

AC 2008-1458: USER ADAPTIVE INTERACTIVE COURSES IN SCORM COMPLIANT LEARNING MANAGEMENT SYSTEMS

Sabina Jeschke, University of Stuttgart

After receiving her M.Sc. in Physics at the Berlin University of Technology in 1997, graduating with distinction, Sabina Jeschke worked as an assistant teacher at the department for mathematics and natural sciences and earned her doctorate in 2004. Holding a scholarship from the German National Academic Foundation, she spent several months of research at the NASA in Moffet Field, CA. In 2000 and 2001, S. Jeschke worked as an instructor at the GaTech (Georgia Institute of Technology, Atlanta). Since 2005, Sabina Jeschke has been associate professor for "New Media in Mathematics and Natural Sciences" and director of the MuLF Center (Multimedia Center for "New Media in Education and Research") at Berlin University of Technology (TU Berlin). Starting in 2001, her Berlin group has been a driving force behind the development of multimedia technologies at the university, implementing multimedia educational elements in the education of undergraduate students, in particular for engineering students. In May 2007, Sabina Jeschke has taken over a full professorship for "Information Technology Services" at the University of Stuttgart and is also acting as scientific and executive director (CEO) of the "Center of Information Technologies" of the University of Stuttgart. Additionally, she holds a co-professorship at the TU Berlin, in particular targeted towards the co-ordination of the design of new curricula in technology-oriented studies and the coordination of several eLearning and eResearch projects.

Olivier Pfeiffer, Technische Universitaet Berlin

Olivier Pfeiffer received his M.Sc. in Mathematics at the Berlin University of Technology in 2002. His thesis in numerical mathematics investigated "Error Control using Adaptive Methods for Elliptic Control Problems in Matlab". He has been working in several eLearning projects at the TU Berlin, beginning as a student assistant in the Mumie project - a platform using new pedagogical concepts to support teaching of mathematics for mathematicians, engineers and natural scientists - at the TU Berlin in 2001, as a research assistant at SFB609 in Dresden from 2002-2004, and is now part of the Team of the MuLF (Center for Multimedia in Education and Research) at the TU Berlin). In the past three years, Olivier Pfeiffer focused on the organization and coordination of the involved teams and contributed to several other eLTR related projects. He is also involved in the planning and application of future eLTR projects at the Berlin University of Technology and the local coordinator at the TU Berlin of the EMECW3 project. His research interest focuses on the development of interactive mathematical objects especially supporting the visualization of complex mathematics and physics related problems.

Thomas Richter, University of Stuttgart

After receiving his Ph.D degree in mathematical physics at the Berlin University of Technology (TU-Berlin), Thomas Richter worked for two years in the private enterprise "Algovision Technology" as project leader for image compression. In 2002, Thomas Richter returned to the TU Berlin, working on Virtual Laboratories at the DFG-funded mathematical research center "Matheon" of the Berlin universities. Since 2003, he continued his research and standardization work on image compression in cooperation with Pegasus Imaging, a privately held company located in Tampa, Florida. In 2006, Thomas Richter moved to the Multimedia Center for "New Media in Education and Research" of the TU-Berlin, extending the development and integration work on virtual laboratories. Since 2007, he holds a position at the Computing Center RUS of the Stuttgart University.

User Adaptive Interactive Courses in SCORM Compliant Learning Management Systems

Abstract

Traditional on-line courses are static: Unaware of the learner, they present the same content to every user that participates in the course, independent of the background and the experience of the learner. Furthermore, content is often static and leaves little freedom to the learner. One might argue that this is because currently applied standards like SCORM 1.2 do not allow much more than static content linked statically within the learning management system.

However, while the upcoming version 4 of SCORM 2004 addresses adaptivity at the level of the learning management system, we present a class of dynamic interactive content objects in this paper that provide adaptivity at the level of the learning objects themselves while also leaving lots of freedom to the learner.

Since the data mining required for adaptivity happens outside of the learning management system, the presented learning objects already provide their full functionality within SCORM 1.2.

