Sonja KABICHER[1] & Michael DERNTL[2] (Wien)

Visual Modelling for Design and Implementation of Modular Curricula

Abstract

European curricula are currently being reorganized to fulfill requirements and recommendations set forth by the Bologna Process. Design, development and rollout of new curricula constitute significant change processes and need cooperation and coordination. This paper describes the use of visual modeling within an interactive online environment (ActiveCC Web) to share knowledge and support coordination among teaching staff during the implementation of a new curriculum. Graph based visualization is used to support the illustration of the curriculum structure and course pedagogies and offers intuitive navigation through the curriculum.

Keywords

Visual Modeling, Bologna-process, curriculum design, active curriculum

Visuelle Modellierung als Unterstützung für Entwicklung und Umsetzung modularer Curricula

Zusammenfassung

Viele Curricula in Europa werden derzeit umstrukturiert, um Anforderungen und Empfehlungen des Bologna-Prozesses zu genügen. Die Entwicklung und Umsetzung neuer Curricula bedeuten enorme Änderungprozesse, welche die Kooperation und Koordination aller Verantwortlichen und Betroffenen erfordern. Dieser Artikel beschreibt die Verwendung von visueller Modellierung innerhalb einer interaktiven Webumgebung („ActiveCC Web“), um Inhalte und Struktur eines neues Curriculums festzuhalten und Koordination zwischen Lehrenden während der Umsetzung des Curriculums zu fördern. Dabei wird mittels graphenbasierter Visualisierung die Curriculumsstruktur sowie pädagogische Aspekte in Kursen abgebildet und eine intuitive Navigation durch das Curriculum ermöglicht.

Schlüsselwörter

Visuelle Modellierung, Bologna-Prozess, Curriclums-Entwicklung, Aktives Curriculum

[1] e-Mail: Sonja.Kabicher@univie.ac.at

[2] e-Mail: Michael.Derntl@univie.ac.at

1 Introduction

European curricula are currently facing change initiated by the so called Bologna Process (EUROPEAN COMMISSION, 2007) (BOLOGNA DECLARATION, 1999) (CONFEDERATION OF EU RECTORS' CONFERENCE AND ASSOCIATION OF EUROPEAN UNIVERSITIES, 2000). Curricula conforming to the Bologna requirements need to be organized into three “cycles” (bachelor, master, and doctorate programs) of modular structure. Student progress in terms of course/module completion is accredited through ECTS points (European Credit Transfer System). Module and course learning goals are formulated in terms of learning outcomes considering professional (subject-specific) and generic (transferable) competences. Furthermore, innovative teaching methods and technology-enhanced learning should be considered and integrated into the curricula (COMMISSION OF THE EUROPEAN COMMUNITIES, 2006) (COMMISSION OF THE EUROPEAN COMMUNITIES, 2005) (TUNING PROJECT, 2004).

Implementing new curricula which adhere to European and national qualification targets requires cooperation and coordination efforts among teaching staff and policy makers to offer coherent study programs. As curricula are mostly communicated in a static document, there is little space to collaborate as a teaching staff community during the rollout of the curriculum.

In this paper we present the concept and implementation of an interactive online environment called “ActiveCC Web” which was developed as part of our faculty’s e-learning strategy project Active Curriculum for Computer Science (ActiveCC). The ActiveCC Web is provided to facilitate transparency, coordination and sharing of knowledge among teaching staff to support a well-coordinated and coherent curriculum implementation. For offering insight into the curriculum’s structure and dependencies among modules and courses, as well as to illustrate pedagogical strategies in courses, an open source graph visualization library (GANSNER, KOUTSOFIOS, et al., 2006) was implemented as an extension to the online wiki.

One of the side effects of this kind of staff networking is the transparent sharing of teaching strategies of our instructors within the ActiveCC Web, which has shown to further the interest in and the adoption of innovative, technology-enhanced teaching and learning strategies. These change processes are facilitated in the project through interviews and workshops with module coordinators and course instructors.

The paper is structured as follows. In the next section we will present and outline the main goals of the active curriculum of computer science project. Section three presents the design- and implementation process as well as the structure and organization of the ActiveCC Web. The final section gives a conclusion.

