Web Presentation and Authoring of Interactive Medical Cases

Ted D. Wade, Monica Younger, Carol Kamin and Robin Deterding

Abstract—We developed web applications to integrate video and other media for presentation and remote discussion of pediatric medical cases. Bulky media are stored on CD-ROM. A web server controls presentation from the CDs and coordinates communication among participants. Initial success led to demand for wider adoption and quicker authoring of cases. We added web administration of participants, and a web-based authoring tool which gives authors the flexible control needed for problem-based learning. In the five-year evolution of this system web discussions helped system design, teacher training, and curricular integration for seven medical schools.

***Index Terms*—e-learning, problem-based learning medical cases, distributed authoring.**

I. Introduction

WE tried to address several problems with how to more effectively teach medical students clinical skills – examining, diagnosing, treating, and counseling patients.

The problems derived mostly from changes in the healthcare system that affect the number and type of patients the students are able to care for and learn about in today's hospital setting. Since hospital stays are generally shorter [1], and only the sickest patients are hospitalized [2], students are limited in what they see on the wards. Students who are on a rural rotation in a primary care setting will not see the same types of patients that students in urban hospitals see. Students doing a rotation in the hospital in the summer won't see the same types of patients that seen by students who do their rotations in the winter. By implementing the cases in the curriculum, all the students -- regardless of where they are geographically or seasonally -- have a chance to learn the same important topics. Additionally, the Council on Medical Student Education in Pediatrics (COMSEP) reports that students do not have enough time to gain competencies in areas such as child abuse, genetic disease, and adolescent issues during a pediatric clerkship [3].

We hypothesized that an effective approach to these educational problems would be to create a *distributed* Problem-based Learning (PBL) program, combining authentic digital video cases to support the pediatric curriculum with collaborative learning [4]. PBL includes tutorial groups consisting of 4-8 students and a facilitator, reviewing a clinical case. The group's activity models clinical reasoning by actively involving students in observation, hypothesis formation, and hypothesis testing. Through collaboration *and* self-directed learning, students activate prior knowledge, while elaborating upon and encoding new knowledge. We decided to see if PBL could be successful when the members of the group were physically remote from each other, and discussion was asynchronous, mediated via the internet.

We have now spent five years in refinement and application of web-based technologies to the solution of these problems. That period was divided into three roughly overlapping stages in which experience led to new objectives for better effectiveness and wider adoption. The objectives were met with new technological solutions, integrated as much as possible with the old ones. An initial focus on case presentation led to an emphasis on group process and administration to make the system more widely available. This was followed by work to facilitate the authoring of cases and their sharing across institutions.

We intend this paper to present a detailed case history of progressive refinement of an internet educational application, showing how substantive issues were met with technological solutions.

II. Case Presentation

A. Rationale .

The initial case presentation design was guided by our hypothesis of how problem-based learning groups could function when geographically dispersed.

We wanted generous use of rich media, particularly video clips from a few seconds to several minutes in length. The media would simulate the sequence of events which normally occur during the diagnosis and initial treatment of a patient.

This sequence includes the initial interview and taking of medical history, a physical examination, issuance of first orders for care and laboratory procedures, study of lab results,

Manuscript received October 25, 2002. This work was funded by the Fund for the Improvement of Post-Secondary Education, United States Department of Education.

T D Wade is with the National Jewish Medical and Research Center, Denver, CO 80206 USA (phone: 303-398-1877; fax: 303-270-2131; e-mail: wadet@njc.org).

Monica Younger is with University of Colorado Health Sciences Center, Denver, CO 80262 USA. (e-mail: Younger.Monica@tchden.org).

Carol Kamin is with the University of Colorado Health Sciences Center, Denver, CO 80262 USA (e-mail: kamin.carol@tchden.org

Robin Deterding is with the University of Colorado Health Sciences Center, Denver, CO 80262 USA (e-mail: deterding.robin@tchden.org

and final diagnosis and treatment plan. Just as in a real case, students would have to respond to one stage, trying to understand its implications, before being allowed access to the clues of the next stage.

Communication among the geographically dispersed students and a teacher (whose role is as a "facilitator") would have to be asynchronous to accommodate a variety of schedules, as well as students' needs to study other sources of information. One complaint that students have is that they are too frequently pulled away from patient care to participate in other educational activities. By allowing the students to join in the learning experience at any hour of the day, they can structure their time to suit their schedule.

As much as possible, we wanted simplicity of use. Minimal technical requirements for any computer involved would serve this end, and make it more likely that participants could work from wherever they happened to be.

B. Methods

1) Choice of platform.

