Jl. of Computers in Mathematics and Science Teaching (2001) **20**(3), 307-321

The WebCOSY System for Course Management in Distance Education

JENS HOEFKENS, LARS DIENING, MARTIN BERZ,
AND BÉLA ERDÉLYI
Department of Physics and Astronomy and
National Superconducting Cyclotron Laboratory
Michigan State University
East Lansing, MI 48824 USA
hoefkens@msu.edu
diening@mathematik.uni-freiburg.de
berz@msu.edu
erdelyi@nscl.msu.edu

The WebCOSY system is an interactive web-based homework and course management system that has been developed and used as part of VUBeam (Berz & Hoefkens, 2001), an Internet based remote Physics M.S. and Ph.D. program at Michigan State University. Besides supporting problems written within the system, it has the unique feature of also supporting processing and grading of student's submissions by arbitrary external programs, and hence provides rather far-reaching flexibility for the assignment of homework problems. The system supports and manages databases for classes, students, scores, graders, as well as homework problems for re-use in other classes. Written entirely in Perl, the system is easily manageable and portable.

The WebCOSY system (Hoefkens, 2001) is the main course management engine of the VUBeam Internet based remote graduate curriculum at Michigan State University (Berz & Hoefkens, 2001). VUBeam is a joint initiative of the MSU Virtual University, the MSU Department of Physics and Astronomy, and the US Department of Energy. It provides graduate education to participants in Beam Physics, including the ability to earn M.S.

and Ph.D. degrees in cooperation with external mentors at US and Foreign National Laboratories.

Beam Physics is a rapidly developing sub field of Physics that is connected to some of the largest and most expensive scientific experimental devices located at major particle accelerator laboratories. As such, the traditional educational clientele is situated predominantly outside major research universities and scattered at many sites in the US and the rest of the world. Because of these specific circumstances, the field has been recognized as a prime candidate for methods of distance education. As a response to this need, the VUBeam course and degree program has been initiated, and it represents the first distance learning Physics program that offers Ph.D. degrees.

The program was launched in 1997, and the degree options were officially implemented by MSU in 1998. Since that time, the program has served nearly 200 course participants from a large number of locations, most of which are listed in Table 1.

Table 1
Participating Sites in the VUBeam Program

Site Name	Country	Participants
Argonne National Lab.	USA	23
Beijing University	China	2
Brookhaven National Lab.	USA	9
Calcutta University	India	3
Campbell University	USA	1
CEBAF (T. Jefferson National Lab.)	USA	11
Cornell University	USA	1
DESY	Germany	1
Dubna Laboratory	Russia	2
Fermi National Accelerator Lab.	USA	10
Kansas State University	USA	2
KVI	Netherlands	4
Los Alamos National Lab.	USA	1
Lawrence Berkeley Nat. Lab.	USA	7
Lawrence Livermore Nat. Lab.	USA	2
Mississippi State University	USA	1
Michigan State University	USA	18
Notre Dame University	USA	2
Oak Ridge National Lab.	USA	1
Sandia National Lab.	USA	1
Saratov State University	Russia	1
Stanford Linear Accelerator Center	USA	2
St. Petersburg State University	Russia	6

(continued on next page)

Table 1(continued)
Participating Sites in the VUBeam Program

Site Name	Country	Participants
Stony Brook Laboratory	USA	1
TRIUMF	Canada	3
Ukrainian College	Ukraine	1
University of Antananarivo	Madagascar	1
University of Chicago	USA	1
Univ. of Illinois, Chicago	USA	1
University of Islamabad	Pakistan	1
University of Texas, Austin	USA	2
Université Laval	Canada	1
University of Helsinki	Finland	1
Unaffiliated		37

Internet-based technology has been used to arrange optimal educational settings for all these participants. Figure 1 shows the information flow of the course, including local delivery, live transmission by way of videoconference and Internet, as well as asynchronous delivery.

