WebTA: Software for creating Web-based Problem Solving Tutorial in Engineering Courses

Stefan Cular, Venkat Bhethanabotla and Babu Joseph[1]
Chemical Engineering Department, University of South Florida, Tampa, Florida

Last revised on: October 4, 2004

Abstract

In this article we discuss the development, testing and evaluation of a *hybrid* learning tool (WebTA) which is a combination of both web-based problem solving tutorial and web-based assessment of learning. The problems are presented in tutorial fashion by breaking up the design-application exercise into smaller step-by-step questions each of which must be answered before proceeding to the subsequent step. On-line help files are provided to guide the students who get stuck at any particular step towards the solution. Answers are checked using a web-interface to engineering software tools such as Matlab or numerical routines written in FORTRAN or C. Immediate feedback regarding the correctness of the answer is provided to the student. A preliminary evaluation indicates that such exercises can indeed help facilitate the learning process in engineering courses.

[1] Author to whom correspondence should be addressed. joseph@eng.usf.edu

Introduction

Traditionally, problem based learning is integrated into engineering education through homework exercises. Homework exercises on relevant design-oriented problems are often simplified so that the answer can be obtained using pencil, paper and calculator. New teaching tools (computer-based instruction systems, intelligent tutoring modules, virtual laboratories, web-based distance learning etc.) have emerged with the availability of computers and the internet. Computer-based instruction systems are developed as a supplement and sometimes as a replacement for traditional lecturing and textbook based learning. Intelligent tutoring systems are designed for students with various knowledge levels. Virtual laboratories are aimed at supplementing physical laboratories [3-6]. In this study, we focus on the use of web-based exercises as a substitute for traditional homework exercises.

One of the most often received requests we get from students in engineering classes is for more problem solving examples in class where there are opportunities to interact with the instructor on specific difficulties encountered while attempting a solution. Unfortunately, the fixed contact time available limits the number of problems that can be discussed in a classroom setting. Working out problems on the blackboard is useful in explaining concepts, but the passive nature of this activity reduces its value as a teaching tool. If we want to actively engage the students in problem solving in class, then classes must be limited in size to achieve good interaction between the instructor and the students. Often we assign a number of exercises to be done outside the class room. In our experience there are a number of roadblocks that hinder learning via this process:

(i) Students often get stuck while solving the problem and unless a TA or Instructor is readily available, it becomes difficult to proceed further.

(ii) We may encourage our students to work in teams, and this works well for some. It can also lead to passivity for some students who might become too dependent on others in the group, rather than thinking through the problem himself or herself.

(iii) Students often have no idea whether they did the problem correctly until it is graded (if at all) and given back. Feedback often comes too late, by which time new topics have taken the attention of the student and the instructor.

(iv) The instructor cannot put too much weight on the homework for assigning course grades if cooperative effort is encouraged since in this case the homework performance may not reflect individual understanding of the subject. Reducing the weight often leads to students not bothering to turn in their work due to time pressures. As a result many students fall behind in learning the concepts.

(v) Good problems are hard to create. So we often reuse problems from semester to semester and in that case students create back files to reduce their workload. This further reduces the value of the exercises.

In this article we discuss the development, testing and evaluation of a *hybrid* approach that combines the web-based assessment approach with a web-based tutorial for learning engineering problem solving. The *assessment component* provides motivation for the student while the *tutorial nature* of the problem solving assignment makes learning interactive with immediate feedback on their own understanding of the subject. The first part of the paper outlines the design considerations of such a tool. We then discuss how such a tool was developed using commonly available tools such as JavaScript and PERL. An example of problem created using such a tool is presented next. The paper concludes with a discussion of the results obtained after employing web-based problem solving tutorials in a course on process control for undergraduate students in chemical engineering.

Instructional Design Considerations

Design considerations from the viewpoint of the student and instructor are discussed separately below. From a student's perspective, we felt the following considerations were important:

1. It should be readily accessible. The web provided an obvious mechanism.
2. Each tutorial exercise should be of reasonable length so that it can be completed in about an hour or less.
3. It should provide immediate feedback so that the student will know if they did the problem correctly.
4. There should be sufficient help available on-line that will allow them to proceed if they get stuck at any step of the problem. Ideally help files should link to lecture notes and or slides if applicable.
5. There should be some incentive in the form credit towards their grade if they worked on a problem (even if they did not get the correct answer). There should be no penalty for multiple attempts at solving the problem.
6. Links to appropriate web-pages such as lecture notes and lecture slides should be available.