2 Active Curriculum for Computer Science

Active Curriculum for Computer Science (ActiveCC) is an e-learning project at the Faculty of Computer Science, University of Vienna. It aims to transform the formal curriculum document (UNIVERSITÄT WIEN, 2007) into a lively, innovative and cooperative online environment aiming to facilitate the implementation of our new computer science curriculum. The main goals of the project are:

– Transparency of learning goals and course contents as a prerequisite for coordination of modules and courses;

– An interactive online environment to facilitate coordination among all involved educators additional to a series of interviews and workshops, where dependencies and detailed course information were collected;

– A representation of the curriculum design in a universally accessible way, meaning it should be accessible anytime and anywhere, addressing all potential users and stakeholders (KABICHER, DERNTL, et al. 2008).

In a nutshell, ActiveCC aims to provide insight into the structure and implementation of the new computer science curriculum to support teaching staff in coordinating their course contents (particularly during the initial rollout phase where a lot of new course content needs to be developed) and to support instructors at the faculty in getting an overview of the curriculum structure, content, content dependencies, module prerequisites, and to efficiently integrate their courses into the program. In this context, "efficiency" means to plan and deliver courses in a way that minimizes redundancies and gaps and to maximize synergies in subject matter and learning outcomes addressed. This is supported by the identification and visualization of dependencies of content and competences among modules and courses. From the student viewpoint, the visualization of the curriculum facilitates experiencing the curriculum as one well-coordinated and coherent program.

The formal curriculum of the Faculty of Computer Science is distributed as a PDF document in which relevant information and instructions are defined. That is, the structure of the curriculum, module descriptions and course titles, brief definition of course contents, credit points, and so forth. This document has legal validity and was used as the starting point and as a template for initializing the ActiveCC Web.

3 Visual Modelling in Curriculum Design

3.1 Design- and Implementation Process

We implemented the online environment on the CEWebS (Cooperative Environment Web Services) platform, which provides an extensible, web service-based architecture for cooperation and learning (MANGLER, 2005). CEWebS offers a wiki module, which was extended with the functionality of directly writing graph visualization code into the wiki. The wiki module itself enables teaching staff to contribute to or modify content and update information on their courses and modules, thus enabling collaborative provision and maintenance of information on

the actual content of the curriculum. Using graph visualization functionality the temporal arrangement of modules and dependency links among modules are visually modelled to show the modules' location and role within the curriculum. As a first step we modelled known dependencies as stated in the curriculum. These formal dependencies define which modules students have to finish before they are allowed to take a particular module. For example, to be allowed to take the algorithms module, students would first need to collect 6 credit points in the introductory programming module.

Identification of the full set of dependencies among modules is not possible by only consulting the curriculum document. As instructors select and develop the topics and resources for their courses, questions may arise like "Did some module in earlier semesters already deal with that topic, and to which degree of detail?" or "Did students already get an introduction to writing project or research reports?" We experienced that sensing the current situation of each module and course through personal interviews and workshops with module coordinators and communicating contents of different modules of the curriculum online made it easer to identify potential synergies and dependencies among modules.

The multitude of influencing elements that shape the content and structure of the ActiveCC Web is displayed in figure 1.

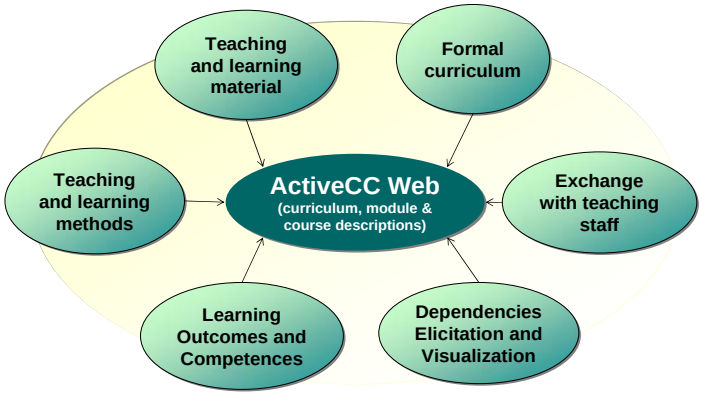

Fig. 1: Overview of the main ActiveCC "ingredients"

For each of the modules and courses data is collected from administrative systems of the university/faculty, learning management systems, courses' websites and through personal interviews and exchange with the module coordinators and course instructors. The ActiveCC Web is initialized and updated with data collected by a project team member. Collecting data and updating the interactive environment is continually repeated in order to document the actual implementation of the curriculum. Figure 2 illustrates the design and implementation process as described above.