Introduction

A SCORM [17] compliant eLearning system consists of two layers: A learning management system (LMS) that regulates access to the contents, administrates the students, their scores and the assessment results, and sharable content objects (SCOs) that implement the content to be learned, let those be by simple text documents, images or interactive content like questionnaires or applets. The learner gains access to the LMS by a web-browser which embeds the content provided by SCOs into a common framework administrated by the LMS. The communication between the SCO and the LMS handled by java script, a simple scripting language that is executed by the browser of the learner. For that reason, the LMS provides information on the user to the SCO, and the SCO returns information on the learning success back to the LMS.

Most SCOs are not very spectacular: Text files, possibly some embedded images and simple multiple-choice questionnaires make up most of the content seen by the authors. Truly interactive applets are rarely seen, and if so, the interactivity is often limited. If compared to lab-based interactive hands-on experiments as found in the curriculum of engineering or scientific studies, this type of learning material has little to offer and goes only little beyond an interactive book. By that, the LMS is degraded to a content management system for web-based data.

Here we present the concept of “virtual laboratories”, a framework for students to freely perform experiments and simulations, quite similar to a real student lab. It is complemented by interactive courses run by an intelligent and user-adaptive course system. The overall system, laboratory and course system, forms a learning object (a SCO) in the sense of the SCORM standard, and thus can be plugged into existing learning management systems like Moodle [11] or Ilias [3]. Remote experiments and simulations are actively used in various experimental sciences, related training courses have also been explored in chemistry, see e.g. [12] and electrical engineering, e.g. by

[19]. Virtual laboratories are also explored as on-shore educational tools to train the technical skills of sailors of the US navy, see [1].

SCORM 1.2 [16] treats all SCOs as separate entities and does not allow them to carry over information on the learning process from one SCO to another, and thus to establish the necessary data to setup an adaptive system. This changed with the SCORM 2004 standard [17]. Possible approaches how to reach adaptivity in SCORM have been studied by Mödritscher, García-Barrios and Gütl in [10], though the authors conclude that the lack of metadata standardization for the SCOs hinders adaptivity of a SCORM driven LMS. A suitable set of metadata has been identified by the “Knowledge On Demand” project [15] within which adaptivity and reusability will be possible [13,14].

The work presented here only reaches adaptivity within a SCO, but does not require a specific ontology of the learning material. Any ad-hoc classification would work as the model is entirely driven by usage statistics. Clearly, if adaptivity should be extended beyond SCO boundaries, standardization would be required to compose SCOs from different vendors into a consistent course.

This paper is organized as follows: We first introduce the concept of virtual laboratories and discuss some of the technological aspects of our virtual laboratory VideoEasel [7]. After that we describe our course system Marvin and provide insight in how it organizes courses and allows adaption to the learner, followed by a section that presents the interface between course system and laboratory. We then describe how this two-component system interacts with SCORM and forms a learning object, and discuss some of the technological aspects. We conclude with future work items.

Virtual Laboratories

The focus of virtual laboratories is to offer highly flexible eLearning environments that offer much more interactivity than traditional static content. Very much like their “real” counterparts, virtual laboratories provide a framework to set up and perform experiments by combining the lab-equipment — here of course algorithms, which aim at either modeling real physical devices, or representing abstract concepts and phenomenon. Please note that a virtual laboratory is different from a simulation: While the latter allows the user access to one single phenomenon, the former is rather a framework to carry out multiple experiments and simulations. That is, virtual laboratories are the computer version of a laboratory that, while providing elementary equipment for experiments, leaves it to the user to combine these to form an experiment.

The virtual laboratory VideoEasel developed at the TU Berlin and University of Stuttgart is one such virtual laboratory: Its main application target is to perform experiments in statistical mechanics and thermodynamics [9], but due to its flexibility experiments in chemistry, biology, mathematics or engineering sciences are available. Fig 1 shows a screenshot of the system running as a SCO in the moodle LMS.

