Hardware/Systems

A Multimedia Document System Based on TEX and DVI Documents

R. A. Vesilo and A. Dunn

1 Introduction

This paper examines the development of a multimedia document system based on TEX. Multimedia document systems involve the design of many complex components including editors, formatters, display systems and components to support the content types used (text, images, graphics etc.). By using TEX to do the formatting, using a standard text editor to enter the document text contents and define the document structure, and modifying a DVI previewer to include support for non-text contents, the amount of effort in developing a multimedia document system is greatly reduced.

Multimedia includes time independent information such as text, graphics and raster images and time dependent information such as audio and video. The added time dimension means that documents are structured not only in space but temporally.

Multimedia information is inserted into TEX documents by means of multimedia macros and the resulting document is called a multimedia TEX document. This document is formatted by TEX to produce an extension of the standard DVI file, called a multimedia DVI file, that contains embedded multimedia commands. Since it could not be foreseen all the different content types that potentially could be used, a flexible approach was taken by using standard terminology, based on a standard model for representing multimedia information, for defining the macros and embedded DVI commands.

In order to present multimedia DVI documents, an existing DVI preview program was modified to interface to the different I/O devices and to support the document constructs needed for multimedia. It uses available image and audio processing software; however, software to integrate the various components had to be developed.

The remainder of this paper describes an overview of the system structure, the structure and definition of multimedia TEX and DVI documents and a description of the presentation program.

2 System Structure

Multimedia systems are integrated systems, which can be defined as a group of programs and components that are logically part of the one system and work together to handle an application. This can involve a sharing of data, the movement of data from one component to another or the use of the same presentational format by the different components. Integrated systems often combine functions previously performed by a series of single-purpose programs, for example, word processing, spreadsheets, communications and graphics.

One of the main trade-offs in system integration is in using off-the-shelf components against using in-house developed components. In a highly integrated approach, which has a consistent user interface and gives the user the impression that they were dealing with an integrated multimedia document, system development can be a large task involving the development of large programs and makes it difficult to take advantage of off-the-shelf components. In order to reduce development effort, the system described in this paper uses a number of existing and off-the-shelf components.

The hardware structure of the system is built around an IBM PC/AT compatible computer under DOS and an Enhanced Graphics Adapter (EGA), which provides an inexpensive platform. It is equipped with:

- a mouse,
- an Ethernet interface,
- an audio interface using the Votan voice card with attached microphone and speaker,
- a Matrix PIP-512 frame-grabber card,
- a PAL-RGB converter,
- a 10 MHz RGB colour monitor (SONY PVM 1300E),
- a video camera (National WVP-A2N).

The system software structure has components grouped into different categories: editing/creation, formatting, presentation and translation/communications. Processing is also categorised according to information type, with text, image and audio processing by different editors, formatting programs and presentation program elements. The image editor also has associated with it a scaling program that is used to format images so that they fit into the space reserved for them. Translator programs convert between multimedia DVI documents and the ISO Office/Open Document Architecture (ODA) Office Document Interchange Format (ODA/ODIF) [ISO8613] but will not be discussed in this paper.

3 Multimedia TEX and DVI documents

3.1 Multimedia Contents Model

To deal with arbitrary content types a general multimedia information model was developed. The model is similar to the model used in the ODA standards. Multimedia information consists of the encoded multimedia data and multimedia attributes that are used to describe the structure of the multimedia data and control the presentation of the data. For example, in an image, the bit sequence describing the actual pixel values of the image would be the multimedia data and the attributes might include the number of pixels per line, number of lines and bits per pixel.

In order to describe different data encodings and attributes a two-level scheme is used. At the first level, multimedia information is divided into generic content types. These include images, audio and graphics. At the second level each content type is split into a number of different content architectures where each content architecture describes a particular method of data encoding and a particular set of multimedia attributes.

This two-level structure is implemented by having each piece of multimedia information described by a multimedia attribute file which contains the associated attributes and the name of the file containing the encoded multimedia data. The attribute files are text files, enabling them to be easily created and modified by text editors while the multimedia data files are usually binary files.

