Simple But Useful Tools for Interactive WWW Development

Robert C. Maher
Department of Electrical Engineering
University of Nebraska-Lincoln
Lincoln, NE 68588-0511
rmaher@unl.edu

Abstract

An important area for World Wide Web (WWW) development is customized educational web pages incorporating interactive features. Unfortunately, it is usually necessary for an engineering instructor to expend a substantial amount of time to learn the peculiarities of programming and debugging web-based interactive applications. To help eliminate this time hurdle a description is given in this paper of a set of software routines which allow an instructor to prepare a set of quiz questions or drill problems in the form of ordinary text files, then to convert the files automatically into hypertext markup language (HTML) with an interactive "forms" interface for inclusion in a web site. The relatively simple examples considered in this paper are intended to serve as templates for those individuals who are interested in probing more deeply into the intricacies of web software. Examples and computer code are also available.

1. Introduction

New methods for delivering educational materials are appearing with each change in computer technology. It is difficult for engineering educators to keep up with the changes, much less master the latest authoring techniques.

Among the recent additions to the educational technology repertoire is the World Wide Web (WWW), or simply, the *web* [1]. The web allows a user to request text, pictures, sounds, movies, and other types of data to be delivered over a computer network from a *server* computer, and then to display the information on the local computer. If one imagines the server stocked with educational materials it becomes clear that there is great potential for teaching and learning to take place electronically.

In order to provide a smaller hurdle for engineering educators interested in using interactive web features, two programs are described in this paper which allow creation and on-line checking of multiple choice quizzes. The programs allow the instructor to prepare a simple text file containing the questions and multiple choice answers (with the correct answer indicated), and then automatically to generate a "radio button" form and form handler for inclusion in a class web site.

The remainder of this paper is organized as follows. First, a short primer on the web is given. Next, the operation of the quiz generator and response checking programs are described, including the simple format used for the quiz question text file. Finally, a brief discussion of the programs, suggestions for additional features, and some concluding comments are given.

2. Interactive Web Primer

The World Wide Web is a very popular mechanism for distributing information on the internet. The web includes features which allow nearly seamless compatibility with prior internet information services, such as FTP and GOPHER, as well as support for "point and click" features via *hypertext*. Some of the implications of the web environment are considered next.

2.1. Hypertext and Browsers

Hypertext refers to an information system in which specific words or phrases in an electronic document contain links to other related documents. Note that the word "document" in this context can refer to text, graphics, images, sounds, video sequences, or any other type of electronic media. The user can choose to follow a link by selecting the desired word or phrase, typically by clicking with a mouse pointer. The hypertext links provide a simple but powerful means for the user to explore a large database of documents in a convenient and intuitive manner. The web is based on a hypertext interface [1].

Web documents are written in a content description format called the hypertext markup language, or HTML [2]. HTML includes a set of commands to identify different parts of the document, such as section headings, enumerated items (bullets), images, and, of course, hypertext links. The most important feature of HTML is that the hypertext links can refer not only to documents residing on the same computer, but to documents located

anywhere that is accessible on the internet. The mechanism whereby the information is transferred is called the hypertext transfer protocol, or HTTP [3]. Each document accessible on the web has a unique address called the uniform resource locator, or URL.

At the user end of the web is a *browser* program, such as Netscape, running on the user's personal computer, workstation, or mainframe. The browser performs two major functions: retrieval from a *server* of the document selected by the user, and display of the contents of the document. Most browser programs have other features such as electronic mail, news readers, and so forth.

2.2 Interaction with the Server

The hypertext framework of the web as described so far is really little more than a fancy page turning scheme. Fortunately, the Web provides mechanisms which allow the response from the server to depend more intimately on the user's requests.

One popular method for the user to provide more specific information to the server is the *form* interface. An HTML "form" consists of several methods for choosing items from lists of options, selecting checkboxes, and typing in words or phrases from the keyboard. The browser program is responsible for displaying the requested form items and for collecting the user's responses.

The server must also cooperate with the form interface if anything useful is to happen. The server receives a file from the browser program containing the user's responses to each of the form queries, along with a specification of what to do with the information. Typically the response to the form is processed by an executable program on the server through the common gateway interface (CGI) system. The form processing program returns HTML information back to the browser program in reply to the form responses [2].

