DOCUMENT RESUME

ED 429 547 IR 019 455

AUTHOR	Siviter, Phil
TITLE	Authoring Tools for Courseware on WWW: The W3Lessonware Project.
PUB DATE	1997-11-00
NOTE	6p.; In: WebNet 97 World Conference of the WWW, Internet & Intranet Proceedings (2nd, Toronto, Canada, November 1-5, 1997); see IR 019 434.
PUB TYPE	Reports - Descriptive (141) -- Speeches/Meeting Papers (150)
EDRS PRICE	MF01/PC01 Plus Postage.
DESCRIPTORS	*Authoring Aids (Programming); Computer Assisted Instruction; *Courseware; Distance Education; Educational Technology; Foreign Countries; Higher Education; Hypermedia; *Material Development; Multimedia Materials; Workshops; *World Wide Web
IDENTIFIERS	University of Brighton (England)

ABSTRACT

This paper describes the W3Lessonware project being carried out at the University of Brighton (England), the purpose of which is to produce an integrated set of tools for the production of World Wide Web-based educational materials. The project and its objectives are introduced, and an overview of the main deliverables is provided. The general strategy which was adopted to elicit an evolving set of requirements is then outlined. Following a summary of how the main applications (structure editor, document editor, imagemap editor) were integrated, the paper draws some conclusions regarding what the project has achieved and points out areas for future work, in particular the need for an evolving library of courseware-oriented Web templates. (AEF)

Authoring Tools for Courseware on WWW : the W^{3}Lessonware Project

Phil Siviter, School of Computing and Mathematical Sciences, University of Brighton, UK
Email Phil.Siviter@brighton.ac.uk

Abstract: This paper describes the W^{3}Lessonware project being carried out at the University of Brighton, England, the purpose of which is to create an integrated suite of tools for the production of world wide web (WWW) based educational materials. We introduce the project and its objectives, and give an overview of the main deliverables. We then outline the general strategy which was adopted to elicit an evolving set of requirements. We say a few words about how we integrated the main applications, and finally, we draw some conclusions regarding what the project has achieved and point out areas for future work, in particular the need for an evolving library of courseware oriented WWW templates.

Introduction

Overview

This paper summarises the W^{3}Lessonware project, an 18-month project managed by UKERNA on behalf of the Joint Information Systems Committee and carried out at the University of Brighton. The aims of the project were to produce a set of tools to facilitate the production of multimedia courseware based on HTML, and therefore deliverable directly on the world wide web (WWW). The official release of the tools can be downloaded from UKERNA's web site at http://www.tech.ukerna.ac.uk/ from which the reader can also obtain project documentation and the other deliverables from the project. The *latest versions* of the tools can be obtained from the W^{3}Lessonware web site at URL http://www.comp.it.brighton.ac.uk/w3lessonware/.

The project team comprised members with expertise in the development of WWW materials, educational and otherwise; the development of non-WWW courseware and courseware management tools; the development of computer based simulations of laboratory experiments, and network management and administration. Following an exercise in the construction of an example of WWW based lessonware in which we paid particular attention to monitoring our use of tools and techniques, and noting the problems which we encountered, we drew up the following broad set of requirements for a tool set aimed at facilitating the development of such materials. We felt that a WWW lessonware processor (W3LP) should include:

a comprehensive, WYSIWYG, HTML editor which could incorporate a variety of media, in a variety of formats, into the HTML document
an imagemap editor enabling the direct manipulation of hot regions and graphical elements on an image
a utility to facilitate the structuring of large collections of documents
a utility to facilitate the management of large collections of documents
utilities to automate (at least partly) the tasks involved in making a large collection of HTML documents, and other files, look and behave like a coherent entity, e.g.,

> managing the links between documents as documents are inserted into and removed from the collection
> supporting the authors' view of the structure of the collection, and hence supporting meaningful operations on that structure (even though in reality the only structure is an arbitrary network of files)
> supporting the presentation of these structures to the user, in the form of common navigational metaphors and their associated icons (next, previous, index, etc.)

