

# Virtualization of Course Structures through Adaptive Internet Techniques\*

E. Köppen, G. Neumann  
Information Systems and Software Techniques  
University of Essen, D-45141 Essen, Germany

**Abstract:** Currently, most curricula in university are based on independent courses and do not take into account existing knowledge and learning preferences of the students. We provide a model which tries to deal with these requirements and provides more flexibility. The model is based on the division of courses into self-contained concepts and user-specific navigation structures.

## 1 Introduction

The curricula of many or maybe most German universities are currently under discussion. On the one hand public universities have to face a growing competition from private universities, on the other hand there is a strong desire from government and industries to shorten the length of the studies and to allow higher flexibility and more specialisations. As a consequence classes become more heterogeneous. Traditional ways of teaching are not longer sufficient to cover the continuous growing requirements on students. These requirements result in a number of design goals we describe in this paper:

- Definition of well defined units for knowledge transfer, which have a much smaller granularity than a course to allow flexibilization of courses.
- Description of the pre- and postconditions of the knowledge units, consideration of existing knowledge and learning preferences of the students by presenting tailored course material.
- Reuse of existing course material.
- Development of tools for students and lecturers

---

\*Published in: Gaul, W. and Locarek-Junge, H. (ed), Classification in the Information Age, Springer Verlag, 1999.

The following sections describe the needed definitions and terms, the resulting structures and an architecture for an implementation. The paper finishes with a discussion of the results and topics for further work.

## 2 Definitions and Terms

First, a number of terms we need to describe our approach are defined.

**Concept:** Smallest unit of knowledge, describes a self-contained, independent idea, can be identified through a keyword, requires often the knowledge of other concepts

**Course Unit:** Presentation which mediates one or more concepts

**Presentation Unit:** physical representation of one or more concepts

**Course:** Sequence of course units which follow a certain theme or teaching goal, realized as a path through the knowledge pool

**Knowledge Pool:** Contains all concepts which are used in the courses of a department

Course units and courses are conflict-free preselected navigation paths over a set of concepts. A navigation path is conflict-free, if none of the pre-requirements of the covered concepts is violated. An example for the above categorization would be:

**Concept:** Polymorphism

**Course Unit:** Object-Oriented Analysis

**Presentation Unit:** Slides and handouts for the concept “Polymorphism”.

**Course:** Systems Analysis and Design

**Knowledge Pool:** Courses of the Department of Information Systems and Software Techniques

On the layer above the courses, a new picture evolves (Figure 1). Typically, a student takes several mostly independent courses. Using a concept-based approach, this distinction between different courses vanishes, and the student is able to progress through the course units based on her experience and preferences.

This structure is similar to the approach chosen by Pilar da Silva et al. (1997), but our emphasis is more on the technical realization and less on cognitive factors.

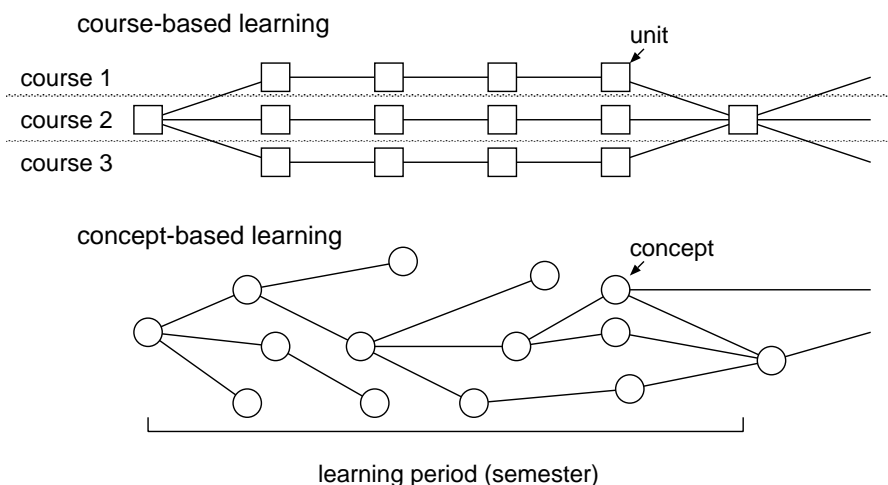


Figure 1: Course-based and Concept-based Knowledge Transfer

### 3 Navigation Structure

The level of experience and the learning preferences of the students have a direct influence on the navigation structure. Learning preferences are expressed as unknown concepts rated with a level of interest and experience is expressed as known concepts rated with a level of confidence.

