In: F. T. Tschang and T. Della Senta (eds.): *Access to Knowledge: New Information Technologies and the Emergence of the Virtual University*. Amsterdam: Elsevier Science and International Association of Universities, 2001, 167-206.

Course Delivery Systems for the Virtual University

PETER BRUSILOVSKY AND PHILIP MILLER

Carnegie Technology Education
4615 Forbes Avenue, Pittsburgh, PA 15213, USA
{plb | plm}@carnegietech.org

Introduction

This chapter analyzes course delivery systems in the context of distance learning using the World Wide Web (WWW) and Internet in general. Many projects are focused on web based education (WBE) systems and tools, covering the spectrum of teacher, student and institutional needs for virtual education. These needs are addressed in traditional, face to face residential, education (we will call this human based education or HBE) by a variety of methods: lectures, textbook, tests and exercises, labs and seminars, office hour contacts with the teachers and teaching assistants (TA), the registrar's office etc.

Systems and teams that produce systems are changing rapidly, but the underlying educational needs are not. The goal of this chapter is to provide the reader with a sense of the landscape of systems and features plus some idea of who the players have been up to this time. For the builder of on-line courses and virtual universities we hope to help shape thinking and provide a good bibliography.

Educational Components

We divide the needs of virtual university courses into four main components: presentation, activities, communication, and administration.

Presentation comprises all functions related to the delivery of new material. It is one core of education. In HBE presentation is achieved through lectures, textbooks, and video.

Activities comprises the learning materials, active and interactive, which involve students in doing something. In most WBE systems activities are assessment oriented; i.e., their goal is assessment of student progress (it includes self-assessment). Assessment, provided in HBE through quizzes, tests and homework, is important in credit-bearing courses and critical in education that diagnoses the state student knowledge and then prescribes ways to move the student to a higher state. A number of WBE systems can also offer activities specially developed to support learning-by-doing. These systems are more concerned with providing student support and are similar to HBE labs.

Communication comprises all ways of communicating (group or one-to-one) between teacher and students, or between student groups. The university student who has problems is able to ask questions of a teacher or of peers. Communication is an important way that teachers and students diagnose and remedy problems. This is done with problem solving feedback, additional explanations and suggestions of additional work. Communication between students is so important it is often taught as a separate skill. In HBE communication is achieved in contact during classroom activities, office hours and informally.

Administration comprises all record keeping activities of registering students, payments, course cancellations, course credits and grades, auditing student progress against degree requirements etc. Administration is performed in HBE by teachers and administrative personnel. While at first it may seem that tasks such as recording payments and student withdrawals are not really a part of education, they most certainly are a part of what allows universities to operate. Without administration, university education would not take place.

Multiple tools and systems have been developed to provide all four components in virtual universities (Table X). Within each component these tools and systems could be further divided into two groups - tools used in course preparation (various authoring[1] and set up tools), and tools used to support a running course (delivery and run-time management tools).

	Presentation	Activities	Communication	Administration
Before start	Authoring tools	Authoring tools	Set up tools	Set up tools
Delivery time	Delivery tools	Delivery tools	Support tools	Run-time tools

Table X. Groups of tools required to prepare and run a virtual university course. Course delivery tools are shaded gray

This review is centered on two groups of tools and systems (shaded gray on Table X) which are required to support what is called *course delivery* for a virtual university course. We will divide these groups of tools into further subgroups and analyze the completeness and sophistication of existing WBE tools and systems. Although other groups of tools are left outside the scope of this paper, we will be talking briefly about presentation and exercise authoring and grade viewing for the purpose of clarifying comment on delivery tools.

What kinds of tools are available

Various authoring and delivery tools used in Web-based education could technically divided into two groups - specialized tools and integrated tools. There are many functions, which could be performed by a tool in the process of design and delivery of a Web-based course. Specialized tools usually support a narrow range of these functions. It could be a single function, such as test authoring, or a small group of functions centered on a particular type of educational material, such as Web lecture tools. Integrated tools are aimed to support a wide range of activities. The goal of all integrated tools is to provide a comprehensive universal environment, which could support at least all most important functions. If this goal is achieved, the integrated tool could be the one and the only tool educational tool required to create and deliver a Web-based course. There are multiple benefits of using integrated tools vs. a set of many independent tools. Very few examples of integrated tools were available just three years ago, but due to the high demand dozens of university and commercial integrated tools are available today. There is a clear tendency now towards developing integrated tools. Existing specialized tools are getting assembled in "kits" and "suites" (CourseWeb Team, 1997; Instructional Toolkit Team, 1997; POLIS Team, 1998; Siviter, 1997). Narrow-range integrated tools are being extended with more functionality. Companies who had a system, which cover only a part of the needs are rapidly enhancing their tools to the level of an integrated tool by development and mergers (like SoftArch with their FirstClass, Centra with its Symposium (Centra, 1998), or Lotus with its LearningSpace (Lotus, 1999).

Who makes the tools

According to their origin, existing authoring and delivery tools could be divided into four categories, where first two categories form a group of *university-level tools* and second two forms a group of *commercial tools*.

- University research-level systems

[1] By *authoring tools* we mean the tools that help put lectures, tests and so forth onto the Web.

- University-supported products
- University-grown commercial systems
- Full-fledged commercial systems

University research-level systems are often quite advanced and demonstrate innovative ideas, but these tools are not supported so other universities and companies are unlikely to put them into the critical path of materials development or delivery. Some of these tools are available as freeware (or through technology license) and are used by teachers in other universities. The following are the most interesting examples of this kind of systems: QuestWriter (Bogley *et al*., 1996; QuestWriter Team, 1998) - Oregon State University, ClassNet (ClassNet Team, 1998; Gorp & Boysen, 1996) from Iowa State University, CourseWeaver (Rebelsky, 1997) and ASML (Owen & Makedon, 1997) from Dartmouth College, ONcourse (Jafari *et al*., 1998) from Indiana University, Online (Rehak, 1997a; Rehak, 1997b) and InterBook (Brusilovsky *et al*., 1998) from Carnegie Mellon University, ARIADNE (Forte *et al*., 1996a; Forte *et al*., 1996b) and MTS (Graf & Schnaider, 1997; IDEALS Project, 1998) from two European University-driven consortia, WebMapper (Freeman & Ryan, 1997) and FLAX (Routen & Graves, 1997a; Routen *et al*., 1997) from De Montfort University

In America and Canada a strong market push helped many research-level tools evolve into *university-supported products*. These systems as products at low cost or sometimes no cost from their home universities. These tools are well-developed and tested. Since installation and maintenance help is available for this group of products, they are quite popular and are used in many universities, mainly in USA and Canada. Some well-known integrated systems of this category are: CyberProf (Hubler & Assad, 1995; Hubler & CyberProf Team, 1998; Lam & Hubler, 1997; Raineri *et al*., 1997) and Mallard (Brown, 1997; Brown & Mallard Team, 1998; Graham *et al*., 1997; Graham & Trick, 1997) from University of Illinois at Urbana-Champaign, Serf™ , from University of Delaware (Serf Team, 1998), WebAssign from North Carolina State University (Titus *et al*., 1998), and Virtual-U from Simon Fraser University (Fisher *et al*., 1997; Harasim *et al*., 1997; Virtual-U Research Team, 1997)

For some *university-grown* tools, the success in their home university leads to establishing a company that usually ships some version of the tool as a *commercial system* and continues the development of this tool on an industrial basis. The most well-known examples are WebCT, from University of British Columbia (Goldberg, 1997a; Goldberg, 1997b; Goldberg, 1997c; Goldberg *et al*., 1996; WebCT, 1999), Web Course in a Box (WCB) from Virginia Commonwealth University (madDuck, 1997), and CourseNet from SUNY (Whitehurst, 1997).

Full-fledged commercial tools systems is now the fastest growing category. Until recently, only two products in this group were available: TopClass and LearningSpace. TopClass (WBT Systems, 1999) started several years ago as a university-grown tool. For a long time it was the only commercial system on the market, and has become very strong and comprehensive. LearningSpace (Lotus, 1999) started from Lotus' strength in database and Web-base server technology. Since that time many commercial tools appeared on the market. Several major players of software industry such as Asymetrix, Macromedia, and Oracle has joined the list. Some often-referred commercial systems are: CourseInfo (Blackboard, 1999), Docent 2.0 (Docent, 1998), The Learning Manager (Campus America, 1998), Symposium (Centra, 1998), WebMentor (Avilar, 1998), Enterprise Education Server (Mentorware, 1998), Web University (IMG, 1998), LearningEngine (Knowledge Navigators, 1999), and IntraKal (Anlon, 1998).

We have to note that the difference in functionality and service between categories of tools is not clear-cut. Some research level tools are stronger and more mature than some commercial tools. However, commercial tools evolve faster. The market pushes them to be more generic and user friendly. Commercialization brings better service. Naturally, a number of tools are in the process of moving from one category to higher one. Some research-based tools are becoming products; some university level products are becoming commercial tools; companies (like Blackboard and ULT) are acquiring some university-grown commercial tools.

Levels of sophistication

To characterize different systems we identify three levels of advancement in each component:

Base, State-of-the-art and Research.

Base level: provides minimum requirements. All usable systems addressing component are not below this level. A system, which offers less is not given credit for the component and will not be able to survive on the market. Still systems offering no more than the Base level may survive because they are well supported, inexpensive, reliable or because customers do not yet demand more sophistication.

State-of-the-art level: the best systems provide this level of functionality. Few systems will be State-of-the-art on all components. Systems with State-of-the-art features are often presented at development-oriented conferences (such as WebNet, ED-TELECOM, or EDUCOM). As this chapter is being written commercial products have some, but not all, State-of-the-art features with the rest being at the base level.

Research level: Experimental systems, often incorporating advanced elements of artificial intelligence or cognitive psychology, are referred to as research level. These systems are not ready to be commercially distributed. Often these systems are primitive in some components, but very advanced in the component, which is of research interest. No commercial systems exist with research level features. This work is typically presented at research conferences.

A few words about advances in authoring and delivery tools are in order. There is a tacit implication that education is better served as systems are more heavily populated with components that are state-of-the-art, better yet when research level advancements are folded in. While the authors generally subscribe to this view they are not prepared to make such claims. It is an empirical question as to the educational impact of "advancements" in Web delivery. Indeed this is, in part, why educational research is ongoing. Similarly advances in content creation tools certainly seem to be a good idea, but user studies (not reviewed here and largely not yet undertaken) will have the final word on this.

The progress in WWW-related technology is fast and many techniques, which were State-of-the-art just a year ago are now only Base level. Essentially when features are incorporated into multiple commercial systems, those features define a new Base level. The life cycle is that Research level techniques move into the State-of-the-art and State-of-the-art features are rapidly incorporated into commercial systems. This is natural and appropriate. Software vendors have user communities that cannot be abandoned. Therefore advances must be painstakingly engineered into systems that continue to support cash paying customers with systems built on earlier technology.

1 Presentation: Learning by Reading and Watching

Existing options and systems for WBE content presentation differ in five main aspects: structure, type of content material, media, authoring, and delivery. *Structure* is the most essential aspect. In the early days of WBE an unstructured presentation or *a heap* was the dominant form of content presentation for Web courses. A heap is essentially an assorted set of static HTML[2] pages connected to the course home page. While many universities still apply a heap approach, an ability to support structured presentation is now a base level requirement. We distinguish two types of structuring the learning material: a hierarchical structuring which we call *electronic textbook* and a sequential structuring which we call *electronic presentation* (here we distinguish two special subclasses *electronic lecture* or a *guided tour*). From our point of view, electronic textbooks and electronic presentation differs from each other essentially, so we will compare existing systems within each of the identified classes.

Type of content material and *media* are closely connected. The currently used types are text, static figures, dynamic figures, audio material, video material, and, very rarely, interactive simulation (this type will be considered separately later). Type usually determines the media used: HTML is used for text, various graphical formats for figures and various canned or streamed

---

[2] HTML, HyperText Markup Language, a page definition language, is a defacto standard for Web publishing. Using HTML allows one to publish a Web page and not worry about the computer, operating system or application software sued by its readers.

formats for audio and video material. There are also several multimedia and animation formats like Shockwave (Plant, 1998) or HyperStudio (Wagner, 1998) to represent dynamic figures and simulation, though simple dynamic figures are more often done with animated GIFs. All kinds of dynamic behavior are brought about by Java and JavaScript technologies. They are also used to "spice" other types of media. Currently, a good number of commercial authoring tools is available to develop audio, video, and multimedia fragments in several formats. However, most of the university level Web courses and systems are still far from using advanced media types.

From the *authoring* point of view we distinguish three levels of advancement[3]. In the original authoring technology all content was developed in plain HTML using a text editor (we call it zero-level authoring). Zero level authoring is now below the base level. Nearly all serious content developers are using either general tools (we call this *level one* authoring) or special educational tools (*level two* authoring).

General tools are not specific for educational application. Using general tools is typical for base level courses and systems. There are two kinds of general tools. With the first authors develop content with a traditional word processor in some standard format (Microsoft Word/RTF, Emacs/LaTeX, etc.). Most word processors have an option to "save as HTML". Alternatively a separate converter could be used to transfer the material into an HTML format. Since this option is especially attractive for producing electronic textbooks we will conider it in more detail later. Another kind of general tool that is numerous is the HTML WYSIWYG[4] editors such as Microsoft Front Page, or Adobe Pagemill which produce content directly in HTML which in turn is displayed as interpreted HTML (typically lovely graphics) to the author. The most advanced of these tools such as commercial NetObjects Fusion (NetObjects, 1998), GoLive Cyberstudio (GoLive, 1998), or university-developed W3Lessonware (Siviter, 1997) and Gentler (Thimbleby, 1997) can handle not only single pages but also groups of pages and complete sites. Education-oriented authoring tools can provide additional support for the authors of course material. A number of current university research projects are aimed at developing more useful authoring tools. These tools are classified into two groups: educational mark-up tools and structured content authoring tools.