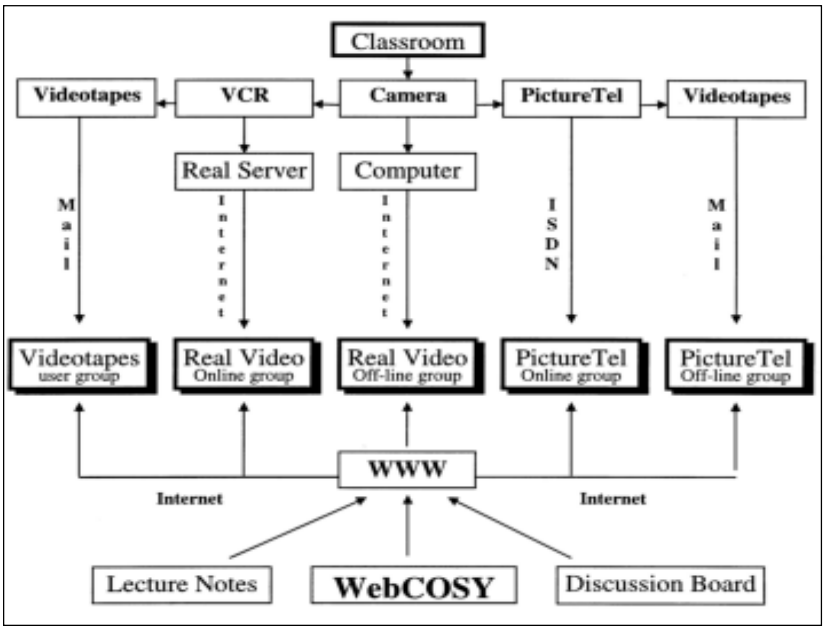

Figure 1. The information flow of Physics 861

To handle the substantial logistic problems involved with the distribution and collection of homework to online students, new methods of interactive homework distribution were developed. A major part of the VUBeam curriculum consists of programming using the COSY Infinity system (Berz, 1999a; Berz, 1999b; Makino & Berz, 1996; Makino & Berz, 1999), a special purpose-programming environment for Beam Physics. While offering the introductory graduate course MSU Physics 861 "Introduction to Beam Physics" for the first time in an online setting (Berz, Erdélyi, & Hoefkens, 1999) it was realized that a major part of the grading time on the homework problems is taken by non-grading operations. In fact the observation was that the grader spends only 15% of the time with actually grading while the remaining 85% are taken by extracting submitted source code out of emails and faxes and actually processing the user's input

It was decided to automate these tasks by using a web interface with a CGI backend. In addition to submission of source code, multiple choice and numerical problems were also used for this course. While the CAPA (Kashy, Morrissey, Tsai, & Wolfe, 1995) system was initially deployed for the latter problems, once the CGI backend was in place it was straightforward to incorporate these in the form of personalized problems into the system. As it stands now, the WebCOSY system is a flexible and powerful tool for remote and onsite graduate and undergraduate instruction.

ARCHITECTURE OF THE WEBCOSY SYSTEM

The WebCOSY system consists of a CGI backend that is written entirely in Perl (Wall, Christiansen, & Schwartz, 1996). It is divided in two parts, namely the student interface and the administrative front-end. They interact using the database backend of the system. The whole system can be used and administered entirely using the HTML front-end.

The WebCOSY program allows posing of problems to students in a personalized fashion. That includes but is not limited to personally randomized numerical quantities. The system currently handles multiple choice and numerical problems and allows the submission of COSY Infinity input files. The former are instantly and automatically graded by the system while the latter are processed on the server and the output of the runs is made available to the users and graders for review.

THE ADMINISTRATIVE FRONT-END

WebCOSY administrators can register new users with the system, store new problems in the problem database, set up new and assign or retrieve problems from the database for inclusion in the courses that fall into their responsibility. Moreover they can add users to these courses, establish co-graders and assign grades to students in their classes. All these functions are implemented using HTML forms with client-side JavaScript verification and the Perl CGI backend.