Design considerations from the point of view of the instructor:

1. It should be possible to design and deploy a problem set without having to learn the details of how to design interactive web-pages. A menu-driven approach is preferable to ease the burden of creating these exercises.
2. There should be an automatic mechanism to keep score and produce a spreadsheet containing the grades.

3. Developing the problem should not be much more difficult than creating a Word document.
4. It should be possible to use random variables in the problem so that each student will get a unique problem.
5. Interface to software tools such as Matlab should be provided to check answers to complex engineering problems.
6. Last, but not least, we felt that the tools for developing and deployment should be based on readily available software so that easy dissemination of both the tools and the exercises created using these tools is possible.

We were able to meet most of these criteria by using a combination of software tools consisting of a web-page development tool such as Mozilla or FrontPage, JavaScript and PERL. Our objective was to provide sufficient tools for the instructors who have some minimal knowledge of how to create and deploy course web pages. The result of about six months of development by a graduate student was WebTA, which provided the essential elements of what we were seeking. These tools are available at: http://www.eng.usf.edu/~cular/interactive/development .

The ability to use random numbers generated in the problem statement, allows the instructor to vary the problem at least numerically from student to student. The ability to access Matlab, C and FORTRAN programs directly from web applications allows us to deploy problems that require complex calculations to arrive at an answer. The latter is a necessity with engineering problems because many of them require solution of nonlinear and/or differential equations. Since it is necessary to provide immediate feedback to the student on the correctness of their answer, solutions to the problem must be generated on the fly as they are created. The Matlab, C or FORTRAN interface provides this capability. However, it also makes it slightly more involved to generate the web-based problem solving exercise. We have stream-lined this significantly by providing templates for the instructor.

Features of a Problem Solving Tutorial Exercise

Figure 1 shows the screenshot seen by the student upon beginning a problem solving exercise. On this page, students are directed to read the problem statement. A mouse click brings forth a set of random numbers that are associated with the problem. The instructor can decide which parameters to randomize and the range of these random numbers. From this point the student can then work through the problem set by answering the questions provided in the bottom frame one by one. The random values used remain constant until the "GET VALUES" button is depressed again.