In contrast to the course units and concepts which are mostly static, the navigation structure is dynamic. It changes whenever a student gains more knowledge about a concept or decides to change his learning preferences. Additional influences are the prerequisites which some concepts may have, resulting in additional concepts which have to be learned.

### 4 Implementation and Infrastructure

The model presented in this paper is targeted at a web-based environment where the course material is available on a web server or distributed as CD-ROMs. Students are able to read the course material on- or off-line using web-browsers. In our implementation, the Extensible Markup Language (XML, see Bray et al. (1997)) is chosen to structure the course units. XML offers more functionality as HTML (Raggett (1997)) for presentation purposes, especially in conjunction with style sheets like CSS (Lie, Bos, (1997)). Furthermore, it provides an excellent means to define structure and semantics of a document through Document Type Definitions (DTDs). Finally, it has a superior definition of link semantics (Maler, DeRose (1998)). We can identify four types of XML document:

- concepts
- course units,

- navigation structures and
- user profiles.

As mentioned above, the system will be used via a web browser. The resulting infrastructure is shown in Figure 2. In this scenario, the web server dynamically generates the user-specific navigation structures based on a unique user ID (implemented using the cookie mechanism, see Kristol, Montulli (1997)) and a corresponding user profile. The generation of the navigation structure is handled by CGI scripts. The resulting documents are XML documents, but it is possible to convert them to HTML documents before transmission to the client. This allows the usage of browsers which are not XML/CSS-enabled.

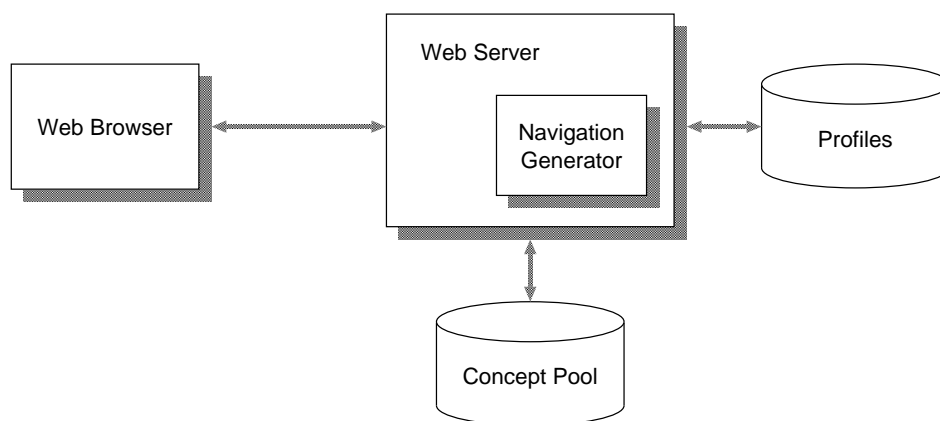


Figure 2: Infrastructure

A more flexible approach however is the usage of a web browser which is able to execute scripts. For this purpose, we are using the Cineast browser (Köppen et al. (1997)). It is written in OTcl (Wetherall, Lindblad (1995)) and handles HTML and XML documents along with CSS style definitions. Its main advantage over commercial browsers is that the Cineast browser is a prototyping environment for web-based applications which has furthermore the ability to execute embedded scripts, permitting access to the document structure and the network functionality provided by the Cineast browser through a programming interface. This interface covers on the one hand most of the Document Object Model (DOM, see Byrne (1997)) and allows the generation and modification of XML documents. On the other hand, the networking layer of the Cineast browser can be accessed through OTcl objects, which makes it possible to fetch and store documents across the Internet.