A clear relationship has been implemented that connects classes with graders. While administrators have full control over the courses that fall into their responsibility they have no power to interfere with classes for which they do not hold the grader privilege. The graders can set up new users and register them with the system using a simple HTML interface that asks for some basic settings like login names, passwords, and user's e-mail addresses.

Problem management is implemented directly on top of the database backend. Graders can retrieve problems from the database using a keyword query and can then include them into their own classes. This particular implementation of the problem database allows the WebCOSY system to grow with the user base. Naturally graders can also store new or changed problems in the database.

Once the necessary problems for a class are stored in the system the grader can set up a new class by either cloning an existing setup or creating a new one and assigning the necessary problems to it. After setting up the deadlines of the problems and adding students the class is ready.

Finally the grading and management of a class is performed using the "Grading Page" displayed in the Figure 2. It shows the enrolled students and makes the assigned problems and submissions accessible for grading. Moreover the list of assigned problems is available to make changes as necessary.

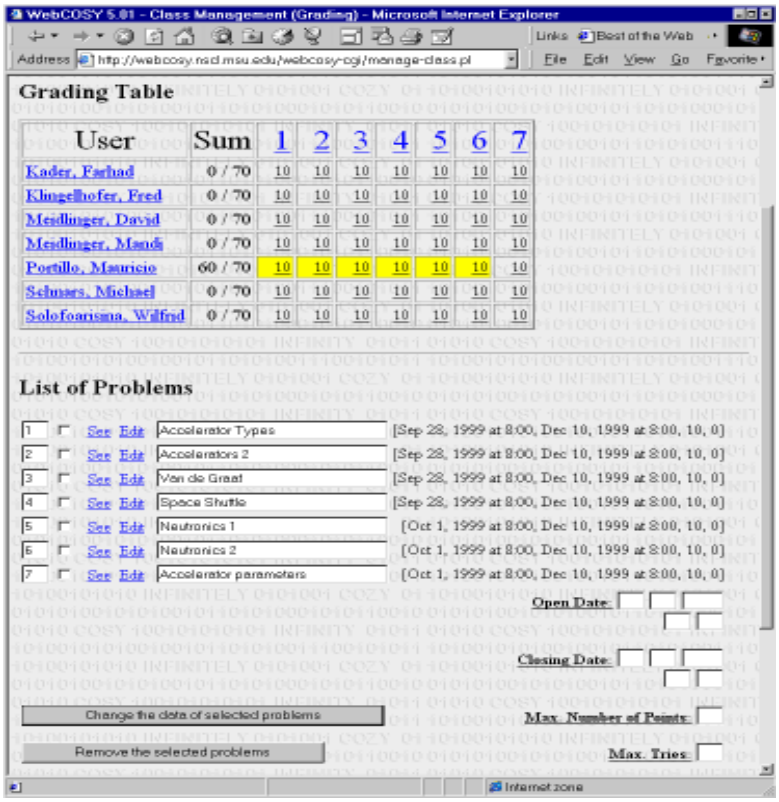

Figure 2. Example of a WebCOSY grading page

THE STUDENT INTERFACE

The student interface of the system is written in Perl and relies only on standard HTML that can be rendered in any complying browser. The student has to log in by supplying a valid user ID together with a corresponding password.

After successful authentication the main page of the system is displayed. It allows the student to either change personal information like name and address or to select one of the classes in which he or she is enrolled. Once a class has been selected the student can work on currently open problems or review previously submitted solutions for already closed problems.

If the student has selected a particular problem, the problem text will be displayed and the problem dependent rendering engine will display the solution entry form. The student now enters a solution and submits the answer. Depending on whether the validity of the problem can be determined by the system without grader intervention or not and whether the processing of the problem can be performed quickly the student either gets instant feedback on the submitted answer or information about a deferment of the grading process. The student can work on other problems. Once the background execution of the processing program is finished the output files are made available to the user for review and the student is given the opportunity to resubmit a solution until either the deadline for the problem has passed or the student is satisfied with the output. While this seems to be a rather simple interface it proves to be very powerful as will be explained in the following sections.