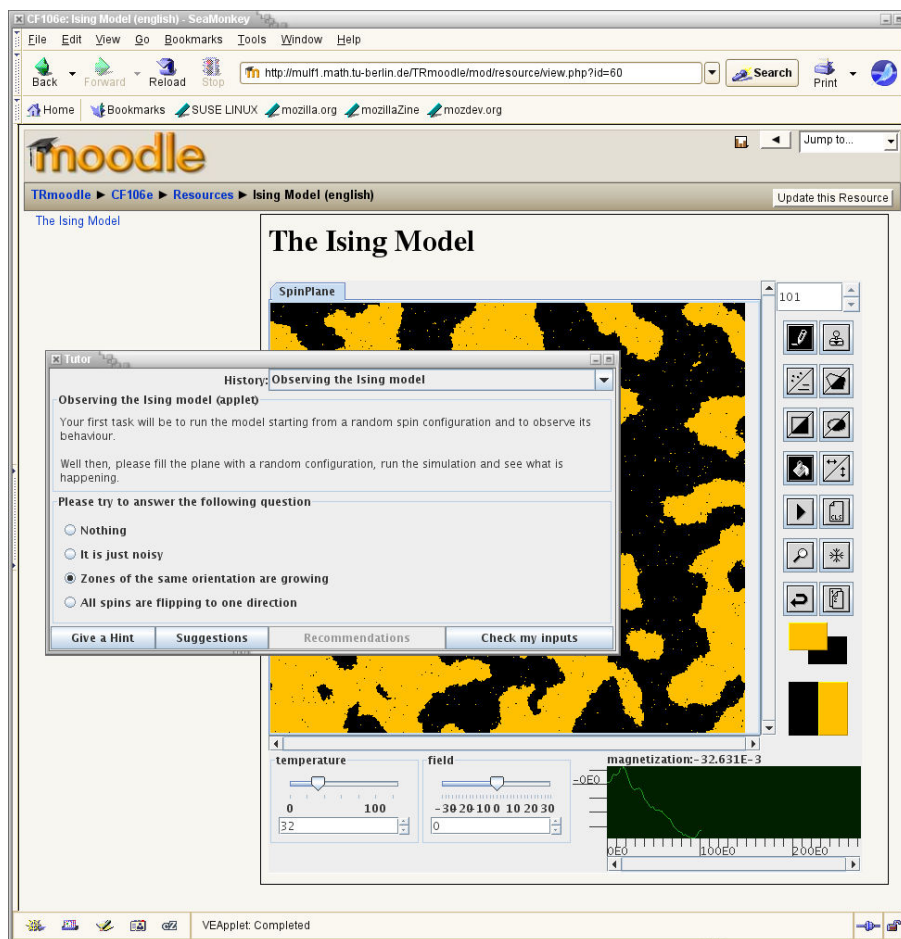


Figure 1. The Ising Model [5] SCO of the virtual lab VideoEasel running in the browser, the Marvin course system in the front.

Technological Aspects

The technology VideoEasel is based on a programmable cellular automaton [18] that not only implements the microscopic laws of the simulated physics, the very same idea is also used to define measurement devices that compute macroscopic observables. Typical experiments that can be carried out here are for example the Ising model of ferromagnetism, lattice gas experiments on thermodynamics, image denoising and much more.

The laboratory VideoEasel follows a classic client/server design: A number-crunching server performing the simulation and one or several clients that visualize the simulation and provide a user interfaces. The clients are either freestanding java applications or applets, even Maple [8] or LabView [4] interfaces exist, turning mathematical software into a user-frontend for the lab. Several users can access the lab simultaneously and either share a session, that is have cooperative access to the same experiment, or work in isolated, separate sessions on distinct setups. A session identifies the complete laboratory setup — it can be understood as the room number of the student's private lab within a virtual campus. Sessions persist for a while even

when the student leaves until, at some time, an automatic garbage collector, the virtual janitor of the campus, cleans up.

Foundations of Adaptive Course Systems

An adaptive course system obviously has to gain some knowledge about its users: That is to one degree static data available from the environment and dynamic data obtainable from the usage patterns of the learner. The first type of data, called the audience in the following, includes for example the course and lecture the student is participating at — engineering, physics, mathematics, to give some examples — the second type of data has to be collected by the system itself by software agents observing the user while the course is running.

Note that even though the wording “audience” suggests that this data classifies students according to their study, this need not to be the case. An audience is simply a classification of users into groups of similar learning style and could also be applied to distinguish users that prefer a rough overview first from those preferring a sequential learning style. As far as our technology is concerned, this is just additional metadata used to select the proper learning material.