We chose CD-ROM as a delivery vehicle for the case presentation and media. The first two cases each required about 400 megabytes of storage, 99% of which was video. We preferred a hardcopy medium over streaming because the application required substantial image quality and quick response time. Streaming video may work for short, low-resolution video clips, but since we were asking users to appreciate sometimes subtle nuances in the videos, we had to use higher resolution video that would not have worked well in a low bandwidth situation [5]. The video format (QuickTime) was chosen for its relative ubiquity among browsers of the time. We chose, and continue to use because of its productivity and power, Cold Fusion middleware for server-side programming.

2) Modes of Participation.

Asynchronous, text-based discussions were implemented using http. Text messaging is relatively easy to use. Text also makes it easy for participants to review past discussions. We decided to do our CD-ROM-based media presentation using html pages as well. An all-web approach would give us economy of development, distribution, and training.

We decided on several modes of student input: postings, orders, and discussions. Each of these modes had a different purpose in the problem-based learning process.

Postings were non-interactive, similar to a diary or personal notes. On a postings page, students would assert what they thought were the facts of the case as currently known, what their current hypotheses were about the diagnosis, and what their own personal learning issues were with regard to the case material. Before each round of discussions, students were expected to update their postings. A posting was unlike a diary in one way. A student could see other students' postings, but only after he or she had made his own. Seeing what others said was intended as a kind of reward, making a virtue of the required act of posting. We also expected that the act of committing oneself to a position in posting would be a stimulus for more vigorous participation in discussion.

Orders were another non-interactive entry, intended to simulate writing orders in a medical record. Because a real medical record is a legal document, students could add to their orders, but never change old ones. Postings could be changed to reflect changes in the student's thinking, but behind the scenes the database kept all versions for further study by educational researchers.

Discussions were just a typical bulletin-board type of application. Discussion threads were limited to one level, both to keep discussion on track and to make the mechanics easier to grasp. We made each student's photo available by a link, as a way to make the interaction more social.

3) Staging Control and Contingent Links.

In the first cases there were several instances where links to further material should only work if the student had already done something else. We called this *staging control.* Since the relevant state information had to persist no matter what computer the student was using, state had to be maintained on the server.

Implementing the contingent links for staging control was a simple matter when the link was to a server page. When the page request came in, logic on the server decided whether to send the requested page or a message saying what the student needed to do first.

It is easy to link from a page on a CD-ROM to a page on a web server. Our challenge was when the link was from a server page to a page on the CD-ROM. This requires (see below) that the server knows how to locate the page on the remote web client. Others researchers have linked web sites to remote CD-ROMs by, explicitly asking the user of the remote PC how to designate the CD-ROM drive on that PC [6,7]. We devised something much more automatic and platform-independent.

To allow access to the requested page in our application's case, the server had to send the browser an HTTP header to redirect to the page, and the header had to include the complete file system path to the page on the browser's CD-ROM drive (Fig 1). The path necessarily is different for different browser PC's. Our solution was to include the path as part of the page request sent by the browser. The path was constructed at runtime using JavaScript. Given the path, the server could construct and send the appropriate redirection back to the browser. In principle this might have been considered to be a security risk – allowing a remote server to tell a browser to open a page on the browser's local drive. In practice browsers never gave us trouble on this issue. We only got into cross-domain browsing issues in a later version of the system.

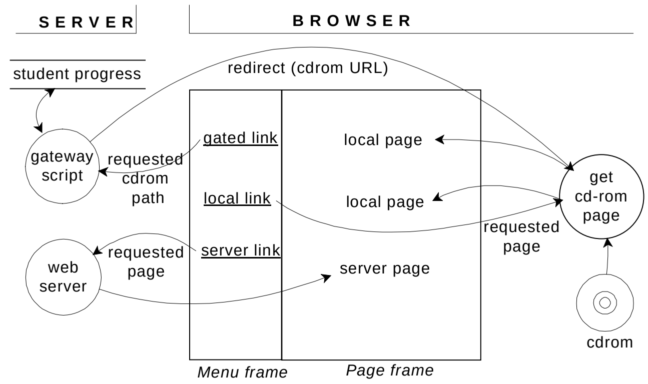

Fig. 1 Contingent versus normal links in a mixed server / cd-rom web application. Contingent ("gated") links send the local path of the desired cd-rom page to a server gateway script. The script checks its database to see if the request is allowed, and, if so, sends back to the browser an http redirection header which contains the local path.

4) *Web-based Project Coordination.*

Because the implementation team was also scattered across a city, and because we wanted a historical record of how our project went, we used a version of the same web discussion system to discuss system design and implementation issues.

