported in the course of learning. The work performed in the context of EML supports only single-loop learning processes, since it does not explicitly support de-contextualization and re-contextualization within a reflection loop. Reflection in that context denotes the conscious reflection of newly acquired content. This type of reflection is explicitly supported by SchoML and ProML.

3. SMAC®-Learning Framework

We have developed a framework for academic teaching which is based on Körner’s “Aufweisanalyse and Ersetzungsanalyse” [16]. The process of De-Contextualization of the framework is comparable to the “Aufweisanalyse”, Theoretical Understanding to “Normierung” and the process of Re-Contextualization denotes the “Ersetzungsanalyse”. The model describes the knowledge acquisition process as a self-controlled/self-managed procedure (see Figure 1).

The framework instantiates Argyris’ concept of single- and double-loop learning [1][2][12][29], since the Reflection and iterative (Re-) Contextualization correspond to the idea of double-loop learning, whereas the other processes implement single-loop learning. The framework also captures Nonaka et al.’s [23] four modes of knowledge conversion, as shown in the center of Figure 1: Internalization, Externalization, Socialization and Combination. Finally, the OADI cycle of individual learning introduced by Kim [19] (observe, assess, design and implement) has been taken into account (see also the center of Figure 1).

A learning process as shown in Figure 1 can be initiated by the learner in a self-managed way [13], as denoted in the figure by self-managed knowledge acquisition process. Knowledge acquisition processes can also be triggered, namely, in case of task-specific knowledge needs.

Operational knowledge for performing actions and tasks is provided by Skills transfer. Skills are considered to be of central importance for each individual action. Skills are synthesized from other categories of knowledge: Firstly, to turn Existing Knowledge into Skills the former is internalized (Internalization [23]) as task-specific operational instructions. Existing Knowledge is based on individual cases rather than abstract specifications. Parts of the Existing Knowledge diffuse to the explicit Background Knowledge which captures part of the Existing Knowledge. The Skills include practical experiences which are also supplied by implicit Background Knowledge. Explicit Background Knowledge is transformed through Transformation & Transfer to implicit Background Knowledge if an activity is repeatedly and practically trained. Experiences from Applications are released from implicit Background Knowledge in a process of Socialization [23]. Socialization describes the process of gaining collaborative Experiences from Applications. By means of Externalization [23] new Information is turned into explicit Background Knowledge. The latter is combined with Existing Knowledge (through Combination [23]). According to Argyris [1][2] the above described process of knowledge acquisition can be interpreted as single-loop learning.

The double-loop learning process addresses another form of knowledge acquisition. Its origin is the field of organizational learning. In order to acquire skills for task accomplishment it follows an extended acquisition loop. Existing Knowledge which is bound to a specific context has to be abstracted and put onto a generic level by the process of De-Contextualization (denoting the generalization of knowledge). In learning practice, this process is the most difficult one and thus, has to be coached by an instructor. Theoretical Understanding can not be “produced” by introducing theoretical contents. It has rather to “build up itself”, since it can be considered as “abstract modeling” of content. As such, it is established step by step throughout the entire learning process (as shown through the unshaped area in Figure 1). Instructors or features of learning environments can only initiate this process, since it is an individual effort to be undertaken continuously along the learning procedure.

The process of Re-Contextualization has been derived from Körner’s “Ersetzungsanalyse” [16]. It denotes the process of shifting theoretical knowledge (in the model termed Theoretical Understanding) from one context to another one. Consequently, Skills enable the execution of novel tasks. In the course of applying knowledge new Experiences can be gained from the Skills. This application also might initiate a Reflection whether the acquired knowledge is suitable. Reflection is considered as a conscious process which might lead to finding a new context for already performed activities. In the course of this proc-
New knowledge might be generated, which step by step migrates (termed Condensation) into Existing Knowledge.

