An Author Usability Trial for the Networked Assessment Toolkit (NEST)

Craig Burton, Engineering Multimedia Unit, The University of Melbourne
aburt@ecr.mu.oz.au
Lorraine Johnston, Computer Science, The University of Melbourne
ljj@cs.mu.oz.au

Introduction

The on-going development of NEST, the Networked Assessment Toolkit, was studied as an example of how a WWW-mounted computer program must be designed to have a pared-down interface that does not attempt to mimic the kind of user control we are used to in stand-alone software.

Originally a UNIX command-line translator for marking up HTML exams, NEST was recently improved so that its authoring capabilities could be used via the WWW. Developed at the University of Melbourne in 1996, NEST was chosen for assessment of its WWW interface to determine how a complex piece of software can support a WWW interface.

A non-trivially complex application, NEST was considered to be representative of the types of applications desirable for access via the WWW.

The philosophies behind NEST, its interface and its function are described below, followed by the rationale for, and outcomes of, a usability study. This paper is most concerned with the areas where technical design considerations have a significant impact on usability issues.

NEST: Structure and Design Philosophy

What is NEST?

NEST was written to allow educators who were not programmers or HTML authors to write complex Common Gateway Interface (CGI) programs via a simple scripting language. It permits the production, delivery and assessment of web-based examinations and surveys. *TopClass*[2] uses similar technology. In both cases, a program runs on the WWW server to provide a basic interactive WWW interface out of standard HTML forms. Neither employ scripting languages at the client such as Java or Javascript. Interaction with the server (and NEST in this case) consists of submitting HTML forms and being served further forms.

It should be noted here that an HTML page may originate from either a text document sent to the client browser via a server, or alternatively may be created by a computer program, and can contain information assembled specifically for that client. The latter HTML pages are called Common Gateway Interface pages, but this difference can be transparent to the client. An HTML form is a page that has the ability to collect data from the client. It may contain HTML commands to offer the client buttons to click or text boxes to fill in. Forms can be created for the client in either an HTML page or a CGI page. When the client *submits* the form, some other basic CGI program must be ready to accept the submission content.

There are already many web tools to make HTML pages (such as *HTMLAssist* and *HotDog*), but there is currently little support for writing CGI pages, or for creating the CGI programs that process data sent from client forms. To address this need, NEST generates CGI pages with forms that also act as the receiving CGI program for handling the submission from the same form. The same program

then processes the data collected and supplies the author with summary information. The users who supplied the data can be given appropriate feedback such as hints or the results of their individual assessments.

To make the package more accessible, an interface was developed allowing the basic authoring abilities of NEST to be available via the WWW. As a special case of a *NEST Exam*, the NEST WWW interface was written with NEST itself. This allows the substitution of different languages and the re-organising of the interface layout and graphic appearance, again without knowledge of HTML or CGI.

NEST Exams are modular in their design. Each consists of a "cover page", which holds general information pertaining to the entire exam, such as its graphical appearance, encryption mode across the WWW, access period and access authority list.

To use the WWW interface, authors must provide details of the exam to build up the cover page for the exam. NEST then requires the author to supply text and images for posing the question stimuli (e.g. pictures to click on in a multi-choice scenario), the correct answer for the question and the question number to position this question in the exam lineup. We refer to this as an *exam content item.*

The process of authoring an exam is to submit the cover page information once, then to submit the content as many times as there are to be questions in the exam. To collect this information from the author, a CGI page called the "header" gathers the cover page information, while a second "content" page gets submitted several times for all the exam content items. Both pages have a similar appearance, and the server takes almost exactly the same time to respond to the submission of each.

NEST was originally built to serve exams to students, to collect their answers and to build assessment reports. To date, about 1500 students have sat engineering exams with NEST, and many more have used NEST quizzes built from past exams, or filled in questionnaires and course appreciation surveys authored with this package. Exams and quizzes are automatically marked and statistically analysed. Questionnaires are collated in a variety of ways and can be weighted for hypothesis testing.

NEST has also been used to web-mount an administrative database as well as an election for the Melbourne University Postgraduate Association, where voters were not all in Australia during the election period.

Design Rationale

NEST seeks to exploit a user's WWW experience and provide a web-page look and feel to what is in fact an interactive computer program. At the same time the design philosophy of NEST attempts to overcome some of the problems arising out of the conflict between "surfing" and user interface navigation. This is, in effect, another *orthogonal extension*[1] to the familiar WWW interface. This section deals with the major differences between this extended WWW interface and the user model of an interface based on their experience with stand-alone software.