Educational markup tools assume the use of some educational mark-up language (HTML-like or LaTeX-like) and a text editor for authoring. Usually an educational mark-up language offers some extensions to HTML. Two illustrations of educational mark-up extensions are: support for rendering formulae as in (Bryc & Pelikan, 1999; Hubler & Assad, 1995; Xiao, 1999) and support for exposing a multi-document structure in one document (Owen & Makedon, 1997). Structured content authoring tools provide additional support for authoring one of the structured types of course material. There are specific authoring tools for developing textbooks, lectures, and guided tours (we will talk about them later). In the near future, education-oriented authoring tools will allow course developers to index all developed material with education-specific metadata.[5]

From the *delivery* point of view we distinguish three levels of sophistication. The base level for Web courses still uses static HTML and media pages with static links. More advanced systems usually add some functionality implemented with simple CGI[6] scripts like adding coherent footer, header, or navigation buttons. However, the state-of-the-art level now uses

---

3 As we stated, authoring issues are left outside of the scope of this paper. Here we provide a very simple classification and a brief review which we need to classify existing Web courses and systems.

4 What You See Is What You Get (WYSIWYG) editors became popular first in text editors when graphics oriented displays emerged (pioneered by Xerox with its Alto system, later commercialized by Apple with the Macintosh). The key feature is that complex editing commands are performed directly and their results (not a marked-up and yet to be compiled text stream) instantaneously appear on the computer's display.

5 Metadata means "data about data". In WBE metadata are used to provide educational information about fragments of learning material such as media files, tests, or lessons. Educational metadata usually includes such fields as author of the fragment, language, difficulty, related topics. Several research groups are working on developing a standard set of educational metadata fields.

6 Common Gateway Interface defines a standard way to transfer data entered by a user into a HTML form from browser to the client, process this data with a special program (called CGI script) on the server, and return the results back to the browser.

database technology with course information and content stored in a database and most of the pages generated on the fly. A system with a database core can provide a lot more functionality while being more easy manageable at the same time. Illustrative of the advantages are generation of tests from equivalence classes of test items, archival of system use and automatic handling of multiple simultaneous file accesses. A number of advanced commercial and university-level delivery systems like TopClass (WBT Systems, 1999), LearningSpace (Lotus, 1999), CourseInfo (Blackboard, 1999), ARIADNE (Forte et al., 1996a; Forte et al., 1996b) or Carnegie Mellon Online (Rehak, 1997a) are database-driven.

1.1 The electronic textbook

The electronic textbook offers a hierarchically structured representation of material. It mirrors a printed textbook with its subdivision into chapters, sections and subsections. The material is usually presented in a text form, extended with figures. Hierarchical structuring implies special hierarchical navigation[7] aids A typical set of hierarchical links includes links to all subordinate sections, a link to the higher level section, and a link to the beginning of the electronic document.[8] Sequential navigation links (next-page/previous-page) are provided in most electronic textbooks. These links are often provided in the form of “arrow” icons. Additional navigation aids include a navigable “path” (a list of direct ancestors usually shown at the top of a page) with the possibility to move in a single click to any direct ancestor and sometimes to any page at the same level, a table of contents, and an index. Pages from the same textbook should have a uniform look sharing a similar design, header and footer. If possible, a glossary of terms should be provided (Barbieri & Mehringer, 1997; Brusilovsky et al., 1998; Goldberg et al., 1996; Langenbach & Bodendorf, 1997; WebCT, 1999).

Base level: We consider the following configuration as the base level for electronic textbooks. A textbook is a tree of static HTML pages with still figures. Simple navigation buttons (previous-next-top) and table of contents are be provided. Base level textbooks are usually produced with general authoring tools. In particular, they could be very efficiently produced using a structured document preparation system and a special converter. Currently a number of advanced converters are available: LaTeX to HTML (Dracos, 1998), RTF to HTML (Hector, 1998), FrameMaker to HTML (Harlequin, 1998). These converters preserve hierarchical structure of the content and generate a hierarchy of HTML pages with a separate table of contents and all basic navigation buttons provided “for free”. A number of early advanced Web-based electronic textbooks were produced using these converters (Antchev *et al*., 1995; Brusilovsky *et al*., 1996; Marshall & Hurley, 1996a; Marshall & Hurley, 1996b).

State-of-the-art level: The state-of-the-art electronic textbooks extend base level in both media and authoring. State-of-the-art textbooks include some choice of more advanced media items such as audio and video fragments, simple animations and dynamic figures developed with commercial-of-the-shelf (COTS) software tools. From the authoring side, state-of-the-art electronic textbooks are usually produced with advanced content authoring tools. It could be either some education-oriented authoring tools mentioned above or a combination of general authoring tools and special tools for uploading (and on-the-fly structuring) of course material. The main reason for application of these tools is to ensure generation of consistent links and provide homogeneous navigation and look-and-feel. A number of university developed tools such as ASML (Owen & Makedon, 1997), WebMapper (Freeman & Ryan, 1997), FLAX (Routen & Graves, 1997a; Routen & Graves, 1997b; Routen et al., 1997), WebCT (Goldberg et al., 1996; WebCT, 1999), CourseWeaver (Rebelsky, 1997) and most of the advanced commercial tools for development and delivery of Web courseware such as TopClass (Schwarz *et al*., 1996;

---

[7] Navigation, in this context, means the traversal of the electronic text. Just as the reader of a book can read one page at a time, check out the table of contents, use footnotes or rely on an index or glossary the reader of the electronic text can jump from place to place in the electronic document. Good navigation aids guide the reader in this process, making it easy to get to related portions of the text and easy to get back into the main flow of the document.

[8] The reader may recognize this hierarchy as the a-cyclical directed graph or “tree”

WBT Systems, 1999), (Blackboard, 1999), LearningSpace (Lotus, 1999) provide one of these authoring options. From the delivery point of view, we still can count textbooks produced from static HTML files as state-of-the-art electronic textbooks. At the same time, most state-of-the-art systems use CGI scripting extensively and a number of them use database technology. We expect that very soon a database core will be a requirement for state-of-the-art level. Databases become increasingly important as system usage increases. While a file system and hand coded CGI is sufficient for managing a few students interacting with simple courses, more sophistication is essential in maintaining large numbers of students in entire universities.

Research level: There are several directions in which research level systems are trying to advance the state-of-the-art level. Most active research direction now is what we call "reusability authoring"[9]: ARIADNE (Forte et al., 1996a; Forte et al., 1996b), MTS (Graf & Schnaider, 1997; IDEALS Project, 1998), Educational Broker (Langenbach & Bodendorf, 1998), and others (Neumann & Zirvas, 1998; Rebelsky, 1997). Reusability authoring is a technology which allows authors and groups of authors to develop pools of various content objects indexed with metadata and to re-use previously authored objects along with new objects to build a new course. This technology implies a database core.

Another research direction is aimed at improving the structure of the educational hyperspace by identifying concepts (i.e., knowledge atoms) behind the content and, as a result, providing more advanced navigation facilities. It includes work on concept-based hyperspace design,(Abou Khaled *et al*., 1998; Adams & Carver, 1997; Brok, 1997; Eklund, 1995; Fröhlich & Nejdl, 1997; Neumann & Zirvas, 1998; Nykänen, 1999; Pilar da Silva *et al*., 1998) and work on concept-based navigation (Brusilovsky & Schwarz, 1997).

An interesting research direction is "intelligent figures." Illustrative of this are Java-based figures in Medtec (Eliot *et al*., 1997), which could be used for both exploration (point to a part of the figure and see some additional information) and quiz mode (get a question - point to a part of the figure). Intelligent figures integrate presentation, exploration, and assessment and further extend the notion of "figures" in a textbook.

The most challenging direction of research is adaptive presentation and adaptive navigation support. The goal of *the adaptive presentation technology* is to adapt the content of a course page to the student's goals, knowledge and other information known about the student and his or her condition of work. In a system with adaptive presentation, the pages are not static, but adaptively generated or assembled from pieces for each user. For example, with several adaptive presentation techniques, expert users receive more detailed and deep information, while novices receive additional explanation. Existing works on adaptive presentation for Web courses cover adapting media types selection to the learner individual traits and connection speed (Carver *et al*., 1996; Danielson, 1997) and adapting educational content to different groups of learners (Lemone, 1997) and learners individual level of knowledge (Ahanger & Little, 1997; Calvi & De Bra, 1997; Eliot et al., 1997; Kay & Kummerfeld, 1994a; Kay & Kummerfeld, 1994b; Yoon *et al*., 1997). Adaptation to the learner individual needs is not limited to a page-level adaptation. A sequence of content from small presentation to whole course cold be individually generated on demand (Ahanger & Little, 1997; André *et al*., 1998; Vassileva, 1997).

The goal of the *adaptive navigation support technology* is to support the student in hyperspace orientation and navigation by changing the appearance of visible links. In particular, the system can adaptively sort, annotate, or partly hide the links of the current page to make easier the choice of the next link to proceed. Adaptive navigation support (ANS) for Web based courses has been researched in a number of projects: (Brusilovsky et al., 1998; Brusilovsky et al., 1996; De Bra, 1997; Nakabayashi *et al*., 1997; Neumann & Zirvas, 1998; Pilar da Silva et al., 1998; Specht & Oppermann, 1999).

1.2 The electronic presentation

An electronic presentation is a way to represent essentially sequential material such as a lecture or a slide show. A main criterion for considering a material as sequentially structured is the presence

[9] This direction of research is only partly relevant to electronic textbooks

of sequential navigation tools. The presentation should have a distinct “top” node with some introduction and a link to the first node. All nodes (we call them “slides”) should contain “next” and “previous” navigation links as well as a link to the top node. A table of content with a possibility of direct navigation to any slide should be also provided. Each slide in a presentation usually has a content and a narration. In this sense a zero-level presentation is just a sequence of slides where both content and narration are authored together as pure text (or text with figures). While this style of presentation is still very popular we will not consider it here since it actually equal to a one-level-deep electronic textbook analyzed above. In this section we will consider two special types of electronic presentations used in Web-based courses: an electronic lecture and a guided tour. An electronic lecture is a sequence of slides extended with audio or audio/video narration. It is modeled after a regular lecture. Usually both the content (slides) and the narration are created by the same author who takes the role of a virtual lecturer. A guided tour is usually a multiple-author presentation. The content of a guided tour slides could be previously developed by different authors and located anywhere on the Web. The role of a guided tour author is to sequence this content and extend it with a narration (here the narration is usually a text). This type of presentation is modeled after a museum guided tour.

1.2.1 THE ELECTRONIC LECTURE

There are two kinds of “electronic lectures” used in distance education: synchronous lectures and asynchronous lectures. Synchronous lectures simply provide a distance access to a real lecture theatre. While this technology is quite popular in distance education, we will not consider it here. A synchronous lecture is a mode of distance education but not an element of a courseware. Aslo, current implementation of synchronous lectures is most often based on special video-conferencing software rather then on a Web-based solution. Asynchronous lectures are recorded and could be viewed at any time. It makes it an eligible courseware element, which can be stored, “owned”, and distributed. A number of advanced suits of tools support both synchronous and asynchronous lectures: a synchronously presented lecture can be recorded, enhanced, and turned into an asynchronous lecture (Eisenstadt & Domingue, 1998; Synnes *et al*., 1998).

Asynchronous lectures come in “single piece” or “chunked” form. A “single piece” lecture is usually a continuous record of a classroom presentation. We will not consider single piece lectures here. First, this type of lecture is much less useful than a full-fledged Web lecture because it is almost impossible neither to read a video-recorded content of presented slides (Abowd *et al*., 1998) nor navigate within the content of the lecture. Second, this type of lecture is technically rather a piece of video or audio learning material than a special form of Web presentation. What we are considering below is a “chunked” asynchronous lecture i.e., an asynchronously presented sequence of slides with video and/or audio narration. We will call this type of electronic lectures “Web lectures”. Though this kind of electronic lectures was originally developed for an older CD-based technology (Dannenberg, 1998), its Web implementation now dominates.

Web lecture is becoming more and more popular technology of presenting course material on the Web. First of all, it is still the best replacement for classroom lectures. Neither textbooks nor handouts adequately replace an up-to-date lecture done by a leading researcher or professional. Secondly, lectures provide distance students with the "feeling of the classroom". Third, it is often the easiest way to place some course content on the Web. Some research claims that Web lectures are at least as efficient as regular lectures (LaRose & Gregg, 1997).

The developers of modern Web lecture technologies are driven by three reasonably different goals. For some developers (Abowd et al., 1998) Web lecture is the way to support regular "classroom" students by recording of "what happened in the classroom". Since these students can always re-play the lecture in their own pace, they can spend less time taking notes and more time understanding the actual lecture. For other teams (Bacher & Ottmann, 1996; Harris & DiPaolo, 1996; Severance, 1998; Stanford Online Team, 1998) the goal is "on-the-fly" authoring, i.e., providing a "fast copy" of a real lecture for students (mainly distance learners) who can't be in the classroom. Yet other developers aimed at providing an archival material aimed specially for distance education (Barbieri & Mehringer, 1997; Dannenberg, 1998). While the goal audience (classroom students only, distance students only or both) may influence different features of Web lectures, any existing Web-lecture systems can serve all three kinds of audiences.