---

**Theoretical Understanding**

- **Externalization**
- **Internalization**
- **Transformation & Transfer**

**Notation:**

- **Type of Knowledge**
  - Existing Knowledge
  - New Knowledge

- **Process**
  - Externalization

**Figure 1: The SMAC® Learning Framework for Knowledge Acquisition**

A framework per se does not indicate or guarantee (in case of leading to technological development) that individual trainees are able to acquire knowledge in a self-organized way. It rather serves as a frame of reference, denoting essential types of knowledge and transitions that are part of self-management in transferring knowledge from material and/or teachers to learners. Consequently, the SMAC® learning framework only works properly, if the learners engage themselves along the transfer process. As an example, the topic “distributed systems” taught as a part of the course “communication systems” (at our department) can be instantiated for the SMAC® Learning Framework as follows: In the professor’s lecture theoretical foundations about the design of distributed systems, fundamentals of networking and concepts for application development are explained and exemplified. In order to make students understand how these concepts work, the concepts are instantiated by UML-modeling methods, Java and Oracle design and programming features in the course of practical exercises. Hereby the newly acquired concepts are trained in a spiral process – including the entire “single loop” process. In order to deepen the understanding of the
concepts and their application, a reflection process has to be occur. Therefore, specific tutorials and courses dealing with selected aspects of distributed computing are held. In these course, students have to cross-examine existing techniques as well as to discuss practical approaches in light of concepts and theories. This way, they are supposed to shift knowledge from a particular context either to a higher level of abstraction or to another context of application. Additionally, dedicated student projects aim to train the acquired knowledge and to fix Skills and Existing Knowledge. Finally, the work on a diploma thesis requires to pass through an entire “double loop” process, building upon the already acquired knowledge.

4. Requirements

The learning framework in Figure 1 describes a wide range of possibilities for learners to acquire knowledge and to engage into the process of transfer in a self-controlled way. From the perspective of an instructor, only the processes for knowledge acquisition can be initiated and the results can be evaluated. The same holds for the structure of course materials. Thereby, the processes of the “singe-loop” – (Transformation and Transfer, Internalization, Socialization, Externalization and Combination of Knowledge) are internal processes in the learner’s mind. Consequently, an instructor cannot support these processes actively. These processes are triggered for each activity as they are required for accomplishing tasks properly.

The main “double-loop” processes – De-Contextualization, Re-Contextualization and Reflection (Extension is considered as an initial process for the Reflection, and Condensation is an internal process) – have to be supported through proper design of course-material structures. In the SCHOLION-project [3][4][5] the requirements to establish “double-loop” processes have been identified. They will be listed in the context of the framework. Note that all requirements are marked with brackets [x] and a letter to allow direct references later on (see section 5).

De-Contextualization: Most hypermedia-based distance-learning environments have lecture character. As such, they support the process of De-Contextualization more accurate than the other ones (Re-Contextualization and Reflection).

- [a] Hypermedia-based learning materials are considered to be an effective way to enable knowledge transfer to the learners [10][15]. This statement is strongly coupled with enabling
- [b] Individualization of the content, which means enabling customization of the learning material to learner needs with respect to mental maps, networked thinking and information linking [7][9]. Enabling feature for the learning material could be hypertext enriched with multimedia elements. Features to implement individualization [3] can be:
  - [b1] Marking implemented through underlining and highlighting of text,
  - [b2] Individual enrichment implemented through textual and multimedia annotations,
  - [b3] Handling of personal annotations enabled through storing all annotations in so called “profiles”, which can be manipulated by the user,
  - [b4] Link management implemented through internal and external links for customizing the courseware or enabling networked thinking, respectively, and
  - [b5] Cascaded Profiling which can be implemented through setting profiles (containing the markups) that are publicly available for other learners or teachers. Consequently, other users can lay profiles on top of the course-material and can walk through the content with another perspective on the content. In case a learner likes another profile, he/she might copy it to his/her private ones and continue to work with this profile. Some of these aspects can be compared to shared global workspaces, as mentioned in [14][20].

- [c] Mutual reflection of the content through discussing the content, not only among learners, but also between students and teachers [25]. Enabling feature could be a Discussion Board [4].
- [d] Various navigation possibilities as proposed in [5] enabled through:
  - [d1] Sequential navigation through the courseware, comparable to “turning the pages” in a book
  - [d2] Hypertext navigation via internal and external links
  - [d3] Direct steps implemented through permanent links to each page/slide of the course-material from a predefined entry (focal point), such as the menu bar
  - [d4] Index or keywords to be set by the teacher (public keywords) or by the learner (private keywords) for guidance, enabling fast access to crucial passages in the material
  - [d5] History function laid over the four navigation concepts mentioned before

- [e] Courseware-specific help as a reply/anticipation to frequently asked questions brought up in the course of (group) discussions.
- [f] Learning from others, implemented through dedicated mechanisms, such as (cascaded) profiling, and linking underlying structures (profiles, contributions to the discussion board).
- [g] Feedback and quality management, implemented through self-assessment tests for feeding back
learning-progress to the learner him/herself, or through external assessment controlled by the coach/instructor.

