MANIC: An Open-Source System to Create and Deliver Courses over the Internet

Agustín Schapira
Computer Science Department
schapira@cs.umass.edu

Kimberly De Vries
English Department
kmarcell@english.umass.edu

Cris Pedregal-Martin
Computer Science Department
cris@cs.umass.edu

University of Massachusetts, Amherst

Abstract

This paper describes MANIC, an open-source, cross-platform system to create and deliver courses over the WWW. MANIC allows instructors to create courses and deliver them to students as HTML slides shown in synchrony with the voice of the instructor in an audio stream. Courses are organized as modular units to teach individual topics; students can drive their learning experience by browsing the material freely and controlling the pace of the presentations. MANIC provides an on-line authoring tool that does not require knowledge of HTML and that makes it easy to create, maintain, reuse, and share courses. MANIC has been used to deliver a variety of courses. Students benefit from around-the-clock access to material; the combination of audio and text also makes it easy to understand and remember the material. Instructors are able to cover more material during their lectures and to adapt the digital material for future use.

Introduction

Computer-assisted instruction provides important advantages both as a supplement to and as a replacement for traditional classroom instruction, from the viewpoint of both students and instructors. It can be asynchronous and self-paced: students can follow course content when they want, and suspend and resume at their convenience. (Likewise instructors update their material anytime.) It can reach more people: a person can work on material from any suitably equipped computer. Instruction can also be customized to each individual or small group. A significant production advantage is that once a particular instructional unit is added to the system, authoring tools allow instructors to easily modify it, reuse, and recombine it with others thanks to its digital nature; this is especially easy when the delivery is on-line (as opposed to CD-ROM, for example). Along these virtues we must also consider limitations: spontaneous student interactions with peers and instructors are made much harder, nonverbal cues are lost, the system has to make decisions based only on “observable” actions of the student, and so forth.

This paper reports on MANIC[1] [1,2], a system for computer-delivered instruction designed to make the most of the advantages outlined above. We chose the World-Wide-Web (WWW) as our delivery vehicle (as opposed to CD-ROMs, broadcast video, etc.) because we wanted to take advantage of its popularity, the low distribution costs, and the fact that many people are already familiar with it. We set out to develop a system in which students would have some freedom to navigate over a course's content, as well as control the pace at which they covered it. We further wanted to build a system where textual course material would be enriched with multimedia content, while still keeping it accessible via a variety of browsers and platforms. We also wanted to develop an authoring tool to make it easy to create, maintain, reuse, and share instructional materials. In order for the system to gain acceptance among both students and instructors, it had to be fast and scale well with a large amount of material and students. Finally, we were interested in sharing our product with others and thus needed to create a system that could be easily distributed, deployed, and customized. The system we describe in this paper is a re-implementation of the first production version [1,2] which maintains and enhances the original functionality but uses a much more flexible database-based, open-source infrastructure. Functional differences are mainly apparent to authors, who see significant improvements in their interfaces. We are confident that the experience reported here, which is mostly based on the production version, also applies to the new database-backed version.

From the students' point of view, MANIC is a repository of instructional material that they can navigate. The material is delivered in the form of Hypertext Markup Language (HTML) pages ("overhead slides," because they usually follow the form of those visual aids) that are enriched with an audio stream that (usually) contains the voice of the instructor explaining

[1] MANIC stands for Multimedia Asynchronous Networked Individualized Courseware

the slides in greater detail[2]. The slides and the audio are synchronized: when the voice starts to explain a new concept, the HTML page in the browser displays the appropriate slide or highlights the current bullet. Unlike other systems with similar goals [3,4,5], MANIC allows students to select what they want via search engines and browsable indices of the material, and to control the pace of the presentation by stopping and resuming the audio. Because the course material is delivered as simple HTML pages, students do not need extra hardware or software to interact with MANIC. (MANIC runs on most versions of Netscape and Explorer on Windows, MacOS, and Linux.) From the instructors' point of view, MANIC is a web-based tool to create slides, aggregate them in modular units, and deliver them with multimedia content over the Internet. (The authoring tool can also be used from a variety of browsers and platforms.) The modular structure of MANIC courses makes it easy to maintain, share, and reuse the material. The same facilities can be used by instructors to share with others the material that they have created. Behind the scenes, a MANIC server is a computer connected to the Internet running a webserver, a database server, and a RealAudio server, together with scripts inside the webserver that coordinate the actions of the other components. Most of the MANIC components are based on free, open-source solutions that run on a variety of platforms and are easy to deploy. Our scripts are also available open-source, which allows other researchers to customize and experiment with MANIC.