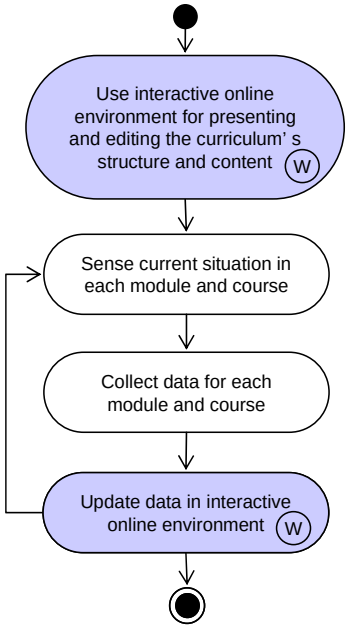

Fig. 2: Design and implementation process of the ActiveCC Web as a coordination tool for the faculty's teaching staff illustrated as an activity diagram. The first and the last activity of the activity diagram are marked as web-based activities ("W"). Activities without any mark do not explicitly have a mode of presence (for example face-to-face, online, or blended).

3.2 Sharing Knowledge and Coordination Support within a Curriculum

The ActiveCC Web is organized into three layers: curriculum layer, module layer, and course layer. The curriculum layer contains general information about the curriculum, for example the overall learning outcomes of the study, the students' professional qualification profile, and a visualization of the curriculum's modular structure, which contains:

– Modules building the core of the curriculum (e.g., modules grouped in the area "Information Technology" like databases, software architectures, software engineering; mathematics, or basics of computer science like modelling);

– Modules that relate to an application area of computer science such as bioinformatics, business, media, medicine, and scientific computing, separated in modules that belong particularly to the application area (e.g., the modules "finance" and "production and logistics" relating to the business/economics) or to the area of interdisciplinary computer science (e.g., the modules "Information system engineering" and "Enterprise information systems").

Figure 3a and 3b give a screenshot of the curriculum structure taken from the ActiveCC Web (note that the figure content is German, however the relevant annotation boxes are in English.) The boxes of the graph are hyperlinked to wiki pages which contain detailed information to the module groups. For instance, when clicking on the box "Scientific Computing", a wiki page is opened which contains detailed information to the curriculum with application area "scientific computing"

including a graph showing all modules (core modules, application area-, and interdisciplinary modules) and dependencies among them (dependencies that are stated in the formal curriculum document and content-related ones, which were identified during the interviews and exchange with teaching staff). The graph is illustrated in Figure 3b. In our bachelor study, all modules are distributed over a full six semester plan. The graph visualization of the curriculum's modules, their belonging to particular subject areas and their dependencies to other modules within the curriculum shall support teaching staff to get an overview of the location and role of their modules and courses.

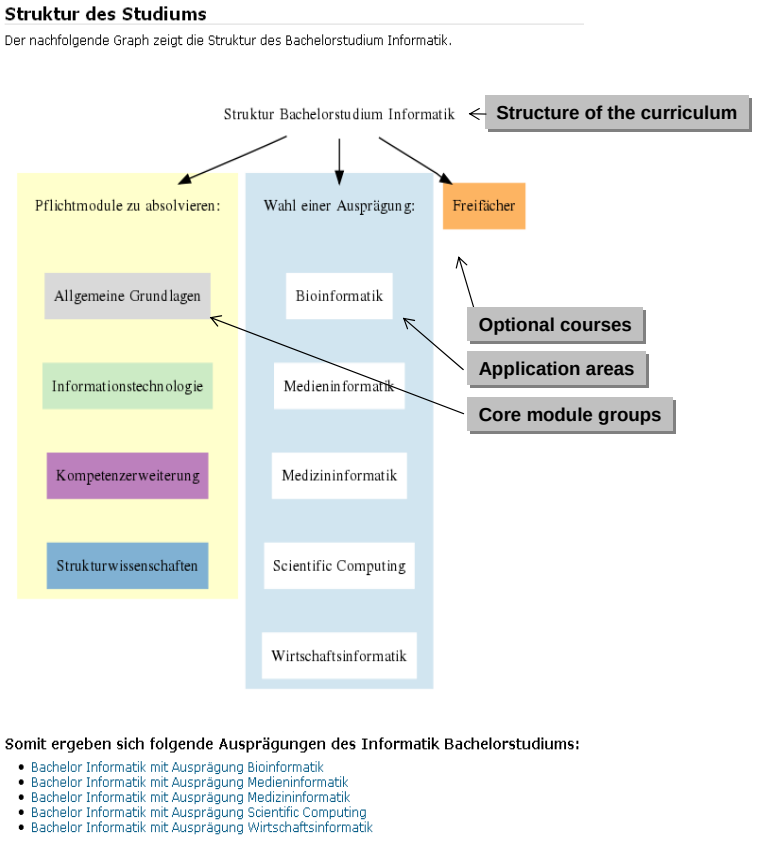

Fig. 3a: Graph visualization of the general curriculum structure of the bachelor computer science.