DESIGN GOALS AND IMPLEMENTATION

The WebCOSY system consists of the student interface and the administrative front-end. They share and communicate by way of the system's database. A separate system that is not directly connected to either of these parts is the execution backend. It is the outstanding feature that distinguishes the WebCOSY system from other web-based interactive homework systems. It allows the automatic processing of user supplied input and the redirection of the output to the Web front-end. While this is currently only implemented for the COSY Infinity programming system, it can easily be extended to any other system that processes user input like compilers or interpreters or even formula manipulators and basic essay evaluation tools.

Figure 3 shows how the output of a background-executed program is made accessible to graders and users. The regular textual output is displayed and any number of produced files can be displayed either directly in the browser or are available for download (note the button "Download the Postscript File" in the lower right corner).

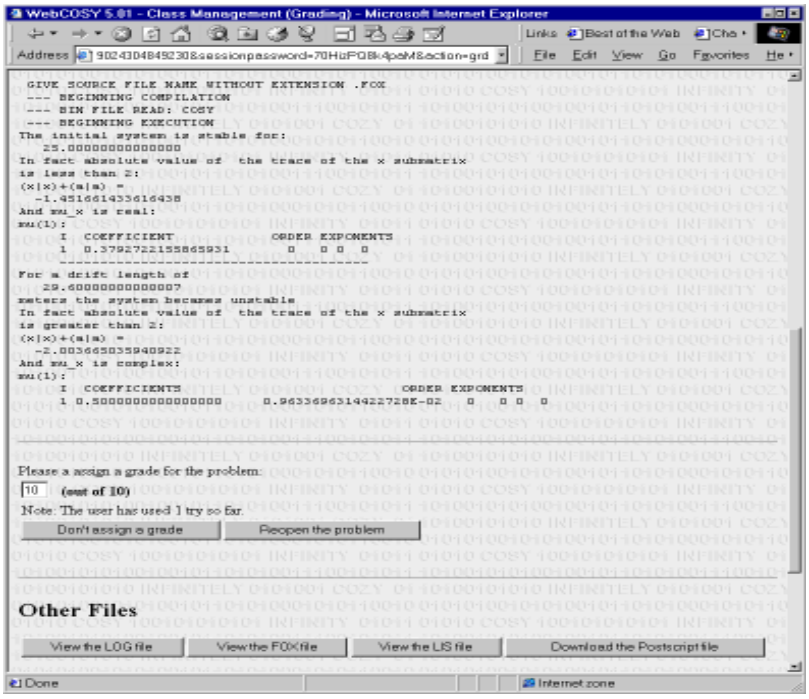

Figure 3. Display of program output

PORTABILITY AND MAINTAINABILITY

Since the expected lifetime of the system is in the range of many years and it is very difficult to foresee the development in today's computer technology, the system has been designed for easy portability to other platforms. It has been used on GNU/Linux, Solaris, and Windows NT using various web servers. To satisfy the portability goal the entire backend of the system has been written in Perl and as such is very portable. The total amount of platform dependent code is about 10 lines and these are grouped at a single location. So installation of the whole system on a POSIX compliant system with an installed Perl environment takes less than 10 minutes.

Two graduate students at Michigan State University (Jens Hoefkens and Lars Diening) have initially developed the system and based on its expected lifetime, the deployment of the system will outlast their presence at that institution. So maintainability has been an important factor during development. The code is well structured, fully commented, and uses object oriented features and standard Perl modules to make it easily maintainable and extendable.