In the rest of the paper we describe in more detail how students and instructors interact with MANIC, and briefly present its internal architecture. We also comment on our experience using MANIC to teach several courses at UMASS.

MANIC courses

Both students and instructors view the material in MANIC as a collection of tree-like structures called courses. Courses correspond to the complete syllabus that an instructor wants to teach, and consist of a collection of knowledge unit (Kn.U.) modules that represent the "main ideas" in the syllabus. For example, the Computer Networks course consists of several independent modules such as Introduction to Networking, The ISO Model, TCP/IP Protocol, etc. Knowledge units are in turn composed of topics. A topic resembles a lecture in content (but possibly not in length) in that, it probably should be seen in its entirety (at least on its first viewing), in order to make sense. It is composed of an audio or video stream punctuated by slides presented synchronously; in some cases, these correspond to a recording of an actual lecture with its supporting slides shown as HTML pages. For example, the The ISO Model knowledge unit explains, in different topics, The Physical Layer, The Data Link Layer, The Network Layer, etc. A sample course structure is shown in Figure 1. Navigation of the course is possible at each of these levels, down to the individual slides, which serve as reference points within a topic.

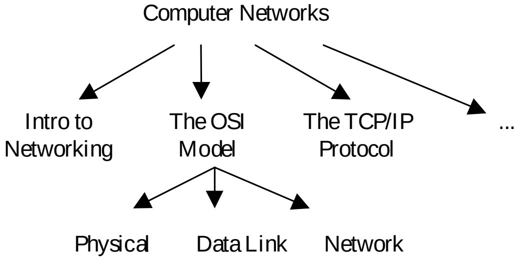


Figure 1. A sample MANIC course

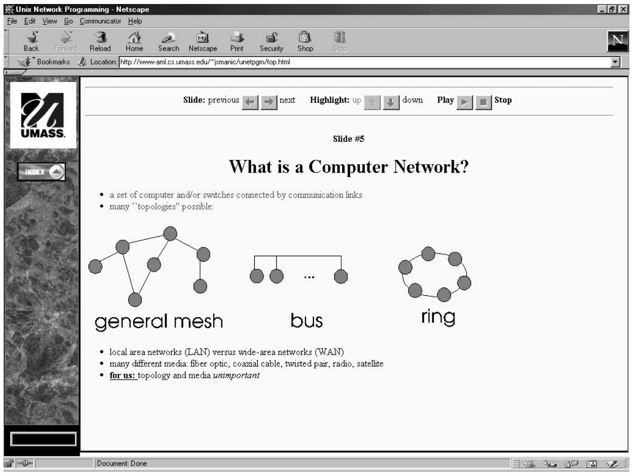
Students' view of MANIC

When students activate a MANIC presentation, their computer starts playing an audio/video stream and shows the slides at the appropriate point of the course. The slides and the stream are synchronized: as the stream advances, the slides change to display the information that the instructor is talking about, without any intervention from the student. Slides usually contain text, images, and a list of bullets that are highlighted as the stream progresses. Figure 2 shows an example screenshot of what a student sees when using MANIC. One frame contains the slide being displayed, together with VCR-like controls that the students can use to control the pace at which they see the material. Another frame, to the left, contains links to a search engine and a table of contents of the course.

Allowing students to play an active role in the way they see the course is important to keep them interested in the material [6]. The VCR-like stream controls allow students to browse the material at their own pace, stopping and starting the stream at will. Using the navigation controls, they can also move back and forward between bullets in the slides or between the slides in the topic they are watching. The slides are shown synchronized with the stream as designed by the instructor, but students can stop the stream altogether and watch the slides at their own pace. This is useful when the student is not interested in the entire explanation but just wants to review an individual slide, or quickly browse a set of slides. In order to allow students to browse the material in a sequential fashion, MANIC provides a table of contents, which allows students to jump directly to knowledge units, topics, and individual slides. For example, a student watching the Computer Networks course for the first time might be

[2] We often stream video and audio; here we do not make a distinction between stream types unless necessary.

interested in seeing the entire presentation sequentially. At a later time, however, she might be interested in some particular slide needed to solve a homework problem. The table of contents allows the student to jump directly to that point in the course. Furthermore, MANIC has a search engine to find slides containing specified keywords in all the MANIC courses to which they have access. For example, a student watching the Computer Networks course can search for the keywords "TCP/IP" to find all the slides that mention that protocol.