Current technologies for producing Web lectures differ significantly in two aspects – the *content structuring* level and the *media level*. A high-quality "archival" Web lecture should allow a fine grained sequential and random access to the lecture content. First, the video/audio stream has to be divided into smallest meaningful chunks, which usually corresponds to a line or a piece of a slide. Chunking is important for slide synchronization, random access to lecture parts and retrievability. Synchronization means that each audio or video chunk has to be associated with a corresponding portion of the slide presentation. With *slide-level* synchronization a chink of narration is associated with the whole slide. With *line-level* synchronization a chunk is associated with each line (or a fragment) of the slide. Random access means that each chunk and slide (or slide line) can be individually addressable via a lecture table of contents. Retrievability means that the user can retrieve lecture chunks satisfying some criteria, for example, containing some keywords. To be retrievable by request, each chunk should be indexed with keywords or domain concepts. This is very important because it enables students having problems with later lectures or exercises to retrieve a helpful piece of teacher's explanation. Finally, an author may be willing to provide annotation, i.e., associate comments, references, and links to additional resources with any chunk of material. Annotations also could be used for finding a relevant lecture piece with full-text search (Stanford Online Team, 1998). On another side, a simple Web lecture provides no line-level synchronization, and no indexing.

Currently very few systems like MANIC (Stern *et al*., 1997a), CALAT (Nakabayashi *et al*., 1996), and JITL (Dannenberg, 1998) support line-level chunking and synchronization and even fewer (Dannenberg, 1998; Smeaton & Crimmons, 1997) support indexing and search. The problem is that reliable chunking, synchronization and indexing requires several hours of manual work for one hour of lecturing. Special advanced synchronized recording software used in AOF (Bacher & Ottmann, 1996), Sync-o-Matic 3000 (Severance, 1998), WLS (Klevans, 1997b), mStar (Synnes et al., 1998), and Class 2000 (Abowd et al., 1998) can to some extent solve the problem of synchronization. This software records the time when the presenter changes each slide or slide line and uses this data to change the slide or to highlight next line at the proper time when re-playing the presentation. Same technology could also be used to build a table of contents, however, it not able to solve the indexing problem and support retrievability.

We distinguish three media levels in existing Web lecture systems: recorded audio, recorded video, and high-quality video. The difference between recorded audio and recorded video is mainly the capabilities the clients. A number of teams specially restricted themselves to audio because their student use low-bandwidth Web connection and because they see little additional value provided by video in exchange for long waiting time. With the new streamline video technologies and increasing bandwidth recorded video will become more and more popular. The difference between recorded video and high-quality video is the required equipment and processing time. High-quality video requires special recording equipment, preferably a studio with a *blue screen* and several hours of processing time for one hour of lectures.

There is an obvious relationship between the goals of developing Web-lectures and the level of used technology. Web-lectures made as a record for classroom students (Abowd et al., 1998) may stay with audio or low-quality video and without fine grained chunking and random access (a teacher is available to solve any problems!) On-the-fly authoring with the goal to provide a copy of a classroom lecture for distance audience as soon as possible can afford only 2-3 hours of processing for one hour of lecturing (Stanford Online Team, 1998) and can use only minimal level of manual content processing and video processing. However, an archival Web lecture specially prepared for asynchronous distance education needs the highest level of content structuring and a high quality video. It currently requires many hours of processing for one hour of lecturing (Dannenberg, 1998).

Base level: A base level Web lecture is currently a sequence of static slides with a a separate audio or video narration file for each slide (Barbieri & Mehringer, 1997; Ingebritsen *et al*., 1997; LaRose & Gregg, 1997; Smeaton & Crimmons, 1997) or video (Bacher & Ottmann, 1996; Harris & DiPaolo, 1996; Radhakrishnan & Bailey, 1997; Stanford Online Team, 1998). Some evaluation (LaRose & Gregg, 1997) claims that such base level Web lectures are as efficient as ordinary face-to-face lectures. Base level Web lectures could be developed with COTS tools such as Microsoft Power Point.

State-of-the-art level: The requirement for state-of-the-art lectures is in-slide synchronization with audio and video. It could be either authored synchronization as in MANIC (Stern et al., 1997a), CALAT (Nakabayashi et al., 1996), JITL (Dannenberg, 1998) or recorded time synchronization (Abowd et al., 1998; Bacher & Ottmann, 1996; Klevans, 1997a; Severance, 1998; Synnes et al., 1998). Authored synchronization is better since it allows a line-level access to the material, but it is much more expensive. Slides or even lines in state-of-the-art systems should be indexed or annotated to allow keyword or full-text search. State-of-the-art lectures are usually developed with special authoring tools (Abowd et al., 1998; Bacher & Ottmann, 1996; Klevans, 1997a; Severance, 1998; Synnes et al., 1998) or technologies (Dannenberg, 1998).

Research level: Research teams working on Web lectures are trying to extend Web lecture technologies in the following directions: developing more powerful authoring tools (Bacher & Ottmann, 1996; Dannenberg, 1998; Klevans, 1997a; Severance, 1998), adding advanced options for finding relevant audio/video chunks (Smeaton & Crimmons, 1997), adding adaptivity to user knowledge to systems with Web lectures (Nakabayashi et al., 1996; Stern et al., 1997a; Stern *et al*., 1997b; Stern, 1997), and using life-like intelligent agents ("virtual lectors") for presenting the material (André *et al*., 1997).

1.2.2 THE GUIDED TOUR

Guided tours or paths is a traditional feature in the area of hypertext systems (Trigg, 1988). A classic guided tour is a sequential path through complex hyperspace extended with some narration. Similarly to its museum predecessors, guided tours could be developed to introduce a user into a complex hyperspace or to express a specific aspect of the available information. Though first works on educational Web guided tours started as early as other trials to use Web for education (Hauswirth, 1995; Nicol *et al*., 1995), this technology was not very popular by now. Possible reasons are lack of material to reuse, legal problems with reusing of previously authored content, and limited applicability of guided tours in university-level education. In addition, a good guided tour requires a special authoring and delivery system. Currently, a few authoring systems for Web guided touts have been developed: FootSteps (Nicol et al., 1995), Hierarchical Guided Tours (Hauck, 1996), Walden Paths (Furuta *et al*., 1997; Shipman III *et al*., 1998; Shipman III *et al*., 1997), Ariadne (Jühne *et al*., 1998) and others (Hauswirth, 1995; Langenbach & Bodendorf, 1997; Wright & Jones, 1997). All these systems allow the tour author to specify a sequence of existing Web pages for the tour and to add a narration. Some systems also support hierarchical (Hauck, 1996) or other branching tours (Jühne et al., 1998). Most of tour development systems provide a graphical interface for designing a tour though some systems requires a tour to be written in a special text-based language (Hauswirth, 1995; Langenbach & Bodendorf, 1997). At the delivery time these systems support user navigation by providing a panel with bi-directional sequential navigation buttons (backward, forward, top), a table of contents, and (for a non-linear tour) a graphical map (Jühne et al., 1998). The narration and the navigation buttons are either shown in the same window with the content of the original WWW page (Furuta et al., 1997; Hauck, 1996; Hauswirth, 1995; Nicol et al., 1995) or in a separate window (Jühne et al., 1998; Wright & Jones, 1997). The latter is probably better from a copyright point of view. An important navigation feature or a guided tour-delivery mechanism is a tool (preferably a simple button) for the user to return back on the trail after "side excursions" to the pages outside the tour (Hauswirth, 1995; Langenbach & Bodendorf, 1997; Nicol et al., 1995). Until recently, a "tour server" with a CGI script was a feature of any tour delivery system. More recent systems rely more on JavaScript (Wright & Jones, 1997) or Java (Jühne et al., 1998; Langenbach & Bodendorf, 1997) functionality.

Since we know only a handful of reported guided tour systems we can't provide a well-grounded level taxonomy for guided tours. It is reasonably clear, though, that a state-of-the-art guided tours should provide a navigation overview (table of contents or map) and "back to the tour" feature, and be developed with special authoring systems. Recent studies report that Web guided tours is a promising educational tool (Shipman III et al., 1998). This technology may become very popular in the near future as soon as COTS authoring systems will appear and a copyright problem will be solved. Re-using existing material is already becoming a better option due to the increase of the amount of already authored material and improvement in Web search technologies. The progress with reusability authoring approach will probably make the guided tour technology very popular since a guided tour over a set of pre-authored resources is the fastest way to develop a customized course (Langenbach & Bodendorf, 1998).

2. Actively Engaging Students: Assessment and Learning-by-doing

Students learn, in part, by performing activities. To go beyond presentation-only courseware - learning-by-reading, learning-by-watching, etc. Web-based education systems facilitate students in *doing* something. There are two major goals in adding various student activities to an educational system. The first goal is assessment: the only reliable way to evaluate students' understanding and progress is to let them do something which could show this understanding and evaluate results. The second goal is learning-by-doing itself. It is through activities like problem solving and exploration that students can come to grips with the subject; intellectually make it their own. The difference between assessment and learning-by-doing activities is generally not clear-cut, for example, such typical activities as problem solving promote learning by-doing while the results could be used for reliable assessment. In current academic and Web-based education practice there is a clear distinction between two groups of activities that we will call *objective* and *applicative*. Objective activities (true/false questions, multiple choice questions, short-answer questions) are designed to check student understanding and involve little creativity. As a rule, objective activities are easily graded automatically. Applicative activities (we will call them exercises) involve students in serious problem solving, development, or exploration. In most of the cases applicative activities must be evaluated and graded by a human teacher or assistant. While objective activities tend toward evaluation and applicative activities are more learning oriented activities can more or less emphasize each of these two major goals. On one end of objective activity spectrum are those used solely for assessment: the results are going directly to the teacher and students not even provided feedback (a precondition of learning by doing). On the other end we have objective activity developed solely for self-assessment. The students are provided with comprehensive feedback and the teacher does not see the results at all. The same spectrum of goals could be observed for applicative activities. There are applicative activities issued solely for assessment and there are applicative activities aimed solely to promote learning-by-doing. Since current technologies for dealing with objective activities and applicative activities are quite different we will consider these types separate in the following subsections.

2.1 Assessment: Quizzes and Objective Activities

Objective tests and quizzes are among the most widely used and well-developed tools in higher education. A classic test is a sequence of reasonably simple questions. Each question assumes a simple answer that could be formally checked and evaluated as correct, incorrect, or partly correct (for example, incomplete). Questions are usually classified into types by the type of expected answer. Classic types of questions includes yes/no questions, multiple-choice/single-answer (MC/SA) questions, multiple-choice/multiple-answer (MC/MA) questions, and fill-in questions with a string or numeric answer. More advanced types of questions include matching-pairs questions, ordering-questions, pointing-questions (the answer is one or several areas on a figure) and graphing-questions (the answer is a simple graph). Also, each subject area may have some specific types of questions.

2.1.1 LIFE CYCLE AND ANATOMY OF QUIZZES AND QUESTIONS

To compare existing options we have analyzed the life cycle of a question in Web-based education (see Table Y). We divided the life cycle of a question into three stages: preparation (before active life), delivery (active life), and assessment (after active life). Each of these stages is further divided into smaller stages. For each of these stages we have investigated a set of possible support technologies.

Life of a question begins at *authoring* time. The role of WBE systems at the authoring stage is to support the author by providing a technology and a tool for question authoring. All authored questions (the content and the metadata) are *stored* in the system. The active life of a stored question starts when it is *selected for presentation* as a part of a test or quiz. This selection could be done statically by a teacher at course development time or dynamically by a system at run time (by probability or according to some cognitive model).

Next, the system *delivers* a question: it presents the question, it provides an interface for the student to answer; it gets the answer for evaluation. At the *assessment* stage, the system should do the following things: evaluate the answer as correct, incorrect, or partly correct, deliver feedback to student, grade the question and to record student performance.

Existing WBE tools and systems differ significantly on the type and amount of support they provide on each of the stages mentioned above. Simple systems usually provide partial support for a subset of the stages. The cutting-edge systems provide comprehensive support at all the listed stages. The power of a system and the extent of support provided is seriously influenced by the level of technology used at each of the main stages - preparation, delivery and assessment. Below we will analyze the currently explored options.

Before	During	After
Preparation: • Author • <u>Store</u> • Select	Delivery: • Present • <u>Interact</u> • Get the answer	Assessment: • Evaluate • <u>Grade and record</u> • Deliver feedback

Table Y. Life cycle stages of a test question. The choice of technology for storing, interacting and recording results usually determines the "testing power" of a WBE system.

2.1.2 PREPARATION STAGE

A question is created by a human author - a teacher or a content developer. A state-of-the-art question has the following components: the question itself (or *stem*), a set of possible answers, an indication which answers are correct, a type of the interface for presentation, question-level feedback that is presented to the student regardless of the answer, and specific feedback for each of the possible answers. Questions used primary for self-assessment may include additional scaffolding information such as "hint" or "help" (Raineri et al., 1997). Finally, an author may provide metadata such as topics assessed, keywords, the part of the course a test belongs to, question weight or complexity, allowed time, number of attempts, etc. This metadata could be used to select a particular question for presentation as well as for grading the answer.