Obviously, to allow the system to adapt to the learner, more than one possible learning strategy towards a learning goal must exist, and the learning elements must have sufficient semantic annotation to detect and select suitable learning paths through the material. The elementary unit of such a learning system, i.e. one elementary learning step is called an asset in the following. Even though this follows the SCORM naming convention, the learning system described here is not compliant to SCORM in any way.

Our course system Marvin [6] structures assets within two hierarchical layers, see Fig. 2: First of all, asset nodes are grouped to exercise units by hard links that define a branch target and by that a reaction of the system given observed user input. Adaptivity to static data is ensured at this level by providing several versions of the same learning material designed for the needs of the target audience. That is, if a certain asset node defines a reaction by providing an asset node to jump to for a specific user reaction, several versions of the jump node depending on the target audience might exist.

On the next higher hierarchy level, exercise units are grouped into mini courses. To this end, each exercise unit defines an abstract learning goal it provides, and also defines a set of preconditions that must be satisfied in advance to make this learning goal reachable. These preconditions are then, of course, satisfied by other course elements defining them as their learning goals. Since a mini course may define several alternative preconditions of which only one has to be known by the learner, alternative learning paths can be used. It is now the aim of our course system Marvin to detect and rate suitable learning paths, and to offer them to the user.

A typical learning unit would be the preparation of a suitable initial condition of the Ising model to demonstrate the effect of spontaneous magnetization, which again consists of the assets of first cleaning the screen, then selecting a tool that fills the screen with suitable data, then configuring that tool, followed by the application of the tool to the screen. Depending on the audience, the

granularity of the assets might differ — more experienced users will prefer shorter and less granular assets, for example.

The adaption to the user now requires some elementary probability theory: The optimal learning path is the one that provides the highest probability of reaching the learning goal given the history of the user, and this probability can be estimated from the observed success ratio of learners having chosen a similar learning path. For details, see [6].

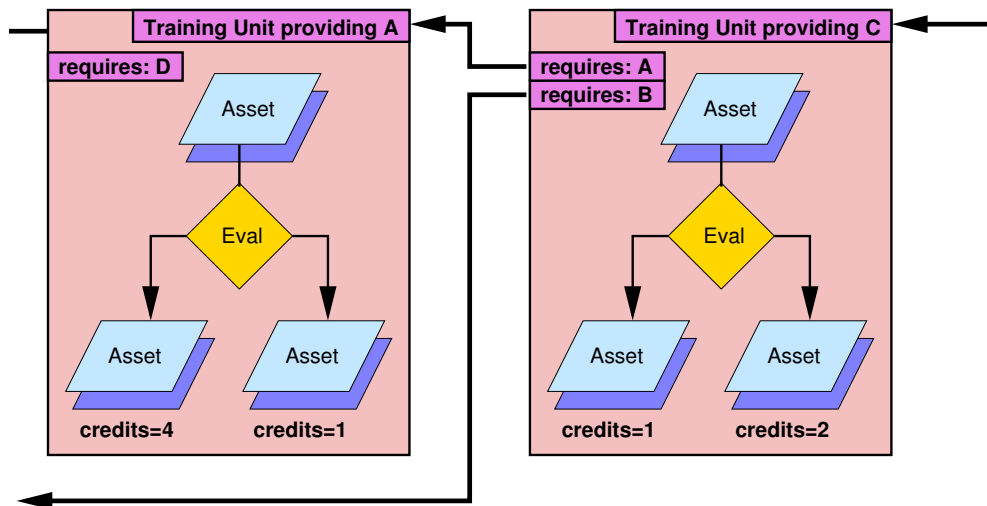


Figure 2. Design of a Marvin course: Asset notes contain elementary assignments to the students. Evaluators assess student behavior and return results on which the decision for the next asset is based. Assets may exist in several versions (shades of blue) that are targeted to varying audiences. Assets are grouped to form training units that are terminated by a credit point assignment. Training units formulate objectives and preconditions which are resolved by the course system building a learning path for the student.