- [h] Building of virtual learning teams [14][20] for mutual reflection and walking through content together.

Re-Contextualization is to put theoretical understanding to practice through exercising exercise task accomplishment procedures. Incentives for learners can be set through

- [i] Practical samples well known from teaching chemistry or physics, heavily using multimedia content.

- [j] Discussions about individual experiences – they can also learning Skills [25].

- [k] Mutual support procedures: Handling joint problems is considered widely valuable [25][20]. Cooperative learning does not only mean to act as or within a group but also to mutually support other learners, to provide feedback, and to participate in discussion.

- [l] Exercises including solutions.

Reflection, in the sense of finding new contexts for existing Skills, is traditionally not supported by conventional CAT/CAL tools. It could be implemented by features to discuss activities or operations using complete discussion-contributions in the form of multimedia presentations [m], showing related contents [n] and use cases [o] as presented in scientific papers or other material. Another implementation could be based on self-assessment procedures, such as multiple-choice tests [p]. However, these indirect ways of reflection lack activity-orientation [q] when self-assessing personal skills, and thus, are of crucial importance for the design of course-material structures.

5. Architecture and Document Structure

In the following, we design a XML structure that meets most of the requirements listed in section 4. The document structure introduced in the following has been designed for the domain of educating the principles of business information systems, and has to be considered as a domain model.

5.1 A Quest for XML

The Extensible Markup Language (XML) [11] is the universal format for structured documents and data of the Web. It meets several demands, that have also been formulated as design goals (as given in [6]). XML can be used straightforwardly over the Internet and supports a variety of applications. It facilitates the implementation of applications that process XML documents. XML documents are legible for a variety of users, in particular, non-programmers, since they provide reasonably clear syntax and semantics. Software engineers are able to design and create XML documents quickly. With respect to education, XML supports additional features, according to [26]:

- Separation of structure, content and layout, hence,
- Preparation of a source-file containing different media (paper, CD-Roms, Internet) to use it in an integrated learning environment,
- Access and selection of arbitrary document-modules or document-positions.

The latter enables the

- Support of content-specific search in the courseware and
- Support of versioning or multilingualism.

We consider further advantages of using XML to be relevant for CAT/CAL developments:

- A single XML source file containing instructional content can be transformed into multiple digital formats, such as Adobe’s PDF, HTML (primarily for web-publication), wireless display languages (e.g. WML 2 standard) or Open eBook Publication Structure [24] which is also XML-based, but also into other XML structures by XSL and XSLT technology.
- Existing accurate software support, primarily for Java and C++ such as XML4J-Parser (IBM’s alphasworks®), JAXP (SUN®), XERCES Parser (Apache), MSxml (Microsoft®) or XSLT-Processors from IBM, Apache or SUN.
- XML itself is platform-independent.

5.1.1 Courseware Architecture

Several different structural approaches have been proposed. In contrast to them we split the courseware conceptually into a course structure (SchoML) containing all static instructional content, and into a structure containing all dynamic data stored in a user profile (ProML). The work of Karjalainen [17,18] has primarily influenced this approach, since it has to be considered to be one of the most comprehensive structures and most suitable for supporting the implementation of frameworks for self-managed learning. Consequently, some parts of her structure, as described in [17][18], have been adopted to the static Scholion Markup Language (SchoML) with her agreement. The fundamental parts according to the learning framework described in section 3 and to our experience with the teleteaching-environment Scholion [5] are new.

Although the idea of user profiles is not new (cf., e.g., [8][14][20]), the idea to save annotations in profiles is novel. Hence, an arbitrary number of courseware-depending profiles (containing the annotations) can be integrated into a single user-profile, and merged into the courseware for displaying it at the client side (see Figure
2). Note: The requirements defined in the last section are merged into the content (brackets [x] and letters).