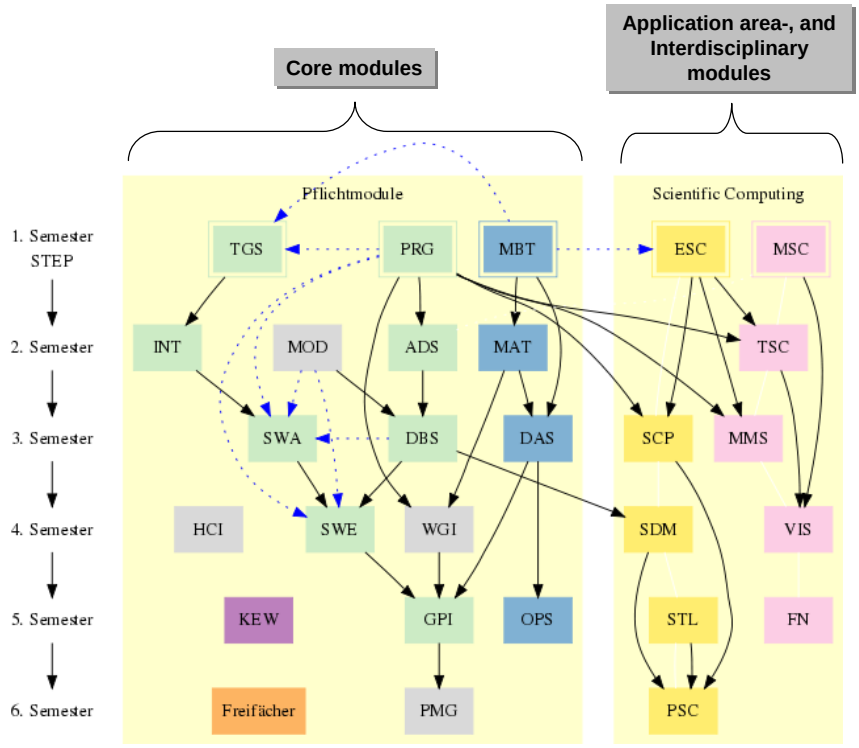

Fig. 3b: Modules and their dependencies (solid arrows are dependencies stated in the formal curriculum document, dotted arrows are identified content-related dependencies by teaching staff) of the bachelor computer science with scientific computing as application area or specialization mode.

For example, by illustrating dependencies with arrows, a module coordinator can easily identify those modules that are defined as prerequisites, i.e., those modules offering inputs for the particular module, and those building upon subject matter/content and competences addressed in the particular module. A content dependency between modules means that one module builds on content or topics covered by another module.

Also, different fill colours for boxes are used to illustrate the association of each module to a particular module group (e.g., green colour is used for modules belonging to Information Technology, blue for modules of Mathematics, yellow for interdisciplinary modules). Thus, module coordinators can easily identify to which module group their modules belong and what other modules are part of the group.

In the official curriculum document modules are described using module ID, title, credit points, semester, a short general description, as well as prerequisites giving information about which modules have to be passed in order to be admitted to the current module. As each module is implemented by a predefined set of courses, course ID, title, credit points and semester of the course(s) are defined in each module. For the ActiveCC Web, we used these module descriptions as basis for a module description template and extended it with the following details:

– A link to the module coordinator's contact information in the university's staff directory;
– A graph visualizing all dependencies that exist between the module described and other modules;
– Formal dependencies (prerequisites as stated in the curriculum document) and content dependencies as arrows between modules in the graphs;
– Subject-specific (or professional) and generic (or meta-cognitive) learning outcomes;
– List of content units and topics;
– Pedagogical approaches that are used in the module's courses;
– Modes of assessment;
– Links to the course description pages in the wiki.

The module wikis shall give teaching staff an idea of the module's integration within the curriculum and support a broad insight into the module implementation, as it summarizes the implementation information of the module's course(s) in a generalized way. The module coordinator's contact information facilitates immediate consultation of the person when coordination is necessary.

Detailed course descriptions are not included in the formal curriculum document to allow some flexibility and space for adaptation, but the most important course information is presented in the university's online course directory (the ActiveCC Web links to that information from the module/course pages). According to our study statutes, courses must be described in terms of learning outcomes, content, pedagogical methods and mode of assessment, language and literature used.

