A Teachware Management Framework for Multiple Teaching Strategies

Christian Süß, Fakultät für Mathematik u. Informatik (suess@fmi.uni-passau.de)

Rudolf Kammerl, Lehrstuhl für Allgemeine Pädagogik (kammer01@fsuni.rz.uni-passau.de)

Burkhard Freitag, Fakultät für Mathematik u. Informatik (freitag@fmi.uni-passau.de)

Universität Passau,
94030 Passau, Germany

Abstract: In this paper we argue that effective teaching is a multi-method process and propose a teachware management framework flexible enough to support multiple teaching strategies. We model e.g. behaviouristic or constructivistic strategies based on a metamodeling approach to learning material. As the common implementation language we introduce the XML-based Learning Material Structure Markup Language which also facilitates the interchange and portability of learning material. Our teachware management framework has already proved its advantages from both, the author's and the learner's perspective.

1. Introduction

Learning processes depend on the individual properties of learners like age, level of experience, knowledge and interest which usally span a wide range. Thus, there is no best teaching strategy that applies to every learning situation. In fact, even for a single learner the best strategy may change over time. Students taking an introductory course on database theory, for instance, should be guided through the material whereas later on they may want to find out on their own about the SQL query language perhaps coached by a human tutor. As another example, students taking a course on the history of music may at one time wish to learn about operas by following a timeline-based tour through the musical epochs while at another time they may prefer to have more or less unstructured material on composers and their works available and to construct an individual map of the world of opera.

Apparently, teachware management systems or computer managed instruction (CMI) systems should support multiple teaching strategies. Moreover, as the examples above show, they should provide an appropriate data model, suitable abstraction mechanisms, and tools not only to authors but also to learners. It is obvious, that a flexible multi-strategy support and powerful application interface both require system properties that are beyond most traditional hypermedia systems and common techniques and tools for computer based teaching.

In this paper, we show how the required structural properties and multi-strategy support can be largely achieved in the PaKMaS teachware management framework (PaKMaS 2000) which is based on a meta-modeling approach (Süß et al. 1999) providing a flexible and adaptable data model and on the XML-based Learning Material Markup Language (LMML). Some of the key features, of course, are a simple yet powerful component model, repository techniques that exploit the meta-data available, the strict separation of contents from structure as well as the capability of adaptive presentation and navigation.

The rest of the paper is organized as follows: In section 2, we have a look at the three most important learning theories and emphasize the need to apply multiple teaching strategies in teachware managment systems. In section 3, we discuss our meta-modeling approach focussing on the concepts needed for the modeling of learning strategies. We also introduce a sublanguage of LMML, the Learning Material Structure Markup Language (LMSML). The benefits of the teachware management framework based on the meta-modeling approach and the LMML for authoring and application of different strategies are discussed in section 4. A comparison with related work can be found in section 5. The paper is concluded with a summary in section 6.

2. Learning theories and teaching strategies

Due to the heterogeneity of the goals aimed at, practical arrangements for teaching and learning are influenced by various didactical and learn-theoretical considerations. Consequently, teachware appreciating this variety of approaches and thus satisfying the requirements of practical teaching has to account for and give support to different learn-theoretical foundations. The latter can be classified into behaviouristic, cognitivistic and constructivistic theories.

2.1. Behaviourism

The behaviouristic approach is based on the assumption that behaviours can be steered by external stimuli. Certain stimuli (S) cause related behavioural responses (R). Stimulus-response pairs (S-R pairs) can be chained and become habits. Teaching goals are to be achieved by presenting selected information and questions or exercises as stimuli, that are expected to cause a certain behaviour, to the learner. If the learner reacts in the expected way his or her behavioural response will be confirmed. The same principle applies to both simple and complex teaching goals. However, in the case of complex goals the entire process is broken into a collection of simpler steps which are normally ordered sequentially. The behaviouristic learning theory is chiefly materialized as a learning strategy by guided tours or linear presentations using multiple choice questionnaires or drill & practice modules. Teaching machines and the model of programmed instruction are early realizations of behaviourism (Skinner 1958). The basic principle of these approaches is the presentation of information followed by a collection of questions and according feed-back to the learner. Authoring systems (Chambers et al. 1980) support the development of behaviouristic teaching material. The overall teaching goal is the acquisition of factual knowledge.