Unix Network Programming - Netscape
File Edit View Go Communicator Help
Back Forward Reload Home Search Netscape Print Security Shop Stop
Bookmarks Location: http://www-aml.cs.umass.edu/~jsmanic/unetpgm/top.html
UMASS
INDEX
Slide: previous next Highlight: up down Play Stop
general mesh
bus
...
ring
Document: Done

Slide #5

What is a Computer Network?

- a set of computer and/or switches connected by communication links
- many ``topologies'' possible:

- local area networks (LAN) versus wide-area networks (WAN)
- many different media: fiber optic, coaxial cable, twisted pair, radio, satellite
- **for us:** topology and media *unimportant*

Figure 2. A MANIC slide, with navigation bar

Students only need a web browser to watch the slides, and a RealAudio plug-in to receive the multimedia stream. Because the slides are delivered as standard HTML code, without any browser-specific tags, students can watch MANIC presentations with all the popular browsers running in most platforms. They only need to have a RealAudio Player, which usually comes pre-installed in most machines. MANIC has been tested on Netscape and Internet Explorer versions 3 and higher, running on various flavors of Windows, MacOS, and Linux.

We have based the overall design of the system in Constructivism, a popular learning theory with roots in the work of Piaget and then extended by Papert. It suggests that learners actively construct their knowledge, that knowledge cannot merely be transmitted [7]. We believe that the features of the MANIC system described above (audio controls, controls to move back and forth between the slides, a table of contents, and a search engine) allow for a greater participation of the students in their learning process.

Instructors' view of MANIC

MANIC includes a web-based authoring tool that allows instructors to easily create instructional material. An instructor creates a slide by filling out forms in a web browser to specify title, header, footer, introductory text, list of bullets, and formatting. Although slides are shown to the students as HTML pages, instructors need not write HTML code; MANIC generates a valid HTML page from the elements supplied by the instructor. Advanced users may embed HTML code if they wish, for example, tags to include images, change fonts, insert a table, etc.

As discussed above, slides belong to a topic inside a knowledge unit, which in turn is contained inside a course. Instructors start by defining a course and specifying general characteristics that all the slides in the course will share: header lines, footers, titles, and various formatting options (colors, fonts, etc.; reasonable formatting defaults are offered). Instructors then create a series of knowledge units and topics that make up the course (see Figure 1). The authoring tool provides facilities to re-arrange the course structure by changing the order of the slides, moving slides from one topic to another, and moving topics from a knowledge unit to another. There are also forms where the instructors can specify the timing information to synchronize slides and the multimedia streams[3].

Elements in every level of the course tree inherit their parent's properties unless otherwise specified. In this fashion, all the slides under a particular knowledge unit, for example, will have the title, header, footer, and format that was specified for the entire course. At any level in the tree, instructors can override the inherited properties of an individual element. For example, an instructor may specify a different background color for all the slides in a particular topic. This inheritance of attributes makes it easy to create and maintain courses. Instructors only have to specify once, for the entire course, the basic components of the slides and the formatting options; they do not need to worry about indicating those properties for every slide in the course. This also makes it easier to maintain a consistent appearance for the whole course, as it is enough to modify a single field in the course properties to change it for all levels below.

In addition, the modular structure and the attribute inheritance mechanism make it easy to reuse and share existing material. For example, looking at the Computer Networks course an instructor could reuse the TCP/IP Protocol knowledge unit in a new short course on UNIX Socket Programming. Reusing that material is as easy as selecting the entire knowledge unit in the context of the old course and copying it to the new one. Furthermore, the instructor need not worry about the formatting of the newly added slides: since they inherit the properties

[3] Audio and video streams are created with different applications and then converted to RealAudio format with tools provided by Real Networks. Lack of space precludes discussion of those processes in this paper.

specified for the entire course, the slides in the "borrowed" knowledge unit will look just like the others in the rest of the course. This copying mechanism also facilitates the collaboration between different authors. An instructor can grant copy access to another and let her take a knowledge unit or a topic and include it in her course. From that point on, however, the copied material is independent from the origin, and can be modified to suit the second instructor's needs.