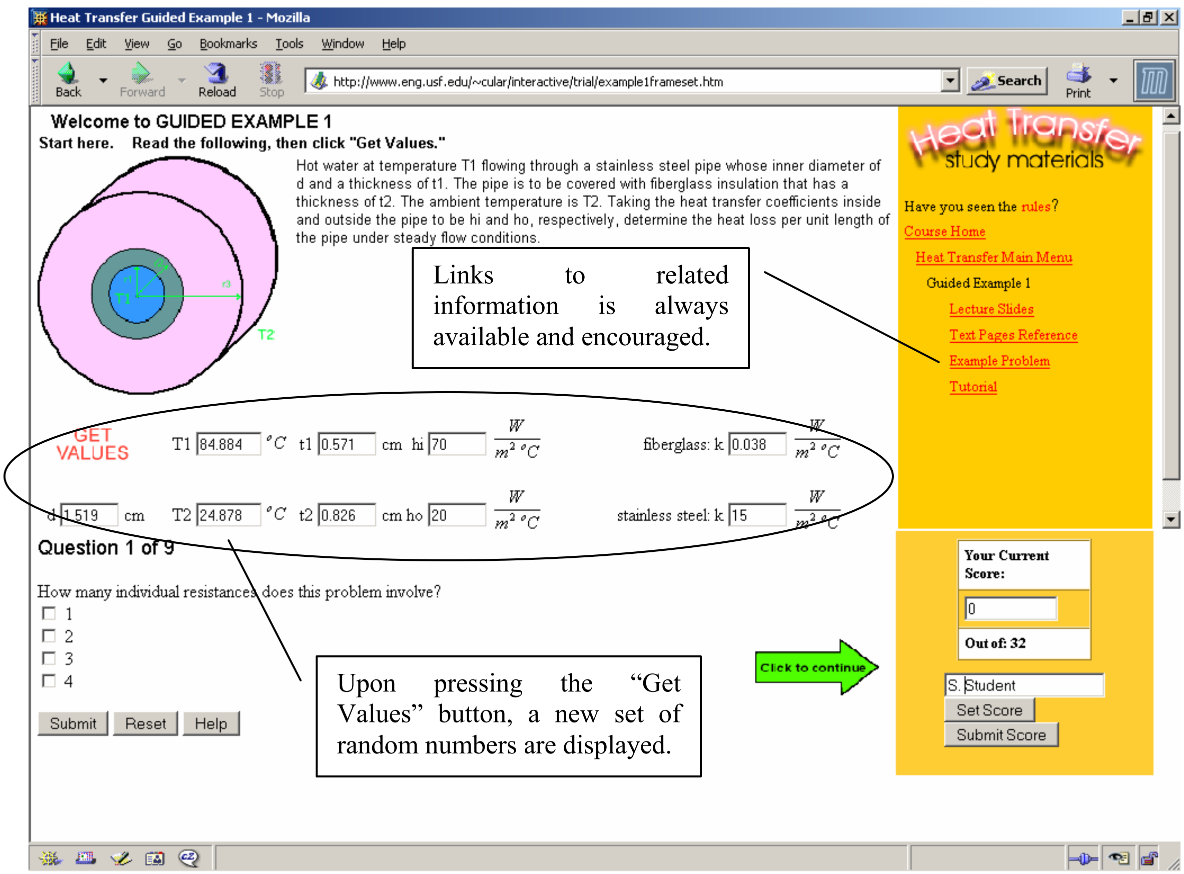

Figure 1. Screenshot of the first page seen by the student on opening a web page for the problem set.

Throughout the problem set, the student has the option of submitting the score to a central database. Questions are typically arranged in a hierarchical fashion with each new question building upon the results from previous questions leading to a solution to the problem posed in the beginning. Figure 2 shows the last question in the heat transfer problem. Here the student is asked for the steady state heat loss. In order to calculate this value, answers to the previous eight questions are necessary. Here in lies the tutorial nature of this assignment. The student is led step by step towards the solution.