2.2. Cognitivism

Cognitivism which relates to the theories of Piaget (Piaget 1977) and Bruner (Bruner 1966) considers the learner as an individual that processes external stimuli actively and independently and that can therefore not be steered by external impulses. As opposed to the behaviouristic view, the cognitivists believe that learning is based on cognitive structures. It is assumed that the learner perceives, interprets and processes impressions in a selective way according to his or her level of experience and personal evolution. The latter is characterized by the total of perception, comprehension and processing patterns and schemata that are at the command of the individual and constitute his or her cognitive structure.

Therefore the question what processes underly the interaction of learning material as the external condition of learning on the one hand and cognitive structure as the internal condition of learning on the other hand receives special attention. Within the cognition-theoretical framework there exist various different positions concerning (among others) the question whether the overall goal should be the construction of a knowledge structure or rather the development of a general problem solving ability or intellectual and moral capabilities, the practical means in both cases being hints and problems in simulations or case studies as appropriate to the individual cognitive structure. As another question it has been discussed how the learning material and related information is processed by the human brain and how content can be adapted accordingly, e.g. by relating notions to their conceptual environment or by mixing visual and textual presentations. The cognitivistic learning theory together with behaviouristic ideas has influenced the Instructional Design (Merrill 1991). However, essential aspects received attention only in connection with constructivistic approaches.

2.3. Constructivism

From the perspective of constructivism the potential of teaching media to steer learning processes is considered even lower than from the cognitivistic point of view. The active processing of external impulses is rated even higher. The learner is assumed to construct reality individually based on his or her subjective experience structures (Maturana et al. 1984). Although constructivism is an offspring of cognitivism it refuses the latter's objectivism (Jonasson 1991). Constructivism considers knowledge not as a reflection of an external reality but rather as a function of perceptual processes. From this point of view learning material is to consist just of collections of information units and tools that form the input of the learning processes built by the individual. It is assumed that learning material can neither control the learning processes nor provide any reflection of reality. Every act of perception creates a new world (Maturana et al. 1984). As one possible realization of the constructivistic approach, learning environments can be provided that support the activity-based, explorative interaction with the environment. Three approaches to computerized learning environment are rooted in constructivism:

1. Cognitive tools are a collection of software programs that use the control capabilities of the computer to amplify, extend, or enhance human cognition (Kozma 1987). Cognitive tools support the learner in the formation of his or her own concepts. The learning environment is therefore formed and modeled by the learner.

2. Learning is understood as a communicative activity in knowledge communities. The construction of reality requires the interaction with the social environment, too. Communication, e.g. in networks, should therefore be supported (Brown 1985).

3. Learning is viewed as coaching based on a master-apprentice relationship (cognitive apprenticeship). The master offers advice but gives the apprentice more and more independence when he or she makes progress (Collins et al. 1989)

2.4. Multiple teaching strategies

Courses and seminars normally contain components belonging to different learn-theoretical positions. We are convinced that all three approaches sketched above should be taken into consideration. However, which learning theory is best suited to support learning processes depends on the teaching goals and the individual properties of the learner, i.e. level of experience and education, age, learning habits. To teach freshmen the foundations of database systems, for instance, could possibly be accomplished best by a guided tour, whereas more experienced students may be better served with an interface to a real database system together with hints, comments, documentation, and a facility to contact the responsible coach, e.g. by email. Consequently, a teachware management system should have the capability to support different teaching strategies. These can be materialized in the form of behaviouristic components like guided tours with multiple-choice questions. Likewise, a constructivistic strategy could be provided by a collection of informative components and tools that allow the learner to construct or change the learning environment, in a certain way being an author him- or herself. Besides, a coaching support and facilities for human communication should be provided. Most CMI systems implement only one single teaching strategy. To overcome this restriction a suitable modeling and implementation of multiple strategies are needed as well as an appropriate toolkit for authors and learners. Of course, a CMI system can provide means to develop and use teaching material. However, its success is largely determined by the knowledge and competence of the author.

