
Hickey, Thomas B., and Terry Noreault. "The Development of a Graphical User Interface for The Online Journal of Current Clinical Trials." The Public-Access Computer Systems Review 3, no. 2 (1992): 4-12.

1.0 Introduction

The Online Journal of Current Clinical Trials (CCT) is a peer- reviewed, interactive electronic journal. The primary form of publication is electronic--no paper version of the journal is planned. In addition to the full text of articles, CCT includes tables, equations, and graphics. There are two interfaces available: (1) a command-oriented ASCII interface, and (2) a graphical user interface. Subscriptions to the journal are currently set at $110 per year.

2.0 Journal Features

The journal is set up as a database that can be searched and displayed by the user interfaces. The ASCII interface (called EPS for Electronic Publishing Service) is based on OCLC's EPIC interface. It offers users, from virtually any terminal, access to the text and the ability to order offline services such as having the article faxed. While this interface works quite well, we expect most subscribers to use "Guidon," OCLC's graphical user interface that gives full access to the database's figures, equations, and typeset text. Guidon is a software package that runs under Microsoft Windows 3.0 and 3.1, and it must be installed on a user's personal computer. It is included in the base subscription fee. This is different than most electronic journals. The CCT is not a set of files that subscribers can FTP from OCLC. It is a database that requires software for access. If you only need to locate and request a document, this can be done by dialing or TELNETing into OCLC's EPS service. If you want to examine whole documents online, download them to your PC, or enjoy the benefits of using a window-based interface, then you need to have Guidon installed on your PC (along with Microsoft Windows 3.0 or 3.1).

Guidon uses a modified version of Z39.50 (1988) to communicate with the database at OCLC. There are several advantages to the use of this protocol: o Other user interfaces are planned for other environments and with other capabilities, but using the same protocol. o Authorization and billing is handled centrally. This both simplifies the interface program and eliminates most security issues about program code that is not under the vendor's control. o Capabilities will grow as we support the newer Z39.50 standards. o Z39.50 provides a clear interface standard to work with, rather than having to develop our own. o It makes it possible for others to use their own interfaces to get to OCLC databases. In the future, we hope to release a programmer's toolkit that will make the development of such interfaces easier.

3.0 Data Communication

The primary mode of data communication is over dial-up lines. The system is usable at 2400 baud, but 9600 baud is better. OCLC wrote a software layer to provide reliable communications from the PC to OCLC. A TCP/IP version that will run over the Internet is planned, with the expectation of substantially better performance. Making the interface work with standard modems was one of our major challenges.

In most cases, we were able to design the program so that the user is not blocked from using the document while waiting for a response over the phone line. While the interface is downloading requested paragraphs, users can scroll through and read sections of the document that have already been downloaded to the workstation or request new sections. Of course, since Guidon runs as a standard Microsoft Windows process, users can use other applications on their PC while long operations (such as downloading and local printing) are going on. Although the initial scheduled introduction date was April 1, 1992, this has been extended to July 1, 1992, allowing us to have more articles available and to further test and refine the software.

+ Page 6 +

4.0 Research

We have been conducting research directly related to this journal since 1982. Early research was focused on the use of sophisticated text formatters, font generation methods, and the problems of translating typesetting tapes into a form suitable for electronic retrieval and display. During the Graph-Text project, OCLC collaborated with the American Chemical Society [1] and John Wiley and Sons [2] to produce experimental CD-ROMs that contained chemical articles. Most of our work involved the TeX typesetting system [3] in some way, and we still use TeX to format CCT. During this period, computer displays and interface software have steadily improved, and there have been similar advances in telecommunications. Many of the problems we originally faced, such as font design for display, have disappeared because commercial software (e.g., word processing software) provides needed capabilities. The ubiquitousness of users exposed to the popular windowing systems on Macintoshes--and now PCs--has greatly lowered barriers that, in the past, would have discouraged use of the type of journal that we have designed.

