

# Developing a Web-based Tutorial System For Physics Education

Andrew G. Baruth  
Department of Physics  
Doane College  
Crete, Nebraska 68333-2496  
[Abaruth@doane.edu](mailto:Abaruth@doane.edu)

## Abstract:

Over the past few years the Internet has given birth to a potential of transforming the educational process. Now information and activities can be accessed anytime and anywhere, taking information out of the formal class setting. Computer assisted instruction allows immediate feedback as well as increased accessibility. For this reason the concept of a textbook has begun to change. We are exploring how to make it easier for instructors to jump into this educational process by helping them with the construction of interactive content. First, we have created a set of reusable pieces that can be used universally. We have created multiple text sections, much like those found in a textbook, and have added different multimedia to enhance their effectiveness. This multimedia can range from graphics to java applications. We have also created a series of computer-graded questions that allow immediate feedback to both the student and the instructor. Next, we have created a method of assembling these pieces into web-based lessons and assignments. For the structural layout of this information we have developed a convenient XML-based coding scheme for representing our information in a database. This XML scheme is used in all categories: questions, text sections, grade sheets, and the lessons themselves. The XML coding allows us to scan through multiple files and folders to find pertinent text sections and questions when producing a lesson. At this time we have added some content to this system in the area of Physics. However, we have developed software that allows instructors to produce their own text sections and questions, with their own multimedia, and assemble them into their own free standing web-based lesson. This feature allows this system to be very dynamic and literally never looks the same twice. At this time we are carrying out a formative evaluation to determine the appeal and user-friendliness of the software. Our ultimate goal is to produce an open source, interactive web lesson creation system for physics courses.

## **Introduction:**

There has been a strong push in recent history to place education more and more into the technological era. Until this time the thought of using our growing technology for education was difficult, because of the rapid pace of hardware and operating system development. Producing curriculum materials for the computer was difficult because of the mismatch between the textbook production cycle and the hardware/software development cycle [1].

Web technology has simplified the task of developing reusable education tools that can be modified quickly depending on the desires of the instructor. For example, a class of Java applets used in physics education called Physlets allows a web page to display an interactive physics simulation that can be easily tailored by an instructor [1]. The instructor can change what is being animated, add graphs, and allow data collection by adding short client side scripts that pass information to the applet itself. The instructor does not have to know or do any Java programming.

With the introduction of Java applets such as Physlets we see a small file that is written in Java and can run within a web page. It takes on the characteristics of an Interactive Physics [2] simulation as well as video analysis. With the development of these applets it makes sense to utilize these applications within some type of electronic document delivered over the World Wide Web. This leads us into the development of online textbooks, web-based tutorials and web-based homework applications. This in turn, brings the dynamic quality of scripting to interactive homework problems.

Running parallel to the development of web technologies in the 1990's we see the creation of open source software [3]. There are several variations of this idea. A well-known version is the GNU license, which allows anyone to use a piece of code without paying a licensing fee so long as any modification is released with the same condition. It is an intriguing idea to carry this open source concept into the curriculum materials area.

Faculty and students at Doane College have been developing content for physics courses, and developing software for using this content with the goal of building an open source interactive textbook [4]. The system is called WebWorks. It utilizes Java applets and other HTML capable features to produce a variable assortment of electronic classroom devices. The system allows instructors to create tutorials, which allow the student to read textbook sections and follow-up with questions that augment and intensify the material. The system also includes an online homework section that allows the instructor to give web-based homework that the students can complete over the Internet. Once completed that homework is automatically scored and immediate feedback for both the student and the instructor is given. The advantage of these electronic applications is the speed of interaction and the dynamic quality of what can be accomplished. Textbooks sections, as well as homework problems, can utilize physlets and images, and they have a completely dynamic structure that can easily be altered by the instructor. The idea of the traditional textbook is being displaced by that of a changeable, interactive, electronic environment.