A further refinement is to have multimedia information defined as either internal or external. Internal multimedia information is presented on the same device as the body of the document while external multimedia information is presented on an auxiliary device, such as an external colour monitor or a loudspeaker. During document viewing, icons are used to indicate the presence of external contents and to activate the presentation of external contents.

A standard method for defining multimedia attribute file formats is used in order to provide consistency, flexibility and extensibility. Attributes appear on separate lines and are of the form:

name = value

Standard attribute names are used where possible, but the diversity of content architectures means there may have to be exceptions to this. Values are associated with a data type which include string, integer, real, boolean and content architecture file extension. Comment lines begin with a ";".

3.2 Temporal Structuring and Scripts

With the added temporal dimension in multimedia documents, there is greater interaction with the user because of the need to control the playing of dynamic content types (in particular, audio in the present case).

Temporal structuring and user interactivity is achieved by means of icons and scripts.

In general, icons are used to indicate the presence of external multimedia information. They can be positioned either in the body of the document and be associated with related document contents or in a special section off the page, on another part of the previewer screen, and be associated with the entire page instead. The usual method of presenting external contents is to select the associated icon and then select the play function. Dynamic contents are a special case of external contents where the contents presentation is time limited. In these cases there is also the option of having the dynamic contents played automatically when moving to a new page, with the selection and playing operations conducted automatically. An example usage might be to have all the audio messages on a page played upon entry to the page.

Scripts are used to link a number of icons together and cause them to be played in sequence and can be used to contain simple audio, graphic and image presentations. Each icon involved has an identifier number and the sequence can be described directly by macro arguments or by a script file. The sequence of icons may played one after the other with or without user intervention when going from one icon to the other. The individual icons in the script can also be played on their own.

3.3 Multimedia macros

The multimedia TEX macros used are shown in Figure 1.

Macros begin with the letters 'mm' to that signify that they are multimedia. In the succeeding letters, 'img', 'geo' and 'aud' stand for images, geometric graphics and audio, respectively; 'int' and 'ext' stand for internal and external contents, respectively; and, 'on' and 'off' specify whether an icon representing external contents is to be appear on the document page or in the special icon area off the page. There are two variants of the script

\mmimgint{id}{file}{arch}{hdim}{vdim}	insert internal image
\mmimgexton{id}{file}{arch}	insert external image, icon on-page
\mmimgextoff{id}{file}{arch}	insert external image, icon off-page
\mmaudioon{id}{file}{arch}{auto}	insert audio, icon on-page
\mmaudiooff{id}{file}{arch}{auto}	insert audio, icon off-page
\mmgeoint{id}{file}{arch}{hdim}{vdim}	insert internal graphic
\mmgeoexton{id}{file}{arch}	insert external graphic, icon on-page
\mmgeoextoff{id}{file}{arch}	insert external graphic, icon off-page
\mmscriptlist{id}{list}	insert a script list
\mmscriptfile{id}{file}	insert a script file

Figure 1: Multimedia macros

macro. One using a 'list' argument and one using a 'file' argument.

The icon identifier number argument 'id' is used to identify icons associated with a script. For those icons not associated with a script the id has to be set to zero.

The 'file' argument specifies the name of the multimedia attribute file. The 'arch' argument specifies the particular multimedia content architecture used.

Macros which insert internal contents (images or graphics) contain arguments specifying the horizontal and vertical dimensions, 'hdim' and 'vdim', of the space to be reserved in the document body to hold the contents.

Macros which insert icons into the document page require space to hold icons but this is reserved within the macro definition. Macros which position icons off the page reserve no space in the document.

The 'auto' argument, which is either 'Y' or 'N', specifies whether or not dynamic contents is to be played automatically upon entering a page.

The 'list' argument consists of identifier number separated by commas.

3.4 Multimedia DVI commands

Multimedia DVI commands are the mechanism by which information about multimedia information in a document is passed to the presentation program and are embedded into DVI files by the multimedia macros as string arguments of the *XXX* command (see [KNU86] for a description of DVI commands). They have the following format:

- The first two characters are 'MM' to distinguish the multimedia DVI command from other possible uses of the DVI *XXX* command.
- The next two characters are a two digit identifier for the particular command. There is a multimedia DVI command corresponding to each multimedia TEX macro.
- This is followed by a list of command parameters separate by blanks.