The options for *authoring support* usually depends from the technology used for storing an individual question in the system. Currently, we could distinguish two different ways to store a question, which we call *presentation format* and *internal format*. In WBE context, storing a question in presentation format means storing it as a piece of HTML code (usually, as an HTML form). Such questions could be also called static questions. They are "black boxes" for a WBE system: It can only present static questions "as is". The authoring of this type of questions is often not supported by a WBE system. It could be done in any of HTML authoring tools.

Storing a question in an internal format usually means storing it in a database record where different parts of the question (stem, answers, and feedback) are stored in various fields of this record. A question as seen by a student is generated from the internal format at the delivery time. Internal format opens the way for more flexibility: the same question could be presented in different forms (for example, fill-in or multiple choice) or with different interface features (for example, radio buttons or selection list). Options in multiple choice questions could be shuffled

(Carbone & Schendzielorz, 1997). It provides a higher level of individualization. This is pedagogically useful and decreases the possibility of cheating. There are two major ways for authoring questions in internal format: a form-based *graphical user interface* (GUI) or a special *question marckup language* (Brown, 1997; Campos Pimentel *et al*., 1998; Hubler & Assad, 1995). Each of these approaches has its benefits and drawbacks. Currently, a GUI approach is much more popular. It is used by all advanced commercial WBE systems such as (Blackboard, 1999; Question Mark, 1998; WBT Systems, 1999; WebCT, 1999). Note, however, that some WBE systems use GUI authoring approach but do not store questions in internal format. Instead, these systems generate HTML questions" right away" and store them as static questions

The simplest option for *question storage* is *a static test or quiz*, i.e., a static sequence of questions. The quiz itself is usually represented in plain HTML form and authored with HTML-level authoring tools. Static tests and quizzes are usually "hardwired" into some particular place in a course. One problem with this simplest technology is that all students get the same questions at the same point in the course. Another problem is that each question hardwired into a test is not reusable. A better option for question storage is a *hand-maintained pool of questions*. The pool could be developed and maintained by a group of teachers of the same subject. Each question in a question pool is usually static, but the quizzes are more flexible. Simple pool management tools let the teacher re-use questions; all quizzes may be assembled and added to the course pages when it is required. This is what we call *authoring time flexibility*. The same course next year, a different version of the course, or sometimes even different groups within the same course may get different quizzes without the need to develop these quizzes from scratch.

An even better option is to turn a hand-maintained pool into a database of questions. A database adds what we call *delivery time flexibility*. Unlike a hand-maintained pool, a database is formally structured and is accessible by the delivery system. With a database of questions not only the teacher can assemble a "quiz-on-demand", the system itself can generate a quiz from a set of questions. Naturally, the questions could be randomly selected and placed into a quiz in a random order (Asymetrix, 1999; Brown, 1997; Byrnes *et al*., 1995; Carbone & Schendzielorz, 1997; Ni *et al*., 1997; Radhakrishnan & Bailey, 1997; WBT Systems, 1999; WebCT, 1999). As a result, all students may get personalized quizzes (a thing that a teacher can not realistically provide manually) significantly decreasing the possibility of cheating. Note that implementation of a database of questions does not require the use of a commercial database management system. Advanced university systems like QuestWriter (Bogley et al., 1996) or Carnegie Mellon Online (Rehak, 1997a) and many commercial systems such as TopClass (WBT Systems, 1999) or LearningSpace (Lotus, 1999) use full-fledge databases such as ORACLE or Lotus Notes for storing their pools of question in internal format. However, there are systems which successfully imitate a database with the UNIX file system using specially structured directories and files (Byrnes et al., 1995; Gorp & Boysen, 1996; Merat & Chung, 1997).

A problem for all systems with computer-generated quizzes is how to ensure that these quizzes include a proper set of questions. The simplest way to achieve it is to organize a dedicated question database for each lesson. This approach, which is, for example, used in WebAssessor (ComputerPREP, 1998), reduce question reusability between lessons. More advances systems like TopClass (WBT Systems, 1999) can maintain multiple pools of and can use several pools for generating each quiz. With this level of support a teacher can organize a pool for each topic or each level of question complexity and specify the desired number of questions in a generated quiz to be taken from each pool.

A database of questions in internal format is currently a state-of-the art storage technology. Research teams are trying to advance it in three main directions. One direction is related to different kinds of *parameterized questions* as in CAPA (Kashy *et al*., 1997), EEAP282 (Merat & Chung, 1997), WebTester (Sapir, 1999), OES (Bryc & Pelikan, 1999), WebAssign (Titus et al., 1998) or Mallard (Brown, 1997; Graham et al., 1997). This allows one to create an unlimited number of tests from the same set of questions and can practically eliminate cheating (Kashy et al., 1997). The second direction of research is related to question metadata. If the system knows a little bit more *about* the question (for example, type, topics assessed, keywords, part of the course a test belongs to, weight or complexity) then the system can generate customized and individualized quizzes by author's or system's request. This means that the authors could specify various parameters for the quiz their student needs at some point of the course: total number of questions, proportion of questions of specific types or for specific topics, difficulty, etc., and the system will generate a customized quiz on demand (that is still randomized within the requirements) (Byrnes et al., 1995; Merat & Chung, 1997; Rehak, 1997a; Rios *et al*., 1998). This option is definitely more powerful than simple randomized quizzes. Systems that make extensive use of metadata really "know" about the questions and their functionality. The third direction of research is the adaptive sequencing of questions. This functionality is based on an overlay student model which separately represents student knowledge of different concepts and the topics of the course. Intelligent systems such as ELM-ART (Weber *et al*., Submitted), Medtec (Eliot et al., 1997), (Lee & Wang, 1997), SIETTE (Rios et al., 1998), Self-Learning Guide (Desmarais, 1998; Khuwaja *et al*., 1996) can generate challenging questions and tests adapted to the student level of knowledge as well as reduce the number of questions required to assess the students state of knowledge.

2.1.3 DELIVERY STAGE

The interaction technology used to get an answer from the student is one of the most important parameters of a WBE system. It determines all delivery options and influences authoring and evaluation. Currently, we distinguish five technologies: HTML links, HTML/CGI forms, scripting language, plug-in, and Java.

HTML links is a simple interaction technology that presents a set of possible answers as list of HTLM links. Each link is connected to a particular feedback page. The problems here are that questions are hard to author (because question logic must be hardwired into course hypertext) and that it supports only yes/no and MC/SA questions. This technology was in use in the early days of WBE when more advanced interaction technologies like Common Gateway Interface (CGI), JavaScript or Java were not established (Holtz, 1995).

The most well-established technology for Web testing which is used now in numerous commercial and university-grown systems is a combination of HTML forms and CGI-compliant evaluation scripts. HTML forms are very well suited for presenting main types of questions. Yes/no and MC/SA questions are represented by radio buttons, selection lists, pop-up menus, MC/MA questions are represented by multiple selection lists or checkboxes. Fill-in questions are implemented with input fields. More advanced questions such as matching pairs or ordering can also be implemented using forms. In addition, hidden fields can be successfully used to hold additional information about the test that a CGI script may need. There are multiple benefits of using server-side technology such as form/CGI technology and a similar server-side map technology that can be used for implementing graphical pointing questions. Test development is relatively simple and can even be done with HTML authoring tools. Sensitive external information which is required for test evaluation (such as question parameters, answers, feedback) may be safely stored on the client side preventing students from stealing the question (the only external information which is required in a well-developed system to evaluate a test is the test ID and the student ID). Server-side evaluation makes all assessment time functions (such as recording results, grading, providing feedback) easy to implement. In fact, the same server-side evaluation script could perform all these functions. The main problem of server-side technology is its low expressive power. It is well suited only for presenting basic types of tests. More advanced types of tests as well as more interactive types of tests (for example, tests which involve drag-and-drop activities) can not be implemented with pure sever-side technology. Authoring questions with server-side evaluation is tricky because a question's functionality is spread between its HTML presentation (either manually authored or generated) and a CGI evaluation script. Another serious problem is CGI-based questions do not work when a user's connection to the server is broken or very slow.

A newer technology for question delivery and evaluation is JavaScript (McKeever *et al*., 1997). The interface provided by the JavaScript interaction technology is similar to the one of form/CGI technology. At the same time, JavaScript functionality supports more advanced, interactive questions, for example, selection of a relevant fragment in a text. With pure JavaScript technology all data for question evaluation and feedback as well as evaluation program should be stored as a part of the question text. It means that a JavaScript question can work in standalone mode. It means that the question is self-sufficient: everything for presentation and evaluation is in the same file, and is a very attractive option for authoring. But it also means that students can access the source of the question and crack it. Also, with pure JavaScript evaluation technology there is no way for recording the results and grades. With all the above features JavaScript technology is a better choice for self-assessment tests than for assessments used in grading. We think the proper place for JavaScript in WBE is in a hybrid JavaScript/server technology. With this technology JavaScript can be used to present more types of questions, do it more interactively and with compelling user interfaces leaving evaluation and recording to be done by traditional CGI for reasons of security (ComputerPREP, 1998; WebCT, 1999).

A higher level of interface freedom can be achieved by using a plug-in technology.[10] The classic example of serious use of this technology in education is the Shockwave player (Macromedia, 1999b) which can run multimedia presentations prepared with Macromedia authoring tools. Currently, Shockwave technology is used in WBE mainly for delivering "watch-only" animations, but this technology is more powerful. In fact, a variety of very attractive Shockwave-deliverable questions could be developed using Macromedia tools with relatively low effort. Some examples could be provided by AST (Specht *et al*., 1997) and Medtec (Eliot et al., 1997). The negative side is the same as with JavaScript: recording assessment results requires connection to the server. Until recently, Shockwave provided no Internet functionality and its users had to apply special techniques (e.g. saving evaluation results in a temporal file). Due to Shockwave communication problems, some teams that started with Shockwave migrated later to more powerful Java technology (Eliot et al., 1997). Currently, after the release of the Authorware Web Player (Macromedia, 1999a) a special WBE-targeted update of Shockwave player, this technology becomes an attractive platform platform for delivering various self-assessment questions.

The highest level technology for question delivery is provided by Java. An important advantage of Java is that it is a complete programming language designed to be integrated with browser functionality and the Internet. Java combines connectivity of form/CGI technology and the interactivity of Shockwave and JavaScript. Any question interface can be developed with Java, and, at the same time, Java-made questions can naturally communicate with the browser as well as with any Internet object (a server or a Java application). Examples of systems which heavily use Java-based questions are FLAX (Routen et al., 1997), NetTest (Ni et al., 1997), Mallard (Graham & Trick, 1997), and Medtec (Eliot et al., 1997). Developing question interfaces with Java is more complicated than with form/CGI technology and it is not surprising that all the examples mentioned above were produced by advanced teams of computer science professionals. However, the complexity will not stop this technology. Java is currently the way to implement a variety of question types non-implementable with form/CGI technology such as multiple pointing questions, graphing questions, and specialized types of questions. Developing Java-based questions can become suitable for ordinary authors with the appearance of Java based authoring systems (Ni et al., 1997; Routen et al., 1997).

2.1.4 ASSESSMENT STAGE

As we noted, the choice of interaction technology significantly influences *evaluation* options. Evaluation is the time when an answer is judged as correct, incorrect, or partially correct (for example, incomplete). Usually, correct and incorrect answers are provided at authoring time, so evaluation is either hardwired into the question like in MC/SA questions, or performed by simple comparison (in fill-in questions). There are very few cases that require more advanced evaluation technology. In some domains correct answers may not be literally equal to a stored correct answer. Examples are a set of unordered words, a real number, a simple math expression (Bryc & Pelikan, 1999; Holtz, 1995; Hubler & Assad, 1995; Xiao, 1999). In this situation special comparison programs are required. Some systems includes special evaluation modules for

[10] Plug-in technology enables independent vendors to extend the browser functionality by developing specially structured programs called *plug-ins*. At start-up time, a browser loads all plug-ins located in a special directory and they become parts of the browser code.

advanced answer matching (Hubler & Assad, 1995). Other systems rely on a "domain expert" such as the Lisp interpreter for Lisp programming (Weber & Specht, 1997) or a computer algebra system for algebra domain (Antchev et al., 1995; Pohjolainen *et al*., 1997; Sapir, 1999) to evaluate the answer. The first two evaluation options are very simple and could be implemented with any interface technology - even JavaScript could be used to write a simple comparison program. If more advanced computation is required (as in the case of intelligent answer matching) the choice is limited to full-function programming with either Java or a server side program using a CGI interface. If a "domain expert" is required for evaluation, the only option currently is to run a domain expert on the server side with a CGI-compliant gateway. In fact, a number of "domain expert" systems (for example, Mathematica computer algebra system) have a CGI gateway.

The usual options for the *feedback* include: simply telling if the answer is correct, not, or partially correct, giving correct answer, and providing some individual feedback. Individual feedback may communicate: what is right in the correct answer, what is bad in incorrect and partially incorrect answer, provide some motivational feedback, and provide information or links for remediation. All individual feedback is usually authored and stored with the question. A system that includes assessed concepts or topics as a part of question metadata can provide good remedial feedback without direct authoring since it "knows" what knowledge is missing and where it can be found. It means that the power of feedback is determined by authoring and storage technology. The amount of information presented as feedback is determined by the context. In self-assessment the student usually receives all possible feedback - the more the better. This feedback is a very important source of learning. In a strict assessment situation the student usually gets neither a correct answer, nor whether the answer is correct. The only feedback for the whole test might be the number of correctly answered questions in a test (Rehak, 1997a). This greatly reduces the student's chances for cheating and student's chances to learn. To support learning, many existing WBE systems make assessment less strict and provide more feedback trying to fight cheating by other means. The only way to combine learning and strict assessment is to use more advanced technologies such as parameterized questions (Brown, 1997; Bryc & Pelikan, 1999; Kashy et al., 1997; Merat & Chung, 1997; Titus et al., 1998) and domain-specific test generation (Eliot et al., 1997; Sapir, 1999; Weber & Specht, 1997) which can generate an unlimited number of questions. In this situation a WBE system can provide full feedback without promoting cheating.