What inertial forces impede the adoption of these new technologies? Fear of change, particularly among faculty, is one that arises [5]:

As a professor of political science at Northwestern University and an avid exploiter of software- and Web-based learning materials, Jerry Goldman believes there exists fear of and resistance to technology in higher education 'everywhere except with students.'

Another problem that impedes this adoption is simply the lack of appropriate content [6]. The key ingredients for successful adoption will be providing content that can be easily used in the web environment.

The remainder of this paper will describe how we are creating a database of physics content and appropriate software for assembling this content into web-based tutorials and web-based homework assignments. We describe the overall architecture of the system, how content can be created or imported, the XML coding scheme we developed for representing content, some formative assessment of using the system, and some plans for future development.

## **Architecture**

The intention is for the instructor to easily, in a step-by-step, well-informed manner, be able to go through the procedures to produce textbook sections, add questions and produce an entire web-based lesson in little time. Once the lesson has been produced the work is done. All questions will now be graded automatically and all scores will be saved and easily viewed by the instructor. For the students, the job is just as simple and easy. They must simply visit a home page that will link them to the various textbook lessons and assignments. Then they must simply go through them as they would in a normal hardcopy textbook. The only difference is now their figures can be dynamic and the student can interact with the assignment, as well as receive their scores immediately. Also, if the professor desires, they can return to the assignment and revise their answers. Once they finish and submit their answers there are no papers to turn in or homework to lose.

### **Instructor Section Architecture**

The architecture is quite simple to the professor's eye. First, the professor starts at a basic homepage, where he or she can link to various parts of the system. The parts that they may choose to venture down are the homework sections, the tutorial sections or they may choose to add some new textbook sections to the system. To start, within the homework section they can add or change questions dealing with certain topics of their choice and then take these questions and construct an assignment. They can pick and choose what questions they wish to include into the assignment, they can be of nearly any type: essay, multiple choice, short answer or a numerical answer. They can also choose the appropriate responses to these questions within this section. Once the assignment is created it becomes a stand-alone web page where the students can visit under a login and password and complete the assignments. The grades are also posted

in this assignment area. Instructors can enter into the scorebook area and pull up the answers in multiple fashions. They can either look at the scores from a student-by-student viewpoint or a question-by-question viewpoint. They can also view all the scores, which are tallied into a single spreadsheet for easy transferring into an Excel© type grade book. All functions dealing with the actual assignment portion of WebWorks will take place within this section of the software.

Next, within the tutorial section they can pick and choose from a nesting topic tree various text sections that might appeal to them for their tutorial. (Figure 1)



Figure 1: Nesting Topic Tree Design

At this point they can either read the text sections in a pop-up window or add them to their tutorial. Once the text section is added, they will be shown the text sections they chose and their appropriate topic heading types and they will also be prompted to give a title to the tutorial they are creating. At this time, they will be able to choose which course (the installation process asks for a course list) the tutorial is for, which will be used for the students' homepage routing section. They will also declare the objectives for the tutorial at this time. They can add prerequisite text sections to the tutorial if they wish, which will appear as links to the student in the final web page. Once these items have been taken care of the instructor will proceed to the final set-up page, where they will choose the appropriate questions to follow the tutorial. The question database for the tutorial section is shared with the assignment section, however the questions in the tutorial section will not be graded, they are merely to help the student grasp the materials better and intensify the material. The tutorial is then created as a stand-alone web

page; the location is printed out to the instructor. The tutorial page is a framed page. The left frame contains links to the objectives, prerequisite links, text sections, questions and a link to the students' homepage. All of the corresponding information is then contained in the right frame. This will be the page that the students can view over the web and read all of the appropriate text sections and answer questions concerning the information in the text.