4.0 Development

The following major events occurred in the CCT development process: 1989 August AAAS and OCLC agree to do an electronic journal. 1989 September Design starts. 1990 January Clinical trials selected as subject area. 1990 May Macintosh prototype developed. 1990 June Focus group interviews with MD's to evaluate the prototype. 1990 December ToolBook prototype on PC. 1991 February C++ file-based prototype started. 1991 September Online document display. 1992 April Initial target for introduction.

+ Page 7 +

1992 April Guidon 1.0 released; mainframe code goes into production. 1992 July Revised target date for introduction. 5.1 Programming Prototyping of Guidon was done using several different systems on both the Macintosh and PC. The production system was written using Borland C++. [4] This was the development group's first Microsoft Windows program, and it took several months before we were really comfortable with programming in Windows, although we had experience using several other window systems. Excluding telecommunications (but including much design and rethinking), the Guidon interface consumed about a year's worth of time for three programmers. The prototyping tools allowed for the exploration of different facets of the interface. Even though many members of the team had significant experience in full text (some with over 10 years of experience), it was still very challenging to blend functionality requirements and technological capabilities into a workable system. The iterative process facilitated by prototyping tools allowed constant refinement of Guidon. Microsoft Windows is a

complicated system, but there are several excellent books available that are a great help in understanding and programming in it. [5] The most difficult part is becoming familiar enough with the Windows calls that you know their capabilities and can find natural methods to accomplish tasks without great difficulty. There are also some things, such as printing, that can be accomplished very easily, but slowly. If reasonable performance is needed under Windows 3.0, the programming task becomes much more difficult. A benefit of the organization of Windows programs is that the look of the interface is greatly controlled by "resources" that can be edited independently of much of the actual code. During the development process, this has allowed us to take a running version of Guidon and edit it to demonstrate and try out different terminology and screen layouts. Once system changes have been agreed upon, they can be quite easily incorporated into the system.

5.2 Testing

Both OCLC and AAAS have invested heavily in interface testing-- OCLC has had one full-time and two part-time people testing various versions of Guidon and EPS for nearly a year. We started doing usability testing as soon as we had working prototypes, and, shortly after that, we started planning the system testing phase. There is always a tension between what is discovered through the testing process that could--and probably should--be changed and what can be changed within the constraints of organizational schedules and budgets. However, many of the usability and programming bugs that are found in testing can be corrected for little cost, and these changes result in a much more satisfactory system. One of the very early tests, in June 1990, was a concept test for both the journal and the technology. A small group of medical doctors from around the U.S. and Canada participated in a full-day focus group discussion in Boston. During this session we tested the basic concept of a clinical trials journal and the functionality of the online delivery system. This was a very productive session and had significant influence on the shape of the final product. One of the surprises we had during the session was the focus group's lack of enthusiasm for publishing original data with articles. Since access to data is one of the things that electronic publishing can do easier than print publishing, we expected that it would be a selling point for both authors and users. The doctors pointed out the great reluctance of researchers in the medical field (and in many other fields) to share the data upon which studies are based. Although we retained a simple method of including data with articles (if available), this session persuaded us to reduce our emphasis on this capability and thereby postpone the development of interfaces to spreadsheets for data manipulation. Another finding was that doctors preferred to view graphics in separate windows from text. Since this simplified both the programming and formatting needed for displaying articles, this was an easy wish to accommodate. After we prototyped a screen display that merged tables with text, we moved tables into separate windows so that they are accessed exactly like the graphics. We also learned more about the typical author and user's computer capability. In general, their computer and searching expertise was better than expected, but their access to electronic mail and networks was fairly low. This finding influenced how the system for reviewing articles was designed. We have found that as systems and software become more complex the testing needed to ensure a reliable service increases enormously. An example of this is the testing Microsoft did with Windows 3.1, the most recent version of Windows, which was reportedly tested at some 10,000 sites.

Although we don't need to test to this extent, we do have to worry about a large number of possible configurations, including: o Printers: support for the most common ones. o Windows versions: 3.0 and 3.1. o Font managers: Windows 3.0, 3.1, and Adobe Type Manager. o Operating systems: DOS 3.3 and 5.0. o Computer models: IBM compatibles, especially 286 and 386 computers. o Modems: support for at least a dozen models. Unfortunately, for many operations such as printing, a problem may show up in only one of the many configurations, such as a 286 microcomputer that is printing at a certain resolution on an Epson printer while it is communicating at 2400 baud. Since many tests take a considerable amount of time, we can only test the software under a few of the possible combinations, so we concentrate on what we expect to be the most common user configurations.