We concluded that the above requirements fell into three main categories:

1. document editing
2. imagemap editing
3. structure editing

Accordingly, we proposed to build three core modules for W3LP - one to cope with each of these three areas. There would therefore be a document editor module (W^{3}HTMLEdit) an imagemap editor module (W^{3}MapEdit) and a structure editor module (W^{3}StructureEdit). Each of these modules would be able to communicate with the other two, as shown in [Fig. 1].

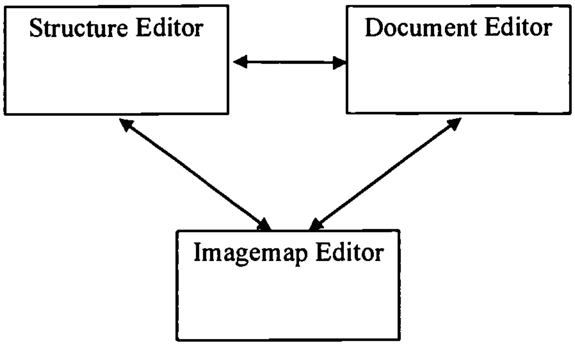

Figure 1: The three core modules of W3LP.

For example, the structure editor would present a view of the lessonware structure with icons representing files, and links between the icons representing links (e.g., HREFs) between the files. Double clicking on one of these icons would then open the relevant file in the document editor. The insertion or removal of links in the document, from within the document editor or the structure editor, would be reflected in the view from the other editor. Similarly, it would be possible to open the imagemap editor from within the structure editor. The structure editor enables the user to select a collection - any subset - of files from their project and pass their URLs to the imagemap editor, which will then create a user selected shape for each URL. The user can then position, size, and adorn the shapes to create a graphical menu whose hot regions hyperlink to the selected URLs. The imagemap editor can then converse with the document editor to enable the user to point to the position within an HTML file at which an imagemap should be placed. The document editor would be capable of displaying any in-line images (e.g., GIFs) within the document, in the same way that they would be seen when viewing the document with a typical web browser (e.g., Netscape, Internet Explorer, Mosaic).

In addition to developing the toolset, we created three examples of WWW based lessonware. These examples helped us to elicit requirements for the tools, and to test the tools. They also now act as templates for future lessonware developments. These examples can be viewed and downloaded from the aforementioned URLs.

Finally, we held two workshops during the course of the project - one in January 1996 and the other in July 1996. The second was multicast live over MBONE, and we hope to be retransmitting an edited version at regular intervals. The aims of these workshops were to get members of the courseware development community involved with the project, in order to:

publicise the project and its deliverables, especially the tools, so that they would become widely used

obtain feedback from the community regarding the tools' specifications, so that the tools would meet users' needs

The final specifications of the tools are lengthy documents and will not be repeated here. They can however be obtained from the UKERNA and Brighton web sites.

Choice of platform and development environment

The tools we have developed are all PC/Windows based, and were developed using the Borland Delphi environment (version 1.0). These platforms were chosen for the following reasons:

We decided that it was desirable to restrict ourselves to a single target platform so that we would make maximum progress with the tools, rather than spreading our effort across two or three platforms and not progressing as far.
The platform with the largest user base (by far) is PC/Windows.
Delphi was the preferred environment because it is based on an object-oriented language, and offered rapid application development facilities unrivalled, at that time, by any other object-oriented environment capable of producing stand alone executable files.

These decisions were the correct decisions in June 1995, when the project began. If we were starting the project today it is possible that we may make the same decisions again, although we would have to make a thorough evaluation of the latest Java development environments which have emerged recently to see if they truly compete with Delphi (and Borland's recently released C++ builder environment). If they do, then their cross platform promise would make them a very attractive proposition. In those circumstances the choice of hardware platform and operating system would be less crucial, although the PC/Windows platform remains the most cost effective, and popular.