Finally, the instructor can also produce a new text section, which will be used in the tutorial section of the system. This function will take information, in a digital format, from the instructor and turn it into a textbook-like section. The instructor will be asked a series of questions, such as: Creator's name, Title, Prerequisite readings. A topic code and name is also selected from an established classification scheme created by the Physics Instructional Resource Association, called the PIRA-X code [7]. The instructor will also add the actual text for the text section at this time. The body of the text section is where the instructor can add all the information in an HTML ready environment. This means that the instructor can add anything to the text section that you could add to any web page. The additions can include, but are not limited to, images, sound files, java applications, tables or dynamic HTML. The instructor can also add any text effects or formatting that is allowed in HTML, like bold, italics, underlined, right or left justified or many others. Once all the information is inserted the instructor will be shown a sample of their text section, they can either continue with their current text section in its current format or they can return to the information entry and update the text section. Upon approval, a text section is created as an XML file and the nesting tree (Figure 1) is updated to include the new text section. The architecture was created to be very facilitating to the instructor and to make the process of creating an assignment or tutorial as elementary as possible.

### **Student Section Architecture**

The architecture for the student is very easy to follow and is very self-explanatory. The student will be able to access the system from a course homepage. They can either choose to do an assignment from this page, or they may choose to do a tutorial. First, if they choose to do an assignment, they will go to the main assignment page, where they can choose the assignment they wish to complete based on a series of questions: such as course name, section number, name, password, assignment type and number. (Figure 2)

**Course:**

**Section:** ☒ S1 ☐ S2 ☐ S3 ☐ S4 ☐ S5 ☐ S6 ☐ S7

**SCode:**

**Password:**

**Assignment Type:**

**Assignment:**

1	2	3	4	5	6	7	8	9	10	11	12	13	14
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
15	16	17	18	19	20	21	22	23	24	25	26	27	28
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
29	30	31	32	33	34	35	36	37	38	39	40	41	42
<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

**Operation:**

Figure 2: Student Assignment Page

They can either choose to view the assignment, which will show them the assignment and the responses previously given, if the student has already completed the assignment, or they can get the assignment. When making the choice to get the assignment, it will take them to the assignment's permanent web page on the server and the student will just go through the questions and hit a completion button when finished. Due to the fact that the assignment is a stand-alone web page, the instructor could also simply have a link that takes the student directly to the page and would not have to go through the routing page. If this is the choice of the professor, the student's ID and password is still prompted in a pop-up window. Upon completion of the assignment, they will then be prompted to an evaluation page, where they will see how well they did. The contents of what they can see are up to the professor. They can either choose to show nothing and simply submit the score, show how many were incorrect, show which problems were incorrect, or even show the correct answers for all incorrect responses. Depending on the choice made by the professor the student will get the feedback and can then choose to return to the assignment and make changes or submit the assignment, which will send the score to the instructor's assignment page.

Next, when a student decides to view a tutorial they are taken to a similar routing page, where they will choose their course and then pick from a list of titles, in that course, the tutorial they want to complete. Due to the fact that nothing is graded in this section, there is no prompt for a name or password. The page itself has previously been described; they will see the title, objectives and prerequisite reading links. They will also see the text sections and their follow-up questions. In the questions section they are able to answer the questions within the electronic document. By pressing a button called "check you answer" an alert box will appear telling them if their response is correct or incorrect. If they are stumped, they can press the button called "correct answer", which will tell them the correct answer in the alert box. The student section was designed to make all work to complete the assignment or tutorial done in the actual reading

of the text and answering of the questions and not in the navigation. So, it is designed to be as elementary as possible.

## **Filing System**

With a multitude of questions and text sections we were required to develop a database to hold all of the information. The file system is probably the most intricate part of the WebWorks system. We developed a method of filing using what we call the PIRA-X system. The PIRA-X system is based on the PIRA (Physics Instructional Resource Association) system, which was developed by the American Association of Physics Teachers and has been implemented nationally [7]. We added the X to signify that we have changed the code system to fit our needs more optimally. In our PIRA-X filing system, there is a four to five digit code that specifies what best describes the question or text section. The first one or two digits specify the general topic, for physics these include: Mechanics, Fluid Mechanics, Oscillations and Waves, Thermodynamics, Electricity and Magnetism, Optics, Modern Physics, Astronomy & Space Physics and Equipment. The next digit, which is actually represented by a letter, is indicative of a subtopic within the general topic. For example, in the field of Mechanics there are subtopics such as Motion in One Dimension, Motion in Multiple Dimensions, Relative Motion, Newton's First Law, Work and Energy, Linear Momentum and many more. The final two digits specify the actual topic number. So, the completed topic code will be 1c10, if it is the topic of Velocity in the general topic of Mechanics and the subtopic of Motion in One Dimension.