Integration into Virtual Laboratories

The Marvin course system by itself neither knows anything about laboratories, nor about SCORM. It only provides a simple GUI presenting the tasks to be performed by the user. However, to collect data on the user, it needs to observe usage behavior. To this end, it interacts with the virtual laboratory VideoEasel in the following way:

The laboratory — or rather, any client that wants to run a Marvin based course — defines a so called Resource Vector containing java references to core objects of the client. In our case these core elements are references to the drawing area, to the tool buttons of the laboratory and references to the internal structures describing the laboratory setup. Since these elements are client specific, they are per se of no use for the course system. The linkage between the client at one end and the course system at the other end is provided by plug-in java classes that are specified within the asset nodes and loaded by Marvin on demand. The so-called Evaluators receive the resource vector from the course system, by that gain access to the client's internals, and return a textual evaluation to the system upon which it bases its decision.

A typical evaluator would, for example, return the current drawing tool selected by the user, allowing the course system whether the proper selection has been made to setup the initial configuration, or would check the configuration on the screen and return a description of it, again to provide input to the course system for checking whether the user performed the desired activity. That is, evaluators do not return yes/no answers; they rather provide a descriptive answer to the course system which then decides what to do according to the asset node.

Technological Aspects

The Marvin course system, as already said, is a separate program and not part of the VideoEasel front-end. Rather, it is loaded as a separate java class from the server as soon as demanded by the user, or by the learning management system. Similarly, the course data and thus the asset nodes are kept in a database on the server, and retrieved via the client-server interface as soon as Marvin requests them. The same applies to evaluators, forming the interface between the laboratory and the course system: To reduce loading time, evaluator code is also kept at the server side and requested only as needed. Because usage patterns must persist sessions in order to allow the system to learn from them, the corresponding statistics are also kept server-side.

On SCORM

A SCORM compliant system [17] consists of a learning management system (LMS) and several learning objects (sharable content objects, SCOs). The LMS administrates the student data, controls access to the learning material (SCOs) and evaluates the learning success of the student. The learning material, i.e. the SCOs, come primarily in the form of HTML pages that are rendered together with the interface of the LMS by the browser of the learner.

At a syntactic level, SCOs are ZIP archives containing the HTML code, XML meta-data defining the purpose of the SCO within the LMS, and any additional data required for the HTML code to display correctly in a browser. The pages defined within the ZIP can retrieve additional information from the LMS via a simple java script interface that on one hand provides details on the learner — the name, an ID and so on — while on the other hand this interface also communicates the learning success back to the LMS.

Integration of VideoEasel into SCORM

The course elements and asset nodes defined by the Marvin system should not be confused with SCOs — they are defined outside of the SCORM standard, even though inspired by it. The VideoEasel laboratory and its courses can also be run as a stand-alone application. Instead, one entire mini course (e.g. on the Ising model of ferromagnetism [5]) within Marvin is to be understood as one SCO, i.e. one component of a larger course, and the credit points obtained by the learner within Marvin make up the assessment results of the SCO and are reported to the LMS.

Since we want to deliver the laboratory as part of the SCORM, the freestanding java interface is of course no longer suitable — instead the Java applet interface of the laboratory is used here, whose code is delivered to the learners' browser as additional data along the HTML page.

The student Id is extracted from the LMS by means of the SCORM interface by java script embedded into the HTML data, and forwarded to the Java applet by the LiveConnect [2] interface between java and java script. This Id is used as session authentication within the laboratory, and ensures that the same student gains access to his personal laboratory setup each time he re-enters the course. As SCORM Ids are unique, each user is granted a separate session within the laboratory. As soon as the user leaves the HTML page, the learning success data is retrieved by LiveConnect in the reverse direction, and delivered back to the LMS.

However, one problem remains: The learning path collected within Marvin and by that also the state of the student within the course represented by the SCO is not part of the laboratory setup and thus neither part of the session — it must be kept elsewhere. Luckily, a SCORM compliant LMS is able to keep “suspend data” for its SCOs in its own database [17], unfortunately only in string (ASCII) format. To this end, we serialize the java classes representing the learning path of the user, and encode the resulting binary data in textual form by mildly expanding the data. The next time the SCO is re-entered, it is checked whether suspend data is available, and if so, the learning state is deserialized from it.

