

Validation of the Learning Machine

Jo-Mae B. Maris

Northern Arizona University, Flagstaff, AZ, USA

jo-mae.maris@nau.edu

Abstract

As Web-based courses become more prevalent, tools need to be created that go beyond electronic page turning. The tools should allow for easy development of Web-based interactive instruction. The Learning Machine is data-driven tutorial software that is based on behavioral education philosophy. Development and presentation use the same database, but separate scripts, so that changes to content do not require changes to the presentation script. This decoupling enables content providers to concentrate on course development.

This paper validates the effectiveness of Learning Machine tutorials as compared with classroom lectures. The experiment conducted to validate the Learning Machine tutorials showed that the tutorials were at least as good as classroom lectures.

Keywords: Web, instruction, tutorial, programmed-learning, experiment

Introduction

Web-based instruction has become the primary delivery mechanism for many courses. A state university offered over two hundred regular session courses over the Web during spring semester of 2005. However many of the delivery tools for Web-based courses are course management tools (WebCT, Vista, Blackboard, eCollege) rather than learning tools. Course management tools provide grade books, course home pages, quizzes, assignment drop boxes, and containers or modules to hold links to instructor provided content. The instructor provided content tends to be slide shows, readings, or multimedia presentations taken from face-to-face courses. The result is a course with limited interaction. Bett (200) refers to these course components as "buttonology." Bett states, "while the buttonology course is sufficient for about 25% of the faculty to add a few Web enhancements to their lectures. It is not a complete foundation for building a high quality on line course" (Bett, 2000).

Going beyond the passive syllabus, class notes, and readings to provide interactive instruction requires expertise in programming or computer-aided-instruction tools that may not translate to Web delivery. Developing computer-aided-instruction can prove to be time consuming and expensive:

"The tutorial, which began on paper as an interactive, feedback-providing brainchild was quickly developing into an electronic page turning exercise. The instructor had good ideas

Material published as part of these proceedings, either on-line or in print, is copyrighted by Informing Science. Permission to make digital or paper copy of part or all of these works for personal or classroom use is granted without fee provided that the copies are not made or distributed for profit or commercial advantage AND that copies 1) bear this notice in full and 2) give the full citation on the first page. It is permissible to abstract these works so long as credit is given. To copy in all other cases or to republish or to post on a server or to redistribute to lists requires specific permission from the publisher at Publisher@InformingScience.org

and nice plans on paper, but she could not implement them with the technological constraints she was operating under" (Harbeck, Schweizer, & Pienkowski, 1998)

"OCBI accepted a contract to produce fifty lessons within one year as part of a computer-based college course. The

size of the project staff grew to nearly sixty people during the year including six content experts, four CAI designers, eight part-time scriptors, six full-time programmers, and over thirty part-time student programmers. Coordination of the programmers and scriptors became a challenging task within the project." (Reed & Smith, 1983).

"Second, research on the cost-effectiveness of CAI shows that while there may be some gains to academic achievement, these are not proportionate to the costs of buying and maintaining computers; tutoring, for instance, was found to produce greater gains for less money" (Cooper, 1998)

If the instructor chooses to develop an interactive Web page from scratch, the page must be programmed as an Active Server Page (ASP), Java Server Page (jsp), or Common Gateway Interface (cgi) using PERL or php. Although Kanten-McCoy and Flanagan content "with today's commercial interests ain the internet and web site development, programmer expertise is no longer necessary for the relative quick and easy development of web pages" (1998). The approach that Kanten-McCoy and Flanagan are proposing requires each page to be written and linked individual. Additionally, the student does not get immediate feedback from the tutorial.

