XHyper: A Hypertext, Online Help System

James J. Hemmer

ABSTRACT

XHyper is a hypertext implementation designed as an online help system for X applications. XHyper presents online help as a single document with hypertext, multiple fonts, paragraph justification, figures, and postscript output. XHyper is fast, efficient, and can be easily integrated into graphical user interfaces. Online help files are stored in FrameMaker's Maker Markup Language, which can be loaded into FrameMaker for creating user's manuals. It is implemented in C++ using Stanford's InterViews.

Jim Hemmer is a senior programmer with Hughes STX Corporation at the EROS Data Center, Sioux Falls, SD (hemmer@edcserver1.cr.usgs.gov). *All work was performed under U.S. Geological Survey contract 1434-92-C-40004.*

Any use of trade, product, or firm names is for descriptive purposes only and does not imply endorsement by the U.S. Government.

INTRODUCTION

The X Window System (X) and increased computer performance has led to the development of robust applications with sophisticated graphic user interfaces (GUI). Providing user's manuals, training, and customer support for these systems require tremendous cost and effort for developers. An advanced online help system could greatly reduce training time and customer support requirements. In addition, such a system could help automate, or eliminate, the repeated generation of user's manuals. Despite great advances in GUI development, little effort has been spent defining online help methods and standards. Traditional online help provides an unappealing view of ASCII files, contains no figures, and has limited browsing capabilities.

XHyper is an online help system that supports:

- multiple fonts and highlighted text,
- paragraph formatting and justification,
- integration of figures, and
- hypertext.

BACKGROUND

The following sections describe the underlying software and concepts used in implementing XHyper.

InterViews

InterViews, developed at Stanford University, provides a variety of C++ classes, built upon Xlib, suitable for almost any X application. The InterViews libraries and applications are excellent examples of C++. InterViews is available via anonymous ftp from interviews.stanford.edu.

In addition to the several source libraries, InterViews contains two powerful applications: *idraw* and *doc*. *Idraw* is a drawing editor similar to MacIntosh's MacDraw. *Doc* is a WYSIWYG document editor, contains a simple table editor, and can import graphics generated by *Idraw* and several types of rasterized images (e.g. TIFF). *Doc*'s functionality and terminology is modeled loosely after the LaTeX document preparation system (1. Linton, and others, 1991).

Maker Markup Language

One of the more popular desktop publishing systems available today that operates on Unix workstations using X, is FrameMaker. To allow users to update documents from other word processors or text editors, FrameMaker supports the Maker Markup Language (MML). MML files are text files that describe the *format* as well as the *contents* of the document. Because the MML is somewhat broad, the current release of XHyper supports a subset of the MML (extensions are added for figures and hypertext).

One advantage of storing online help in the MML, is a user's manual could be easily generated from these files without extensive desktop publishing. The offline and online help of the system would reside in one location, eliminating potential inconsistencies between online and offline documentation and reducing maintenance costs.

Hypertext

Hypertext can be defined as the creation and representation of links between discrete pieces of data. When these data can be graphics or sound, as well as text or numbers, the resulting structure is referred to as *hypermedia* (2. Parsaye, and others, 1989).

Hypertext is an extremely useful component in an online help system. Instances where hypertext can be implemented include:

- cross-reference links between a word, phrase, or figure and another section of the document relating to the word, phrase, or figure;
- abbreviation expansions; and
- lateral and sequential movement within the document.

Cross-reference links are essential for an online help system. Frequently, references are made to issues discussed in other sections of the document. With a paper manual, the table of contents shows the actual page number and the user must turn to that page or even consult a different manual. In a traditional online document, the user must sequentially page to the desired document, which could require reloading one or more additional files. With hypertext, links can be established that allow the user to immediately access the referenced section, even across multiple files.

Traditional online documents are envisioned as a sequential list of pages. Using hypertext, documents could be organized in a tree structure, with each node representing a section of the document; the user can move across siblings, up to the parent, or down to the child. Figure 1 illustrates possible perusal links in a three-section document featuring hypertext.