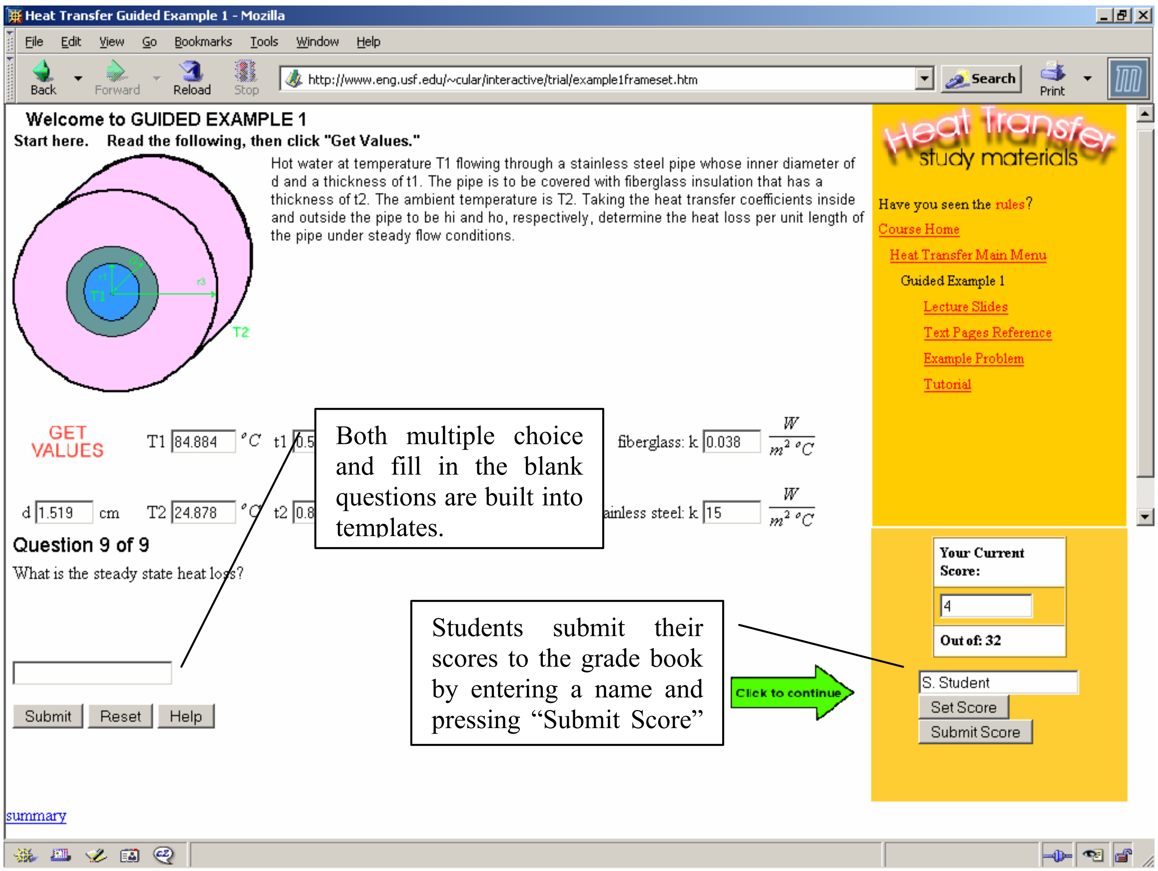

Figure 2. Last question on the heat transfer problem set.

The instructor may provide links to lecture notes and or lecture slides that are relevant to the problem at hand. In addition links to text page references are also suggested so that the student may prepare to tackle the problem. The box at the bottom right shows the current score for the student. While this may seem trivial, this assessment component is an important motivating factor in promoting student participation. When a student hits a submit button the system compares the answer with the correct answer and provides immediate feedback to the student so the he or she may attempt to find the correct answer if their first attempt failed.

After discussions with an educational psychologist (Eison, 2003) we decided on the following simple scoring strategy:

1. A student is given three chances to get the correct answer. If the correct answer is obtained the score is 4 points. If the student fails even after 3 tries, then a score of 1 is given and the student has to move on to the next question.
2. A student may go back and restart the problem from the beginning and get another score for the same problem set. The best score from all attempts is taken as the final score for the student.

A student may use help at any time. The first time the help button is pressed the system displays a hint provided by the instructor. The second time, it provides more detailed suggestions to arrive at the correct answer.

We also created a template for use by instructors for creating problem sets. Figure 3 shows part of the instructor's template for creating a multiple choice question. This will make it easy for the instructor to convert an existing problem set into a web based tutorial.

Class room assessment of Web-based Problem Solving Tutorials

We conducted a preliminary evaluation of the utility of these web-based problem solving tutorials in one of the required chemical engineering courses: Introduction to Automatic Control. In the entire class, 27 students increased their score as a result of doing web-based exercises whereas 16 decreased their score. The average changes in the scores of different groups of students are shown in Table I.