General strategy

In this section, we describe the strategy we adopted for the execution of the project. Our approach was both iterative and collaborative. We planned the construction of three realistic examples of W^3Lessonware, for two main purposes:

1. to elicit and document tools requirements (or, to find out "the hard way" what tools would be useful)
2. to test the tools developed so far

The first example, produced at the start of the project, was primarily for purpose (1). The third example, produced near the end, was primarily for purpose (2). The 2nd example, produced midway through the project, served both purposes equally well.

From our experiences with the first example instance of W^3Lessonware, we defined a first tool set (project name: "toolset one"). We published a report on the W^3Lessonware example (available from the web site) and another on toolset one (also available from the web site). The idea was to get feedback very early on from the courseware development community. We wanted their ideas and opinions regarding tools requirements.

We then began specification and development of the core tools in the suite, regularly releasing updates to these tools on WWW, and asking for feedback from users.

Six months into the project, in January 1996, we held our first workshop (heavily oversubscribed) at which we presented the tools developed thus far, made useful personal contacts and received invaluable feedback and ideas. It is worth noting that the feedback received from this and the second workshop was greater in both quantity and quality than that which was received via WWW, email and mailbase over the entire duration of the project. Personal contact has proved to be more valuable than virtual contact - by several orders of magnitude.

Following the first workshop we entered our second major iteration, redefining the toolset (toolset two) and making substantial changes to the specifications of our core tools. A further six months work (which included the second and third W^3Lessonware examples) and WWW publishing brought us to our second workshop, in July 1996 (again heavily oversubscribed). This workshop was transmitted live over MBONE, and recorded in the University of Brighton TV studios. (We intend to re-transmit an edited version at regular intervals, both on MBONE and on the UK superJANET video network.) Once again we obtained invaluable positive feedback and constructive criticism from the delegates, which we acted upon in the next, and final, major iteration of the tools development.

The final period of the project saw us refining the tools to their current state of, if not perfection, then at least of professional standard and very real utility.

With hindsight we believe that this approach was basically correct, in that it enabled us:

to elicit external ideas and opinions, and to incorporate these into the tools

to be flexible in our specifications, as WWW developments occasionally threatened to overtake us

to make the user community aware of what we were doing, and thereby enhance the probability of the tools actually being used by a large number of people.

Integrating the Applications

The three main applications in the toolset (W^{3}HTMLEdit, W^{3}MapEdit and W^{3}StructureEdit) communicate with one another using the Microsoft Dynamic Data Exchange protocols (DDE). The HTML editor in particular acts as a DDE server to the other two applications, enabling the user to indicate simply, where within an HTML document a source anchor, or a destination anchor, or an imagemap, should be placed. (The imagemap editor also acts as a server to the structure editor during an operation which provides semi automation for the creation of graphical menus.) The details of the protocols used are documented in the HTML editor's on-line help. This information may be of use to other developers if they want to write applications to behave as clients in DDE conversations with W^{3}HTMLEdit, or if they want to write their own HTML editors to interact with the other W3LP tools in place of W^{3}HTMLEdit. (Such an editor must support the same protocols as W^{3}HTMLEdit in order to ensure correct operation of certain features of the structure and imagemap editors.)

In considering the requirements for inter-application communications, we did of course consider using OLE. However we eventually decided to use DDE instead for the following reasons:

We were developing for Windows 3.1 - a 16 bit platform - using Delphi 1.0, which produces 16 bit applications. Support for developing OLE clients is good in this environment, but support for developing OLE servers is poor. Placing OLE server capability in W^{3}HTMLEdit would have taken a long time and other aspects of the project would have suffered in the trade off.

The actual communication required to implement the above scenarios is very simple - DDE is perfectly adequate for this.

Should there be a further iteration of the tools, we would aim them at 32 bit environments, such as Windows 95 or NT, and develop them with Delphi 2.1, which has greater support for OLE server development. We would then use OLE to implement advanced features; for example, in-place activation of imagemaps from within the HTML editor, automatically invoking the imagemap editor (which would then be an OLE server).