XHYPER

XHyper is an object-oriented application written in C++ using the InterViews package. The object-oriented design allows additional features, such as line and paragraph spacing, to be easily implemented. Because it is X-based, it can be run locally or remotely, and so only one version of XHyper need reside on a local or wide area network.

XHyper is light-weight and efficient. Because multiple files are supported, only specific portions of a large document need to reside in memory simultaneously. Several sessions of XHyper may be run simultaneously, allowing different sections or even different documents to be viewed at one time.

Hypertext tags (detailed below) can be placed on the command line specifying which portion of the document to initially load and display. Only the MML file containing the requested section is

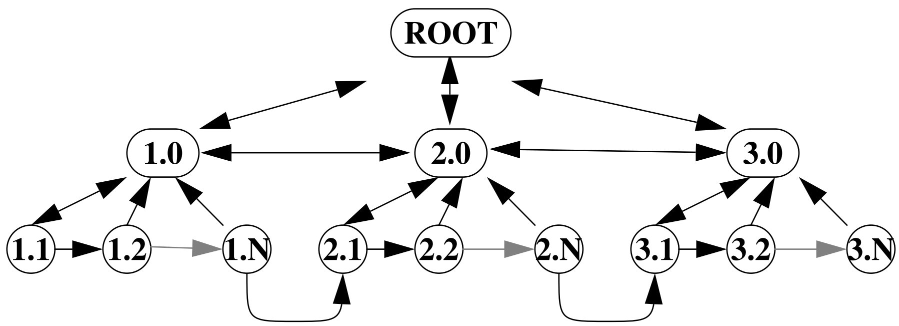

FIGURE 1: Hypertext document links

initially loaded. As other files are requested via paging or hypertext, those files are automatically loaded and the display updated. As files are loaded, they are held in memory for display if referenced later. Loading files is fast and transparent to the user.

Document Format

As previously mentioned, the XHyper documentation format is a subset of Framemaker’s MML. All commands and definitions are placed between “<“ and “>” and are not case sensitive. Although a small subset of the total MML, XHyper’s object-oriented design allows additional MML features, such as interline spacing and indentation, to be easily implemented. Below is a description of the currently supported commands.

Comment

All comments begin with “<Comment” and end with “>”. Comments have no effect on the document’s appearance. Comments can be of any length, including multiple lines. Example:

```
<Comment *** this is a two
  line comment >
```

Font Definition

Font definitions create a tag (or variable) for setting the current font.

Format:

```
<!DefineFont  tag
  <Family     family>
  <pts        pts>
  <style>
```

```
   >
```

Example:

```
<!DefineFont     ft>
   <Family       Times>
   <pts          10>
   <Bold>
>
```

Usage:

`<ft>`**This text is Times 10 point bold.**

Paragraph Definition

Paragraph definitions create a tag (or variable) that sets the style of all paragraphs following the tag. At present, only the font, hypertext, alignment, leading space, and spacebefore are supported. The hypertext definition is an XHyper *extension* to the MML. Placing "<hypertext>" in the paragraph's definition denotes that these paragraphs can be referenced as hypertext destinations.

Format:

```
<!DefinePar      tag
   <font         tag>
   <hypertext>
   <Alignment    alignment>
   <Leading      pts>
   <SpaceBefore  pts>
>
```

Example:

```
<!DefinePar      Section
   <ft>
   <hypertext>
   <Alignment    Justify>
   <Leading      6pts>
   <SpaceBefore  12pts>
>
```

If a given paragraph is to be a hypertext destination, the paragraph's tag must also include the hypertext tag. Following is a paragraph of type "Section," which may be referenced by the hypertext tag "mml."

Usage:

```
<Section  mml>
```

Figures

Idraw is the graphical editor delivered with InterViews. Figures, created using *Idraw*, can be linked to XHyper documents between paragraphs. The figures are left in separate files because they are in Postscript, not the MML. Because figures are linked to the document, they can be updated using *Idraw*. To insert a figure, use the command:

```
<Figure "filename">
```

Include

The “Include” command is similar to the “#include” preprocessor command in the C language. The named file is inserted at the given position.