To add another dimension to the topic code we have also made the system print the name of the school in the file name, so if the file is transferred to another server it is still distinguishable. The new filename will look like 1c10\_doane. Now, there may be more than one text section or question having to do with the topic, in this case Velocity. So, the final part of the file name ends in a two-digit number, which ascends with the addition of a new file to the topic. The final filename will now look like 1c10\_doane\_01.txt, which specifies the first text section or question in the topic Velocity under the sub-topic heading One-Dimensional Motion and under the general topic heading of Mechanics.

When a question file or text section file is created it will be classified by this category and saved in this manner. In addition, any image files that are associated with a certain question or text section will be named similarly, but instead of a two digit last number it has a three digit last number. So, an image file will have a name similar to 1a10\_doane\_001.jpg, this symbolizes the first image file for a text section or question in the Mechanics general topic heading and Motion in One Dimension subtopic heading.

The filing system itself then utilizes this PIRA-X system to save the text sections and question files into a folder system. The main folder will be titled “questions” or “textSections” and within this folder there will be a folder for each general topic number. So, for physics there are nine file folders named 1 through 9. Inside these folders are all of the .txt files that represent the questions and text sections named by the PIRA-X system. With this system the files can quickly and easily be located and brought up to add to an assignment or tutorial lesson. The system simply looks at the first or first two digits to determine the file folder it can be found in and then

quickly goes to the requested file folder and quickly scans to find topic code, once found it searches to find the full file name from those matching topic codes. In the actual system this is advantageous because the files can also be brought up by their subject matter, so the instructor only needs to see the files that pertain to the subject they are working with. Once they choose the main topic it will show the subtopics and once they choose a subtopic it will only show files in that subtopic. You will find that the entire system's filing method is based upon this idea of a four to five digit code, which is based on the PIRA system.

## **Purpose of Server Scripts**

The WebWorks system runs due to its use of a number of server scripts to do its job. These scripts are all written in Perl, which allows efficiency in this type of application, because it is all text based. The way that the system works is that an anonymous Internet user can gain access to a stand alone, HTML based web page. As they enter information or pick from information they will be directed to the Perl based server scripts. These scripts will take this information and run it through the scripts found on the server and then spit information back out to the user. So, there is nearly a script for each main function, as the user travels from web page to web page. The scripts themselves can be broken up into three major areas, which are the same as the break up of the software: assignment system, tutorial system and text section addition scripts. There are also a number of administrator scripts that help to get the application onto the server and in working order.

## **Assignment System Server Scripts**

These scripts can be found in the MICS Conference proceedings of 2000 [8].

## **Tutorial System Server Scripts**

These scripts are the ones found within the tutorial system and are the ones that will eventually set-up the actual tutorial, based upon the input of the instructor.

### ***readSections.pl***

This script takes the list of text sections chosen within the nesting tree (Figure 1) and prints them out into a web page so the instructor can view them and decide if he/she wants to incorporate them into their tutorial

### ***lessonSetup.pl***

This script takes the list of text sections and saves them into its memory. It then prints out a new HTML based GUI that asks for the Title of the tutorial, the course that the tutorial is for, and the



objectives of the tutorial. As an instructor you can then pick any possible prerequisite readings you would like to have in the tutorial. Upon submission the script saves this new information into its memory and passes the information to the next script.