EXTENSIBILITY AND PROBLEM PLUG-INS

Besides the currently implemented categories of problems the system is open for future extensions in both the problem types and the display of content. The system uses native HTML as the problem statement language. This allows the inclusion of virtually any current and upcoming web-based content including but not limited to Java applets and multimedia presentations. This allows the WebCOSY system itself to remain lightweight while not restricting the content developer.

Moreover using object oriented techniques provided by the Perl programming language a new problem type can be plugged into the system by just supplying three routines that determine how the problem is described, rendered, and graded. This again allows the system to be very flexible with respect to future extensions, standards, and applications.

Example

The preferred way of setting up problem descriptions is to actually write them in the Perl programming language and use the Perl function EVAL to evaluate the problem. The following is a small example of a numerical problem from the system. It is written in Perl and uses the WebCOSY provided function random_number (a,b,s) that returns a random number between a and b in steps of s. The random seed is based on the user's login name and hence every user is presented with a "personalized" problem statement.

The additional tokens GIVE, TRY, and RESULT are used. The latter announces the content of the variable $vol to be the correct result. Note that the token RESULT has an optional third parameter that specifies the desired accuracy of the result to be accepted. This accuracy goal can be specified in percent and may be either relative or absolute. The system default is 5% relative accuracy. The meaning of the other tokens will become clear in the next section.

```
# Test problem. Demonstrates the ability to include hints.

$radius = random_number(3,4,0.1);
GIVEN("radius",$radius);

$vol = 4/3 * $CONST{pi} * $radius**3;
RESULT("Volume",$vol);
```

```
if ($TRY > 0) {
  GIVEN("hint","The volume of a sphere of radius r is
        <i>V = 4/3 Pi r^3</i>");
}
```

PROBLEM DESCRIPTIONS

The WebCOSY system stores the problem descriptions as HTML code. That allows for inclusion of external documents like images and streaming media. While this keeps the internal overhead of the system small it allows for feature rich user experiences.

Using the very well documented and widely used programming languages Perl and HTML to set up problems makes it very easy for new users of the system to set up problems. A sample problem description for the previously given problem statement could be as simple as given in the next example.

Example

```
<img src="http://webcosy.nscl.msu.edu/images/some-image.gif">
Calculate the volume of a <u>sphere</u> with radius &$radius; meter.
&$hint;
```

For the student this example will then be rendered much like shown in Figure 4.

Variables that have been declared using the token GIVEN allow the inclusion in the HTML problem statement. They have to be referenced by their name and have to be included in the standard notation for special HTML characters. They will be replaced by the Perl backend before being sent to the client.

A special feature is the variable TRY that maintains a counter for how many times the user has tried to submit the problem. It allows for conditional inclusion of text in the problem statement. Its main usage is for additional hints that are shown to the user only if it seems to be necessary.