Now, consider implementing one interactive page using Microsoft's ASP. ASP requires that the developer know Visual Basic, Java, or C# in addition to knowing Hyper Text Markup Language (html). An example of a simple ASP page is shown in Figure 1.

```
<%@ Language="VBScript" %>
<% Option Explicit %>
<% 'Page to get user's name %>
<html>
  <head>
    <title>Sample ASP Page</title>
  </head>

  <body lang="en-us">
    <h1>Sample ASP Page</h1>
    <p>This page demonstrates data entry and response.</p>
    <% 'If userName is null, then display form;
      'otherwise greet user by name.
      Dim userName 'Variable to store user's name
      userName = Request.Form("txtUName")

      If userName = "" Then
        'Display form on this page (demo.asp).
      %>
    <form name="frmDemo" method="post" action="demo.asp">
      <p align="center">Please, enter your name:&nbsp;
        <input name="txtUName" type="text" width="30"></input><br>
        <input name="cmdSubmit" type="submit" value="Submit">
        </input>
      </p>
    </form>
    <% Else 'Greet user %>
    <p>Hello, <%=userName %>.&nbsp; Thank-you for helping
      with this demonstration.</p>
    <% End If %>
  </body>
</html>
```

Figure 1: Simple example of ASP page

The ASP shown in Figure 1 does not use any previously stored data or retain any data provided by the user. To retrieve or store data usually involves accessing a database. Once a database is introduced into the processing, the developer would need to know a database query language, such as SQL. Each additional element introduced into the development of an interactive Web page makes the task more difficult. If the content provider relies on what-you-see-is-what-you-get (WYSIWYG) Web page development, then a page similar to the simple example in Figure 1 is not possible. The goal of the Learning Machine was to provide teachers of Web-based tool for developing interactive content without having to learn programming.

Learning Machine

The Learning Machine had its beginnings in Ford Foundation sponsored education research of 1960. The author was a subject in one of the research projects in Colorado. The project used plywood boxes. Each box contained a paper scroll. The boxes and their mechanisms fascinated the pupils in the study. Reading the text and answering the questions was secondary to the box with its primitive and failure prone spindles. The progression from answer to the next text was an adventure. Would the scroll tear? Would the spindles break or come out of their sockets? Would you get to open the box and fix the mechanism? However, the idea of getting to skip material one already knew stayed with the author.

The Learning Machine project involved developing a Web-based learning tool. The tool is based on the behaviorism philosophy of learning. One of the best known researchers in the area of behaviorism was B.F. Skinner. The learning machines developed by Skinner and advanced by many others (Pettijohn, 2004) operate on the principle of rewards for demonstration of comprehension of material presented.

Educational Philosophy

The Learning Machine is based on behaviorism. Behaviorism was chosen from the five prominent educational philosophies (Shaw, 2001) because it gives guidance in the delivery of curriculum as opposed to the content of curriculum. Behaviorism reasons that we learn through “response to internally or externally generated physical stimuli” (Shaw, 2001).

The behaviorism philosophy was applied by Skinner and others in the development of programmed-learning. In programmed-learning, the student is presented a concept and then asked about the concept. If the student answers the question correctly, the student is rewarded by moving forward to the next concept. If the student answers the question incorrectly, the student is presented remedial material and asked the question again. This approach to content delivery was introduced to the author as a Skinner box. The Learning Machine is an automated Skinner box.

Technical Overview

The Learning Machine is data-driven. All of the content and navigation for a tutorial and its slide show are stored in a database. The database also stores student progress information. The database used by the Learning Machine is given in a link to the Learning Machine script. Thus, each tutorial has its own database. Using a separate database for each tutorial and associating the database in a hyperlink to the script isolates the Learning Machine from the content of its tutorials. Therefore, changing the content within a tutorial or changing tutorials requires no changes to the Learning Machine’s code. The goal of decoupling of content and delivery script is to reduce production time for tutorials.

Content is developed using a content editor script. The content editor manages all of the relationships between concepts, questions, and remediation for a tutorial within its database. The content provider enters text, image references, and text formatting through a Web form. The Web form,

shown in Figure 2, has areas for entering each concept's title, descriptive text, follow-up question with multiple-choice responses, and remediation for each incorrect response. The format of one question per concept helps content provider limit the scope of each screen to one succinct topic.