C. Results

We [4] compared groups meeting "virtually" (using the online/CD-ROM system) to other groups, meeting face to face, but using either text-based or video-based presentations of the same cases, and to control groups without PBL experience. In performance on similar problems, and on self-rated confidence about the case material, PBL students were better than controls, with no difference seen between face-to-face and virtual groups. When we examined the transcripts of group interactions for semantic content [8] students in the virtual groups engaged in less "rapport building" behavior, but higher in "critical thinking". When asked in focus groups, students said they preferred face-to-face groups. Students in remote areas nevertheless appreciated the ability to collaborate with their peers and a faculty member [9] .

Technical support issues were fairly rudimentary. Students often waited until the day the case was beginning before attempting to determine whether their web browser had the required media plug-ins. Students using older computers often need to update the plug-ins. In a later version we countered this problem by providing the installers for the plug-ins on the same CD as the cases, so that if the students needed the plug-ins, they could install them without waiting for the computer download.

A smaller problem involved browser cookies. If the browser was configured to not accept cookies, the server-based session control would not work properly. Once the browser settings were corrected, most users had no trouble using the application.

Students and faculty thought it was burdensome to have to check frequently for new messages. Most participants were more familiar with the model of working through a self-study CD-ROM tutorial. Even though the discussions were asynchronous and the application was designed to be used to suit one's own schedule, in order for a meaningful dialogue to take place, users must participate in the discussion a handful of separate times during the week. The concept of integrating the group discussion over a set period of time had to be introduced early to the learners.

III. Refinement: Group Process

A. Rationale

1) *Enrollment.*

In the project's first stage about 70 students per year participated in the virtual cases, in groups of 4 or 5 students plus 1 or 2 facilitators. We also wanted to demonstrate the system to faculty from other universities. Some of these were interested in having their students try the cases. There was talk of having students from different institutions work through a case together. To handle the enrollment chores, and looking ahead to when we would focus on disseminating the system to other institutions, we needed an application for administering participation.

2) *Participatory Roles.*

When demonstrating the system we did not want students to be concerned with whether other educators were "visiting" the case. We also needed to have educators visit a case specifically to learn how to be facilitators themselves. We clarified the concept of roles in participation to include

- *teachers* – Who would be exempt from stage control, could participate in any group any time.
- *students* - Who could only participate in a given case during the week when their groupmates did
- *auditors* – The new category, who could participate any time, were subject to stage control like students, and not listed in the roster

Student behavior and attitudes made us want to enhance the social dimension of the virtual groups as much as possible. One of the strengths of PBL is that students learn from each other. We reasoned that we wanted them to feel as much a part of a group as possible.

B. Methods

1) *Multi-level Administration.*

We created a web-based group administration tool because a web service at one institution would often be supporting usage by participants at another, remote institution.

The administration application had two tiers. The upper tier allowed a new institution to be defined in the database. The institution would then have one or more group managers defined. In the second tier, group managers would define study groups. The members of a group would be expected to work through one or more cases together.

A person defined in the database could be in multiple groups, and have different roles in each. For example, a professor could be an auditor in one group while learning how to facilitate PBL. Later the professor could be a teacher for other groups.

Each institution would be responsible for entering their own people into the database. When defining a group, a group manager could pick from a list of their institution's people, but, with slightly more effort they could include in the group people from other institutions. The overall intent was that administrative burden would be spread across institutions, but that institutions could cooperate in training teachers or even in educating students.

Other administrative tasks involved scheduling when a groups of students could undertake each case, and automating the upload of participants' portraits.

2) Social Enhancement.

To enhance the sociality goal, we decided to make each person's photo appear next to each of their discussion messages. There is evidence [10] that if students see their colleagues' photos and begin to sense each other as members of the same small group, it can help to overcome some of the constraints of web-based conferencing and make the interaction feel more like a face-to-face group.

Another aspect of sociality was the slow cycle time for replying to a discussion message. To shorten the interval between messages, we decided to add email notification about new messages, and to make it possible for learners and facilitators to easily check and reply to messages even without using the CD-ROM.

3) New Version Costs

Even without these last changes, the group administration enhancements meant that the application itself had to change We thus incurred the costs of what was, in essence, a major version release for the application.

C. Results

1) Media Control and Versioning

In releasing the new version we had to re-issue the CD-ROMs for the cases, including new masters for our partner institutions. We realized that there needed to be a clearer distinction between the hardcopy case material and the server's case presentation side. CD-ROMs for new cases tended to replicate parts of what should have been a central engine.

2) Need for Consortium

The net result of the version two changes was a more refined, attractive system. The high costs of developing video cases meant that we still had only two cases implemented. However, other institutions were showing more interest in using the system. We decided to use their interest to build a consortiuum of institutions that would each develop new cases and share their cases with others.

3) Self-study.

Continuing success also meant that we had a need to demonstrate the case material to colleagues who were not part of an actual PBL group. Meeting this need by hand-crafting special, non-networked CD-ROMs was too laborious. We needed a simple way to unlock the content on a CD-ROM – in effect, "re-purposing" the material for simple demonstrations or self-study.