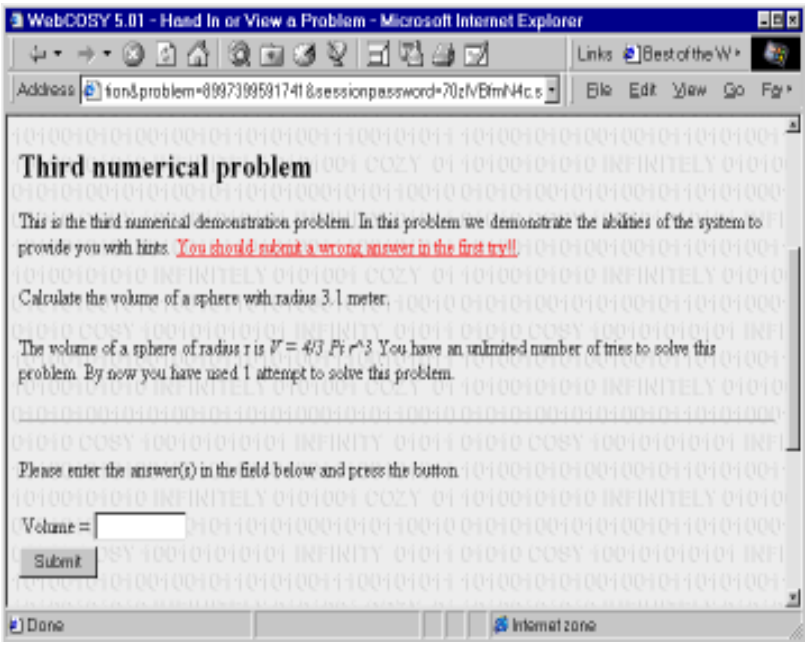

Figure 4. Rendering of a numerical problem

SCALABILITY

Deployment of the system in graduate and undergraduate remote and onsite instruction at Michigan State University requires it to be able to support multiple courses and user bases in the range of several hundred students. This has been achieved using a scalable database backend for the WebCOSY system. It not only stores the user and class information but also the previously assigned problems. This has the positive side effect that the usability of the system measured in the available number of tested problems grows exponentially with the number of courses it has been used for.

Moreover, the database backend allows the reuse of previously registered classes using a cloning feature. That simplifies the deployment of the system especially for online courses that are offered independently from the local academic schedule throughout the year. All the cloning of a class requires is the change of the deadlines of the problems and the addition of new users.

STANDARD COMPLIANCE

While it is tempting to build an interactive web-based environment using the most recent technologies like HTML 4.0 (Raggett, Le Hors, & Jacobs, 1998), media plug-ins, Java and JavaScript, the WebCOSY system intentionally avoids this by basing the student interface only on a subset of the HTML 3.2 (Raggett, 1997) standard. Nevertheless as explained earlier it is still possible to use these features by including them into problems statements.

This decision is based on the intended target audience of the system. It has been developed for a large-scale remote graduate instruction in Beam Physics with international participants. Since it had to be assumed that participants might not have access to the latest computer technology the system is based on a least common denominator approach while still being open for future extensions. However, this restriction does not apply to the administrative front-end of the system. It makes extensive use of more recent features like client-side JavaScript to reduce the processing load on the server and assure a more stable working environment.

Security

The implementation of the WebCOSY system in an educational environment requires tight security measures. It has to be guarded against external intrusion and malicious behavior of legitimate users. Moreover, the integrity of the used database is of prime importance since it includes the users' scores that directly affect their final grades. Finally the system may be accessed from computer pool settings that do not assign separate cache files for different users and do not provide individual cookie files, so no private information may be stored in the pages or cookies.

To this end the WebCOSY system uses a combination of web server security mechanisms, password authentication, and a token system of one-time passwords to authenticate users and enforce access restrictions within the system. However, by far the biggest threat to system security comes from the unique features of the system itself. Since users submit source code that is compiled and executed within the system it has to be able to establish secure environments that do not permit malicious or accidental violations of system integrity. This has been implemented using the "chroot" system call of the UNIX environment. If a particular system does not offer this facility it is highly recommended to use an external server to process the user input.

Ease of Use

The WebCOSY system uses an intuitive user interface both for the students and the administrative backend. It has a complementary online help system that greatly reduces the support necessary for students and allows for a quick introduction to the system for administrators. Every web page generated by the system contains explanations and hints to the available options and commands. Furthermore the administrative backend uses separate browser windows to display additional help on input formats. The online help system can be disabled at any time which is meant to be an option for experienced users.

The system maintains a database that keeps track of user preferences and options and by default every user is assigned to the "Demonstration Group." This is a special class that contains samples of all the available problems. It allows users to get experienced with the system without worrying about the grades. This group does not have any due dates nor does it limit the maximum number of tries for problems.

Performance

In its current implementation state the system is capable of serving several hundreds of users and classes simultaneously. The system itself is rather lightweight on the processing power of the server and currently it serves about 100 users in 10 classes on a single low end Pentium system running the GNU/Linux system.

If any one of the classes and involved problem types requires CPU intensive computation for the processing of the students input, the performance of the system may become an issue but generally the database backend allows the system to perform well even in high load situations.