Purpose

The goal of this research project was to validate the Learning Machine's tutorial presentation. Although the Learning Machine is based on proven educational philosophy, its implementation had not been proven. For the Learning Machine to be trusted content delivery mechanism, its implementation of programmed-learning needed to be tested. The experiment compared the performance of students using the Learning Machine with those receiving traditional lecture. The Learning Machine includes a slide show feature that allowed both groups to view the same written material.

Figure 2: Example of content editor page

Problem

The Learning Machine will be considered valid if the students using it have a mean performance is not significantly worse than traditional on-site students' performance. One would like to say that the Learning Machine would be considered valid if the students using the learning machine did the same as the traditional on-site students. However, the null hypothesis must include the equality. Thus, this study is in the awkward position of using the null hypothesis rather than the alternate as the answer to the research question. The hypotheses are:

$$H_0: \mu(\text{Web}) = \mu(\text{lecture})$$

$$H_A: \mu(\text{Web}) \neq \mu(\text{lecture})$$

If the Web-based tutorial works exceedingly well, then a stronger statement may be made: students perform better using the Learning Machine than they do attending lecture. Here the question can be addressed by the alternate hypothesis:

$$H_0: \mu(\text{Web}) \leq \mu(\text{lecture})$$

$$H_A: \mu(\text{Web}) > \mu(\text{lecture})$$

Method

The method used to validate the Learning Machine was an experiment. Two networking tutorials were written using the content editor for the Learning Machine. During the Fall 2002 semester, networking course students had the opportunity to participate in the experiment that compared Web-based instruction to classroom lecture. The following sections present the experiment's plan. Both the Web-base instruction and the traditional on-site lecture were presented the same visual aids and topic content.

Selection of Participants

The students in the two sections of Networking were asked to volunteer to participate in the experiment. The students were given consent forms with the options to participate or not participate. Students consenting were assigned to Group A or Group B by a random draw.

Split Sessions and Tasks

Two lectures during the semester were designated split session. During the split sessions, students in the experiment either attended lecture or did a Web-based tutorial. The lecture and tutorial covered the same material. Students doing the tutorial were not allowed to attend the lecture. Students attending the lecture did not have access to the tutorial. Two lectures were used so that the students in each group participated in both the lecture method and the Web tutorial method of instruction.

Testing of Results

To determine whether an effective difference in learning existed between the two instruction methods, the students were tested over the split session material. The class session after each split session a quiz over the split session material was given. The session after each split session was not another split session. All students took the same quiz.

Results

Forty-eight students participated in the experiment. Each group contained twenty-four students. Before Quiz 1, Group A had lecture and Group B did the tutorial. Before Quiz 2, Group B had lecture and Group A did the tutorial. Quiz 1 had a significantly ($p = 0.000000000166$) lower mean than Quiz 2 (see Table 1). Standardized scores were used to test the hypothesis, since standardization will remove the difference in the means between the quizzes. The means and standard deviations for each quiz were used to standardize the scores.

Table 1: Means for Raw Quiz Scores

	Quiz 1	Quiz 2
Group A mean	10.5000	15.7
Group B mean	10.5833	14.9
Quiz mean	10.7273	15.5

Table 2: Means for Standardized Scores by Treatment

	Lecture	Web
Group A mean	0.0650626	0.069736
Group B mean	0.1768293	-0.04121
Method's mean	-0.120946	0.014265

The standardized scores were then sorted by treatment and group. Table 2 shows the mean scores for the Web tutorial and the lecture by group. Table 2 includes the mean for each delivery method. The standardized scores ranged from high of 2.0820034 to low of -3.10821.