If a test is performed purely for self-assessment then generating feedback could be the last duty of a WBE system in the "after-testing" stage. The student is the only one who needs so see test results. In the assessment context the last duty of a WBE system in the process of testing is to *grade student performance on a test and to record these data* for future use. Grades and other test results are important for teachers, course administrators, and students themselves (a number of authors noted that the ability to see their grades online is the most student-appreciated feature of a WBE system). Early WBE systems provided very limited support for a teacher in test evaluation. Results were either sent to the teacher by e-mail or logged into a special file. In both cases a teacher was expected to complete grading and recording manually: to process test results and grade them, to record the grades, and to ensure that all involved parties get access to data according university policy. This lightweight support is easy to implement and it does not require that teachers learn any new technology. For the latter reason this technology is still used as an option in some more advanced systems (Carbone & Schendzielorz, 1997). However, a system that provides no other options for grading and recording is now below a state-of-the-art. A state-of-the-art WBE system should be able to grade a test automatically and record test results in a database. It also should provide properly restricted[11] access to the grades for students, teachers, and administrators. Many university-level systems (Bogley et al., 1996; Brown, 1997; Carbone & Schendzielorz, 1997; Gorp & Boysen, 1996; Hubler & Assad, 1995; MacDougall, 1997; Ni et al., 1997; Rehak, 1997a) and almost all commercial level systems (Lotus, 1999; WBT Systems, 1999; WebCT, 1999)provide this option in a more or less advanced way. Less advanced systems usually store the grades in structured files and provide limited viewing options. Advanced systems

[11] Restrictions are usually determined by university policies. For example, a student may not be allowed to see grades of other students or a teacher could be allowed to change the automatically assigned grades.

use database technology to store the grades and provide multiple options for viewing the grades and other test performance results such as time on a test or a number of efforts made. Database technology makes it easy to generate various test statistics involving results of many students on many course tests. In a Web classroom, where student-to-student and student-to-teacher communication is limited, comparing statistics is very important for both - teachers and students to get the “feeling” of the classroom. For example, by comparing class average with personal grades a student can determine class rank. By comparing class grades for different tests and questions a teacher can find too simple, too difficult, and even incorrectly authored questions.

2.1.5 LEVELS FOR QUIZZES

Base level: A base-level WBE should satisfy three requirements: administer tests, grade tests and store results. Any Web-based testing technology that is unable to record testing results are below Base level. Self-assessment orientation can hardly be considered as an excuse for the absence of tools for recording and viewing grades. As it was noted above, even in a self-assessment context where tests are not expected to be graded there are multiple benefits of maintaining a record of student test results.

State-of-the-art level: To qualify for State-of-the-art a WBE system should provide several advancements over a base level system (Table Z). The system should support at least all the basic question forms: true/false, MC/SA, MC/MA, and fill-in. Authoring system should be provided - at least for the development of these basic question types. Questions should be stored in a pool that allows at least authoring-time flexibility in selecting questions for a test. Test results should be automatically graded and recorded in a database for further access by students and teachers. Grade-viewing tools allowing teachers to view grades in several basic ways should be provided. Competition between state-of-the-art systems is providing better authoring tools: test assembly from questions pools; multiple question types; options for restricting questions by time and repeated attempts; and multiple ways to view results.

Research level: Research systems investigate advanced Web-based testing options in several directions. Here is a summary of main research directions discussed above.

- *Metadata:* Several research teams are working on tools which can support the author in adding metadata to questions (Forte et al., 1996a; Forte et al., 1996b) and selection tools which either help the author in selecting a proper question from a pool at authoring time (Forte et al., 1996a; Forte et al., 1996b; Graf & Schnaider, 1997), or use metadata to generate a customized test on demand using criteria supplied by the author (Byrnes et al., 1995; Merat & Chung, 1997; Rehak, 1997b).
- *Variety-interactivity:* Extending variety and interactivity is accomplished using Java and similar technologies. It includes developing more interactive types of questions (Graham & Trick, 1997; Ni et al., 1997; Raineri et al., 1997) and developing authoring tools which support a greater variety of questions (Ni et al., 1997; Routen & Graves, 1997a; Routen et al., 1997). From another side, some teams are working on extending the variety of question types by providing more advanced evaluation technologies such as advanced comparison (Hubler & Assad, 1995; Raineri et al., 1997) and “domain experts” (Antchev et al., 1995; Bryc & Pelikan, 1999; Pohjolainen et al., 1997; Sapir, 1999; Weber & Specht, 1997; Xiao, 1999).
- *Parameterized questions:* This work (Brown, 1997; Bryc & Pelikan, 1999; Kashy et al., 1997; Merat & Chung, 1997; Sapir, 1999; Titus et al., 1998) covers several related issues: authoring systems and languages for creating parameterized questions, generation of a parameterized question, and evaluation.
- *Artificial intelligence:* Artificial intelligence technologies are being adapted for test generation from a knowledge base (Eliot et al., 1997; Specht et al., 1997; Weber et al., Submitted) and adaptive question sequencing (Desmarais, 1998; Eliot et al., 1997; Khuwaja et al., 1996; Lee & Wang, 1997; Rios et al., 1998).

	Basic	State-of-the art	Research
Types of questions	2-3 of basic types	All basic types	Greater variety

Authoring	COTF tools	Authoring tool or language	Authoring for advanced questions
Storage	Any	Pool, database	Enriched with metadata
Selection	Static	Authoring time or delivery time flexibility	Metadata for test assembling, intelligent sequencing
Interaction	CGI	CGI, some Java, restricted questions	Interactive questions, extensive use of Java
Evaluation	Simple matching	Simple and advanced matching	Intelligent comparison and domain experts
Grading/recording	Any form	Database	
Viewing	Simple	Advanced	

Table Z. Levels and features for Web-based testing

2.2 Learning-by-doing: exercises

A nature of learning-by-doing is to push students to do something which requires application of knowledge being taught: to write an essay or a program, to draw a diagram, to translate a phrase into a foreign language, to solve a problem, to run an experiment with a simulator. An exercise is aimed at achieving an expected result that could be evaluated, if required. Typically an *artifact* is produced. It could be a program, a problem solution, an essay, a set of simulation parameters - anything which could be created and evaluated. Still within these limits there are many parameters which distinguish different exercise settings. The size could differ from small, several minutes long activities to projects lasting days or weeks. An exercise could be assigned by a teacher or selected by the student. Teacher evaluation may or may not be expected.

From a technical point of view it is important to distinguish exercises where the result is a "executable" artifact (such as a program or simulation) from those where the result is static such as an essay. If an exercise result is executable the student can use the computer to self-check. The usual cycle of work with an executable artifact is: develop - run - view - submit - evaluate - grade. If a student is not satisfied with the result at any of these stages the process can be interrupted, returning to development. If the artifact is not executable, the only way to get a reliable evaluation is to submit it to the teacher. Here the cycle of work is simply develop - submit - evaluate - grade. An exercise with a "executable" result enables a student to perform many develop-run-view cycles before involving the teacher.

One of the ways to compare WBE systems is to classify them in regards to the support they could provide to the students on various stages of work with exercises. From this point of view, almost all WBE systems which support exercises fall into two groups: assessment systems and learning environments. Assessment systems assist students in submissions. It is expected that the desired artifact is developed outside the system. What the system provides is an interface to deliver the solution to a teacher - either an HTML form box to paste the artifact (or a part of it (Hitz & Kögeler, 1997)) if it could be represented as a text or an interface to upload a file with the solution. For these kinds of systems assessment is the goal and learning-by-doing is more a by-product.

In contrast, learning environments are more oriented towards learning-by-doing and involve little or no teacher assessment[12]. These systems are trying to assist the student on different stages of their work on an exercise. At the design stage learning environments provide an interface to

[12] This distinction is empirical. It is possible to have learning environments that are also rich in assessment. But such instances are just now coming onto the scene.

develop the solution such a simple editor or *grapher* (McKenna & Agogino, 1997), a structural or diagram editor (Suthers & Jones, 1997), or a domain-oriented design interface, for example, for developing a microprocessor program (Graham et al., 1997). Other kinds of design support environments includes "virtual labs" where the student can solve a case or design an experiment (EDesktop, 1997a; Faulhaber & Reinhardt, 1997; Johnson & Shaw, 1997; Johnson *et al.*, 1998; Pearce & Livett, 1997). At run time learning environments "execute" the designed artifact - a program, a mathematical expression (Xiao, 1999), an "experiment" and deliver a static result to be viewed, for example, simulation result (EDesktop, 1997b; Tardif & Zaccarin, 1997). Some learning environments are not limited to showing the final static result - they could show either continuous or interactive animation of a running artifact - a computer program (Bilska *et al.*, 1997; Brown & Najork, 1996; Brown & Najork, 1997; Brusilovsky et al., 1996; Campbell *et al.*, 1995; Ibrahim, 1994; Jehng *et al.*, 1997; Naps, 1997; Procopiuc *et al.*, 1996; Rodger, 1997), a mathematical object (Xiao, 1999), or a simulation (Hampel *et al.*, 1998; Kirkpatrick *et al.*, 1997; Marshall & Hurley, 1997; McKenna & Agogino, 1997; Milton *et al.*, 1997; Warendorf & Tan, 1997). At the evaluation stage some learning environments could evaluate developed artifacts and either simply judge it as correct/incorrect (Brusilovsky et al., 1996; Graham & Trick, 1997) providing a canned feedback or provide an intelligent diagnosis with explanations and analysis of student's misconceptions (Brusilovsky et al., 1996; Okazaki *et al.*, 1997; Yang & Akahori, 1997). Assessment-oriented environments can also calculate the grade (Foxley *et al.*, 1998). Most advanced intelligent learning environments could provide evaluation and assistance on any step of problem solving. Examples are ADELE (Johnson & Shaw, 1997; Johnson et al., 1998), D3WWW-Trainer (Faulhaber & Reinhardt, 1997), Belvedere (Suthers & Jones, 1997), and PAT Online (Ritter, 1997).

Currently assessment systems and learning environments constitute two different approaches to exercise implementation in WBE systems and we will consider them in more details separately. Potentially, any WBE system could combine assessment and learning-by-doing support. However, very few existing systems combine features of assessment systems and learning environments. Some assessment systems provide more support for development, for example, letting a student edit a program, run it, and see the results in a browser window (Barbieri & Mehringer, 1997; Hitz & Kögeler, 1997). From another side, some learning environments can formally evaluate the developed artifact. Also, all systems with automatic evaluation such as WebCeilidh (Foxley et al., 1998) or ELM-ART (Brusilovsky et al., 1996) and all systems with intelligent diagnosis (Brusilovsky et al., 1996; Okazaki et al., 1997; Suthers & Jones, 1997; Yang & Akahori, 1997) are bridging the gap. Any of the above system provides evaluation support without teacher's involvement. These systems promote learning-by-doing by increasing the number of design-evaluate cycles even for the domains without executable artifacts (Okazaki et al., 1997; Suthers & Jones, 1997; Yang & Akahori, 1997).

2.2.1 ASSESSMENT SYSTEMS

The functionality of learning-by-doing assessment systems is similar to that of test systems. It includes authoring, selection, delivery, submission, evaluation, feedback and grading. For most of these steps there is no difference between assessment and test systems, so whatever was discussed about tests in the previous chapter is also relevant for assessment systems. Authoring and delivery for exercises is usually simple, authored as HTML text enhanced with figures (and probably some media items) and delivered with a form box or another interface for submitting the result. Existing assessment systems differ from test systems and from each other in respect to the type and level of support provided on submitting and evaluation stages. Some systems still use non-web options for submitting the answer - it is either sent by regular or custom e-mail to the teacher, or uploaded to a special site by FTP. Regular options are either a form/CGI interface with a box for entering the solution or a simple browser interface for uploading the solution. The first option is better, but it supports only text solutions. If the solution is, for example, an MS Word file, uploading is the only option. More advanced systems are trying to combine some development support with submitting interface, for example, compile the program to be submitted, to run it, and to view results (Barbieri & Mehringer, 1997; Hitz & Kögeler, 1997; Mehringer &

Lifka, 1998). Some other systems make special provisions to integrate a Web system with a stand-alone problem-solving environment, for example, a programming environment. The student is expected to develop the solution in this stand-alone system, copy it, and paste into the browser window (Lawrence-Fowler & Fowler, 1997). To simplify the work with two applications, a WBE system could provide special buttons on an exercise page to launch all required stand-alone systems such as an editor or a compiler. (Westhead, 1996).

Regular evaluation options for both base level and state-of-the-art level systems is "manual" evaluation and feedback generation written by a teacher. More advanced systems provide better interfaces for the teacher to evaluate submitted work. On the same screen the teacher can view the result and enter both grade and feedback. Less advanced systems expect the teacher to maintain a paper log with grades and enter them all in once.