2.3 Creating a Form-Based Web Application

An educator interested in creating a form-based web application must create the HTML file describing the form and also the form handler program on the server. Creation of the HTML is not inherently difficult, although it does require some experience and experimentation. Creation of the executable form handler is more complicated, requiring knowledge of a suitable high level programming language, a compiler (if necessary), and the cooperation of the "webmaster" or system administrator for the computer that is used as the server. This level of effort can dissuade an instructor from creating interactive web materials, even on an experimental basis.

3. Automatic HTML For Simple Quizzes

In order to assist those interested in getting started with interactive web applications, a simple quiz scenario is presented in this section. The form generation program described here is intentionally quite simple and is intended primarily to illustrate the concepts. Some suggested extensions and improvements are described in Section 4.

3.1 The Instructional Scenario

> *An engineering instructor would like to have a short quiz for the students after every lecture in order to assess the degree to which the major topics were understood. Taking class time to administer the quiz is inconvenient, and collecting and individually scoring a written take-home quiz is also undesirable. Thus, the instructor would like to be able to make a text file containing the questions and multiple choice answers, then somehow to generate the HTML quiz and the answer key for automatically checking and tabulating student responses.*

The proposed solution is to create two self-contained programs. The first program reads the quiz text file, creates the HTML file, and also generates an answer key. The second program is the form handler which runs on the server and compares the submitted form responses to the appropriate answer key. Thus, the same HTML converter and form handler can be used without modification by an arbitrary number of quiz text files.

3.2 The Text File Format

The required text file format used here is deliberately quite elementary. The text file begins with a line indicating the quiz identification, which is also the name of the file that contains the answer key. Following the ID line is the single line title of the quiz which will be displayed as a heading when the quiz HTML is viewed with a web browser. The remainder of the quiz text file contains the series of questions and potential answers that make up the multiple choice queries, as described next.

Each new quiz question in the file is indicated with a capital-Q in the first column. The wording of the question can begin on the same line after the "Q" and can continue onto as many lines as necessary by leaving the first

column of the additional lines blank. After the question text is complete, the series of potential answers are given by placing a capital-A in the first column of the line and giving the desired answer text. The correct answer is indicated simply by placing an asterisk in the column after the 'A', i.e., "A*". There currently is no provision for including comments or other notes, but such a feature could be added without substantial effort. An example of a quiz text file with a single question is given in Figure 1.

```
quiz1.ans

 This Is Example Quiz Number One

 Q  A 10kohm resistor conducts an unknown
  current.  If the voltage across the
  resistor is 1 volt, what is the
  current?
 A 10 amps
 A 10 milliamps
 A* 100 microamps
 A 1 milliamp
```

***Figure 1**: Example quiz text file*

3.3 The FORMGEN Program

Once the text file is prepared the hard work for the instructor is over. What remains is to convert the quiz text file into HTML form. This process is accomplished automatically with the program FORMGEN.C. FORMGEN was written in the C programming language simply because it is familiar to the author. Any language that is compatible with the server system could have been used for this purpose (Pascal, Perl, shell script, etc.).

FORMGEN uses two command line arguments: the name of the quiz text file to read and the name of the answer key file to create. The HTML produced by FORMGEN is printed to the standard output device, which typically is redirected to a file.

FORMGEN produces the HTML tags for the header and body of the file and inserts the question and answer text lines as a form structure. For each question a line is generated in the answer key file indicating which of the answers was labeled as correct by with the "A*" signal by the instructor. FORMGEN also creates a form request for the student's name and identification number which could be used for logging purposes, if desired.

The HTML output of FORMGEN for the quiz text given in Figure 1 is shown in Figure 2. Note that the "action" field (line 7) of the HTML must be edited to specify the URL of the appropriate form handler program for a given server location.

For each question on the quiz the number of the correct response, denoted by the instructor in the quiz text file with an A*, is recorded automatically in the answer key file (Figure 3).

```
<html><head>
<title> QUIZ TEST </title></head>
<body>
<h1>This Is Example Quiz Number One</h1>
<hr>
<form method="POST" action=" http://www.engr.unl.edu/cgi-
bin/FORMCHK" >
<input type="hidden" name="FORMID" value="quiz1.ans">
Enter your name : <input type="text" name="student"> <p>
Enter your Student ID Number: <input type="text"
name="ssnumber"> <p>
1:   A 10kohm resistor conducts an unknown
  current.  If the voltage across the
  resistor is 1 volt, what is the
  current?<p>
<ol>
<li> <input type="radio" name="A1" value="R1"> 10 amps
<li> <input type="radio" name="A1" value="R2"> 10 milliamps
<li> <input type="radio" name="A1" value="R3"> 100
microamps
<li> <input type="radio" name="A1" value="R4"> 1 milliamp
</ol><hr>
<b> You can either <input type="reset" value="cancel"> this
attempt and try again, or you can <input type="submit"
value="hand in">
your answers now .</b>
</form>
</body></html>
```