Considering these requirements we elaborated a course description template for the course wikis with the following items: course ID, title, credit points, semester, language, learning outcomes that relate to the facilitation and improvement of subject-specific and generic competences, course content, pedagogical methods, a graph visualizing the activities in a course illustrated as an activity diagram using the coUML visual design language (DERNTL & MOTSCHNIG-PITRIK, 2007), mode of assessment, and links to resources of the course (e.g., link to the course's e-learning platform, the course's website, etc.).

Figure 4 illustrates an example of a course information wiki page. It is planned to elaborate and provide activity graphs using coUML notation for each course of the curriculum as a next step of the project. The ActiveCC wiki pages support teaching staff in sharing and updating detailed information about their courses and to get a picture of other courses of the modules. This kind of transparency of implementation of the curriculum helps to reduce redundancy of teaching content, and it supports exchange and coordination of, for example, teaching methods and learning goals.

Additional features of the ActiveCC Web are a custom sidebar exposing page-independent links for structural navigation as well as breadcrumb navigation. Furthermore, there is a legend for colours used consistently throughout the online environment. Extended functionalities, like a questionnaire, online response form, or forum can be implemented easily if needed.

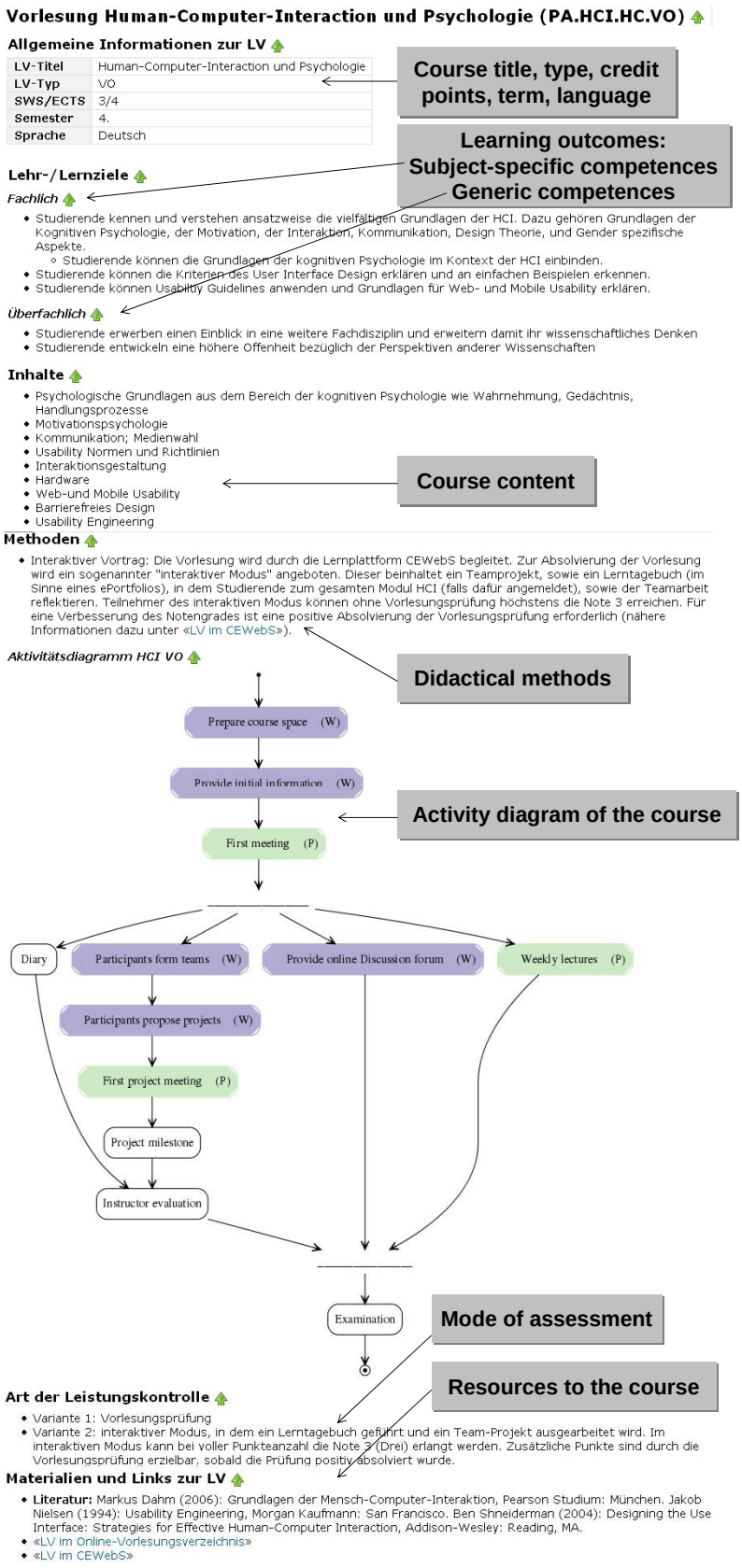

Fig. 4: Course wiki illustrating the description of the lecture “Human computer interaction and psychology”

4 Conclusions

In this paper we presented an approach to collaborative design, implementation and visualization of curriculum structure and content. This is technically realized in an interactive online environment which we developed and used in the faculty's e-learning project "Active Curriculum for Computer Science". The primary goals of this approach are:

- Offer of an easily accessible and intuitively editable virtual space to facilitate working together on curriculum content;
- Extended with functionality that meets specific coordination and visualization requirements;
- Provision of transparency of the curriculum's implementation, curriculum structure, course and module content, and pedagogies adopted by instructors;
- Illustration of module groups, modules and dependencies among modules (as stated in the formal curriculum document and as found during workshops and interviews with coordinators and instructors;
- Visualization of the learning activities in courses
- Visualization of the modules' integration within the curriculum;
- Facilitation of the sharing of knowledge collected within the teaching community by offering a virtual space where information can be provided and edited by everyone;