The course material (based on SchoML) contains all static courseware data. It is stored into a database, containing all course materials, all user data and discussions. The user profiles [b] (based on ProML) are also stored in the database and represent the dynamic part of the courseware. For displaying the courseware at the client side, the course material and the course profile are merged with a DOM-Parser, and subsequently converted to HTML [a] through XSL (eXtensible Stylesheet Language) processing. Consequently, the HTML-pages can be presented in a traditional HTML browser [d1] and need no specific software. Media files are not loaded to the client automatically, but are represented in the course material as link references into the database which are loaded only if needed. A media file can also be loaded into the database from the client side in the case of annotation(s). Discussions [c][j][m] are dynamically created and therefore, not part of the course material itself. In order to enable references into the course material, each contribution to the discussion board requires an own static ID [3].

5.2 Scholion Markup Language (SchoML)

The Scholion Markup Language - DTD and the Profile Markup Language - DTD were designed with XMLSpy Version 3.5 [30]. In the following the SchoML is described. The notation of the figures and the figures stem from the “Schema Design View” of XMLSpy 3.5. The attribute lists and the attributes are not shown in this simplified view.

![Figure 3: SchoML basic structure level](image)

In Figure 3, the Coursematerial acts as the root element of the model. The main level of the model includes general information on the course (Geninfo), educational guidance (Eduguide), meta information (Metainfo) and the Cont-modules. It that part of a course material where the actual learning content resides (see also [18]).

5.2.1 General-, Meta- and Guidance-Information

The Geninfo captures some fundamental information about the course material, such as the Name and the Code of the level of the course. Compinfo can provide information about comparable or similar courses. Schedule includes term information – such as “chat hour with the instructor” or question time. In the Effort-tag the amount of learnable content and the effort for learning are estimated. Information about any official degree or diploma achievable through the course is included in the Apprinfo.

Most of the tags in the XML structure contain Hypertext [a][d2] at the lowest level. The Hypertext-tag describes the granular structure of tables, text, media [i], links and structuring information required for describing difficult content. We distinguish simple from complex Hypertext for the different tasks, differentiated through the complexity of the underlying tag set.

![Figure 4: General Information (GENINFO)](image)

Some educational guidance through the course is provided through the Eduguide-properties. The Proceedorder recommends an order when learning content. Some contact information is provided in the Contact-tag. The corresponding information has to be generated from the database, because it is not only static, but also needed at the discussion forum. Some additional information can be placed at Addinfo. Trainingguide contains didactical training techniques for particular Modules (or more elementary elements) which can be linked directly through the Hyper-
text-structure. Trainingtechs provides guidance in technical matters, such as suggested software or other tools to be used.

Figure 5: Educational Guidance (EDUGUIDE)

Metainfo provides some meta information about a course. It contains additional information, such as Objectives fragmented to LearningObjectives and Motivation of the course, Contentkeywords describing the outlines of the course, a Modulelist generated from the modules in the ContModules and an Abstract outlining the course.

Figure 6: Metainformation (METAINFO)

5.2.2 The Content for Learning

The actual content is described in the content modules (ContModules). It contains a Modulelist (already described above), the Module itself, a Keywordlist and an Assessmentsblock which are termed Module.

Figure 7: Content Modules (CONTMODULES)

Moduleinfo contains some general information about the module and is described separately. The Directory provides a list with links to all mentioned Chapters in the Module. The Abstract again outlines the content of the module. The Summary provides another kind of abstraction of the content, pointing out the most essential parts of the content and summarizing findings. The Assessmentblock contains Selfassessments for the learner represented by Multiplechoicequestions with one Question and an arbitrary number of Answers (of course one or more right answers) for each question. Externalassessment by the instructor may be delivered through Multiplechoicequestions or through Openquestions, enriched with a reference to the instructor for correction and evaluation. The Keywordlist contains a list of Keywords, referring to elements of titles or directly to the content.

Figure 8: Module (MODULE)

The Moduleinfo contains general information such as the Modulename, the Author and the Moduledefinition or the Objectives, but also some instructional guidance in Effort and Learningtech (at the level of chapters). Order suggests an order of how to learn the module(s).

Figure 9: Module Information (MODULEINFO)

Chapters are composed of Abstracts, Slides and Helpslides. The hierarchy of the Slides and of the Helpslides is envisaged to be constructed by the use of an attribute list defining the level of the Slide and the level of the SlideHead. From the view of granularity, a slide can be compared to a specific page of a book and should be displayed as a separate unit for sequential navigation at the client. Each Slide owns public Keywords and each Slide as well as each Module, Chapter or Para is equipped with an unique id for enabling direct navigation steps. Exercise and Examples open the possibility to complete a theoretical part in the Paragraphs with explanatory Examples (either in textual form or in another codality) and/or practical Exercises to be performed by the learner. Paragraphs either consist of Media exclusively or are represented by Paratext, which is represented by Hypermedia. Some deeper reflection of the learned and prac-
ticed content can be provided through Reflecting literature [n][o][q], applied through links to www or media-links (e.g. to pdf-files). All these operational and reflecting tasks are supported through interaction between the students and the instructor using the discussion board. Anticipating frequently asked questions in the discussion board, the static character of the structure has been opened and a new Helpslide [e] explaining content can be created.