IV. Refinement: Authoring

A. Rationale

A major revision of the software was needed to serve the new needs of the consortium and to correct user interface, operational, development and deployment deficiencies. It was a long-standing goal of the original development team to be able to freely share cases and continue to enhance learning in medicine. Thus an authoring tool was necessary to make it simpler for partner institutions to put cases together in this unique format.

1) Navigational Structure.

The navigational structure of the original cases had never changed. It consisted of two menus in separate frames, plus various other links placed within content pages to guide students to make their own postings, messages or orders at appropriate stages. Overlaid on this were help pages directing the students to do certain things on certain days of the week, and some of these pages had their own navigational links. The philosophy had been to make small changes to the original design in response to student complaints or confusion. This incremental and *ad hoc* approach did not give students a sense of where to look in order to know what to do next. They always had to read very carefully. What probably made the system usable was that, overall, a virtual case followed the same stages of investigation as real and textbook cases – a structure drummed into students in various ways throughout their medical education.

2) Application Factoring.

The lack of a clear navigational structure also meant that it was very hard to imagine how one might separate content from the presentation engine, or how to structure the authoring of a new case. We needed a new and clarifying concept of just *what a case was* in order to proceed.

3) Publishing Structure.

At another level we needed a publishing structure that would allow different organizations to create and share cases. No matter who created a case, any partner should be able to host it on their own web site. And we needed to accomplish this sharing within an atmosphere of changing intellectual property concepts and changing university understanding of their own interests.

B. Methods

1) Task of a Case Authoring Tool.

Our clarifying concept was that a case is a series of pages which students visit and interact with in a series of stages. We call the stages a "schedule" even though the flow of events is primarily relative to each other, and not to a clock. Case designers need flexibility in the constraints they choose for a schedule. The structured unfolding of clues is what a facilitator needs to synchronize and thus harmonize the work of the student group. We therefore realized that what we needed for authoring was a tool to *string pages together* into such a schedule. The actual creation of pages, and of the content for those pages, requires readily available tools whose complexity is far too expensive for us to re-create.

2) Scope of New Version.

We began by mocking up a new page design, with a single menu on one side. This menu was to implement the schedule. It would take the place of all the previous menus, schedule-help pages, and *ad hoc* links. We posted this mockup, plus lists of suggested improvements and features, on our team's web board so that we could have preliminary online discussions before making decisions in face-to-face meetings. Only when we had a fairly clear idea of look and feel did we begin the design of:

- a new case-serving engine,
- an authoring program,
- and the integration of the existing administrative application with authoring into a kind of system console.

In Fig. 2 we show the overall data flow of the new system. Case authors create most pages using standard tools on their own workstations. They upload those pages, which are placed in an authoring database, along with information about the case structure (the schedule). At any time the information in the database can be used to *compile* the case into a set of files. You test the compilation by having a browser open the start page. For testing, the case files can be put on a CD-ROM, or on any disk drive available to the browser's workstation.

The case-serving engine is the same for all cases. It only needs a couple of small files containing the compiled case schedule. It uses a second database, the operations database, to hold state information about who is using the case, what stage they are in the case, and what their role is in their group.

A completed case is always used by copying its associated files onto a CD-ROM. Each participant uses their browser to access the case content and schedule from their own copy of the cd-rom. The cd-rom also has -- hard-coded of course --the URL of the case on the server. This means that a given compilation and cd-rom is made for use with a particular server. The alternative we rejected would have been to make a general-use cd-rom that would ask the student to either: select from a set of servers (this could be only those servers known to be available at compilation time) or type in a URL.

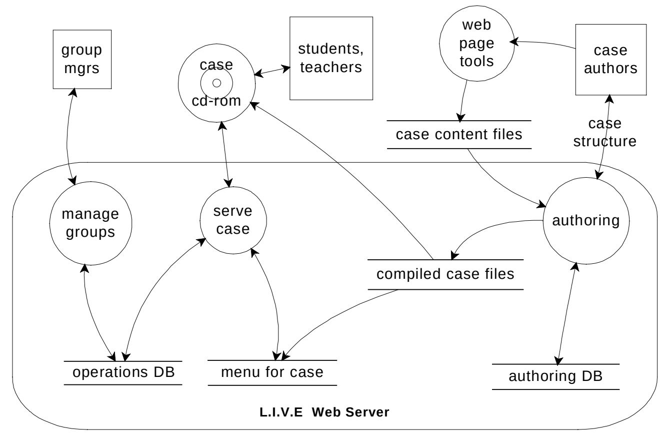

Fig. 2 Data flow for the LIVE system. There are three web applications: group management, case service, and case authoring. A case compiled on the server is transferred to a cd-rom for use by teachers and students.