5.0 Database Construction

Articles are peer reviewed using a bulletin board system at AAAS, to which all the editors and reviewers have dial-up access. One of the goals of AAAS is to reduce the time taken to publish articles as much as possible without sacrificing the rigor of the peer-review process. WordPerfect is used during this review process as much as possible. After an article is accepted, AAAS sends to OCLC (via the bulletin board system) an SGML version of the article and the original graphics (if they are not machine readable, they may have to be physically mailed). OCLC then completes the SGML markup--in particular, OCLC completes the tagging of tables and equations as well as a number of other details. Currently, this tagging is done manually. After the SGML tagging of the article is completed and validated, the figures are scanned and the article is typeset. We are using TeX for this, so the SGML file is run through a program to convert it into TeX and format it. The resulting output is reviewed. After the output looks acceptable, it is faxed to both AAAS and the author for review, any needed changes are incorporated, and the database is built.

Although we realize that this is ambitious, our goal is to have articles available within 24 hours of their acceptance. To accomplish this, we need to be able to finish the SGML coding and formatting within six hours, and to have the formatting reviewed by AAAS and the author within two hours. The article will then be loaded into the database overnight. Even if this schedule is not met, we will have the information available to users within days of acceptance rather than the weeks or months that paper journals require.

6.0 Conclusion

We believe that CCT represents a new and important advance in electronic publishing that offers significant advantages over both paper publication and simple file transfer. We have done our best to reduce barriers to its use by extensive testing, listening to users, and trying to anticipate possible methods of use. In the future, we hope to support more journals, to add interfaces running on platforms other than Microsoft Windows, and to respond to the changes that users are bound to request. About the Authors Thomas B. Hickey, Consulting Scientist Two, OCLC Online Computer Library Center. Internet: th@rsch.oclc.org. Terry Noreault, Director, Division of Reference and Database Services, OCLC Online Computer Library Center. Internet: terry@rsch.oclc.org. References and Notes 1. Laura Buddine and Elizabeth Young, The Brady Guide to CD-ROM (New York: Prentice Hall, 1987), 270-276. 2. Thomas B. Hickey, "Using SGML and TeX for an Interactive Chemical Encyclopedia," in Proceedings of the 1989 National Online Meeting (Medford, NJ: Learned Information, 1989), 187-195. 3. Donald E. Knuth, The

TeX Book (Reading, MA: Addison-Wesley, 1984). 4. Borland International, Inc., Borland C++ Version 3.0: Users Guide (Scotts Valley, CA: Borland International Inc., 1991).

+ Page 11 +

5. Charles Petzold, Programming Windows: The Microsoft Guide to Writing Applications for Windows 3 (Redmond WA: Microsoft Press, 1990); Peter Norton and Paul Yao, Peter Norton's Windows 3.0 Power Programming Techniques (New York: Bantam, 1990); and Jeffrey Richter, Windows 3: A Developer's Guide (Redwood City CA: M & T Books, 1991). Appendix I. OCLC's CCT Personnel Mark Basham Graphics and database translation Rebecca Babyak Mainframe software and EPS interface Lisa Cox Database building William Curry Telecommunications Steve Driscoll Telecommunications Kevin Flash Early development Kim Fortney Telecommunications W. Richard Hale Project management John Handley Graphics Robert Haschart Interface Thomas Hickey Research and interface architecture Edward Hoare Testing Lynne Kellar Project management and database architecture Ralph LeVan Interface Daniel Meseroll Testing Carol Miller Testing Terry Noreault Project management Kevin O'Conner Source control and mastering Michael Prasse Prototyping and interface design Georgia Tobin Prototyping, typesetting, and documentation Richard Tobin Prototyping, database, setup, and operation Ron Vu Testing William White Mainframe software Lori Yoder Database input

+ Page 12 +