5.3 Profile Markup Language (ProML)

The tree displayed in Figure 11 shows the DTD element structure of the ProML-DTD, also adopted from the schema design view in XMLSpy 3.5. The attribute lists and attributes are not displayed in this simplified view.

A UserProfile [b] can be owned by a single user or by a user-group [f][b][k]. It contains, besides the date of creation, a name and an ID, UserDependent- and CourseDependent information. UserDependent information abstracts from the courseware – it is laid over all course materials and contains, for instance Bookmark-Groups containing Bookmarks (which refer to course material entries). A user can have GroupMembership [f][b][k] in arbitrary UserGroups. The latter refer to group-discussions and to specific content parts at the Module- or Chapter-level. Additionally, the UserDependent profile-information contains all history information (CourseHistory) logged from learning processes which facilitates history-navigation [d5] on the one hand. On the other hand a static evaluation concerning learning duration (stored as an attribute) for each slide and of course can be extrapolated to Module or Chapter level. Thirdly, the history enables the reconstruction of all learned content of a user, valuable for resuming this content and putting it onto a CD-ROM or into a pdf-file with all annotations for future use. This information can be used for CourseProfiles and Keywords [d4] that are considered to be CourseDependent information. A CourseDependent element always belongs to a specific course material. Keywords are text-dependent references into the textual part of a course material. CourseProfiles contain Annotations [b2] which can be Text Annotations (hidden or directly displayed textual annotations in the context of the material), Links [b4] (internal ones refer to id’s in the course material or external ones referring to www-URLs), Marks [b1] (highlighting or underlining text), DiscussionLinks [f] (links directly from the content to suitable discussion contributions in the discussion forum – represented in the course material as a symbol-link) and MediaLinks (Links to each type of Media such as video, audio or images – represented in the course material as a symbol-link). CourseProfiles can be public (available for other users) or private [b5][f]. Storing all personal information in a profile enables the individual modification of each entry [b3].

Figure 10: Chapter Level (CHAPTER)

Figure 11: Profile Markup Language (ProML)
6. Conclusion

Since traditional CAT/CAL environments do not lead to an increase in effectiveness and efficiency of transferring and acquiring knowledge actively, fundamentally didactic concepts to overcome these deficiencies are required. Based on learning frameworks such as SMAC® novel development can be grounded on generic domain ontologies. These meta-representations allow to design activity support for effective knowledge transfer and acquisition, relying on common (domain) structures. The SMAC® learning framework indicates how self-managed learning and collaborative teaching have to be designed. It has guided the development of a corresponding environment for tele-education.

The courseware structure we have developed and presented in this paper implements fundamental, consolidated requirements. However, the list of requirements has to be completed by inspecting and evaluating several other and partially consolidated learning environments, as well as exploring several domains. Furthermore we will try to extract a domain-independent courseware structure developed top-down based on the SMAC® Learning Framework and bottom-up integrating our experience from the domain of Business Information Systems.

Acknowledgements. We have to thank Anna Karlajainen and Rainer Born for their valuable inputs and reflections.

7. References

[28] Süß, C.; Freitag, B.: Entwicklung und Nutzung von Teachware, Das Passauer Knowledge Management System (PaKnMaS); In: Rudolf Kammerl (Hrsg.), Computerunterstütztes Lernen, München, Oldenburg, 2000


http://www.eml.ou.nl/introduction/articles.htm

NATO ASI Series F, Vol. 67 Springer Verlag, Berlin/Heidelberg 1990, p. 3-25

http://www.w3.org/XML/downloaded 25th of May 2001

http://www.w3.org/TR/2000/REC-xml-20001006

http://www.openebook.org/; downloaded 25th of May 2001

http://eml.ou.nl/introduction/articles.html; downloaded 12th of September 2001


http://www.xmlspy.com; downloaded 20th of May 2001

http://www.w3.org/XML/