3) Sharing Cases.

The mechanics of sharing cases are simple. (Fig. 3) To make a case suitable for use with server A, an author must login to server B where the case was authored, enter the URL of server A into a form, and then compile the case for server A. The author then obtains the compiled files and makes cd-roms from them.

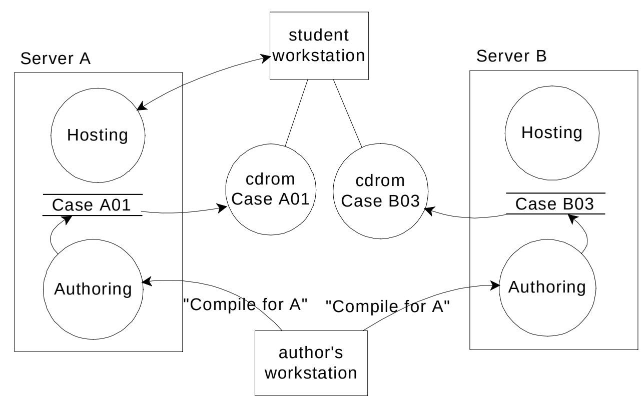

Fig. 3. Example of sharing cases to another institution's server. The author makes both cases -- Case A01, authored on server A, and Case B03, authored on server B -- available for use by students connecting to server A.

We added an encrypted unlocking code to the compiled case. If a person with the CD-ROM enters the code, they can see and use the entire case schedule menu except for the pages which come from a server. Thus any CD-ROM can be used by an authorized person for self-study.

The consortium partners adopted an agreement for sharing cases and related software. The partners share a perpetual, royalty-free, non-transferable license to all the material. The originators of the original system (the hosting and authoring software described in this section) retain the right to license it to others. The partners must agree on terms of licensing follow-on enhancements (such as new cases) or educational services to other parties. The partners also agree to consultation in advance of any publishing or application for research funding based on the shared software.

4) Authoring Work Flow.

The authoring program creates an inventory of pages that can be placed on the schedule for a case. Some pages are provided -- as either "canned", unmodifiable, pages or as examples of appropriate kinds of pages. An author uploads to the inventory both original pages and modified versions of the example pages.

The author creates the schedule by selecting pages from the inventory, arranging them in a sequence, and creating subgroups of pages called steps. Generally all the pages of a step must be completed by the student before she can go on to the next subgroup. The author can fine-tune the schedule by designating which pages of a stage are or are not required, and by setting gateways which students can only pass by typing in specific kinds of responses to the case. Fig. 4 shows the user interface for the scheduling task. The author sees a representation of the case schedule, complete with indications of staging control and help links. Pages from the inventory appear in a drop-down list so an author can insert them into the schedule, as well as rearrange and rename any page on the schedule.

The authoring program lets you create the case in an iterative fashion - you may go back and forth between the different tasks: page inventory, page scheduling, and compiling, until the case takes the shape desired.

In order for authors to upload web pages into the authoring system, we had to make provision for uploading the needed media files as well. The system firsts uploads the html page. It then parses the html page, looking for references to media files. It changes the page so each reference assumes that the media file is on the local directory. It creates a list of the needed media files, looks to see if they are in its inventory, and presents the author with a page where they can upload new or replacement media files.

5) Integration of Server and CD-ROM Control

Fig. 5 shows the new way we coordinate server and CD-ROM to achieve staging control. In the original system (Fig.1) the server was contacted only if a page was needed from it. In the new version we want to keep a history of every page which the student visits, and the order in which they were seen. This history necessarily has to reside on the server. So every time the student clicks a menu item to request a page, a dispatch script on the server is notified. It consults a student participation database, and if the rules of the case allow the student to see the page, it is either sent directly from the server, or a redirect is sent to get it from the cd-rom. The dispatcher then updates the student participation database. If the student is not authorized to see the page yet, the dispatcher sends back an explanation of what the student will need to do first.

On the client side, the arrival of a new page causes the schedule menu to refresh itself. The menu needs to show where the student currently is, so a representation of the menu state is kept in the browser cookie storage. We made the menu frame local to the browser to avoid problems with some privacy features on browsers. Browsers have various rules about whether a page from one domain, X (such as a particular web server) can pass information to a page from another domain, Y (such as a CD-ROM). Such information might be a new page, or just the value of a JavaScript variable. We found that if the menu frame and content frame both originally were created with content from the CD-ROM, then the menu frame could communicate with the server directly, and also could cause the contents of the content frame to be replaced with a page from either source. If one had a menu frame that came originally from a server, it would be blocked from making changes to the content frame.