Given the speed at which knowledge advances and curricula change, it has become increasingly difficult and expensive to keep course material up-to-date. A design goal of the MANIC project was thus to develop an authoring tool that would make it easy to create and maintain course material, on the one hand, and that would allow to take advantage of resources from existing courses and reuse them in novel situations, on the other. The web-based authoring tool that we have created makes it possible to produce instructional material and deliver it over the WWW, even for instructors who are not familiar with the HTML language. The modular structure of the course and the powerful inheritance mechanism make it easy to maintain the courses and keep the material up-to-date. The same modular structure and the inheritance mechanism also encourage instructors to leverage existing resources and share them and reuse them in novel situations.

The architecture of MANIC

In this section we explain the internal structure of MANIC (its "architecture") and give the rationale for the choices we made. The MANIC system comprises the MANIC server and the MANIC clients. In order to impose the least requirements on the clients (to foster MANIC's accessibility and ease of use), the clients do not use any MANIC-supplied software. Any computer with a network connection, a web browser, and a streaming client (RealAudio player), can become a MANIC client by simply connecting to a MANIC server.

The MANIC server has three major components running on a UNIX server connected to the Internet (see Figure 3):

- A database engine, which stores (and provides to the web server as requested) all the instructional material, except the audio and video streams.
- A webserver, which understands the HTTP protocol in which clients and the server communicate, and executes scripts to integrate the functionality of all components.
- A streaming server, which handles the streaming of audio or video to the clients, as well as generating synchronization events to advance slides. The streams are stored as files on the server computer.

We built MANIC by the careful integration of these off-the-shelf components via a set of scripts, which realize our desired architecture by connecting components and their functionality in an efficient and modular manner. These scripts run in response to client requests and are executed within the webserver. Their function is to coordinate the actions of the individual components to produce meaningful responses. Depending on the type of client request, the scripts connect to the database server either to store information or to retrieve and process instructional material, and then instruct the webserver to return appropriate responses to the clients. The scripts also cooperate with the streaming server to deliver the multimedia material as needed. For example, consider a student watching a MANIC presentation. On the student's machine (a MANIC client) a browser is open and displaying a slide and a streaming client is running, playing the audio for the current lecture; at the MANIC server, the streaming server is sending the client the current segment of audio. While the student watches without interrupting the presentation, the stream generates synchronization events, which cause the client to request the next slide. On the server side, a script connects to the database and retrieves the appropriate material, which the webserver returns to the student's browser, and a new cycle in the interaction is completed.

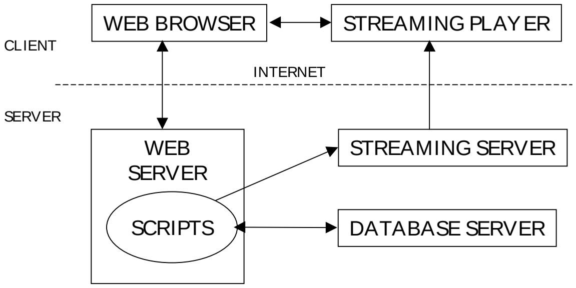


Figure 3. MANIC architecture

The original implementation of MANIC stored all the material in a hierarchy of UNIX files on the server. That approach suffered from limited scalability (limits on the number of courses and the number of simultaneous users, both in performance and in administrative and maintenance load), as well as limited modularity (it is harder to reuse or modify courses, as it requires manipulating and modifying large numbers of interrelated files, whose relationships are complex and implicit). The use of a database server gives both increased scalability and higher modularity. Our system imposes relatively simple demands on the its database engine, so we picked the open-source MySQL for its wide availability, price (it is free), ease of deployment, and robustness. We were concerned about the limitations of conventional CGI scripts, which require creating a process (forking, in UNIX parlance) to serve each request that arrives at the webserver [8]. To address that, we adopted Apache, a free, widely adopted webserver that includes interpreter modules for a variety of lightweight script languages. We wrote the scripts that customize our webserver in PHP, a simple but powerful scripting language. Requests to Apache cause its internal interpreter to execute a PHP script that interacts with MySQL. Overhead is low because the PHP interpreter is always on a live thread within Apache, and the communication with MySQL is done on an open pool of buffers. For the streaming component only proprietary solutions were available, and we chose RealNetworks for its wide availability.