Analysis

To test the hypothesis, a paired t-test was used. The probability of a greater t-score was computed using the Excel t-test function. The arrays in the t-test function were the standardized “Lecture” quiz scores and the standardized “Web” quiz scores. The two-tailed t-test yielded a p-value of 0.363358151. Based in this p-value the null hypothesis of (Web) = (lecture) cannot be rejected. This failure to reject is not the same as saying that the means are equal.

However, the Web grand mean is greater than the lecture grand mean, so the hypothesis that the Web performance is greater than the lecture performance can be tested. Using a one-tailed, paired t-test, the Excel function yielded a p-value of 0.181679076. This p-value is above the very weak = 0.20 level of significance. Since the null can be rejected at the 0.20 level of significance, there is evidence that the Web tutorials were not worse than the lectures. Since the goal of this experiment is not to prove Web tutorials are better than lectures, the very weak level of significances can be useful in determining that the Web tutorials were at least as good as the lectures.

Conclusions

The experiment showed that the tutorials from the Learning Machine were as beneficial to students as attending the lectures. The experiment did not prove that tutorials from the Learning Machine were superior to lectures.

Discussion

This validation gives Web instructors a tool for engaging students in learning that goes beyond page turning. The tutorials allow teachers to concentrate on content rather than implementation. The tutorials do require time to prepare, but do not require programming knowledge. The preparation time is spent in crafting concise presentation of concepts, pertinent questions, and responsive remediation.

Future Plans

With the tutorial validated, the next plan is to improve the content editor and increase the tracking capability of the tutorial. The content editor needs to have WYSIWYG editing of concepts, questions, and remediation. Also, drag and drop inclusion of images is essential for making the content editor user friendly. Another feature that should be added is the ability to rearrange concepts either through reordering items in a list or rearranging images. The primary plan is to make more use of the Learning Machine.

References

- Bett, S. (2000). Buttonology and beyond. Retrieved December 15, 2004 from <http://victorian.fortunecity.com/vangogh/555/dist-ed/Bb-orientation.html>
- Cooper, C. (1998). Does it compute? The relationship between educational technology and student achievement in mathematics. *Policy Information Report*, POLICY INFORMATION CENTER, Research Division Educational Testing Service Princeton, New Jersey 08541-0001. Retrieved March 11, 2005 from <http://www.iste.org/>
- Harbeck, J., Schweizer, M., & Pienkowski, N. (1998). Developing a computer assisted tutorial: A beginner's experience. *Proceedings of Instructional Technology Conference 1998*. Retrieve March 11, 2005 from <http://www.mtsu.edu/~itconf/proceed98/jharbeck.html>
- Kanten-McCoy, R. & Flanagan, J. (1998). Web-based tutorial training for a regional network of clinical providers. AMIA. Retrieved February 16, 2005 from <http://www.amia.org/pubs/symposia/D003995.pdf>
- Norgard, C. (2002). Institutional plan for distance education and off-campus instruction. University of Houston-Victoria. Retrieved December 15, 2004 from <http://www.thecb.state.tx.us/distanceEd/UHVInstPlan.htm>
- Pettijohn, T. F. (2004). "Programmed Learning," *Psychology: A ConnectText* (4th ed.). McGraw-Hill. Retrieved December 16, 2004 from <http://www.dushkin.com/connecttext/psy/ch06/prolearn.mhtml>
- Reed, M. J. & Smith, L. H. (1983). Developing large CAI packages. *CALICO Journal*, 1 (3). Retrieved March 11, 2005 from http://calico.org/journalarticles.html#Volume_1_Number_1_June_1983
- Shaw, L. J. (2001). Five educational philosophies. Retrieved Fall 2001 from <http://edweb.sdsu.edu/people/LShaw/F95syll/philos/phbehav.html>

Biography



Jo-Mae Maris teaches hardware, operating systems, networking, and management information systems at Northern Arizona University. She was the information technology coordinator for InSITE 2005 Conference.

She will be glad to share any scripts discussed in this paper. The home page for the Learning Machine is <http://www.cba.nau.edu/facprjs/learn>.