***Figure 2**: HTML generated from the text of Fig. 1.*

The answer file is stored on the server so that the form handler program can compare the responses submitted by the user to the correct responses.

```
quiz1.ans
A1=R3
```

***Figure 3**: The contents of the answer key file*

3.4 The FORMCHK Program

The server must be configured to understand the quiz answers submitted by the user and to provide some sort of feedback. This is the job of the CGI program FORMCHK. FORMCHK was written in C for convenience.

The FORMCHK program is specified to the server along with the data returned by the browser when the user selects the "submit" button of the quiz form. The server

launches FORMCHK and passes it the user's quiz responses. FORMCHK first looks for the FORMID tag, which specifies the name of the answer key file, then compares the user responses to the desired responses one by one. FORMCHK creates an HTML file on the fly that is sent back to the user indicating which answers were correct and incorrect. After the HTML is returned, FORMCHK exits and the user can either try the quiz again or move on to something else.

Since the required answer key file is specified in the form response data, the FORMCHK program can be used to check a new quiz simply by placing the new answer key in the directory containing all the other keys. Thus, once the local webmaster has installed FORMCHK it should be possible for the instructor to maintain and update the quizzes without any ongoing intervention.

3.5 Summary of the Process

The process an instructor would use to generate form-based quizzes with the software described in this paper is summarized as follows.

1. Compile the FORMGEN.C program on the computer of choice.

2. Create a quiz text file according to the proper format. Choose something short and simple to get started.

3. Run FORMGEN on the quiz text file to create the quiz HTML file and the answer key file.

4. Compile and install the FORMCHK.C program as a CGI application on the available web server system. Help from the local webmaster is probably needed

5. Move the quiz HTML and answer key files to the appropriate locations on the web server, and make sure the quiz is accessible by the desired users and that the answer key is inaccessible. The local webmaster may be needed to help with this, too.

6. Finally, let the users know thequiz URL!

4. Going Beyond Simple Quizzes

Simple multiple choice quizzes are fine, but they certainly don't represent the best the web has to offer. For example, several extensions could be made to the FORMGEN and FORMCHK programs described in this paper in order to make them more versatile.

- Have FORMCHK create a log file of the quiz scores, then produce statistics on the results for each question.

- Provide a mechanism to include figures, pictures, or diagrams as part of the quiz.

- Create a quiz "hint" file with suggestions to be returned in cases where the student selected an incorrect answer.

- Allow the instructor to create a quiz text file with N questions, of which M (where M<=N) questions are selected on-the-fly at random and in random order.

In short, once the basic procedures and mechanisms are understood, creating more sophisticated, useful, and interesting instructional applications can be an incremental process.

5. Conclusion

In this paper a set of two C programs for generating and checking on-line multiple choice quizzes have been described. The first program, FORMGEN, allows an instructor to convert a text file containing quiz questions and answers into an HTML form for use in a WWW site. The second program, FORMCHK, runs on the web server and performs the quiz checking and feedback functions. The programs, though simple, are intended to assist engineering instructors who are interested in experimenting with interactive web-based instructional materials. Source code for the programs and related information will be made available on-line via URL: http://www.engr.unl.edu/~rmaher/fie96/ .

Link:

- Programs FORMGEN.C and FORMCHK.C for WWW-based quizzes (FILENAME: FPRGTAR.Z, FORMAT: TAR+COMPRESS)

References

1. Boutell, T., "World Wide Web Frequently Asked Questions," URL: http://www.boutell.com/faq/
2. Graham, I., "Introduction to HTML," URL: http://www.utirc.utoronto.ca/HTMLdocs/NewHTML/htmlindex.html
3. World Wide Web Consortium, "Hypertext Transfer Protocol (HTTP)," URL: http://www.w3.org/hypertext/WWW/Protocols/Overview.html