3. A teachware management framework

3.1 A metamodeling approach to hypermedia teachware

We have developed a meta-modeling approach to hypermedia teachware (Süß et al. 1999), which is flexible enough to let the teachware management system PaKMaS support multiple teaching strategies (see section 2.4.). The bottom layer of our meta-modeling architecture consists of the subjects to be taught or learned. One layer up, hypermedia learning material are describing the given domain of application, e.g. database theory or the world of opera. They use domain-specific means, which are specified in domain-specific models and describe the conceptual content as well as the modular structure of learning material and different teaching strategies of a given domain of application. Finally, the common abstract meta-model specifies learning material in an abstract way. This includes aspects also of teaching strategy by specifying how both, authors and learners, can form learning material in general. Learning material consists of learning *modules* that are either *basic modules* or *structure modules* containing other modules. The former are the terminal nodes of a polyhierarchical hypermedia structure similar to directories or books, but allowing *modules* to belong to more than one super module. Indeed, each module has to be contained in at least one structure module. By composition, they dynamically adapt learning material to the needs of different groups of learners at the content level and realize the different teaching strategies of section 2 using appropriate meta-data. Basic modules, however, are the smallest self-contained units or building blocks of the modular structure of learning material. Representing conceptual units, they contain various content objects, which can be behaviouristic, e.g. drill & practice instructions and multiple choice questions, or constructivistic, e.g. coaching remarks, simulations or interfaces to real world systems.

3.2. Learning Material Markup Language

To represent learning material incorporating different teaching strategies, we need a data format, which is suitable for storing the content and structure of the material and which allows to access the stored modules or data, very much like objects in a database system. Moreover also the authors and learners should find it easy to read and use the information stored in the modules. As described in (Süß et al. 1999) we have developed a XML representation for course content and structure representation and interchange: the Learning Material Markup Language (LMML). It comprises two sublanguages, the Learning Material Content Markup Language, which is a syntactical representation of the conceptual content and the Learning Material Structure Markup Language, which represents the modular structure model. To implement multiple teaching strategies, it uses special elements, e.g. *guidedTour* or *collection*, which are declared in the Document Type Definition (DTD) (PaKMaS 2000) and which have strategy attributes with possible constant values *beh* (behaviouristic) or *con* (constructivistic) respectively. On the one hand, the use of the XML-based LMML faciliates the interchange and portability of learning material (Süß et al. 1999). On the other hand, using metadata like those mentionend above, a teachware management system like PaKMaS supports authors wanting to create learning material using multiple teaching strategies as well as learners using the presented material.

4. Supporting multiple strategies

4.1. Behaviouristic strategies

Within our framework, teaching material can be presented hierarchically using a book metapher with chapters and sections as *structure modules*. They can also be presented sequentially starting at a first *basic module* using a guided tour metapher (Fig. 3). Both represent behaviouristic strategies where the author explicitly plots the route through the given learning material.

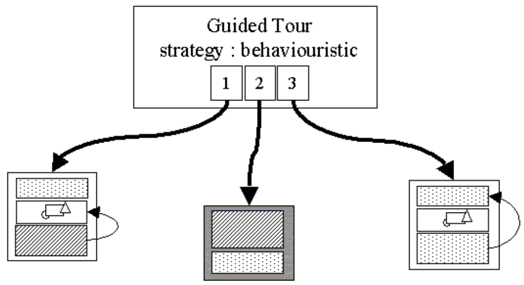

Figure 3: Behaviouristic strategy using a guided tour

If an author, for example, wants to create a section of a booklike course or a guided tour, she can create new behaviouristic *basic modules* using *course units* defined in LMCML which can be extended e.g. by *question* elements from the Tutorial Modelling Language (TML) (Brickley 1999). She also can query the systems repository for suitable basic modules, i.e. for modules with *strategy* attribute (see section 3.2.) set to *beh* (behaviouristic). PaKMaS uses these meta-data to answer questions like "Which basic units about the SELECT statement are appropriate to be used in a guided tour on SQL?". For the *modules* found, the system can suggest a linear ordering or check the consistency of a sequence created by an author using the modules *prerequesite* and *objectives* attributes. As guided tours and sections or chapters are *structure modules* which impose a linear ordering to the children modules a teachware managment system can offer navigation facilities with which the learner can sequentially walk step-by-step through the given *basic modules*.

4.2. Constructivistic strategies

Our framework is flexible enough to support constructivistic strategies, too. It provides *structure modules* like *indexes*, *glossaries*, *time lines* or even just unordered *collections* (see Fig. 4) of learning material.