If indeed the compilation and execution of the submitted source code solutions takes a significant share of the servers processing power it is possible to execute these jobs on remote machines or even a cluster thereof. This is even advantageous under security considerations since this separates the processing of potentially malicious user input from the server that runs the system.

USER EXPERIENCE

Users and course organizers alike have been highly satisfied using the WebCOSY system. The deployment of the system dramatically reduces the time spent on administrative aspects of course management and thus gives the graders more time to improve the direct student-teacher relationship. Moreover the availability of tested and reviewed problems and content allows the teacher to easily improve the course presentation.

"I have found this system and the problems themselves to be excellent in conception and design." Students praise the system for its easy to use interface and the ability to work on homework assignments at their convenience. Additionally they like the instant feedback and instant gratification on submitted answers and well tested problems that usually have gone through several independent deployments and review cycles.

CONCLUSION

Using the WebCOSY system for large-scale remote instruction in Beam Physics has proven very successful. It has simplified the management of large student numbers at many different locations and repeated classes, and the system is now used on a day to day basis for current and future courses of the Beam Physics curriculum at Michigan State University. It also has been successfully deployed at the US Particle Accelerator School (USPAS, 2001).

Tight inclusion in the Web makes the system very extensible, portable, and powerful. Using HTML and Perl as the native problem description languages guarantees a powerful system and allows for smooth transitions to future standards and multimedia rich content. As it stands now the WebCOSY system is the only online homework system that allows running of user supplied input trough arbitrary programs for automated execution or grading. This feature makes it extremely useful for any kind of setting where a significant part of the homework assignments consists of user submitted source code, as in basic computer science classes. The built in numerical and multiple-choice problems even allow the system to be used in most undergraduate courses. For purposes of demonstration an interactive tour of the system has been provided to the readers of this article (Hoefkens, 2001).

References

Berz, M. (1999a). *Applications of modern map methods in particle beam Physics*. San Diego, CA: Academic Press.

Berz, M. (1999b). *COSY INFINITY version 8 reference manual.* National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, MSUCL-1088 Revised, 1999. [Online]. Available: http://cosy.nscl.msu.edu/

Berz, M., Erdélyi, B., & Hoefkens, J. (1999). Experiences with interactive remote graduate instruction in beam physics. *Journal of Interactive Learning Research*, *10*, 49-58.

Berz, M., & Hoefkens, J. (2001). *The VUBeam homepage*. [Online]. Available: http://vubeam.nscl.msu.edu/

Hoefkens, J., (2001). *WebCOSY Test Page*. [Online]. Available: http://webcosy.nscl.msu.edu/ (Login "demo", password "webcosy").

Kashy, E., Morrissey, D.J., Tsai, Y., & Wolfe, S.L. (1995). *CAPA—A versatile tool for science education.* National Superconducting Cyclotron Laboratory, Michigan State University, East Lansing, MI 48824, MSUCL-971.

Makino, K., & Berz, B. (1999*). COSY INFINITY version 8*, *nuclear instruments and methods A* (427) 1-2, 338-343.

Makino, K., & Berz, B. (1996). *New features in COSY INFINITY*. Paper presented at the 1996 Computational Accelerator Physics Conference, Williamsburg, Virginia.

Raggett, D. (1997). *HTML 3.2 reference specification*. World Wide Web Consortium. [Online]. Available: http://www.w3.org/TR/REC-html32

Raggett, D., Le Hors, A., & Jacobs, I. (1998). *HTML 4.0 specification*. World Wide Web Consortium. [Online]. Available: http://www.w3.org/TR/1998/REC-html40-19980424

USPAS, (2001). *The US particle accelerator school*. [Online]. Available: http://www.indiana.edu/~uspas/

Wall, L., Christiansen, T., & Schwartz, R.L. (1996). *Programming Perl*. Sebastopol, CA: O'Reilly.