Conclusions and Future Work

In terms of its original objectives, we consider the project to have been a success. The core toolset, comprising the structure editor, the imagemap editor and the HTML editor, presents users with an integrated suite of tools which makes creating lessonware for WWW a much easier process than it would otherwise be. The toolset is, as intended, of great use not only to the HTML novice, but also to the experienced user.

For example, a novice user can incorporate an imagemap into an HTML file without needing to know anything about <IMG> elements, USEMAP attributes, NAME attributes, shapes or co-ordinates. All they need to do is draw their hot regions on top of their image, specify URLs for each region, and point and click at the position within the HTML file at which the imagemap should appear.

Experienced users will also appreciate the time savings which the toolset can give them. For example, creating sequences of documents, and then editing those sequences by inserting and / or removing documents from the sequence, can be a very time consuming process requiring the editing of page counters in each document in the sequence, plus the editing of next/previous links (four links in three documents when inserting a new document into the sequence). The structure editor does all of this work for the developer, who simply drops the new document onto an existing sequential link.

The HTML editor enables developers to incorporate all of the major HTML elements, even if they do not know the syntax for those elements - they just supply the required attribute values (e.g., a destination URL, or the colour they want for their background) and the editor supplies all the mark-up. This not only eases the cognitive load on the developer, it also speeds up the whole process of writing HTML documents - even for experienced HTML writers. The built-in browser view also enables very fast switching between HTML and browser without the need to manually refresh / reload the document to see the effects of recent changes.

Naturally there are areas where we feel the tools could be improved. Any WWW project must face the fact that the WWW is evolving very rapidly indeed. Features which are commonplace now, such as Frames, Java, Javascript, ActiveX, did not exist when this project started. As these and other developments occurred we have had to decide whether and by how much, to modify the project's specific implementation goals in order to remain faithful to its original aims - of enabling the widespread exploitation of WWW for educational purposes. For example, we decided quite late in the project to incorporate some basic support for the creation and editing of frames within the structure editor. Time and resources are finite however, so some of the other planned features had to be set aside as a result. However, the frames feature has received very positive feedback wherever it has been shown, and overall we feel we have made the right compromise. We do of course intend to continue to develop and refine the toolset, as time and funding allows.

Thus we feel that the tools are a success, whether measured in terms of the project's own objectives, or in terms of how useful they are to anyone developing up to date, HTML-based resources.

The two workshops were also hugely successful, judging by the feedback we received from the delegates. In the second workshop especially, delegates who had never before used an imagemap (for example) were delighted at how easily they were able to generate one and incorporate it into their HTML (and the tools have been significantly improved since then to make it even easier). The workshops also provided an opportunity for personal contact between members of the courseware development community which we feel certain will prove fruitful in the future.

One area highlighted in the workshops - once again, the second workshop in particular - was the production and use of templates - ready made starting points for lessonware development which developers could use, and then tailor to their own purposes. The project has produced some templates for this purpose as planned, but although we always knew that templates were a significant resource, the feedback we have received has made us reassess the magnitude of this significance. We now believe that although sophisticated tools are an essential part of any developer's arsenal, the availability of a large, diverse collection of customisable W^3Lessonware templates, at a number of levels of abstraction (from a component on a single page to an entire course structure) would provide the greatest single boost in WWW lessonware development productivity.

To this end we are proposing a further project to develop such a library, along with tools for browsing the library, extracting and customising the templates, and putting them together to form coherent items of courseware for downloading to the developer's own site for completion. If our proposal is successful, we will look forward to building on the success of the W^3Lessonware project, to provide the community with a service consisting of a continuously evolving template resource base combined with courseware construction tools which are exceptionally easy to use (even by people with no technical knowledge of HTML). We hope that this will hasten the adoption of the WWW as a teaching and learning medium by an increasing number of education providers, and thus make a significant contribution to a more flexible educational provision for an increasing number of education consumers.