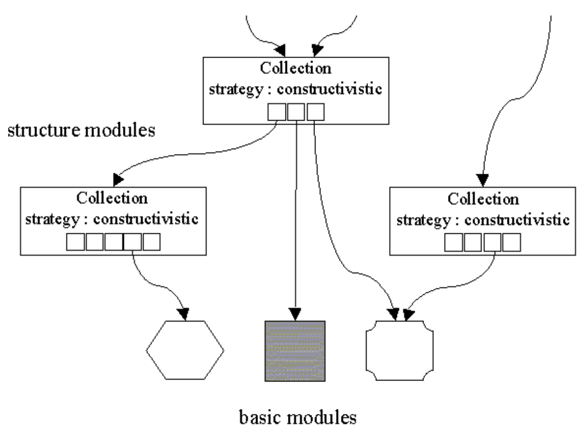

Figure 4: Constructivistiv strategy using collections

They all allow direct, non-hierarchical access. Special *basic modules* representing *coaching units* give some hints to the learner on the appropriate usage of the available material. PaKMaS supports cognitive apprenticeship (see section 2.3.3) using these modules and offering easy to use communication facilities with human coaches. Furthermore communication via email and chat with other users of PaKMaS as well as posting to appropriate newsgroups managed by PaKMaS support the creation of knowledge communities (see section 2.3.2.). A learner, for example, can freely browse through *basic modules* on composers or operas in any order, e.g. alphabetical or by date of birth. If provided by the author, she can use coaching remarks or hints in a coaching unit. Furthermore, she can communicate with coaches or with other learners in this domain of application.
Our framework allows the wrapping of simulations or real life systems into basic modules using LMCML. This way a learner, for example, in a database course can find out about SQL by herself using an interface to a real world database (Fig. 5). As another example, she can learn about harmony in a music course using a composing system thus acquiring complex knowledge by herself. It should be noted that we do not address the support for creation of constructivistic programs like simulations etc.

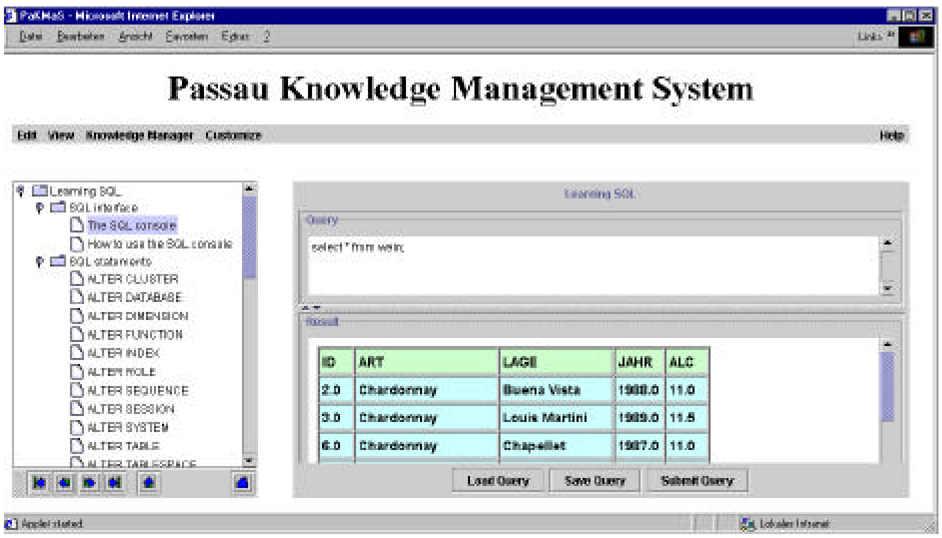

Figure 5: Collections with coaching unit, SQL interface and descriptions of SQL statements

In addition to the simple possibility to annotate learning material with individual notes, PaKMaS provides cognitive tools which are most important for the application of constructivistic strategies (see section 2.3.1). With these tools, learners can structure and form their learning environment with respect to the meta-model. For example, they can use existing *structure modules* to form own *collections* and to create own *guided tours*. Note that although the latter represents a behaviouristic strategy, creating a *guided tour* is a constructivistic learning step. To model their own learning material, learners can use existing content objects like *definitions* or *examples*. Beyond that learners can create new types of structure modules like *slide show* or new types of conceptual units like *slides*. Thereby learners can continuously form their learning environment with respect to the metamodel. Students in a music course, for example, can construct their own “map” of the world of opera using own concepts.

