CMU-ITC-91-098

Multimedia Data Formats: Issues and Instances

Michael McInerny

12-Jun-91

Abstract

Multimedia data formats are the permanent representation of a multimedia document and thus encode the information in that document. Users are limited in their expression ultimately by what the data format permits the application to store. Therefore, it behooves the system designer to be aware of the issues involved in creating and using data formats. This paper sketches some of the user requirements of multimedia, discusses the variety of issues concerning data formats (both common and media specific), and examines a few key formats in use today.

[1] This work was performed as a joint project of Carnegie Mellon University and the IBM Corporation. The views and conclusions contained in this document are those of the author and should not be interpreted as representing the official policies of the IBM Corporation or Carnegie Mellon University.

Table of Contents

Introduction

The success of a multimedia system depends partly on three factors: its ability to exchange documents with established monomedia applications, access and organize the information in those documents, and exchange documents with other multimedia systems. These factors necessitate an understanding of multimedia data formats and how these formats impact storage and retrieval.

In addition, as multimedia capable systems grow in usage, data formats are being called upon to support novel applications. In particular, users want to freely mix media, reuse parts of other documents, and organize their multimedia documents with information management systems.

In this paper, we will examine the user requirements for multimedia, outline the key technical issues concerning the storage and interchange of each medium, and survey some of the prominent data formats and how they stand against the technical issues.

Using Multimedia

While a digital computer represents all information in terms of numbers (by definition), to require users to deal with media such as text and drawings in terms of their numeric representations would be counterproductive. With the advent of bit-mapped displays, applications can now present images that more closely match the cognitive expectations of the user. For instance, rather than print a list of vectors a computer-aided design (CAD) application can project a drawing of the design. While arbitrary media are possible, there are four broad classes of media a user is likely to encounter: text, drawings, audio/video, and documents composed of several media.

Formatted Text

Modern text editors let a user add typographical information (often called ‘‘styles’’) to their text and show the effect of these styles on-screen as the user works with the text. For instance, **emboldened** words look heavier, titles appear in larger type, and subscripts are smaller and sit below the baseline.

Some editors, recognizing that most styles are used to convey the structure of the document or additional semantics of the text, try to capture this information directly: an author builds a hierarchy of paragraphs, subsections, sections, and chapters; or marks passages as quotations. The text editor then induces the appropriate stylistic layout, using author-adjustable parameters.

Rasters and Drawings

Like a tile mosaic, raster images allow a user to manipulate the image down to the picture element (*i.e.* pixel), or more broadly, as with a brushstroke. Rasters are a flexible medium, as indicated by the plurality of methods for working with them: paint programs allow artists access to a digital canvas via simulated brushes and pencils; scanners and frame grabbers allow photographs and video stills to be captured; image editors allow pictures to be cropped, cut, pasted, rotated, and filtered.

A more structured approach to imaging can be found in drawing editors. Instead of manipulating the pixels, a user defines the image in terms of its component geometric shapes, such as lines, polygons, and circles. More sophisticated drawing editors work in three dimensions and provide the ability to specify surfaces and solids, even texture, material, lighting sources and manufacturing tolerances. Almost all allow the user to package a set of specifications together for reuse as a component, creating a personal or organizational library of designs.

The process of turning a drawing into an image is called *rendering*. Every drawing program is capable of at least enough rendering to allow the user to interactively manipulate the drawing. But there exist accurate and computationally expensive rendering operations such as ray-tracing and radiosity which can also be employed to generate photo-realistic images. Rendering a drawing into a raster freezes a perspective on the drawing, and opens up the possibility of an artist altering the image.

Audio and Video

Digital audio and video allow sound and motion pictures to be recorded, stored, and played back under computer control. Furthermore, because the media are sampled digitally, the computer has very fine control over how the sample is played back and when. One of the chief advantages that this technology has over computer-controlled tape or laser disc is that careful programming can eliminate the seek times, or pauses, when jumping from one point in the sample to another. Another advantage is that the samples can be digitally reprocessed and filtered, both to remove noise and perform other interesting effects. Digitizing video/audio also permits intermixing of other digital media, e.g., putting a graphic or some text on a video image.

Basic digital audio and video editors use a splice operation to assemble sequences, much like their tape or film counterparts. Filtering and other signal processing effects are becoming more popular, and start to take advantage of the flexibility of the digital medium. The most interesting audio/video applications allow an instructor to create programmed or interactive sequences.

Multimedia Documents

Once an author has access to several different media, he will want to compose a multimedia document, that is, a document which uses more than one medium (e.g. an article with figures). The ability to compose media varies from system to system: some systems allow only an aside, where the non-text media can be requested to appear in another window; others allow in-line media, but only in alternation with the text; the richest multimedia editors allow any media to contain any other media *ad infinitum*, so that a text can contain a table which contains more formatted text within which lies a raster. The Andrew Toolkit [Palay88] is an example of such a system, and it was used to produce this paper.

Technical Issues

In order to compare data formats, it is necessary to enumerate the technical issues against which the formats can be evaluated. There are some issues common to all data formats, but each media type also adds new issues. For example, continuous time media (like video or audio) have a real-time performance aspect to their presentation, unlike static media (like text or rasters).

Common Issues

All data formats can be evaluated on the issues of interchange, organization, their ability to support external references, and scale.