Main research efforts here are centered on automatic evaluation. So far, the most visible progress is achieved for mathematical exercises where an evaluation module could rely on modern computer algebra systems available via a Web interface. A number of existing Web-based systems in such areas as algebra or calculus are able to evaluate automatically student solutions to a wide range of exercises – from to simple questions to relatively big problems (Bryc & Pelikan, 1999; Pohjolainen et al., 1997; Sapir, 1999; Xiao, 1999). Automatic evaluation for computer programming is another well investigated area. There are some pre-Web systems like Ceilidh (Benford *et al*., 1993) or TRY (Reer, 1988) and Web-based systems like WebCeilidh (Foxley et al., 1998; Marshall & Hurley, 1996b; Marshall & Hurley, 1997) and ELM-ART (Brusilovsky et al., 1996) which evaluate student programs by running them against test data. Similar technology could be used for automatic evaluation of other executable artifacts. Web-based systems in areas with non-executable artifacts have to rely on artificial intelligence techniques for automatic evaluation and diagnosis (Okazaki et al., 1997; Suthers & Jones, 1997; Yang & Akahori, 1997).

Base level: A base-level WBE should satisfy very modest requirements: being able to deliver exercise results to the teacher for evaluation (e-mail is a legitimate option) and providing an interface for entering feedback and grades. As in the case of tests, there should be an interface for viewing grades.

State-of-the-art level: A state-of-the-art WBE should provide a Web-based student interface for submitting results and an integrated teacher interface for evaluation and entering grades. Special provisions (at least hints and help) should be made to integrate a WBE and a stand-alone problem-solving environment (such as a programming environment). Alternatively, for executable artifacts some interface for running a solution and viewing results in a browser should be provided. Similar to state-of-the-art test systems, state-of-the art learning-by-doing assignment systems should also provide authoring time or run-time flexibility for selecting exercises from a pool of exercises as well as various grade viewing options. Finally, advanced systems should integrate communication facilities with exercise interfaces to let a student cooperate with other students in working on the exercise or to discuss it with a teacher.

Research level: As we already note, main research direction is automatic evaluation which includes auto-checking and auto-grading systems in mathematics (Bryc & Pelikan, 1999; Pohjolainen et al., 1997; Sapir, 1999; Xiao, 1999) and programming (Brusilovsky et al., 1996; Foxley et al., 1998; Marshall & Hurley, 1996b; Marshall & Hurley, 1997), systems which could check various properties of a developed artifact such as plagiarism or compliance to standards (Foxley et al., 1998), and systems which can do intelligent analysis of solutions (Brusilovsky et al., 1996; Okazaki et al., 1997; Yang & Akahori, 1997).

2.2.2 LEARNING ENVIRONMENTS

Unlike learning-by-doing assessment systems, existing Web-based learning environments are rarely oriented toward assessment of user performance. The goal here is to promote learning-by-doing. Moreover every learning environment should allow students not only to work on exercises which require some kind of result, but also support free exploration for example, playing with existing examples. In the latter case no assessment is possible because there is no result to assess. Learning environments are a well-investigated domain in pre-Web education. The main problem for developers of Web-based learning environments lies not on the level of ideas but on the level of technology. The Web is still a less than perfect platform for implementing highly interactive learning environments. Here we just provide a review of various technologies which could be used for implementing Web-based learning environments.

To start with, the oldest and the easiest to implement rely on "helpers". A helper is an application which is called by the browser to work with a special type of file. The keys here are to declare all design artifacts in a particular area as files of a special kind and to assign as a helper for this type of stand-alone design environment such as a CASE deign tool (Lawrence-Fowler & Fowler, 1997), LogicWorks (McCartney *et al*., 1997), or simply a text editor (Westhead, 1996). After that, clicking on a link could launch the desired environment with an example to play or a partial solution already loaded. Alternatively, clicking on a link could just launch the required helper and an example to play with has to be copied from the Web page and pasted into the stand-alone environment as in Mathlab (Tilbury & Messner, 1997) or Pascal program (Lawrence-Fowler & Fowler, 1997) . This technology is simple and efficient. The only technical problem here is that all client computers must be properly configured.

The first "pure Web" learning environments were developed with form/CGI technology for algorithm animation (Campbell et al., 1995; Ibrahim, 1994), and program visualization (Brusilovsky et al., 1996). Currently form/CGI technology can not be considered relevant for implementation of interactive animations because each step of interaction requires communication with the server. However, if the environment is not providing animated interactive simulation but only shows the result of executing a program (Brusilovsky et al., 1996) or an experiment (EDesktop, 1997a; EDesktop, 1997b), then this server-based technology could be very relevant.

Currently, with development and maturation, Java is becoming a dominant technology for developing Web-based learning environments, mainly for supporting interactive simulations and animations. There are multiple examples using Java for interactive and one-shot simulations in such fields as data structures (Marshall & Hurley, 1997; Warendorf & Tan, 1997), microprocessor simulation (Graham et al., 1997; Graham & Trick, 1997), algorithm animation (Ben-Ari, 1997; Bilska et al., 1997; Brown & Najork, 1996; Brown & Najork, 1997; Procopiuc et al., 1996; Rodger, 1997), simulations for chemistry (Milton et al., 1997), engineering (Kirkpatrick et al., 1997), physics (McKenna & Agogino, 1997) and electronics (Tardif & Zaccarin, 1997). Java could also be used to provide full-scale Web labs for solving a case (Faulhaber & Reinhardt, 1997; Johnson & Shaw, 1997), or running an experiment (Pearce & Livett, 1997). In some cases when a simulation engine already exists in a stand-alone form or is just too big to be completely implemented in Java, the best solution is a combination of Java and CGI technologies where Java is used to support interaction (Faulhaber & Reinhardt, 1997) or to render animation (Naps, 1997) while cycle intensive processes are done on the server side. A very unique example of connecting a Java interface with an external "simulation engine" is described in (Kirkpatrick et al., 1997). Here the simulation engine is a real device and Java provides a safe interface to play with it over the Web.

With the maturity of Macromedia's Shockwave technology traditional interactive multimedia authoring tools developed by this company are becoming a very attractive alternative for developing Web labs. While Java stays as the only tool for developing complex Web-based learning environments, Shockwave technology supported by excellent authoring tools could often provide a better choice for developing small and medium size interactive simulations and exercises.

Web-based learning environments are a new and labor-consuming. Most of the developments in this area should be considered as research. The only systems that could be placed on the state-of-the-art level are simple Java-based or Shockwave-based animations and simulations. With the current level of authoring support, any reasonable development team could develop it. We find no examples of Base level Web-based learning environments.

3. Conclusion

This paper provides an analysis of a group of tools required for building a virtual university, which we call course delivery tools. We have tried to analyze various aspects of currently existing tools and technologies. We have also tried to provide some kind of grading by placing various tools and systems into three levels of advancement. We expect that this review could help Virtual University practitioners to judge various tools available for course delivery. We also hope that the paper will be useful for researchers on Web-based education helping them to find a place of their work in the overall picture, preventing some teams from "re-inventing the wheel" and showing most attractive and demanding directions of research for new teams.

References

Abou Khaled O, Pettenati M C, Vanoirbeek C & Coray G *1998* MEDIT: a Distance education prototype for teaching and learning. *WebNet'98, World Conference of the WWW, Internet, and Intranet*, Orlando, FL, November 7-12, 1998, *pp. 503-508*

Abowd G D, Atkeson C G, Brotherton J, Enqvist T, Gulley P & LeMon J *1998* Investigating the capture, integration and access problem of ubiquitous computing in an educational settings. *CHI'98*, Los Angeles, CA, 18-23 April, 1998, *pp. 440-447*

Adams W J & Carver C A, Jr. *1997* The effects of structure on hypertext design. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 1-6*

Ahanger G & Little T D C *1997* Easy Ed: An integration of technologies for multimedia education. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 15-20*

André E, Rist T & Müller J *1997* WebPersona: A Life-Like Presentation Agent for Educational Applications on the World-Wide Web. *Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education*, Kobe, Japan, 18 August 1997, *pp. 78-85*, http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Andre/Andre.html

André E, Rist T & Müller J *1998* Guiding the user through dynamically generated hypermedia presentations with a life-like character. *International Conference on Intelligent User Interfaces, IUI'98*, San Francisco, CA, January 6-9, 1998, *pp. 21-28*

Anlon *1998* IntraKal, Anlon Systems, Inc., *http://www.anlon.com/aboutset.html*

Antchev K, Luhtalahti M, Multisilta J, Pohjolainen S & Suomela K *1995* A WWW Learning Environment for Mathematics. *4th International World Wide Web Conference*, Boston, USA, December 11-14, 1995, http://www.w3.org/pub/Conferences/WWW4/Papers/89/

Asymetrix *1999* Librarian, Asymetrix Learning Systems, Inc., Bellevue, WA, *http://www.ASYMETRIX.com/products/librarian/*

Avilar *1998* WebMentor, Avilar Technologies, Inc., Laurel, MA, *http://avilar.adasoft.com/avilar/msubfrm.html*

Bacher C & Ottmann T *1996* Tools and services for authoring on the fly. *ED-MEDIA'96 - World conference on educational multimedia and hypermedia*, Boston, MA, June 17-22, 1996, *pp. 7-12*

Barbieri K & Mehringer S *1997* Techniques for enhancing Web-based education. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 27-32*

Ben-Ari M *1997* Distributed algorithms in Java. *ACM SIGCSE bulletin 29 (3): 62-64*

Benford S, Burke E & Foxley E *1993* Learning to construct quality software with the Ceilidh system. *Software Quality Journal 2: 177-197*

Bilska A O, Leider K H, Procopiuc M, Procopiuc O, S. H. Rodger, Salemme J R & Tsang E *1997* A Collection of Tools for Making Automata Theory and Formal Languages Come Alive. *SIGCSE Bulletin 29 (1): 15-19*

Blackboard *1999* CourseInfo, Blackboard Inc., *http://company.blackboard.com/CourseInfo/index.html*

Bogley W A, Dorbolo J, Robson R O & Sechrest J A *1996* New pedagogies and tools for Web based calculus. *WebNet'96, World Conference of the Web Society*, San Francisco, CA, October 15-19, 1996, *pp. 33-39*

Brok E *1997* Hypertext and semantic nets: Experiences with modular authoring. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 106-111*

Brown D J *1997* Writing Web-based questions with Mallard. *FIE'97, Frontiers in Education Conference*, Pittsburgh, PA, November 5-8, 1997, *pp. 1502*

Brown D J & Mallard Team *1998* Mallard, University of Illinois at Urbana-Champaign, Urbana, IL, *http://www.cen.uiuc.edu/Mallard/*

Brown M H & Najork M A *1996* Collaborative Active Textbooks: A Web-Based Algorithm Animation System for an Electronic Classroom. *IEEE Symposium on Visual Languages (VL'96)*, Boulder, CO, Sept. 3-6, 1996, *pp. 266-275*, http://www.research.digital.com/SRC/JCAT/vl96

Brown M H & Najork M A *1997* Collaborative Active Textbooks. *Journal of Visual Languages and Computing 8 (4): 453-486*

Brusilovsky P, Eklund J & Schwarz E *1998* Web-based education for all: A tool for developing adaptive courseware. *Computer Networks and ISDN Systems 30 (1-7): 291-300*

Brusilovsky P & Schwarz E *1997* Concept-based navigation in educational hypermedia and its implementation on WWW. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 112-117*

Brusilovsky P, Schwarz E & Weber G *1996* ELM-ART: An intelligent tutoring system on World Wide Web. In: Frasson C, Gauthier G & Lesgold A (eds.) *Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 1086* Springer Verlag, Berlin, *pp. 261-269*

Bryc W & Pelikan S *1999* Online Exercises System, University of Cincinnati, Cincinnati, OH, *http://math.uc.edu/onex/demo.html*

Byrnes R, Debreceny R & Gilmour P *1995* The Development of a Multiple-Choice and True-False Testing Environment on the Web. *Ausweb95: The First Australian World-Wide Web Conference*, Ballina, Australia, April 30 - May 2, 1995, http://elmo.scu.edu.au/sponsored/ausweb/ausweb95/papers/education3/byrnes/

Calvi L & De Bra P *1997* Using dynamic hypertext to create multi-purpose textbooks. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 130-135*

Campbell J K, Hurley S, Jones S B & Stephens N M *1995* Constructing educational courseware using NCSA Mosaic and the World-Wide Web. *Computer Networks and ISDN Systems 27 (6): 887-895*

Campos Pimentel M d G, dos Santos Junior J B & de Mattos Fortes R P *1998* Tools for authoring and presenting structured teaching material in the WWW. *WebNet'98, World Conference of the WWW, Internet, and Intranet*, Orlando, FL, November 7-12, 1998, *pp. 194-199*

Campus America *1998* The Learning Manager, Campus America, Calgary, Canada, *http://www.campuscan.com/*

Carbone A & Schendzielorz P *1997* Developing and integrating a Web-based quiz generator into the curriculum. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 90-95*

Carver C A, Howard R A & Lavelle E *1996* Enhancing student learning by incorporating student learning styles into adaptive hypermedia. *ED-MEDIA'96 - World Conference on Educational Multimedia and Hypermedia*, Boston, MA, June 17-22, 1996, *pp. 118-123*

Centra *1998* Symposium, Centra Software, Inc., Lexington, MA, *http://www.centra.com/product/index.html*

ClassNet Team *1998* ClassNet, Iowa State University, *http://classnet.cc.iastate.edu/*

ComputerPREP *1998* WebAssessor, ComputerPREP, Inc, Phoenix, AZ, *http://www.webassessor.com*

CourseWeb Team *1997* CourseWeb, Pennsylvania State University, State College, PA, *http://projects.cac.psu.edu/CourseWeb/*