It is important to note that users in this NEST trial were assumed to be computer literate, with experience using stand-alone packages. We also assumed that users had basic web navigation skills, i.e. they were familiar with both HTML widgets (such as scrollbars, clickable buttons and text boxes) as well as the more traditional windowing interface widgets (such as windows and dialogue boxes).

The "page" interface presents interactions with NEST as a series of HTML pages. We made use of the fact that many complex activities can be reduced to a series of linear steps; which is most helpful to less experienced users. A recent development that allows for the creation of WWW

"Wizards"[6] likewise extols the benefits of streamlining tasks into simple steps, much like the installation "Wizards" that accompany software distributions.

A simple HTML page also has a smaller *footprint* than other design alternatives. The footprint of a page that arrives at the client is the sum of its content volume (in bytes) as well as the time taken to execute any scripting languages that might be embedded in it (such as Java). A small footprint, form-based page is desirable and is also a robust method of talking to the client, since the user can resubmit the page if a fault is detected, such as a timeout. Further, the page model is in keeping with the user's expectation of a browseable system and their experience of common HTML forms.

Our approach reflects the philosophy of Shubin and Meehan[5]

> "The World Wide Web is a new application development platform. For better or worse, people come to it with expectations based on their experience with other platforms such as Windows, Macintosh and UNIX. This experience makes up a user's model. Unfortunately, that model conflicts with the model of navigation of the Web."

They conclude it is important to hide "the split between the browser and the application".

Other models of interaction over the WWW could be used, which would present a different appearance to the user. For example, Kindlund's[3] WWW-mounted administration program presents a new window on top of the browser window, which then behaves like a local program. In fact, the application must then juggle the user's requests with the distant server. We chose not to follow Kindlund's path, as we sought to exploit the user's familiarity with the browser, and did not want to introduce other control elements. Also, such special control of the browser requires a large footprint with a greater amount of information being sent to the client and the penalty of execution time of a complex script.

The Study

A usability study was carried out for both the authoring interface for NEST as well as for its presentation interface (exam, quiz or questionnaire). This identified a number of design issues which are relevant in the production of usable WWW interfaces.

Twenty-two students participated in the evaluation of each of these interfaces during two two-hour workshops held a week apart. Their objective was to give feedback regarding usability issues in the interfaces.

The students were in their fourth semester of the Information Systems degree at the University of Melbourne, and this was part of their experimental work in the subject "Human Computer Interfaces".

Initial instruction was given to the class as a whole about the purpose of the package and how they could access it on the WWW. They were also given written directions on what they were expected to do. Students worked in groups of two or three on the project.

In the first workshop, they were to develop a set of questions for a questionnaire. They were allowed to choose from question styles which have: a single answer from a multiple-choice set; multiple answers from a set of possible answers; or a text answer.

While one attempted to generate the questionnaire, the other person recorded the usability deficiencies which they noticed. In lectures they had been introduced to Nielsen's heuristic

guidelines[4]. Hence, for this evaluation they were directed to use these heuristics to assist them with identifying problems.

For the second workshop they were tasked with answering questionnaires constructed by other groups. While doing so, they were required to note any potential problems with the interface, again using the heuristics to help them find potential problems.

For each session students were given evaluation sheets to record problems, severity rating and comments.

Observed Usability Issues

Both oral and written feedback from the usability assessment sessions helped identify several important issues. These are discussed below. The usability studies indicated that the approach used in the development of the NEST WWW interface can, indeed, produce other usable interfaces.

Disorientation Effects
Even when the user had found the required information early on the page, they were often uncertain what to do, unless the end of the page was immediately visible.

A typical page for a question displays the question information first then any possible answers relevant for that type of question. After this, and at the bottom of the page, there was a "Make" button to indicate completion of the task of dealing with that question.

Long pages of information (e.g. those containing wordy options for answers to questions) were observed to cause a loss of orientation for the users. Where they needed to scroll to find further information, we observed extra scrolling, apparently in an attempt to collate all the information.

If pages were limited to a shorter length, these orientation problems would not arise. On the other hand, permitting only shorter pages requires the overall material to be split into smaller pieces. A potential outcome of this is lack of control over how to divide items, or a limitation on the content of the items.