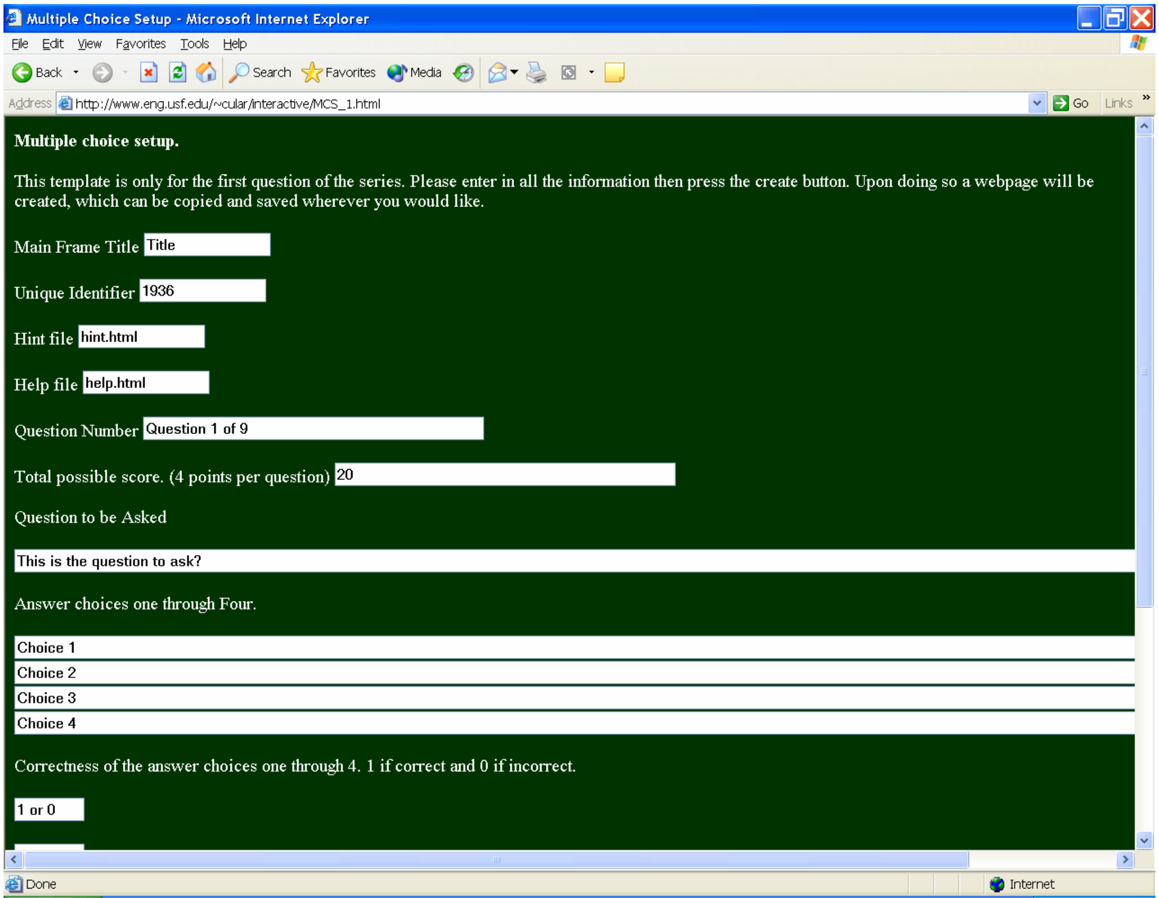

Figure 3. Screen shot of an Instructor's template for creating a multiple choice question.

Table I. Comparison of Scores on Paper vs. Web-based Exercises.

	Paper Exercises	Web Exercises	Change in Score
Top 25% of class	81.6	90.3	+9.3
Mid 50%	59.1	58.2	-.9
Bottom 25%	33.6	41.3	+8.3

This seems to imply that the primary benefit of the web-based exercises is for the top 25% and the bottom 25%.

The results of the survey indicate a positive bias for the web-based exercises. The features that the students appreciated the most were: immediate feedback, and the scoring system which allowed multiple attempts without penalty. The technology seems to be stable enough that the students experienced little or no difficulty in accessing and solving these web-based exercises.

Conclusions

Our experience with providing web-based problem solving tutorials indicates that it provides a convenient way to promote learning while assessing. The problems are a little more difficult to create and publish but once done, they can be reused. The advantages of web-based tutorials include: immediate feedback to the student, access to help on-line, links to most relevant background reading material, automated scoring and randomized numbers which make problems unique to every student. Grading time is reduced thus allowing the instructor and TA to focus on more direct interaction with the student. Access to on-line assessment allows the instructor to pinpoint areas of concern so that he or she can focus on these topics immediately to resolve student misunderstandings and misconceptions.

The main disadvantage is that more time is needed to prepare the problem set and the help files. Since this is a one-time investment, the burden will be eased if the course is taught repeatedly and if there are large number of students in the class. Also if it is a course that is taught widely, then it may be possible to share the burden of problem generation among the peers teaching the same topics.

Our initial experience in using this approach in the classroom has been quite positive. Hence we are moving ahead with plans to develop more of these hybrid tutorial exercises in our core classes.

Acknowledgment
Partial support was provided through a grant from NSF, Grant No. 0230476

References

1. W. R. Crozier, Individual Learners: Personality Differences in Education, Taylor & Francis e-Library ed., New York, 2002.
2. R. Delisle, How to Use Problem-Based Learning in the Classroom, Alexandria, 1997.
3. Y. Li, E. I. LeBoeuf, P. K. Basu, L.H. Turner, Development of a Web-Based Mass Transfer Processes Laboratory: System Development and Implementation, Computer Applications in Engineering Education, 11 (2003) 25-39.
4. D. Shin, E. S. Yoon, S. J. Park, E. S. Lee, Web-Based Interactive Virtual Laboratory System for Unit Operations and Process Systems Engineering Education, Computers and Chemical Engineering, 24 (2000) 1381-1385.
5. S. E. Yarbrough, R. B. Gilbert, Development, Implementation, and Preliminary Assessment of Virtual Laboratory, Journal of Professional Issues in Engineering Education and Practice, 125 (1999) 147-151.
6. K. D. Forbus, P. B. Whalley, J. O. Everett, L. Ureel, M. Brokowski, J. Baher, S. E. Kuehne, CyclePad: An Articulate Virtual Laboratory for Engineering Thermodynamics, Artificial Intelligence, 114 (1999) 297-347.
7. P. J. Fleming, M. E. Paulaitis, A Virtual Unit Operations Laboratory, Chemical Engineering Education, 36 (2002) 166-169.