Each parameter consists of a letter, followed by '=', followed by the command value. The parameters used are shown in Figure 2.

The D, F, A, H, V and U parameters are user supplied while the rest are provided by the macro body.

3.5 Detailed multimedia macros

Details of representative macros are given below. The other macros are similar with minor changes.

The definition of an \mmimgint macro, which inserts an internal image into the document, is as follows:

```
\catcode`\@=11
\def\@mmimi#1#2#3#4#5{%
    \special{MM01 D=#1 F=#2 A=#3 H=#4 V=#5}}
\def\mmimgint#1#2#3#4#5{\null
    \dimen0=#4\dimen1=#5\null
    \edef\h{\the\dimen0}\null
    \edef\v{\the\dimen1}\null
    \hbox to \dimen0
        {\vrule
         height\dimen1 width0pt\null
         \@mmimi{#1}{#2}{#3}{\h}{\v}\null
         \hfil}}
\catcode`\@=12
```

The \catcode`\@=11 command is used to allow private macro names to be defined for convenience. It is cancelled by the \catcode`\@=12 command at the end of the macro definitions. The #1 argument of \mmimgint is the identifier argument. The #2 argument is the name of the attribute file. The #3 argument is the multimedia content architecture. Space for the image is reserved by means of a \hbox with image dimensions passed by the #4 and #5 macro arguments: #4 is put into register \dimen0 and made

Identifier	Description	Format
D	Multimedia content identifier	Integer
F	Attribute file name	Valid file name
A	Content architecture	Three character string
H	Horizontal size	Valid TEX dimension
V	Vertical size	Valid TEX dimension
X	Horizontal size of icon	Valid TEX dimension
Y	Vertical size of icon	Valid TEX dimension
I	Icon to be inserted	Single character (Y or N)
U	Automatic content presentation	Single character (Y or N)
O	Icon on the page	Single character (Y or N)

Figure 2: Command parameters used in DVI commands

the width of the `\hbox`; and, `#5` is put into register `\dimen1` and made the height of a `\vbox` of zero width to define the height of the `\hbox`. This is done to position the `\special` command at the reference point of the `\hbox`. The `\@mmimi` macro is a private macro to invoke a `\special` command which inserts an *XXX* command, containing the multimedia DVI command, into the DVI file.

In the `\mmimgexton` macro, the `\hbox` is used to reserve space for the icon rather than for the internal contents, with the X and Y multimedia arguments being used instead of H and V. The I argument is set to 'Y' to indicate that an icon is to be inserted by the presentation program and the O argument is set to 'Y' to indicate that the icon is on the document page.

In the `\mmimgextoff` macro, no `\hbox` is inserted to reserve space. The I argument is still set to 'Y' but the O argument is set to 'N'.

4 Presentation Program

The multimedia document presentation program was implemented by extending an existing DVI preview program developed in-house at the University of Sydney, which was written in C and runs on an IBM PC compatible computer. The main functions of the existing previewer were to parse DVI documents, access fonts, display characters at the correct screen position, navigate through the document, scroll document pages and provide help facilities. Modules were added for it to have a mouse-based, graphical interface and hardware and software to support image display on the main screen, image display on an external monitor, audio presentation and user interactive control of audio and image presentation.

The extensions to the existing DVI previewer are such that the principles can be applied to a general previewer, provided that the previewer has access to screen-interface facilities that support mouse movement, allow icons to be displayed and mouse clicks on icons to be detected.

The screen layout consists of four sections. The largest section, in the centre of the screen, is used to display the document contents. At the right hand edge are two narrow vertical columns, one containing mouse buttons to activate menu functions such as playing of external contents, navigating through the document and exiting the program, and the other column is used to display the icons declared as "off the page". At the bottom of the screen are a number of message boxes to display the current mode of the previewer, information about the duration of dynamic content types and error messages.