Page Length
Page length is related to the volume of information being displayed. This in turn has an effect on the time it takes to transmit the information.

A designer has thus to trade off manageable page lengths versus the optimum size for a bundle of information. This, in turn, affects the upper limit for the number of items permissible per page.

In NEST one may offer up to seven possible answers to a question, and some subjects found this was too limiting.

Deficiencies in HTML
Subjects noted the problem of radiobuttons in HTML. They also found similar problems with check boxes. Once a radiobutton was selected, it was not possible to retreat to the "none-selected" state. It would be possible to use check-boxes with a checking script at the server to overcome this, but this is not in keeping with the design philosophy for NEST of minimising processing.

One of the problems with the widgets in an HTML form is that non-selection or non-entry of information can not be used as a "no" answer. In the client-server arrangement, anything missing in the transmission is simply not processed. Thus, NEST does not allow the generation of exam

questions where, for example, a single checkbox is offered for a YES/NO type question. Presumably, the user would click for YES and leave un-clicked for NO. In a windowing interface where the application is tightly coupled to its application, this kind of interaction is more reliable. However, with the WWW, not all clients allow the estimation of whether a transmission is incomplete.

Confusion over naming

Names used on buttons may be quite brief, but they have a non-trivial effect, as was found in this study.

"Make" was the name chosen for the button to indicate completion of a question. This word caused confusion, as it was not seen to be intuitive. It was suggested that wording such as "complete/finalise question" would have been more explanatory.

A similar complaint was levelled at "Upload". Perplexity is common over whether one is "uploading" or "downloading", mostly depending on the local jargon. In the present situation, the intention was to provide a local file to be sent to the server for use in the questionnaire. Such confusion can be avoided by using a more explicit description such as "Choose file".

Cooperative work

The WWW accessibility of NEST allows remote authenticated authors access to a common questionnaire. This made the co-authoring of questions very easy, since both authors could access the file.

However, there were deficiencies noted here, in that under the current setup, there are no locks on shared files nor some other method of resolving conflict over shared writing.

This is serious for WWW applications, as all program actions should be allowed to occur in parallel with the activities of other system users.

Browser incompatibilities

There were complaints from our authoring subjects that they could not use *Internet Explorer3.0* for this trial. Although the design of NEST tried to take browser incompatibilities into account, this was not entirely successful.

At the time of the study, only *Netscape* provided a facility to send a private file to the server. While this was not implemented in any other browser at the time, it was used because of its utility and the fact that other *Netscape*-only additions to HTML, such as frames, had been readily adopted by the W3 Organisation in the past.

NEST now uses a workaround by offering a standard text box to the author for them to give the URL of an image or other file for NEST to upload. The image must be world readable and be served from a URL that NEST can reach. This means that the author must manually transfer their possibly-private image files to a public site. While this is less than ideal, the workaround also means that very large files (such as MPEG video) can be uploaded into the server. The *Netscape* upload facility is not stable for files of many megabytes.

Another problem arose with the difference between the UNIX and PC specifications of file names, ie. "/" versus "\", and the use of spaces in file names. As author-provided application names must eventually become URLs, if they contain characters that UNIX shells interpret, or if they contain spaces, they must be modified. Arbitrary file names can be served as URLs, but spaces (at least) are coded up with special escape characters. The handling of these characters is not uniform over all platforms. It was thus necessary to advise our Authors not to use spaces, shell metacharacters (/,

&,|,-), etc. Permitted file names were thus reduced to combinations of alphanumerics and the underscore character.

Help facilities

Although subjects were somewhat critical of the precise content of the help available while using NEST, they were enthusiastic about the ease of accessing the online help. Of special mention is the natural support for context-sensitive help that can be linked directly to pages of a WWW interface.

The simplicity of HTML for providing basic image and hyper-linked information means that this is the medium of choice for delivery of resources that support the WWW interface such as on-line help, mail, demonstrations, purchasing and upgrades.

Further, generation of tutorial and help information is also greatly simplified, as states of the WWW interface can be "saved" as HTML and used as examples. To do this, pages of the interface (the HTML form statements) are copied off the interface, and pasted into the help pages. With a proforma of a different coloured background, and with the "quoted" section of the interface in some sort of thick-bordered table, this activity prevents the need to capture screens and crop out the parts of the interface.