Format:

```
<Include "filename">
```

Include files typically define a common set of paragraph and font definitions used throughout a document.

Font Style

The font style of the current family can be changed within a paragraph. For instance, to bold a phrase, place <bold> before the phrase and place <plain> after the phrase.

Example:

```
<bold> XHyer Document <plain> illustration.
```

Hypertext Implementation

The current hypertext implementation in XHyper supports the definition of links between phrases and individual sections within the document. Every document used by XHyper must have a table of contents defined in the file “TOC”. The TOC file lists the individual filenames, section names, and hypertext tags used to reference a specific section of the document. TOC records are in the following format:

```
section_number   filename   mml_tag
```

Comments can be included, but all other MML commands are ignored. The following is an example of a TOC file:

```
<Comment***Table of Contents for project.mml>
<Comment
Sect  FilenameHyper  Tag
----  ------------   -------->
1.0   section1.mml   Intro
1.1   section1.mml   MML
```

```
1.2   section1.mml   InterViews
2.0   section2.mml   XWindows
```

To cross-reference a given section, type the “hyper” command:

```
<Hyper tag>some_phrase<endhyp>
```

For example, a cross-reference to section 1.1 would appear as:

```
<Hyper MML>Marker Makeup Language<endhyp>
```

The phrase would appear in inverse video (i.e., white with a black background), and clicking on the phrase with the mouse would update the display to the start of section 1.1. Tags are used instead of section numbers, which allows section numbers to change as a document is updated. Future releases of XHyper may have section numbers which permit the user to peruse laterally within a document. Features such as document perusals and abbreviation expansions will also be developed.

Objects

The implementation consists of the following objects (i.e. classes):

- HyperViewer - top level window and buttons
- TextView - document display
- FontTag - MML font variables
- ParaTag - MML paragraph variable

In addition, the following classes from the documentation editor, *doc*, were modified and integrated. Although heavily revised for XHyper, the code originated from *doc* and the InterViews disclaimers remain in the source:

- Command - sends button requests to the HyperViewer
- PSFigItem - Idraw figure display

The following classes were taken verbatim from *doc*:

- Figure - figure of specific format
- IdrawImage - Idraw figure
- PageBorder - displays a layered border around the TextView

Figure 2 identifies some of the components in a typical XHyper display.

HyperViewer

One instance of HyperViewer is created per session of XHyper. The HyperViewer reads the table of contents and creates an instance of TextView for each MML file requested. The displayed TextView is placed within an instance of PageBorder. Below the PageBorder are four buttons that perform various functions. The buttons are instances of InterView’s *Button* class. Each button has

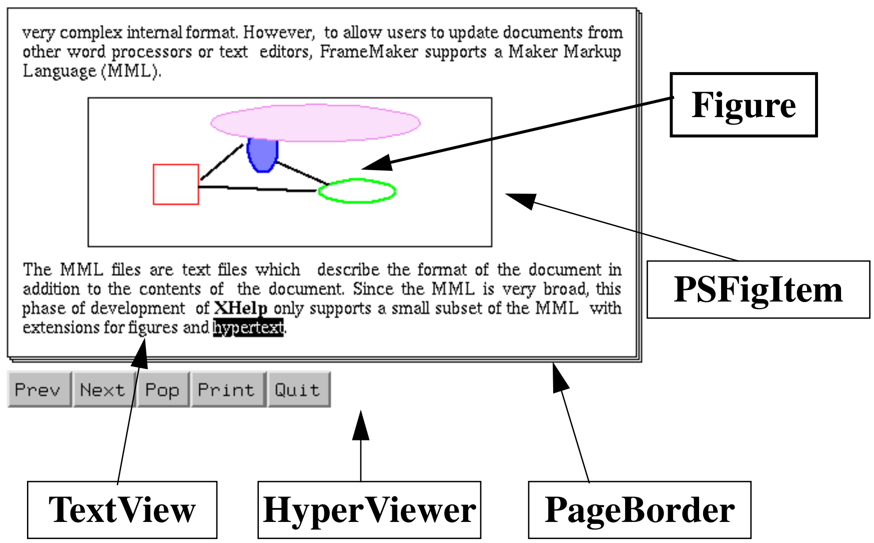

FIGURE 2: Class display.