6) Case User Interface

Fig. 6 shows the compiled case schedule menu as a PBL participant would use it. The current page is highlighted on the menu. A given page's entry on the schedule can have a graphic marker which, if moused, will display a short explanation of that stage of the case. Schedules can be long, having maybe 15-30 steps, so the scrolling position of the schedule menu is maintained whenever a new page is loaded into the display frame.

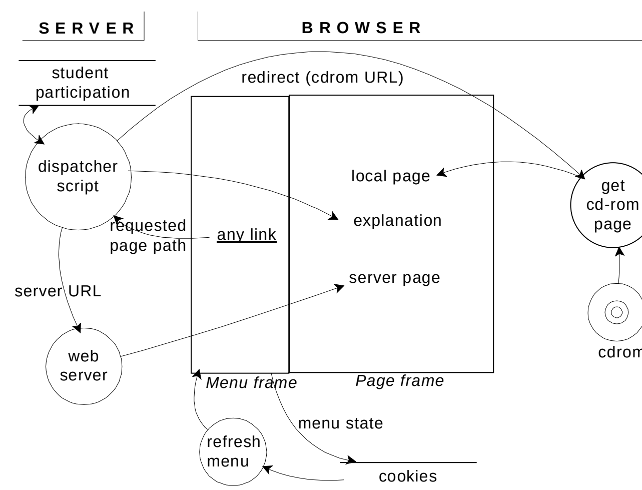

Fig. 5. Integration of server and cd-rom control. The dispatcher on the server handles requests for both server and cd-rom pages, and maintains a database of student participation (pages viewed, responses made). After each selection the menu frame, containing the case schedule, refreshes itself using state information stored in cookies.

C. Results

1) Conversion Costs.

As of this writing, the server software, including authoring, has been installed at two medical schools. Three different institutions have piloted use of the system with their students, and three more have definite plans to use it. We do not yet have enough experience with the new system to point out its particular successes or failures. We are still convinced that growth of the program required the major revision, but we are still paying the price for conversion.

One price is that five cases in various phases of completion have had to be reconfigured to work in the new authoring system. Other cases are being created from the beginning in the new format. Another price of conversion is that our partners, except for one, have yet to integrate our technology platform into their operations. When a school does convert over to the new platform, all the CD-ROMs it has been using have to be replaced.

Creation and maintenance of a valid, current CD-ROM pool is problematic. More often than we would like, a student gets an out-dated or non-functional CD-ROM. We think that we are still years away from having adequate bandwidth to deliver clear, smooth, high-resolution streaming media to our participants. We therefore have to live with the problems of hard-copy. When we do convert to streaming, no doubt major software revisions will be needed.

2) Authoring Costs.

Authoring of a new case is still expensive, even with the authoring tool, because so much effort occurs outside of the tool in case design, media production, and peer review. We do not yet have any idea how the authoring tool will help reduce the time and effort of case production. Of course efficiency is not its only function. The tool also helps to impose a predictable structure, appearance, and usability to cases, which should make them easier to learn and teach with.

3) Curriculum Integration.

Another problem in the dissemination of our system is the relatively conservative nature of medical school curricula. It is particularly hard to get educators to take student time away from one established piece of the curriculum and allocate it to new material or a new modality. Students tend to spend 8-10 hours per L.I.V.E. case. Finding time for that takes an curriculum advocate with real commitment to achieving change. We have planned a study of the process of change – adopting our particular educational innovation – itself.

4) Evaluations.

Lacking an experience base, we can not yet say if our user interface re-design has made it easier for students to navigate.

Now that we have six other medical schools who are implementing the cases and the technology into their curriculum, we have several studies that are currently in progress or are in the planning stages. For example, one study examines how the participation in the virtual PBL experience affects the feedback directed to medical students in a clinical environment. Another study underway is examining the effect of group size on different aspects of learning in this environment.

5) Future Improvements.

Our experience, both anecdotal and from systematic research, suggests that our system can still be improved in specific ways. Among other changes we would like:

- An easier, integrated, way of having private discussions among facilitators about how to help particular students who are working through a case. We have been using a general purpose discussion board for that, but it is not integrated with the L.I.V.E. system, and so could be more convenient and effective.
- Better integration of the non-interactive postings and the interactive discussions. Students and teachers need to cross reference between these, yet they appear on separate pages. . Instructors would like more control over the types of non-interactive postings students are asked to make There are complex issues of who sees what, when they see it, and what actions a user would take to reference something already said. As always, limited room on a display is a factor
- Modifications of the authoring tool to use in situations other than problem-based learning. For example: A variation for testing which does not allow students to see each others postings; A way to integrate multiple, short cases for use in continuing medical education for physicians

V. Conclusion and Expectations

From a technological point of view the project has evolved in stages, where the ideas for each stage usually became clear only after experience with the previous stages. We certainly borrowed from elsewhere educational concepts and technological techniques in coming up with improvements, but much of this was new ground, and experience was the best guide to design change.