At the moment, because the interface "quotes" look (and feel) like the actual interface (as opposed to being images of it), some of the subjects were confused, commenting that they could not work out whether they were on an interface page or in a help page.

However, the advantage of this method of graphical representation of the interface is the small footprint for help. This is contrast to the current trend where the help facilities for software frequently outweigh the software itself in size. It is also easy to maintain, as only re-pasting sections of the interface is required. We suggest that there are other ways to present the interface extracts, say by using reduced fonts, or by employing unusual colours or with an unusual border. This would remind the users that they are in a help page, and not the interface.

User Model Confusion

Incompatibility of the user model with the model presented here caused difficulties for some users. PC-savvy subjects were confused by one element of the NEST design which resulted from an attempt to be economical with the layout.

On PC machines, widgets such as checkboxes and radio buttons can be selected by clicking on the text next to them (we call this their "textual element"). This is also true of browsers on that platform.

Our users expected the words beside a checkbox to be part of the clickable area. Instead, they were confused by two things: only the button area itself was selectable, not the text beside it, and the interface was set up so that a context-sensitive help link was part of the checkbox text. Attempting to "check the box" instead called up the help mechanism. This was a deliberate design decision in an attempt to save space.

Conclusions

The aim of this study was to validate a page-based HTML approach to serving a complete user interface over the web. Our main design rationale was to exploit the user's familiarity with the web familiarity. The activity of WWW navigation is necessarily subject to transmission delays, and users are mostly content to accept this. However, similar delays for a single-user windows-based application would not be acceptable. Therefore we expected that users would accept a web-based interface to an application if it had the "look and feel" of navigation.

For a simple page-based WWW model, embedded script was not needed, nor were HTML4.0 frames exploited. This technology was avoided in as much as it is still evolving, and as it blurs the WWW-navigation/window-using experience for the user. Such technology was also avoided because it results in a larger footprint (download volume and/or greater page rendering/execution time) in what may already be a fairly slow operation, depending on the complexity of the WWW-mounted application.

One of the compromises of this page model is that all interaction with the server must be reduced to a series of steps. However, this simpler model of interaction may be more desirable as it increases the equity of usability for the application. Cited examples are the recent developments of "Wizards" for streaming complex activities.

Nielsen[4] documented that a simple approach to identifying usability problems can have decided benefits. In the present situation, subjects were asked to do a heuristic evaluation of the interfaces involved with the WWW-based NEST examination tool. Nielsen indicates that such an evaluation is not particularly useful with only one evaluator, but that as few as five evaluators can find a significant number of problems. We used 10 pairs of evaluators for our usability assessment, and uncovered a number of issues.

Overall, all the subjects in the study were able to use the new interface and complete a complex interactive authoring task in a reasonable time. Difficulties we document fall into two categories; those which we could predict based on the spartan nature of the HTML widget set and those arising out of the interface design itself and its conflict with users' navigation or user interface skills.

References

[1] Fox, A, Gribble, S D, Chawathe Y, Polito, A S, Huang, A, Ling, B and Brewer, E A. (1997) "Orthogonal extensions to the WWW user interface using client side technologies", *UIST*, Banff, Canada.

[2] Graziadei, B. (1997) "TopClass announcements and news", *http://cpdt.itec.suny.edu/cpdt/tc/tcnews.html*.

[3] Kindland, (1990) "Navigating the applet-browser divide", *IEEE Software*, 107-108, May.

[4] Nielson, J. (1993) *Usability Engineering*, AP Professional, New York.

[5] Shubin, H and Meehan M. (1997) "Navigation in web applications", *Interactions*, 13-17, Nov-Dec.

[6] Tidwell, D and Fucella, J. (1997) "Taskguides: Instant wizards on the web", in *Crossroads in Communications, Proceedings of the 15th Annual International Conference on Computer Documentation.* SIGDOC, Oct 1997 263-272 IBM Almaden Laboratory.

Acknowledgments

The authors would like to thank the following people for contributing to NEST and its derivative projects: Dr. Robert Barnett, Jennifer Port, Matthew Higgins, Bernie Kirby, Samantha Sharp, Karsten Haley, Sandy Munro, Yi-ren Ng amd the student participants from "Human Computer Interfaces".

This document is an excerpt from "Will World Wide Web interfaces be usable", same authors, unpublished.