### ***questions.pl***

This script takes the PIRA-X topic codes from the chosen text sections and searches through the question file databases to find all matching topics' questions. It then prints out the questions into an HTML based GUI that asks the user to choose the questions they would like to add to the tutorial. Upon submission it takes all information passed to this script and then passes it, along with the question information, to the final script.

### ***createLesson.pl***

This script simply does what it says. It takes all of the information that has been saved and passed along and with that information it compiles a framed tutorial page, preciously described, and saves it onto the server as a stand-alone web page. It then updates the student routing pages so that the student can find the page within the routing system quickly and easily when needed. It then, in an HTML based GUI, prints out the location of the new page and then provides a link to that page.

## **Text Section Addition Server Scripts**

These scripts are also found in the tutorial system, but they are all used for the making of new text sections and uploading these text sections into the database so that they might be used in the tutorial system.

### ***imageUpload.pl***

This script allows the instructor to upload any images that might be needed in the text section onto the server and saves them into a known location so that the system can automatically link to the image from the tutorial. The user can show it which image file on their local hard drive they want to use and it will upload the picture from the local hard drive to the server. Once it uploads the picture it shows, in an HTML based GUI, the picture that has been saved under the name given, to verify that you uploaded the right picture. It also prints out the location of the image on the server.

### ***lessonMaker.pl***

This script takes all of the information from the new text section, which was entered in at newTextSection.htm, and relays it back to the user to verify that it is what they want. At this point they will see the Title, Author, Creation Date, Topic, Topic Code, Prerequisites, and

filename. You will also see the text of the text section in its HTML format, including all images, applets, or other HTML features. If the user wishes to make changes, they may still do so by returning to the previous page. If the user likes the text section as it stands, they can submit it to the next script.

### *goodCheck.pl*

This script does a lot of behind the scenes work. First, it takes all of the information taken from newTextSection.htm and assembles it into the proper XML document including all necessary tags, which you will see more of in the next section. Once it saves the information into an XML document it then recreates the nesting tree system (Figure 1) to include the new text section in its proper place. So, now when the user goes to the nesting tree system (Figure 1) they will be able to choose their new text section to add into their tutorial.

## XML Coding of Data

XML has become a widely used language due to its versatility and ease of use for storing data. The WebWorks system also uses this language to store all of the information on the system. XML is so easy to use because it allows you to develop your own tags for the information and make up some of your own rules for the language as long as it stays within a few guidelines. All of the pieces of information are stored in this manner for later recognition by the Perl server scripts. The text section is the one directly dealt with in the tutorial side of the system. First, the tags that are used within this section include (Figure 3):

<textBookSection>
<topicCode>
<title>
<creationDate>
<version>
<text>
<prerequisite>
<author>

Figure 3: Tags for text sections

The <textBookSection> tag is the outermost tag and specifies the type of file you might be looking at. It serves very similarly to the <HTML> tag that you will find at the head of an HTML based document. The remaining tags fall within this tag and are a subset of the <textBookSection> tag. The first one in their order is the <topicCode>. The <topicCode> tag contains the PIRA-X code, discussed previously, and the topic name associated with the PIRA-X code. For example, if the PIRA-X code was 1c20, the topic name associated with that code is Uniform Acceleration. Next, the <title> tag simply contains the title to the actual text section; this will be printed out anytime a search for the text sections is made. An example of this is in the nesting tree system (Figure 1). Next, the <creationDate> tag contains the date the text section was made. This date is written in the form year, month, day, which appears as 20020226