a command associated with it that is passed to the HyperViewer via an instance of *Command*. For example, Next sends the command "Page Forward" to the HyperViewer. The buttons are:

- Prev - view previous page
- Next - view next page
- Pop - returns to hypertext source location
- Print - generate Postscript dump of current file
- Quit - exit XHyper

HyperViewer also receives hypertext requests from the presently displayed TextView. If a different TextView is requested, it is updated to display the hypertext's requested section. To aid in processing hypertext, the HyperViewer places character offsets in the internal table of contents structure. These offsets are the number of characters from the beginning of the MML file where the section begins. They are defined as a file is read and identify the start of a Hypertext section. Hence, the table of contents' structure in HyperViewer could be defined as:

Section	Filename	Tag	Offset
1.0	section1.mml	Intro	0
1.1	section1.mml	MML	1021
1.2	section1.mml	InterViews	1781
2.0	section2.mml	XWindows	0
3.0	section2.mml	Hypertext	Not loaded

HyperViewer is efficient, as it only creates TextViews for those files specifically required by the users. As mentioned earlier, a hypertext tag may be placed on the command line. For example:

```
XHyper    MML
```

The table of contents, listed above, would be loaded, and the section referenced by MML (i.e. section1.mml) would be located. An instance of TextView would be created, the file loaded, and the display updated to the section referenced by MML. If the user was to page beyond the end of "section1.mml," a new instance of TextView would be created that would display "section2.mml." However, if the user hypertexted from section1.mml to section3.mml, section2.mml would not be loaded, only section3.mml.

All paging and hypertext requests are handled by the HyperViewer, which processes each request by displaying the requested TextView instance and updating the display with the requested portion of TextView.

TextView

The TextView class is responsible for reading and translating the MML file, and for creating instances of the FontTag and ParaTag classes that represent the MML font or paragraph definitions. TextView separates characters into lines, and lines into pages; then it receives instructions from HyperViewer per the requested update. Likewise, TextView informs HyperViewer of hypertext requests.

As the MML files are read, TextView maintains a list of the Paragraph and Font variables defined. Each of these variables is represented by an instance of ParaTag and FontTag respectively.

As TextView reads a given file, if a section with a corresponding hypertext is discovered, HyperViewer is instructed to update the table of contents' structure with the character offset of the first character in the section.

For each figure requested, an instance of PSFigItem is created, which then creates an instance of the Figure class.

FontTag and ParaTag

The FontTag class defines a given font variable. Similarly, the ParaTag class defines a paragraph variable. The ParaTag can also contain a pointer to a previously defined FontTag, which would define the font used with that paragraph.

Integrating XHyper With GUI's

XHyper was specifically designed for GUI's with several different user interface panels, each performing a specific function. Integrating XHyper into new or existing GUI's is straightforward. The online help document would consist of sections relating to each panel in the application (the hypertext tag could be the name of the GUI panel). If help is requested from that panel, a session of XHyper would be created, using the *fork* and *exec* commands, with the panel's hypertext tag as a command line argument. XHyper would then display the section of the document relating to that tag.

Each panel would have a Help button that passes the name of the panel to a help callback similar to:

```
void help_callback(w, client, call)
Widget       w;
caddr_t   client;
caddr_t   call;
   {
   extern Display* theDisplay;
   if (fork() == 0){
       close(ConnectionNumber(theDisplay));
       execl("./XHyper","XHyper",(char*)client, NULL);
       }
   }
```

This flow is illustrated in Figure 5.

Future Enhancements

Enhancements to XHyper to be addressed include:

- moving laterally within the document,
- resizing the window, and
- paragraph margins.

Although figures are supported, XHyper does not support hypermedia via figures because the user cannot link objects in a figure with other areas of the document or other figures. This enhancement is quite feasible and should be addressed after the above enhancements are completed.