Figure 3 shows the architecture usage of the learning environment when the Cineast browser is used. In this approach, the generation of the navigation structure is realized through scripts embedded in the course material. The

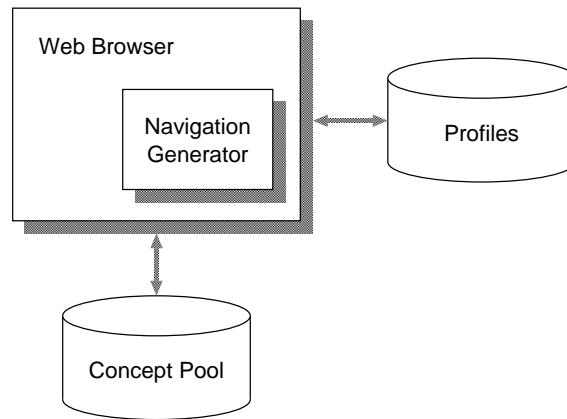


Figure 3: Infrastructure Using an Enhanced Web Browser

user profile and the concept pool can be used off-line if the material is stored on the client side.

The static course material itself is created using XML/SGML enabled editors like Emacs, or generated from existing material, e.g. LaTeX or HTML sources.

## 5 Conclusion

It is obvious that static course material cannot meet the requirements which result from differing existing knowledge and learning preferences, increasing demand for more flexible course structures and changing course contents. We presented an approach that divides courses into the underlying concepts and connects these concepts using a dynamically created navigation structure. As a result, we gain the ability to create individual courses which reflect the experience and preferences of the students. They can be used for on- and off-line learning using web techniques such as XML, CSS and embedded scripts together with complying browser and server software. From the lecturer's point of view, a pool of concepts is built rather than complete courses. This pool of concepts can be used as a basis for courses not only of a single department but of a whole faculty.

However, open for further work and discussion are the following aspects:

**Realisation of exercises and tests:** Currently, there is no mechanism for an automated review of exercises since this works only reasonably well with simple questions like multiple choice questions. Additionally, working on exercises in groups is difficult to implement. Here, a more sophisticated technical foundation is needed. Another difficulty is the organisation of tests, where the test itself is ideally done under supervision at the university.

**Annotation of course material by the students:** The students should

be able to annotate the course material and share the annotation with other students. This aspect is covered in a research project currently in progress at our department.

**Creation of an infrastructure for a “learning community”:** In conjunction with the previous aspect, the establishment of a “learning community” is a long-term goal. Students should be able to share not only their thoughts regarding course materials through annotations, they should also be enabled to evaluate, discuss and criticize the work of the department, providing feedback for improvements.

## References

BRAY, T. et al. (1998): *Extensible Markup Language (XML)*, W3C Recommendation, <http://www.w3.org/TR/REC-xml>.

BYRNE, S. (1997): *Document Object Model (Core) Level 1*, W3C Working Draft, <http://www.w3.org/TR/WD-DOM/level-one-core-971009>.

KÖPPEN, E. et al. (1997): *Cineast – An extensible Web Browser*. Proc. WebNet 97, Toronto.

KRISTOL, D., MONTULLI, L. (1997): *HTTP State Management Mechanism*. Standards Track Protocol, Network Working Group, RFC2109, <ftp://ftp.nic.de/pub/doc/rfc/rfc-2100-2199/rfc2109.txt>.

LIE, H.W., BOS, B. (1997): *Cascading Style Sheets, level 1*, W3C Recommendation, <http://www.w3.org/TR/REC-CSS1>.

MALER, E., DeROSE, S. (1998): *XML Linking Language (XLink)*, <http://www.w3.org/TR/WD-xml-link>.

PILAR DA SILVA, P. et al. (1997): *A Simple Model for Adaptive Courseware Navigation*, Proc. WebNet 97, Toronto.

RAGGETT, G. (1997): *HTML 3.2 Reference Specification*, W3C Recommendation, <http://www.w3.org/TR/REC-html32.html>.

WETHERALL, D., LINDBLAD, C.J. (1995): *Extending Tcl for Dynamic Object-Oriented Programming*, Proc. of the Tcl/Tk Workshop '95, Toronto.