From an educational point of view the project has been remarkable for how much educational objectives have been the driving force, rather than technological innovation for its own sake. Even so, the internet milieu and particularly the power of internet applications enabled us to apply the powerful problem-based learning paradigm to distance learning situations, solving practical educational problems along the way. Since Problem-based Learning has been used in other disciplines besides medicine [11, 12], perhaps other educators would be interested in our approach.

References

[1] D.A. Shugars, E.H. O'Neil, J.D. Bader (Eds). *Healthy American: practitioners for 2005, an agenda for action for U.S. health professional schools*. Durham, NC: The Pew Health Professionals Commission, 1991.

[2] J. Cohen, (1998). Leadership for medicine’s promising future. *Academic Medicine*, vol. 73, no. 2, pp. 132-137.

[3] A Olson, R Berkow, & J Woodhead. “The national general pediatric clerkship curriculum: Progress I implementation,” presented at the Annual Meeting of the Association of Medical School Pediatric Department Chairs and Council on Medical Student Education in Pediatrics in Bar Harbor, Florida, 1998.

[4] C.S. Kamin, R.D. Deterding, B. Wilson, M. Armacost. M.A. Breedon, 1999 “The development of a collaborative distance learning program to

facilitate pediatric problem-based learning," *Med. Educ. Online*, 4, 2. Available: http://www.med-ed-online.org/t0000008.htm

[5] G. Lorenzo "Technological minimalism: A cost-effective alternative for course design and development." *Distance Education Report*. v. 5, no. 7, p 8. Apr 1 2001.

[6] N Mattheos, A Nattestad, & R. Attström "Local CD-ROM in interaction with HTML documents over the Internet." *European Journal Of Dental Education* v.**4,** no.3, pp. 124-127.

[7] T. Bacro B. Gilbertson, J. Coultas "Web-delivery of anatomy video clips using a CD-ROM." *Anat Rec*. 2000 Apr 15, v.261, no.2, pp.78-82.

[8] C.S. Kamin, P.S. O'Sullivan, M. Younger, & R. Deterding "Measuring critical thinking in problem-based learning discourse." *Teaching & Learning in Medicine* 2001; v.13,:pp.27-35.

[9] C.S. Kamin, R. Deterding & M. Lowry, "Students perceptions of a virtual PBL experience." *Academic Medicine*, to be published 2002

[10] S. Malikowski. "Interacting in history's largest library: web-based conferencing tools." *Web-Based Instruction* Educational Technology Publications Englewood Cliffs, NJ 1997, pp 283-294.

[11] D. Boud & G. Feletti. *The challenge of problem-based learning*. St. Martin's Press, New York, 1991.

[12] LA Wilkerson & WH. Gijselaers, "Bringing PBL to higher education: theory and practice." *New Directions for Teaching and Learning*. Issue 68, Jossey-Bass, 1996, pp 101-104.