for February 26, 2002. Next, the <version> tag contains a number that corresponds to the version of that text section, if the text section is modified the number increases by one unit for each modification. Next, the <text> tag is one of the most extensive and important tags. It contains XML CDATA, which includes all of the HTML based text and HTML based tags. So, images are tagged as if they were in HTML and the Java applet functions can be placed within this section. Any parser of XML will not look at anything within the CDATA tag, so the Perl script ignores this section and the tags within, it just considers the text inside one complete unit. This is beneficial because not all HTML tags are legal in XML, so this avoids any problems of interference between the two languages. The next tag is the <prerequisite> tag; this tag holds the file names for any prerequisite readings that might be associated with the current text section. This way the tutorial can display links to prerequisite pages by viewing this tag. Finally, the <author> tag has two inputs, which are the institution name and the author name. The final text section XML document takes the following form (Figure 4):

```
<textBookSection>
  <topic code="1c20" name="Uniform Acceleration"/>
  <title>Linear Motion</title>
  <creationDate>20020226</creationDate>
  <version>1</version>
  <text><![CDATA[Sample Text dealing with Linear Motion]]></text>
  <prerequisite>1c20_doane_01.txt</prerequisite>
  <author institution="Doane College" name="Andrew Baruth"/>
</textBookSection>
```

Figure 4: Sample XML Document

The XML document is quite versatile and is used throughout the program in other places, such as the question files and the assignment files. These XML document setups can be found in the MICS Conference proceedings of 2000 [8].

## Formative Assessment of Tutorial Creation

Although we are not quite yet prepared to consider the project to be in its final stage, we do feel that it is at the point where extensive testing and observation is in order. We decided that it might be beneficial to introduce the system to some instructors and students and hopefully get some of their reactions based upon using the system.

## Instructor Initial Reactions and Comments or Improvements

After speaking to some instructors and allowing them to view and manipulate the system, we got some beneficial ideas for improvement on the system as well as some of the things that were quite beneficial. It was a general consensus that the system itself was a good idea and showed valuable potential for educators. They enjoyed the fact that the user did not need to know the languages it uses to run the system on the server side. The GUI (Graphical User Interface) is pretty simple, yet it has a professional look and conveys the needed information rather

effectively. The instructors found that for the most part, the system was easy to navigate and little outside instruction was needed to get them from point A to point B within the system. It was felt that feedback was essential as much as possible, and to exploit all the feedback possibilities. Instructors tend to like feedback because they want to know that they have done the right thing without question. With this in mind, most felt that an increase in feedback was still essential, this includes button pressing feedback, link jumping feedback and generally more of an indication as to what happens when their screen changes.

Upon discussion with some professors a new startling discovery was found that added a seemingly major flaw to the system, not in programming but in concept. Although the concept of the online textbook is intended to revolutionize the textbook idea, there are some aspects of the textbook that many instructors want to retain in the system. The major idea that they wish to keep in all types of textbooks is the concept of section numbers and with that comes equation numbers and figure numbers. Due to the dynamic quality of the system it is possible to place many different text sections, some of which have no apparent correlation, into the same tutorial document. For this reason, it is difficult to come up with a scheme that could indicate the section numbers as we see in a traditional textbook, which also makes it hard for referencing within the document. This problem is much easier to overlook though than the problem of equation numbers and figure numbers. With the current system if you have three equations in each text section and you have three text sections within a tutorial, you have the potential for three equation number ones and so on. So, the question comes down to whether we actually need the equation and figure numbers in the document, and if so, how can we number them cleverly enough so that they will all be distinct within a single page. This proves to be one of the biggest issues that we have to deal with currently and we need to make some decisions on the appearance of the system, especially in the area of numerical headings.

### **Student Initial Reactions and Comments or Improvements**

Upon showing some students the system, as they would see it in a classroom, I received some very positive feedback. Many students asked when the system would be in place on our Doane College server so they could have it incorporated into their introductory and upper-level physics courses. Most students felt that the information within the tutorial was all necessary and that the GUI of the tutorial was effective and easy to navigate. They like the fact that they can get to any part of the tutorial from the navigation bar in the left hand frame of the tutorial. They thought that it was important to have the objectives section as well as the prerequisite page links. They felt that the text section format was effective for reading, even on a computer screen, which is sometimes tedious. The part that they enjoyed the most though was the final section of questions. They thought that it was very effective to have un-graded questions at the end of the tutorial to intensify the material. They liked how they could get to a question and if they couldn't remember some information they could simply hit the text section link in the left frame and go to the text section they wanted to see and then by hitting the questions link in the left frame they could go right back to the question section. It was effective to have the two choices for the output on the questions. They could choose whether to see if they answered the answer correctly or to see the actual correct answer. This way they could work towards the answer and know immediately if their thinking was right on or not. If their thinking wasn't on, they could