The system supports images in two in-house formats. Although other image formats are available, these two were readily available at the time and easy to incorporate into the system. The modular nature of the system means that other formats can also be added without too much difficulty. One image format is based on the FITS image format [WGH81], used for representing images in radio astronomy and can support colour. The other image format is for simple black and white raster graphic bit-map images. The Halo graphics library is used to display both image formats.

Audio is processed in two formats: unstructured and structured audio. Unstructured audio stores voice messages as binary files recorded using a stand-alone record program. The attributes in the multimedia attribute file are based on the voice algorithms used by the VOTAN voice card and include a flag to indicate compressed or uncompressed voice coding, the amount of compression, whether silent mode encoding is used, the duration of message in tenths of seconds and the length of voice file in bytes. In order to play an audio passage the audio icon is

selected by the mouse (if automatic playing is not enabled) and then the play menu function is activated from the right hand menu column. Playing can be interrupted through the keyboard.

Structured audio uses a three-level hierarchy based on passages, paragraphs and sentences created using a structured voice editor [Ng89]. The editor includes commands to insert, copy, paste, delete and record voice at either the passage, paragraph or sentence level. The voice editor is integrated into the presentation program to allow passages to be played under the control of the structured voice editor interface.

Graphics is supported in the form of Computer Graphic Metafiles (CGM) [ISO8632] but only to a limited degree due to the lack of an editor to create CGM metafiles.

A two-stage process is involved in displaying document pages. During the first stage, the DVI page is scanned, the multimedia DVI commands for the page are extracted, parsed and stored in a list and the text, icons and internal multimedia contents are displayed. Each internal (and external) multimedia content type and architecture has a separate module which is accessed by a standard interface. By separating multimedia dependent processing from the other types of processing, multimedia information can be handled generically, making the system easy to expand to include new content types and architectures.

During the second stage, mouse clicks are processed to display external multimedia information. This requires accessing the multimedia attribute and data files and calling the different device interface libraries. An escape mechanism is provided to save the current page display, invoke a new display and return to the previous display so that device specific user interfaces can be used.

5 Conclusion

This paper has examined the design of a multimedia document system based on TEX and DVI documents. The system used a number of off-the-shelf and existing components to reduce development effort. However, more attention needed to be paid to issues in integrating the system components as a result. The `\special` command was found to be an important technique for achieving this integration.

By applying standard methods for defining multimedia macros and multimedia DVI commands based on a general multimedia model, a flexible and extensible multimedia document was developed. The system developed can easily be extended to include new content types and architectures.

The presence of dynamic contents in a document means there is more interaction between the user and the document. Two mechanisms for structuring dynamic contents were icons and scripts. By a combination of different multimedia DVI commands a range of interactions can be produced.

The effort in modifying the document preview program is reduced by using standard interfaces and modularity in the design. Once program changes are made, new features can be made available to users by copying existing macro definitions and making minor changes.

Further information regarding the multimedia document system can be be obtained by contacting R. Vesilo at Macquarie University by electronic mail at rein@macadam.mpce.mq.edu.au.

Acknowledgments

The work conducted was part of a PhD thesis at the University of Sydney. The author would like to acknowledge his supervisors David Skellern and Robin King, the CSIRO Division of Information Technology, who helped fund the project and the Commonwealth Postgraduate Award Scheme.

References

[ISO8613] ISO 8613, Information Processing Systems - Text and Office Systems - Office Document Architecture (ODA) and Interchange Format. 1989.

[ISO8632] ISO 8632, Information Processing Systems - Computer Graphics - Metafile for the Storage and Transfer of Picture Description Information. 1987.

[Knu86] D. E. Knuth. TEX: The Program. Addison-Wesley, Reading Massachusetts, 1986.

[Ng89] D. Ng. Multimedia Communications. Masters thesis. University of Sydney, 1989.

[WGH81] D. C. Wells, E. W. Greisen, and R. H. Harten. FITS: A Flexible Image Transport System. Astron. Astrophys. Suppl. Ser., Vol. 44, 1981, pp. 363–370.

◇ R. A. Vesilo
School of Mathematics, Physics, Computing and Electronics
Macquarie University
NSW, 2109, Australia

◇ A. Dunn
School of Electrical Engineering
University of Sydney
NSW, 2006, Australia