5. Related work

The Instructional Material Structure Description Language (IMSDL) allows to specify instructional models (Silberhorn et al. 1999). In contrast to IMSDL, we differentiate between different types of *conceptual units* or different types of *content objects*. Thus, we have the concepts and syntactical representations of *coaching units* or *coaching remarks* which are necessary in constructivistic strategies. We also differentiate between different types of *structure modules* to realize structures like *books* or *collections*. Finally, using a meta-modeling approach, we distinguish different domains of application and their specific needs with respect to teaching strategies. (IEEE P1484 1999) proposes a text-based programming language to describe courses which consist of blocks and assignable units. The former are the largest structural elements in a course and are comparable to *structure modules*, wheras the latter are programs or whole lessons launched by a CIM system. In contrast to our work, the order in which a student may go through a course is only implicitly given by the order of the blocks and assignable units or their prerequisites data. In the XML-based TeachML (Wehner et al. 1999), a course is the container of learn blocks, which contain any number of didactical paths. The latter contain at least one and at most four facts and one optional exercise. They seem implicitly to specify behaviouristic *guided tours*. In our approach, teaching strategies are explicitly specified allowing the use of learning strategies different from behaviouristic *guided tours,* too. Finally, (Teege 1999) also proposes a XML-based TeachML. Here, the author recursively composes learning material from more or less large modules using XSLT. The resulting hierarchical structures implicitly realize behaviouristic strategies.

6. Conclusion

In this paper we argued that effective teaching requires the use of multiple teaching strategies. The meta-modeling approach to hypermedia teachware which we have proposed allows the domain-specific modeling necessary for the authoring and application of different teaching strategies by providing appropriate structure modules, conceptual units, content objects and meta-data. We discussed how these concepts can be implemented using the XML-based LMML. On the one hand the Learning Material Content Markup sub-Language of LMML syntactically represents different conceptual units like coaching units or drill & practice units and different content objects like coaching remarks or instructions. The sublanguage Learning Material Structure Markup sub-Language on the other hand implements behaviouristic strategies using guided tours or books as well as cognitivistic or constructivistic strategies. Furthermore, appropriate meta-data are used by a teachware management system like PaKMaS to support authoring as well as usage. Finally, relying on the properties of XML, LMML can facilitate the portability and exchange of learning material.

References

Brickley, Dan (1999). *Tutorial Modelling Language* (TML 5.0), http://www.ilrt.bris.ac.uk/netquest/about/lang/

Brown, J. S. (1985). Process versus Product: A Perspective on Tools for Communal and Informal Elctronic Learning. *Journal of Educational Computing Research* 2 1.

Bruner, J.S. (1966). *Toward a theory of instruction*. Cambridge, Massachusetts: The Belknap Press of Harvard University Press.

Chambers, J. A. , & Sprecher, J. W. (1980) *Computer Aided Instrcution: Current Trends and Critical Issues*. Communications of the ACM 6 23.

Collins, A., & Brown, J. S., & Newman, S. E. (1989): Cognitive Apprenticeship: Teaching the Crafts of Reading, Writing and Mathematics. Resnick, L. B. (ed): *Knowing, Learning and Instruction*. Hillsdale, NJ.

Goldfarb, C.F., & Prescod, P. (1998): *The XML Handbook*. Prentice Hall.

IEEE P1484 (1999). *Learning Technology Standards Committee (LTSC), CBT Interchange Language Working Group*, http://ltsc.ieee.org/doc/wg11/CMI-Sem1.doc

Jonasson, D. H. (1991). Objectivism versus Constructivism: Do we need a new philosophical paradigm? *Educational Technology: Research & Development* 39 (3).

Kozma, R. B. (1987). The Implications of Cognitive Psychology für Computer-Based Learning Tools. *Educational Technology* 11 27.

Maturana, H.R. & Varela, F. J. (1984). *The Tree of Knowledge*. Boston, MA: New Science Library.

Merrill, M. D. (1991). Constructivism and Instructional Design. *Educational Technology* 5 31.

PaKMaS (2000). *Passau Knowledge Management System*, http://daisy.fmi.uni-passau.de/pakmas/

Piaget, J. (1977). *Epistemology and psychology of functions*. Dordrecht.

Silberhorn, H., & Gaede, B. (1999). *IMSDL: Instructional Material Structure Description Language*, 7th BOBCATSSS Symposium on Learning Society, Learning Organisation, Lifelong Learning, Bratislava.

Skinner, B.F. (1958). Teaching Machines, *Science* 128.

Süß, Ch., & Freitag, B., & Brössler, P. (1999): Metamodeling for Web-Based Teachware Management. *Advances in Conceptual Modeling*. ER'99 Workshop on the World-Wide Web and Conceptual Modeling, Paris, France, Nov. 1999, Proceedings, Springer LNCS 1727.

Teege, G. (1999). *Targeted Reuse and Generation of TEAching Materials*, http://www11.informatik.tu-muenchen.de/proj/targeteam/

Wehner, F. & Meißner, K. (1999): Ein Dokumentmodell für Kursdokumente in Webbasierten Virtuellen Lernumgebungen, *Proceedings of "GeNeMe99"*, Dresden, Germany.