keep trying until they used the correct logic and came up with the right response. If they were completely stumped, then they could simply press a button and get the correct answer. The system itself took very little instruction and was pretty much self-explanatory, which they believed was important. As a student they just want to do the work and not have to get caught up in computer issues. They felt this system was much like using a traditional textbook, but with some added bonus features. Most students had very positive things to say and little ideas for improvement, which I will take as a compliment.

## **Future Plans for Further Development**

After taking some time in discussion with the instructors and students, it is time to take a look at what the future of WebWorks holds. One serious step we must take is to do extensive testing on multiple platforms. The WebWorks system was designed to not only work on Windows based servers, but it was designed to also work on Macintosh and Unix based servers. We have, up to now, done some extensive testing on Windows based machines, but have not taken any time to test in any other platform. On a similar note, we would also like to see how the pages are viewed in different browsers. We have done lots of testing in Microsoft Explorer, but little testing has been completed in Explorer for Macs or Netscape browsers.

We plan to follow-up on some suggestions from the instructors in the area of section numbers, figure numbers and equation numbers also. We think that it might be most effective to take the information found in the text tag, which is now unparsed in Perl and contains all HTML tags, and break it apart into smaller XML tags that can be manipulated within the tutorial. This way we might be able to arrange figure and equation numbers to add from 1 to a max value throughout the entire tutorial and not just within a single text section. This way each figure and equation will have a unique value associated with it. We also would like to develop a unique way to produce chapter and section numbering possibilities. However, this problem is more in concept than practice and might take some intuitive development.

We would like to develop a new way to database the text sections, questions, etc... Right now it is just a flat filing system consisting of file folders name cleverly so that they are easy to find. We would like to use some database software that might make our system a little cleaner and more consistent. We also hope that a new database system will cause a faster program. Little research has been done in our department dealing with packaged database, so extensive research is required if we want to employ such a database into our system.

We hope to get the current system employed into the Doane physics department curriculum, in at least a part, as soon as possible. It seems that many students see a use for such a system in and outside the classroom to augment the learning within the classroom. So, we are currently working on adding in text sections in various physics topics and possible questions to follow-up the text sections. With this continuing effort we hope to have a developed system that works on the Doane campus and can be used and widely tested. Upon these tests and use at Doane we hope to make the system's package available online to whomever might like to incorporate it into their classroom at no cost to them, keeping this mindset of the open source software movement.

## References

1. Christian, W. & Belloni, M. (2001). *Physlets: Teaching Physics with Interactive Curricular Material*. New Jersey: Prentice Hall, Inc.
2. InteractivePhysics.com (2002). Retrieved February 28, 2002, from <http://www.interactivephysics.com/>
3. Raymond, E. S. (2001). *The Cathedral & the Bazaar*. Cambridge, MA: O'Reilly & Associates.
4. Wentworth, C.D., Plano Clark, M. (1999). Developing A Public-Domain Question Database for Physics Courses. In *Small College Computing Symposium 1999 Proceedings*. La Crosse: University of Wisconsin - La Crosse.
5. Bollentin, W. (1998). Can Information Technology Improve Education? *Educom Review*, 33 (1), p. 50-52.
6. Simon, E. (2001, September). Are e-books Ready for the Classroom. *Syllabus*, 15 (2), p.28.
7. PIRA Demonstration Bibliography (1998). Retrieved February 28, 2002, from <http://www.physics.ncsu.edu/pira/>
8. Hays, J., Wentworth, C.D. (2000). Improving A Web-based Homework Assignment System for Physics Courses. In *Midwest Instruction and Computing Symposium 2000 Proceedings*. St. Paul: University of Minnesota St. Paul.