- Coordination support by providing insight into module and course implementations, content and pedagogical methods. This transparency can be used for improved coordination among teaching staff in order to implement the curriculum as a coherent program with well-coordinated content, competence facilitation and pedagogical methods.

Although the ActiveCC Web is part of an institution-specific project, its design- and implementation as well as the adaptation of the curriculum in a wiki with extended graph visualization functionality can be applied by other institutions and to other curricula as well. For instance, the Department of History at the University of Vienna is currently feeding the data of their own new curriculum by adopting the ActiveCC approach.

References

Bologna Declaration (1999): Joint declaration of the European Ministers of Education.

Commission of the European Communities (2005): Commission Staff Working Document Towards a European Qualifications Framework For Lifelong Learning. Brussels, Belgium: 48.

Commission of the European Communities (2006): Implementing the Community Lisbon Programme - Recommendation of the European Parliament and of the Council on the Establishment of the European Qualifications Framework for Lifelong Learning. Brussels, Belgium.

Confederation of EU Rectors' Conference and Association of European Universities (2000): The Bologna Declaration on the European space for higher education: an explanation. Retrieved 20-02-2009 from http://ec.europa.eu/education/policies/educ/bologna/bologna.pdf.

Derntl, M. & R. Motschnig-Pitrik (2007): coUML – A Visual Language for Modeling Cooperative Environments. Handbook of Visual Languages for Instructional Design: Theories and Practices. L. Botturi and T. Stubbs. Hershey, PA, Information Science Reference: 155-184.

European Commission (2007): The Bologna process. Retrieved 19-12-2007 from http://ec.europa.eu/education/policies/educ/bologna/bologna_en.html.

Gansner, E. R., Koutsofios, E., et al. (2006): Drawing graphs with dot.

Kabicher, S., Derntl, M., et al. (2008): Approaching Inclusive Universal Access on the Computer Science Curriculum Level. 37th ASEE/IEEE Frontiers in Education Conference. Saratoga Springs, NY, IEEE.

Mangler, J. (2005): CEWebS - Cooperative Environment Web Services, Faculty of Computer Science, Univ. of Vienna.

Tuning Project (2004): Tuning Educational Structures in Europe. Retrieved 19-12-2007 from http://www.tuning.unideusto.org/tuningeu/.

Universität Wien (2007): Studienplan Bachelorstudium Informatik, Fakultät für Informatik, Universität Wien.

AutorInnen

Sonja KABICHER || Universität Wien || Fakultät für Informatik || Research Lab for Educational Technologies

http://www.cs.univie.ac.at || http://www.pri.univie.ac.at/rlet

sonja.kabicher@univie.ac.at

Michael DERNTL || Universität Wien || Fakultät für Informatik || Research Lab for Educational Technologies

http://www.cs.univie.ac.at || http://www.pri.univie.ac.at/rlet

michael.derntl@univie.ac.at