Danielson R *1997* Learning styles, media preferences, and adaptive education. *Workshop "Adaptive Systems and User Modeling on the World Wide Web" at 6th International Conference on User Modeling, UM97*, Chia Laguna, Sardinia, Italy, June 2, 1997, *pp. 31-35*, http://www.contrib.andrew.cmu.edu/~plb/UM97_workshop/Danielson.html

Dannenberg R *1998* Just-in-Time Lectures, Carnegie Mellon University, Pittsburgh,

De Bra P *1997* Teaching Through Adaptive Hypertext on the WWW. *International Journal of Educational Telecommunications 3 (2/3): 163-180*

Desmarais M C *1998* Self-Learning Guide, CRIM, Montreal, *http://www.crim.ca/hci/demof/gaa/introduction.html*

Docent *1998* Docent, Docent, Inc., Mountain View, CA, *http://www.docent.com/contact/index.html*

Dracos N *1998* LaTeX2HTML, University of Leeds, *http://www.cbl.leeds.ac.uk/nicos/tex2html/doc/latex2html/latex2html.html*

EDesktop *1997a* Virtual EarthQuake, California State University, Los Angelos, *http://vearthquake.calstatela.edu/edesktop/VirtApps/VirtualEarthQuake/VQuakeIntro.html*

EDesktop *1997b* Virtual Flylab, California State University, Los Angelos, *http://vearthquake.calstatela.edu/edesktop/VirtApps/VflyLab/IntroVflyLab.html*

Eisenstadt M & Domingue J *1998* KMi Studium, Open University, Milton Keynes,

Eklund J *1995* Cognitive models for structuring hypermedia and implications for learning from the world-wide web. *Ausweb95: The First Australian World-Wide Web Conference*, Ballina, Australia, April 30 - May 2, 1995, *pp. 111-116*, http://elmo.scu.edu.au/sponsored/ausweb/ausweb95/papers/hypertext/eklund/index.html

Eliot C, Neiman D & Lamar M *1997* Medtec: A Web-based intelligent tutor for basic anatomy. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 161-165*

Faulhaber S & Reinhardt B 1997 *D3-WWW-Trainer: Entwiklung einer Oberfläche für die Netzanwendung*. München: Technische Universität München, 31-40.

Fisher B, Conway K & Groeneboer C *1997* Virtual-U developmental plan: Issues and process. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 352-357*

Forte E, Forte M W & Duval E *1996a* ARIADNE: A framework for technology-based open and distance lifelong education. *7th EAEEIE International Conference "Telematics for Future Education and Training"*, Oulu, FInland, June 12-14, 1996, *pp. 69-72*

Forte E, Forte M W & Duval E *1996b* ARIADNE: A supporting framework for technology-based open and distance lifelong education. *Educating the engineer for lofelong learning. SEFI Annual Conference '96*, Vienna, Austria, September 11-13, 1996, *pp. 137-142*

Foxley E, Benford S, Burke E & CEILLIDH Development Team *1998* WebCeilidh, University of Nottingham, *http://www.cs.nott.ac.uk/~ceilidh*

Freeman H & Ryan S *1997* Webmapper, a tool for planning, structuring and delivering courseware on the Internet. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 372-377*

Fröhlich P & Nejdl W *1997* A Database-Oriented Approach to the Design of Educational Hyperbooks. *Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education*, Kobe, Japan, 18 August 1997, *pp. 31-39*, http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Froehlich/Froehlich.html

Furuta R, Shipman III F M, Marshall C C, Brenner D & Hsieh H-w *1997* Hypertext paths and the World-Wide Web: Experience with Walden's paths. *Eight ACM International Hypertext Conference (Hypertext'97)*, Southampton, UK, April 6-11, 1997, *pp. 167-176*

Goldberg M W *1997a* CALOS: First results from an experiment in computer-aided learning of operating systems. *SIGCSE Bulletin 29 (1): 48-52*

Goldberg M W *1997b* Using a Web-based course authoring tool to develop sophisticated Web-based courses. In: Khan B H (ed.) *Web Based Instruction*. Educational Technology Publications, Englewood Cliffs, New Jersey, *pp. 307-312*

Goldberg M W *1997c* WebCT and first year: student reaction to and use of a web-based resource in first year computer science. *ACM SIGCSE bulletin 29 (3): 127-127*

Goldberg M W, Salari S & Swoboda P *1996* World Wide Web - course tool: An environment for building WWW-based courses. *Computer Networks and ISDN Systems 28: 1219-1231*

GoLive *1998* CyberStudio, GoLive Inc., *http://www.golive.com/*

Gorp M J V & Boysen P *1996* ClassNet: Managing the virtual classroom. *WebNet'96, World Conference of the Web Society*, San Francisco, CA, October 15-19, 1996, *pp. 457-461*

Graf F & Schnaider M 1997 *IDEALS MTS - EIN modilares Training System für die Zukunft*. München: Technische Universität München, 1-12.

Graham C R, Swafford M L & Brown D J *1997* Mallard: A Java Enhanced Learning Environment. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 634-636*

Graham C R & Trick T N *1997* An innovative approach to asynchronous learning using Mallard: Application of Java applets in a freshman course. *FIE'97, Frontiers in Education Conference*, Pittsburgh, PA, November 5-8, 1997, *pp. 238-244*

Hampel T, Keil-Slawik T, Ferber F & Müller W H *1998* How to bring cooperative structures and hypermedia into the field of technical mechanics? - Our experiences. *WebNet'98, World Conference of the WWW, Internet, and Intranet*, Orlando, FL, November 7-12, 1998, *pp. 380-386*

Harasim L, Calvert T & Groenboer C *1997* Virtual-U: A Web-based system to support collaborative learning. In: Khan B H (ed.) *Web Based Instruction*. Educational Technology Publications, Englewood Cliffs, New Jersey, *pp. 149-158*

Harlequin *1998* WebMaker™ , Harlequin Inc, Cambridge, MA, *http://www.harlequin.com/products/dpp/webmaker/*

Harris D & DiPaolo A *1996* Advancing asynchronous distance education using high-speed networks. *IEEE Transactions on Education (August)*

Hauck F J *1996* Supporting Hierarchical Guided Tours in the World Wide Web. *5th International World Wide Web Conference*, Paris, France, May 6-10, 1996, http://www5conf.inria.fr/fich_html/papers/P30/Overview.html

Hauswirth M *1995* Pooh's guided tour service Wien, *http://www.infosys.tuwien.ac.at/GuidedTour/GuidedTour.html*

Hector C *1998* RTFtoHTML 4.10, Sunrise Packaging Inc., Blaine, MN, *http://www.sunpack.com/RTF/*

Hitz M & Kögeler S *1997* Teaching C++ on the WWW. *ACM SIGCSE bulletin 29 (3): 11-13*

Holtz N M *1995* The Tutorial Gateway, Carleton University, Ottawa, CA, *http://www.civeng.carleton.ca/~nholtz/tut/doc/doc.html*

Hubler A W & Assad A M *1995* CyberProf: An Intelligent Human-Computer Interface for Asynchronous Wide Area Training and Teaching. *4th International World Wide Web Conference*, Boston, USA, December 11-14, 1995, http://www.w3.org/pub/Conferences/WWW4/Papers/247/

Hubler A W & CyberProf Team *1998* CyberProf, University of Illinois at Urbana-Champaign, Urbana, IL, *http://prof.ccsr.uiuc.edu/*

Ibrahim B *1994* World-Wide Algorithm Animation. *Computer Networks and ISDN Systems 27 (2): 255-265*

IDEALS Project *1998* MTS, Zentrum für Graphische Datenverarbeitung e. V., Darmstadt, Germany, *http://ideals.zgdv.de/*

IMG *1998* Web University, The Information Management Group, Chicago, IL, *http://www.imginc.com/webuniversity/*

Ingebritsen T, Brown G & Pleasants J *1997* Teaching biology on the Internet. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 235-240*

Instructional Toolkit Team *1997* Instructional Toolkit, University of Virginia, Charlottesville, VA, *http://cti.itc.Virginia.EDU/toolkit2/*

Jafari A, Mills D & ONcourse R&D Team *1998* ONcourse, Indiana University, *http://www.weblab.iupui.edu/projects/ONcourse.html*

Jehng J-C J, Tung S-H S & Chang C-T *1997* Visualization strategies for learning recursion. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 532-538*

Johnson W L & Shaw E *1997* Using Agents to Overcome Deficiencies in Web-Based Courseware. *Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education*, Kobe, Japan, 18 August 1997, *pp. 70-77*, http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Johnson/Johnson.html

Johnson W L, Shaw E & Ganeshan R *1998* Pedagogical agents on the Web. *Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98)*, San Antonio, TX, August 16-19, 1998, http://www.isi.edu:80/isd/ADE/ITS98-WW.htm

Jühne J, Jensen A T & Grønbæk K *1998* Ariande: a Java-based guided tour system for the World Wide Web. *Computer Networks and ISDN Systems 30 (1-7): 131-139*

Kashy E, Thoennessen M, Tsai Y, Davis N E & Wolfe S L *1997* Using networked tools to enhanse student success rates in large classes. *FIE'97, Frontiers in Education Conference*, Pittsburgh, PA, November 5-8, 1997, *pp. 233-237*

Kay J & Kummerfeld R *1994a* Adaptive hypertext for individualised instruction. *Workshop on Adaptive Hypertext and Hypermedia at Fourth International Conference on User Modeling*, Hyannis, MA, Aug 17, 1994, http://www.cs.bgsu.edu/hypertext/adaptive/Kay.html

Kay J & Kummerfeld R J *1994b* An individualised course for the C programming language. *Second International WWW Conference*, Chicago, IL, 17-20 October, 1994, http://www.ncsa.uiuc.edu/SDG/IT94/Proceedings/Educ/kummerfeld/kummerfeld.html

Khuwaja R, Desmarais M & Cheng R *1996* Intelligent Guide: Combining user knowledge assessment with pedagogical guidance. In: Frasson C, Gauthier G & Lesgold A (eds.) *Intelligent Tutoring Systems. Lecture Notes in Computer Science, Vol. 1086* Springer Verlag, Berlin, *pp. 225-233*

Kirkpatrick A, Lee A & Willson B *1997* The engine in engineering - development of thermal/fluids Web based education. *FIE'97, Frontiers in Education Conference*, Pittsburgh, PA, November 5-8, 1997, *pp. 744-747*

Klevans R L *1997a* Automatically Generated Web-based Multimedia Presentations: The Web Lecture System (WLS) North Carolina State University, Raleigh, NC, *http://renoir.csc.ncsu.edu/WLS/white_paper/wls.htm*

Klevans R L *1997b* The Web Lecture System (WLS). *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 669-670*

Knowledge Navigators *1999* LearningEngine™, Knowledge Navigators International, Inc, Halifax, Canada, *http://knav.com/content.htm*

Lam M M & Hubler A W *1997* NetworkTA: The CyberProf conferencing system, a complex systematic approach to creating a collaborative learning environment. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 613-618*

Langenbach C & Bodendorf F *1997* A framework for WWW-based learning with flexible navigation guidance. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 293-298*

Langenbach C & Bodendorf F *1998* An education broker toolset for Web course customization. *Journal of Universal Computer Science 4 (10): 780-791*

LaRose R & Gregg J *1997* An evaluation of a Web-based distributed learning environment for higher education. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 1286-1287*

Lawrence-Fowler W A & Fowler R H *1997* Heterogeneous information resources and asynchronous workgroups: Creating a focus on information discovery and intergation in computer science. *FIE'97, Frontiers in Education Conference*, Pittsburgh, PA, November 5-8, 1997, *pp. 833-837*

Lee S H & Wang C J *1997* Intelligent hypermedia learning system on the distributed environment. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 625-630*

Lemone K *1997* Experiences in virtual teaching. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 306-311*

Lotus *1999* LearningSpace, Lotus, Cambridge, MA, *http://www.lotus.com/products/learningspace.nsf*

MacDougall G *1997* The Acadia advantage adademic development centre and the authomatic courseware management systems. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 647-652*

Macromedia *1999a* Authorware Attain Web Player, Macromedia, Inc., *http://www.macromedia.com/software/authorware/productinfo/webplayer/*

Macromedia *1999b* Shockwave player, Macromedia, Inc., *http://www.macromedia.com/shockwave/*

madDuck *1997* Web Course in a Box (WCB), madDuck Technologies, Richmond, VA, *http://www.madduck.com/wcbinfo/wcb.html*

Marshall A D & Hurley S *1996a* Delivering Hypertext-based Courseware on the World-Wide-Web. *Journal of Universal Computer Science 2 (12): 805-828*

Marshall A D & Hurley S *1996b* The Design, Development and Evaluation of Hypermedia Courseware for the World Wide Web. *Multimedia Tools and Applications 3 (1): 5-31*

Marshall A D & Hurley S *1997* Courseware development for parallel programming and C programming. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 689-697*

McCartney R, Weiner B & Wurst K R *1997* Delivering a lab course in a Web-based learning environment. *FIE'97, Frontiers in Education Conference*, Pittsburgh, PA, November 5-8, 1997, *pp. 849-855*

McKeever S, McKeever D & Elder J *1997* An authoring tool for constructing interactive exercises. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 695-696*

McKenna A & Agogino A *1997* Engineering for middle schools: A Web-based module for learning and designing with simple machines. *FIE'97, Frontiers in Education Conference*, Pittsburgh, PA, November 5-8, 1997, *pp. 1496-1501*