CONCLUSION

XHyper is an object-oriented, hypertext, online help system. Integrating XHyper simplifies, or even eliminates, the burden of generating user's manuals, which reduces cost to developers and leads to better products and lower software costs. By supporting the MML, XHyper documents can be directly loaded into the FrameMaker desktop publishing system. In addition, because XHyper generates Postscript output, offline help could be generated directly from XHyper.

The object-oriented design provides enhancement capabilities (e.g., hypermedia). Only the InterViews and X libraries are required to install XHyper, and both are freely distributed and available via internet.

XHyper is available via anonymous ftp from InterViews.stanford.edu under pub/contrib. Included is a small X application, xapp, that demonstrates the use of XHyper for online help.

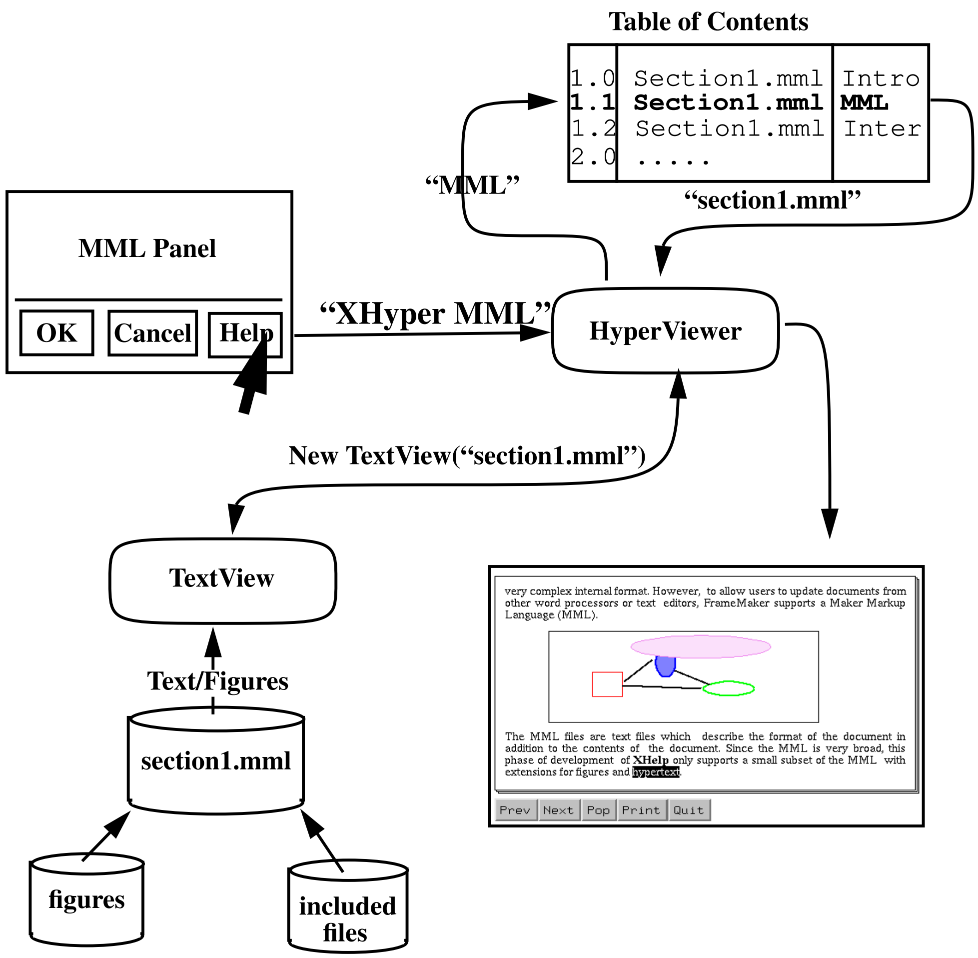

FIGURE 3: GUI help data flow.

BIBLIOGRAPHY

1. Linton, Mark and Calder, Paul (1991). Glyphs: Flyweight Objects for User Interfaces, Center for Integrated Systems, Stanford University.
2. Parsaye, M. Chignell, S. Khoshafian, H. Wong, (1989). Intelligent Databases. John Wiley and Sons, Inc..