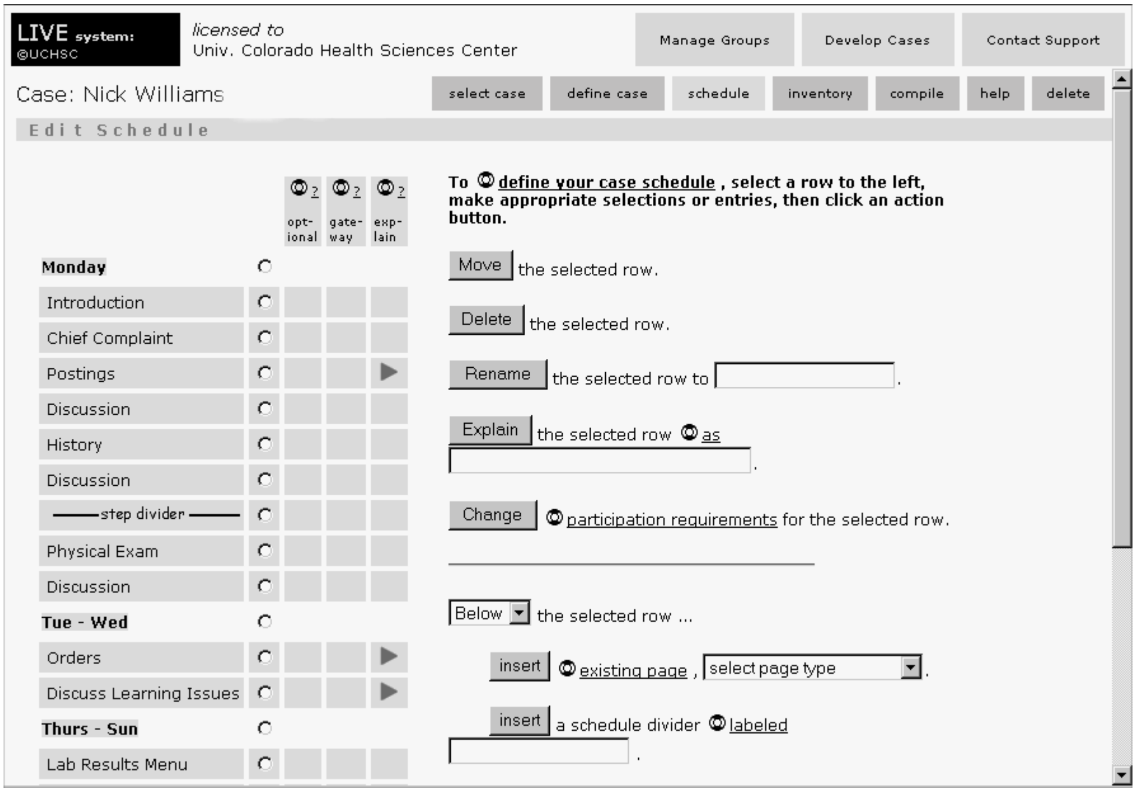

Fig. 4. Editing a case schedule in the authoring program. The schedule is on the left, while the actions to edit it are on the right. Schedule dividers divide the schedule into sections, and can also serve as reminders of when a task is to be done. A section usually must have all its menu items completed before the student can move on.

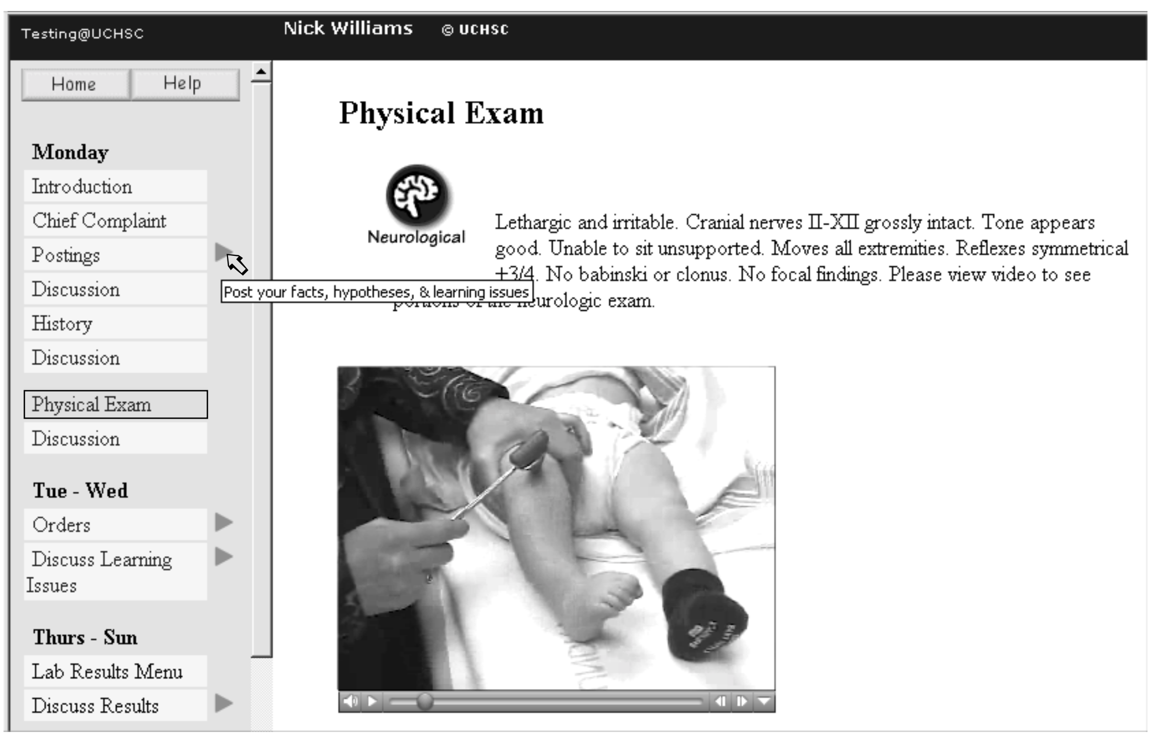

Fig. 6. How a case schedule appears to a student. This is the compiled, cd-rom, version of the case in Fig. 3. The menu indicates that the student is currently at the Physical Exam step. The menu shows vertical gaps between stages. A student must finish each step (i.e., page) in a stage before being allowed to go on to the next stage. Labels in some of the gaps indicate time span when the student is expected to complete subsequent stages. The cursor is over an icon offering an explanation of the Postings step.