With this architecture (Apache + PHP + MySQL + RealAudio Server + lightweight clients) we were able to build most of MANIC meeting our 'production' goals - efficiency and wide availability- and our 'research' goals -the ability to make all the software open so others can install it, use it and modify it.

Experiences using MANIC

Since its inception, the MANIC system has been used to deliver a variety of courses at the University of Massachusetts, Amherst. We have used MANIC in two different ways: to provide computer-mediated support to classes at the university, and to create and deliver stand-alone tutorials, independent of any existing class. Both the creation and delivery of these courses offered us valuable knowledge about the most effective ways to author and deliver instructional content over the Internet, as well as feedback from students and instructors on what worked best for them. In this section we describe our experiences with two MANIC courses: a stand-alone tutorial on technical writing, and an introductory literature class taught in the English Department at UMASS. For more detailed accounts of our experiences using MANIC, see [9].

An example of a stand-alone course is our tutorial on technical writing. Many scientists and technical professionals must write memos, progress reports, or other documents to communicate within their workplaces, and many must also address more general audiences in grant proposals, conference papers, etc. We created a MANIC technical writing course to provide a basic introduction to the practice of technical writing for those scientists and technical researchers. The course speaks generally about effective writing practice, with a focus on how to determine which stylistic rules apply in different situation. Examples were provided from real documents produced for well-known groups such as the National Science Foundation, and links were provided to the specific guidelines of the NSF, IEEE, etc. As one of the first courses authored from scratch, rather than being adapted from an existing course, the tech-writing tutorial was far more labor-intensive than earlier courses had been. The text, graphics, examples, and audio-track were all created for this course, which was time-consuming but allowed us to tailor them for the Internet experience. For example, data collected from earlier MANIC courses had suggested that most students would view a course only for about 30 minutes before switching to another activity, and that they often used the material in a non-linear way, focusing on topics of interest and skipping others [10]. Working from this information, we designed shorter, tightly focused segments for the presentation and made use of the keyword search feature to let students easily locate topics of interest. E-mail feedback suggests that users of the tutorial find it helpful and easy to use, and our usage logs show that hundreds of distinct users from domains around the world are repeatedly accessing the tutorial, and thousands are accessing it from educational domains[4].

More recently, we have experimented using MANIC presentations to support a literature course, English 131/Society and Literature. This particular section of English 131 explored the connection between works of fantasy and their societal context. Fantastic ideas have inspired not only works of fiction, but also a large body of material in other media, such as visual art, music, and film and the MANIC system offered a better way of presenting this material to students. In this way, students had access to visual and musical examples throughout the term, rather than only on the days they were brought into class. The online portion of this class also covered some material that might otherwise have to be presented in lectures, thus allowing more time in class for discussion. Student feedback regarding this class was almost unanimous in agreement that the combination of text, graphics, and an audio track unique to MANIC, make the material more memorable and easier to understand than material presented through only one of these media. They further found that having the material available at any time is extremely helpful, particularly when they wish to review material later in the semester. Students also enjoyed and profited from being able to engage actively with the material through increased class discussion, rather than spending a large portion of time in the classroom listening to lectures.

The teacher who created these two courses also saw benefits in several ways from using MANIC. During the semester, she was able to cover more material in a wider variety of media than usual, and found that while creating the on-line material was initially time-consuming, it is now much easier to update and adapt for future use. Further, because students viewed the material

[4] Current usage statistics for this course are available at http://www-aml.cs.umass.edu/~itw/stats/

online (as opposed to a videotape or CD), they were much more likely to contact her and each other via e-mail with questions or comments, thus creating a more cohesive classroom community. Finally, being able to use a multimedia approach helped many students who were inexperienced or who lacked confidence in interpreting literature, approach the subject through a combination of the less comfortable medium of text and the more familiar media of speech and images.