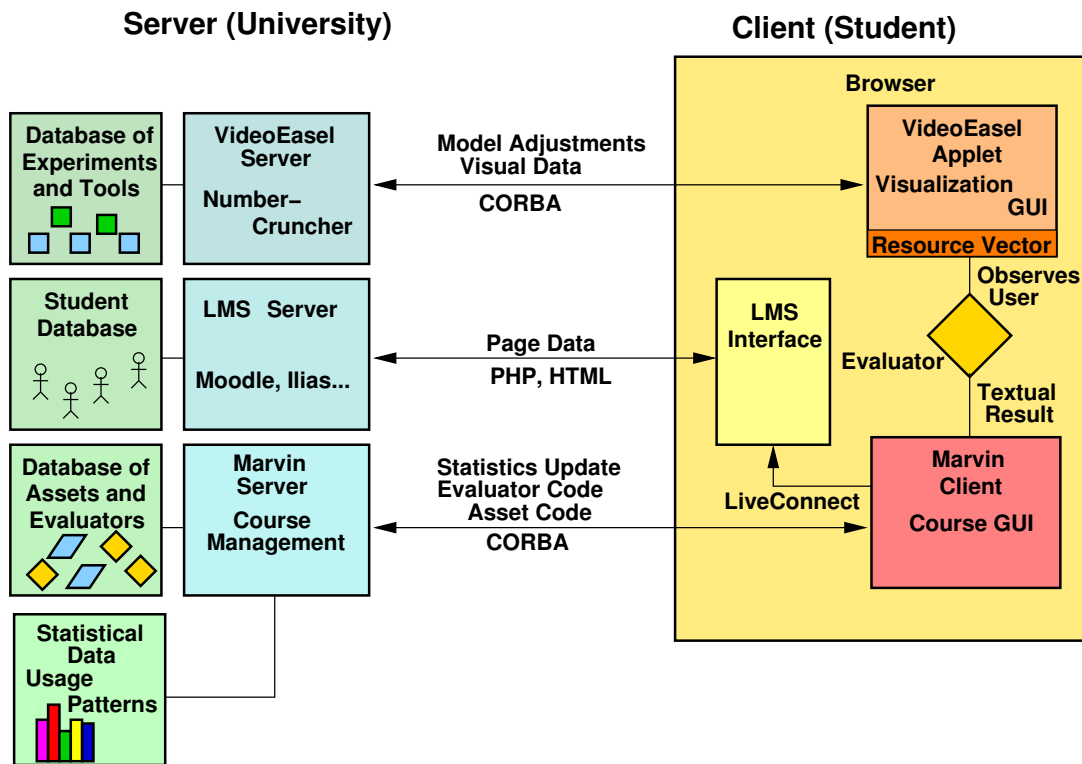


Figure 3. Overall architecture of the system: On the server side, the LMS provides content and the applet core code, the laboratory implements the physics engine, the Marvin server does the course management within the SCO. On the client side the LMS page contains the applet, which dynamically loads the client part of Marvin into the browser, which again loads course nodes and evaluators, the latter moderate between the Marvin course manager and the laboratory core.

The overall system is completely transparent to the user: Learners can leave and enter the SCOs at any time they wish, and will return to the laboratory and the course within Marvin as they left it before.

Re-Usage of Course Data

One of the design goals of modern eLearning systems is the re-usability of contents. SCORM reaches this goal by the separation of the learning material into sharable content objects that, due to their classification by meta-data, can be re-used in various courses. The same applies to the mini-courses within Marvin, too: Meta-data describe requirements and preconditions of their sub-structures. The difference between a SCORM driven course at large and a Marvin course at small scale is that we introduce agent software that is able to combine these course elements dynamically into courses at run time. Even more, not only the course elements are re-usable, the “evaluator” agents forming the interface between the laboratory and the course system are re-usable.