Mehringer S & Lifka D *1998* The virtual workshop companion: a Web interface for online labs. *WebNet'98, World Conference of the WWW, Internet, and Intranet*, Orlando, FL, November 7-12, 1998, *pp. 647-652*

Mentorware *1998* Enterprise Education Server, Mentorware, Inc., Sunnyvale, CA, *http://www.mentorware.com/*

Merat F L & Chung D *1997* World Wide Web approach to teaching microprocessors. *FIE'97, Frontiers in Education Conference*, Pittsburgh, PA, November 5-8, 1997, *pp. 838-841*

Milton J, Thomas B & Yaron D *1997* Java Programs for Physical Chemistry, Carnegie Mellon University, Pittsburgh, PA, *http://www.chem.cmu.edu/milton/Gamelan/*

Nakabayashi K, Maruyama M, Kato Y, Touhei H & Fukuhara Y *1997* Architecture of an intelligent tutoring system on the WWW. In: Boulay B d & Mizoguchi R (eds.) *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*. IOS, Amsterdam, *pp. 39-46*

Nakabayashi K, Maruyama M, Koike Y, Fukuhara Y & Nakamura Y *1996* An intelligent tutoring system on the WWW supporting interactive simulation environments with a multimedia viewer control mechanism. *WebNet'96, World Conference of the Web Society*, San Francisco, CA, October 15-19, 1996, *pp. 366-371*, http://www.contrib.andrew.cmu.edu/~plb/WebNet96.html

Naps T L *1997* Algorithm visualization on the World Wide Web - the difference Java makes. *ACM SIGCSE* *bulletin 29 (3): 59-61*

NetObjects *1998* NetObjects Fusion, NetObjects, Inc., *http://www.netobjects.com/products/html/nof.html*

Neumann G & Zirvas J *1998* SKILL - A scallable internet-based teaching and learning system. *WebNet'98, World Conference of the WWW, Internet, and Intranet*, Orlando, FL, November 7-12, 1998, *pp. 688-693*, http://nestroy.wi-inf.uni-essen.de/Forschung/Publikationen/skill-webnet98.ps

Ni Y, Zhang J & Cooley D H *1997* NetTest: An integrated Web-based test tool. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 710-711*

Nicol D, Smeaton C & Slater A F *1995* Footsteps: Trail-blaizing the Web. *Third International World-Wide Web Conference*, Darmstadt, 10-14 April, 1995, *pp. 879-885*

Nykänen O *1999* Visualizing Navigation in Educational WWW Hypertext by Introducing Partial Order and Hierarchy. *ED-MEDIA/ED-TELECOM'99 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Seattle, WA, June 19-24., http://matriisi.ee.tut.fi/~onykane/papers/navtutor/2444.html

Okazaki Y, Watanabe K & Kondo H *1997* An Implementation of the WWW Based ITS for Guiding Differential Calculations. *Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education*, Kobe, Japan, 18 August 1997, *pp. 18-25*

Owen C B & Makedon F *1997* ASML: Web authoring by site, not by hidnsight. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 677-682*

Pearce J M & Livett M K *1997* Real-world physics: a Java-based Web environment for the study of physics. *AUSWEB'97, The Third Australian World Wide Web Conference*, Queensland, Australia, July 5-9, 1997, *pp. 211-215*, http://ausweb.scu.edu.au/proceedings/eklund/paper.html

Pilar da Silva D, Durm R V, Duval E & Olivié H 1998 *Concepts and documents for adaptive educational hypermedia: a model and a prototype*. Computing Science Reports, Eindhoven: Eindhoven University of Technology, 35-43.

Plant D *1998 Shockwave! Breathe new life into your web pages*. Ventana Communications Group,

Pohjolainen S, Multisilta J & Antchev K *1997* Matrix algebra with hypermedia. *Education and Information Technologies 1 (2): 123-141*

POLIS Team *1998* POLIS, University of Arizona, Lexington, KY, *http://www.u.arizona.edu/ic/polis/*

Procopiuc M, Procopiuc O & Rodger S H *1996* Visualization and Interaction in the Computer Science Formal Languages Course with JFLAP. In: *Vol. 29* IEEE, Salt Lake City, Utah, *pp. 121-125*

Question Mark *1998* Perception, Question Mark Corporation, Stamford, CT, *http://www.questionmark.com/*

QuestWriter Team *1998* QuestWriter, Oregon State University, *http://iq.orst.edu/doc/final/QWHome.html*

Radhakrishnan S & Bailey J E *1997* Web-based educational media: Issues and empirical test of learning. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 400-405*

Raineri D M, Mehrtens B G & Hubler A W *1997* CyberProf™ - An intelligent human-computer interface for interactive instruction on the World Wide Web. *Journal of Asynchronous Learning Networks 1 (2): 20-36*

Rebelsky S A *1997* CourseWeaver: A tool for building course-based Webs. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 881-886*

Reer K A *1988* The TRY system or How to avoid testing student's programs. *ACM SIGCSE bulletin 20 (1): 112-116*

Rehak D *1997a* A database architecture for Web-based distance education. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 418-425*

Rehak D R *1997b* Carnegie Mellon Online: Web-Mediated Education. *FIE'97, Frontiers in Education Conference*, Pittsburgh, PA, November 5-8, 1997, *pp. 1510-1516*

Rios A, Pérez de la Cruz J L & Conejo R *1998* SIETTE: Intelligent evaluation system using tests for TeleEducation. *Workshop "WWW-Based Tutoring" at 4th International Conference on Intelligent Tutoring Systems (ITS'98)*, San Antonio, TX, August 16-19, 1998, http://www-aml.cs.umass.edu/~stern/webits/itsworkshop/rios.html

Ritter S *1997* Pat Online: A Model-tracing tutor on the World-wide Web. *Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education*, Kobe, Japan, 18 August 1997, *pp. 11-17*, http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Ritter/Ritter.html

Rodger S H *1997* Visual and Interactive Tools: JFlap, Visual and Interactive Tools JFLAP, Pâté, JAVAA. PA, Duke University, Durham, NC, *http://www.cs.duke.edu/~rodger/tools/tools.html*

Routen T & Graves A *1997a* FLAX: Interactive courseware on the Web, De Montfort University, *http://www.cms.dmu.ac.uk/coursebook/flax/*

Routen T W & Graves A *1997b* Activating the Student: Authoring Interactive Exercises in Java with Flax *http://www.cms.dmu.ac.uk/~twr/flaxp/activating.html*

Routen T W, Graves A & Ryan S A *1997* Flax: Provision of interactive courseware on the Web. *Cognition and the Web '97*, , *pp. 149-157*, http://www.cms.dmu.ac.uk/coursebook/flax/

Sapir M *1999* The WebTester and the Linear Algebra WebNotes. *ALN Magazine 3 (1)*

Schwarz E, Brusilovsky P & Weber G *1996* World-wide intelligent textbooks. *ED-TELECOM'96 - World Conference on Educational Telecommunications*, Boston, MA, June 17-22, 1996, *pp. 302-307*, http://www.contrib.andrew.cmu.edu/~plb/ED-MEDIA-96.html

Serf Team *1998* Serf, University of Delaware, Newark, DE, *http://www.udel.edu/serf/*

Severance C *1998* Sync-O-Matic 3000, Michigan State University, East Lansing, *http://www.egr.msu.edu/~crs/projects/syncomat/*

Shipman III F M, Furuta R, Brenner D, Chung C-C & Hsieh H-w *1998* Using paths in the classroom: Experience and adaptations. *Ninth ACM International Hypertext Conference (Hypertext'98)*, Pittsburgh, USA, June 20-24, 1998, *pp. 267-276*

Shipman III F M, Marshall C C, Furuta R, Brenner D A, Hsieh H-W & Kumar V *1997* Using networked information to create educational guided paths. *International Journal of Educational Communications 3 (4): 383-400*

Siviter P *1997* Authoring tools for courseware on WWW: the W3Lessonware project. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 513-516*

Smeaton A F & Crimmons F *1997* Virtual lectures for undergraduate teaching: Delivery using real audio and the WWW. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 990-995*

Specht M & Oppermann R *1999* ACE - Adaptive Courseware Environment. *The New Review of Hypermedia and Multimedia 4: 141-161*

Specht M, Weber G, Heitmeyer S & Schöch V *1997* AST: Adaptive WWW-Courseware for Statistics. *Workshop "Adaptive Systems and User Modeling on the World Wide Web" at 6th International Conference on User Modeling, UM97*, Chia Laguna, Sardinia, Italy, June 2, 1997, *pp. 91-95*, http://www.contrib.andrew.cmu.edu/~plb/UM97_workshop/Specht.html

Stanford Online Team *1998* Stanford Online, Stanford University, *http://stanford-online.stanford.edu/*

Stern M, Steinberg J, Lee H I, Padhye J & Kurose J *1997a* MANIC: Multimedia asynchronous networked individualized courseware. *ED-MEDIA/ED-TELECOM'97 - World Conference on Educational Multimedia/Hypermedia and World Conference on Educational Telecommunications*, Calgary, Canada, June 14-19, 1997, *pp. 1002-1007*

Stern M, Woolf B P & Kuroso J *1997b* Intelligence on the Web? In: Boulay B d & Mizoguchi R (eds.) *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*. IOS, Amsterdam, *pp. 490-497*

Stern M K *1997* The difficulties in Web-based tutoring, and some possible solutions. *Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education*, Kobe, Japan, 18 August 1997, *pp. 1-10*, http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Stern.html

Suthers D & Jones D *1997* An architecture for intelligent collaborative educational systems. In: Boulay B d & Mizoguchi R (eds.) *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*. IOS, Amsterdam, *pp. 55-62*

Synnes K, Lachapelle S, Parnes P & Schefström D *1998* Distributed education using the mStar environment. *Journal of Universal Computer Science 4 (10): 803-823*

Tardif P-M & Zaccarin A *1997* An interactive Java applet for teaching discrete-time system theory. *FIE'97, Frontiers in Education Conference*, Pittsburgh, PA, November 5-8, 1997, *pp. 1517-1520*

Thimbleby H *1997* Gentler: a tool for systematic web authoring. *International Journal on Human-Computer Studies 47 (1)*

Tilbury D & Messner W *1997* Development and integration of Web-based software tutorials for an undergraduate curriculum: Control tutorials for Matlab. *FIE'97, Frontiers in Education Conference*, Pittsburgh, PA, November 5-8, 1997, *pp. 1070-1075*

Titus A P, Martin L W & Beichner R J *1998* Web-based testing in physics education: Methods and opportunities. *Computers in Physics 12 (Mar/Apr): 117-123*

Trigg R H *1988* Guided tours and tabletops: Tools for communicating in a hypertext environment. *ACM Transactions on Office Information Systems 6 (4): 398-414*

Vassileva J *1997* Dynamic Course Generation on the WWW. In: Boulay B d & Mizoguchi R (eds.) *Artificial Intelligence in Education: Knowledge and Media in Learning Systems*. IOS, Amsterdam, *pp. 498-505*

Virtual-U Research Team *1997* Virtual-U, Simon Fraser University, Vancouver, *http://virtual-u.cs.sfu.ca/vuweb/*

Wagner R *1998* HyperStudio, Roger WagnerPublishing, Inc., *http://www.hyperstudio.com/*

Warendorf K & Tan C *1997* ADIS - An animated data structure intelligent tutoring system or Putting an interactive tutor on the WWW. *Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education*, Kobe, Japan, 18 August 1997, *pp. 54-60*, http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Warendorf/Warendorf.html

WBT Systems *1999* TopClass, WBT Systems, Dublin, Ireland, *http://www.wbtsystems.com/*

WebCT *1999* World Wide Web Course Tools, WebCT Educational Technologies, Vancouver, Canada, *http://www.webct.com*

Weber G, Brusilovsky P & Specht M *Submitted* ELM-ART: An adaptive educational system on the WWW. *User Modeling and User-Adapted Interaction*

Weber G & Specht M *1997* User modeling and adaptive navigation support in WWW-based tutoring systems. In: Jameson A, Paris C & Tasso C (eds.) *User Modeling*. Springer-Verlag, Wien, *pp. 289-300*

Westhead M D *1996* EPIC: Building a structured learning environment. *WebNet'96, World Conference of the Web Society*, San Francisco, CA, October 15-19, 1996, *pp. 480-485*

Whitehurst G J *1997* CourseNet, Worth Publishers, New York, *http://129.49.71.236/coursenet.htm, http://www.whitehurst.sbs.sunysb.edu/Coursenet/CourseNet/default.asp*

Wright C & Jones R *1997* A Novel Architecture for WWW-based Learning: the implementation of guided tours on the WWW. *5th Annual Conference on the Teaching of Computing*, Dublin, Ireland, August 1997, *pp. 263 - 265*

Xiao G *1999* WIMS, Université de Nice, Sophia Antipolis, *http://wims.unice.fr/~wims/wims.cgi?lang=en*

Yang J C & Akahori K *1997* Development of Computer Assisted Language Learning System for Japanese Writing Using Natural Language Processing Techniques: A Study on Passive Voice. *Workshop "Intelligent Educational Systems on the World Wide Web" at AI-ED'97, 8th World Conference on Artificial Intelligence in Education*, Kobe, Japan, 18 August 1997, *pp. 99-103*, http://www.contrib.andrew.cmu.edu/~plb/AIED97_workshop/Yang/Yang.html

Yoon I, Mah P, Cho C & Shin G *1997* An adaptive visual programming tool for Web application. *WebNet'97, World Conference of the WWW, Internet and Intranet*, Toronto, Canada, November 1-5, 1997, *pp. 576-581*