Conclusions

Instructors find the MANIC system convenient because it allows them to create and deliver rich multimedia instructional material over the Internet. The simple web interface and the modular structure supported by MANIC make it easy to for instructors to maintain, reuse, and share their content. MANIC's delivery mechanism helps students because it allows them to control the pace and the content of their instruction, and because it enriches their experience with multimedia elements that appeal to different styles of learning. Because MANIC does not require any special software or hardware, students can access it anywhere, anytime. The architecture of MANIC ensures that the system can respond quickly to user requests and can scale well in the presence of many courses and clients. And since it is implemented with off-the-shelf products, it can be distributed and deployed easily; because it is open-source, it can be modified to suit other users' needs.

We are currently working on improvements to MANIC to satisfy requests from instructors and students. In order to facilitate the collaboration among students and with the instructors, we are adding a bulletin board system where students can pose questions, share their discoveries, and work with the instructors to solve problems. A new annotation system will allow students to take notes that will remain associated to the material being watched for later review. We are also instrumenting MANIC with powerful logging to keep track of the students' interactions with MANIC. Detailed reports about these interactions will allow instructors to detect problematic areas in the content (e.g., units where students to take longer than expected), to identify problems that individual students might have (e.g., a student doesn't seem to understand a particular slide because of missing prerequisite knowledge), to monitor which students view what sections, etc. Together with the logging, we are adding capabilities to the authoring tool to enable instructors to divide the students in groups and deliver different content to each group. For example, an instructor will be able to choose to divide a class into a Standard group, an Honors group, and a Remedial group, and then define that some topic should only be seen by students in the Honors section, or some review material by the Remedial students. A more dramatic step in the direction of personalized will be to maintain a model of each individual student and adjust the content according to that model. That ongoing research is further described in [11,12,13].

Acknowledgements

We would like to acknowledge the support of the National Science Foundation through grants EIA-9502639 and EIA-9979833, and the support of the Centers for Real-Time, Intelligent Complex Computing Systems at the University of Massachusetts, Amherst. We also thank Ivon Arroyo for her clear explanations of instructional theory.

References

[1] The MANIC system. Available online at http://manic.cs.umass.edu

[2] M. K. Stern, J. Steinberg, H. I. Lee, J. Padhye, J. Kurose. MANIC: Multimedia Asynchronous Networked Individualized Courseware. In Proc. Of Educational Multimedia and Hypermedia (1997).

[3] MediaNet. Available at http://www.cs.cornell.edu./Info/Projects/MediaNet

[4] RealProducer. Available at http://www.realnetworks.com

[5] Berkeley Continuous Media Toolkit. Available at http://bmrc.berkeley.edu/frame/research/cmt

[6] R. Hannafin and H. Sullivan. Learner Control in Full and Lean CAI Programs. In Educational Technology Research and Development, volume 43, number 1, 1995.

[7] B. G. Wilson. Reflections on Constructivism and Instructional Design. In Instructional Development Paradigms. C.R. Dills & A.A. Romiszowski, Eds. (Englewood Cliffs, NJ, March 1997)

[8] Ph. Greenspun. Philip and Alex's Guide to Web Publishing. Morgan Kaufmann Publishers, 1999.

[9] W. R. Adrion. Developing and Deploying Software Engineering Courseware in an Adaptable Curriculum Framework. In Proc. Of the International Conference on Software Engineering (2000).

[10] J. Padhye and J. Kurose. An Empirical Study of Client Interactions with a Continuous-Media Courseware Server. In IEEE Internet Computing (April 1999).

[11] M. K. Stern. Web-based Intelligent Tutors Derived from Lecture-based Courses. Unpublished Dissertation Proposal, Computer Science Department, University of Massachusetts, 1997. Available at http://www-aml.cs.umass.edu/~stern/papers/proposal.ps

[12] M. K. Stern and B. Woolf. Curriculum Sequencing in a Web-Based tutor. In Proc. Of Intelligent Tutoring Systems (1998).

[13] M. K. Stern and B. Woolf. Adaptive Content in an Online Lecture System. To appear in Proceedings of International Conference on Adaptive Hypermedia and Adaptive Web-based Systems (2000). Also available at http://www-aml.cs.umass.edu/~stern/papers/ah2000.pdf