Conclusion and Outlook

The system at hand describes a SCORM based eLearning system that is adaptive within the SCOs. However, since usage pattern statistics are collected by the laboratory sessions for each user and made available to the Marvin server regardless of the particular user, the system can optimize learning paths for each Marvin driven SCO with the experience made by all users. Exchange between SCOs is, however, impossible within the SCORM 1.2 framework. SCORM 2004 provides advanced services and data exchange between SCOs for data describing the same student, and thus knowledge obtained from one SCO could be carried over to another. As far as both SCOs represent Marvin driven courses, this is simply a straightforward merging process of the collected statistics.

Another research topic is of course to gain experience on the efficiency of the course system and its adaptivity described above. As we have completed the SCORM interface only recently, not much data is yet available. However, the virtual laboratory itself — though as standalone application without it’s embedding into a LMS — has been deployed successfully in several courses at the TU Berlin.

Acknowledgments

A special thanks goes to W. Randelshofer for providing a simplified JavaScript interface to SCORM.

Bibliography

1. Y. Cherner, A. Lotring, R. Klein, and T. Campbell. Innovative Simulation-Based Online System for Learning Engineering and Training Sailors’ Technical Skills. In Conf. Proc. CD, 2006 Ann. Conf., Chicago, ASEE Proc. ASEE, 2006.
2. D. Flanagan. JavaScript: The Definitive Guide. O’Reilly, 2006.
3. ILIAS Learning Management. <http://www.ilias.de/> .
4. National Instruments. Labview. www.ni.com/labview/ .
5. E. Ising. Beitrag zur Theorie des Ferromagnetismus. Zeitschrift für Physik, 31:253–258, 1925.
6. M Jeschke, S. Jeschke, O. Pfeiffer, R. Reinhard, and T. Richter. Intelligent Training Courses in Virtual Laboratories. volume 1 of Proc. of World Conference on Educational Multimedia, Hypermedia and Telecommunications (EDMEDIA), pages 2415–2422. AACE, 2006.

7. S. Jeschke, T. Richter, and R. Seiler. VideoEasel: Architecture of Virtual Laboratories on Mathematics and Natural Sciences. Proc. of the 3rd Int. Conf. on Multimedia and ICTs in Education, June 7-10, 2005, Caceres/Spain, Badajoz/Spain, June 2005. FORMATEX.
8. S. Jeschke and Th. Richter. Experiments in Statistical Mechanics: A Maple Front-End for Virtual Laboratories. 2006.
9. S. Jeschke, Th. Richter, H. Scheel, and Ch. Thomsen: On Remote and Virtual Experiments in eLearning in Statistical Mechanics and Thermodynamics. pages 153–158, 2007.
10. F. Mödritscher, V.M. García-Barrios, and C. Gütl. Enhancement of SCORM to support adaptive E-Learning within the Scope of the Research Project AdeLE. 2004.
11. The Official Moodle Project Page. <http://moodle.org/>.
12. R. Moros, F. Luft, H. Papp, and W. Bailey. VIPRATECH -Das online verfügbare Praktikum Technische Chemie. In K.P. Jantke, K.P. Fähnrich, and W.S. Wittig, editors, Von e-Learning bis e-Payment 2004, Tagungsband LIT '04, LNI, pages 322–328. Akad. Verlagsgesellschaft Aka GmbH, Berlin, 2004.
13. D. Sampson. Current and Future Research and Technology Developments in e-Learning. Industry and Higher Education Journal, 16(4), 2002.
14. D. Sampson, C. Karagiannidis, and F. Cardinali. An Architecture for Web-based e-Learning Promoting Re-usable Adaptive Educational e-Content. Educational Technology & Society Journal, 5(4), 2002.
15. D.G. Sampson. Knowledge On Demand. <http://kod.iti.gr/> , 2000.
16. Sharable content object reference model (scorm) version 1.2. <http://www.adlnet.gov/downloads/> , 2001.
17. Sharable content object reference model (scorm) 2004. <http://www.adlnet.gov/downloads/> , 2006.
18. T. Toffoli and N. Margolus. Cellular Automata Machines. MIT Press Cambridge, 1987.
19. H.D. Wuttke and K. Henke. In K.P. Jantke, K.P. Fähnrich, and W.S. Wittig, editors, Von e-Learning bis e-Payment 2005, Tagungsband LIT '05, LNI, pages 481–490. GI, Bonn, 2005. J.