Interchange versus Storage

Many applications choose to store their information in a private, native format, often for reasons of space or time efficiency. These applications usually feature an option to write the document in a transportable fashion to be used for interchange with other applications. Data formats optimized for storage often take advantage of machine-specific encodings to increase speed and decrease storage costs; implicit and application-specific references are used; and hardware characteristics are assumed. Formats for interchange are careful to either avoid these assumptions or document them explicitly. For instance, numbers can be written out in ASCII and tags can be generated for use as targets of references. Frequently the resulting datastream is readable and can even withstand some careful editing. Unfortunately, this is not always true: the ISO abstract syntax notation (ASN.1 [ISO-8824]) has a binary encoding (BER [ISO-8825]) that does not lend itself to human interpretation and is difficult to modify [Rosenberg91].

A data format designed for storage can become an interchange standard if enough applications support it. In addition, interchange formats are sometimes used as native storage formats, so that an application has interchange capabilities ''built-in''.

Internal Organization

The ways of organizing the internal layout of a datastream encompass format, separation of data from form, and sorting. Older formats mimic a deck of punched cards with fixed length and fixed format records while newer formats use a grammar with data tucked between delimiters or keywords. Structured documents raise the issue of whether to mix structure and data, or to group all the raw data and use pointers from the structure back into the data. Sorted data are easier to search than unordered data, but the sort operation can be computationally expensive.

External References

The ability to *name* or point to a component or subsequence of a medium is dependent upon the structure of datastream: whether it supports tagging and clustering, or uses compression. *Tags* allow the author or reader to name a point or a region of interest within the document, which will be adjusted accordingly across edits. *Clusters* are linked components, which are preserved across edits like tags but can refer to a set of interesting regions. Compression is used to reduce storage requirements, but it interferes with external references since it may become necessary to decompress some or all of the data in order to extract the target region. Most data formats simply are not designed to allow for arbitrary access to the document.

Scalability

The ability of a format to handle large and small documents is its scalability. There are often hard limits to the size of the information being represented: number of dimensions, extent of the dimensions, resolution, total size. In addition to considering how well a format deals with large or complex documents, it is also interesting to note how well it deals with very small or simple documents. If the format requires a lot of overhead in coding the document, authors of applications which generate small documents will probably forego the complicated format in favor of a simpler, less general format or even create a private format.

Static Media Issues

While static media (like text) lack the performance aspect of continuous time media, in many ways they are more sophisticated than their real-time cousins, owing to their need to convey as much information as concisely as possible. The most common types of static media are formatted text, rasters, and drawings.

Text Issues

Page Description versus Layout Rules

When storing formatted text, the datastream can represent either the *structure* of the document or the *formatting*. A document's structure includes text of the document and indicates the layout rules. It is easy to edit and it preserves an author's semantic structure (e.g. SGML). The document's formatting indicates how the text should be positioned on a page and is useful for printing (e.g. PostScript). Some systems allow both types of information to be stored (e.g. ODA).

Deep or Shallow Markup Semantics

Text editors with *shallow* markup semantics allow text to be marked with style macros, which translate a tag like ''chapter'' into ''new page, bold, bigger, and left justify'' typographic conventions (e.g. ATK). While this is sufficient for creating a printed page, it fails to capture the notion that the paragraphs and subsections following the chapter title, up to the next chapter, are logically part of the chapter. In other words, the semantics of ''chapter/section/subsection'' as a hierarchy are completely lacking. An editor with *deep* markup semantics actually organizes the text internally as a hierarchy (e.g. IBM's Quill for SGML). Deep markup semantics produce a datastream that can be unambiguously parsed without assumptions as to the semantics meant by the style information.

Raster issues

Bi-level versus Continuous Tone

A *bi-level* raster[1] uses only two colors, usually black and white. A *continuous tone* image allows the use of many colors to express the image. Certain compression techniques, such as run-length encoding, work well on bi-level rasters (e.g. in Group 3 fax) and miserably on continuous tone images. Likewise the converse holds true for techniques like discrete cosine transforms (DCTs, e.g. as used in JPEG). [Chen84] Bi-level images are usually smaller, since they have an obvious binary encoding. However, fractional scaling of bi-level images suffers from aliasing, while continuous tone images can use antialiasing techniques to overcome this problem.

Bitplane Depth

The number of colors available in an image is capped at an upper bound of two raised to the number of *bit planes* which make up the raster. Thus, an 8-bit deep raster has a palette of 2^8 or 256 colors, and a 24-bit deep raster has 2^{24} (over 16 million) colors (e.g. TIFF).

Color Lookup Tables

A raster can use a means of indirection called a *color lookup table* (CLUT), to increase the apparent depth of the image (e.g. GIF). For instance, a picture of an apple on a red tablecloth will need many different shades of red, but few shades of blue and green. So, a CLUT can be created to reflect this imbalance in the color distribution, and the pixels in the raster will be assigned the index value of the appropriate color in the CLUT. This technique allows 24-bit images to be compressed to 8-bits, with minimal loss of information.

"Alpha Channel" Information

It is also possible to allocate additional planes to encode information about the pixels in the image. This so-called *alpha channel* information can be used as a mask to indicate what parts of the image are in the foreground versus the background, the level of transparency in different regions, even range or distance from the camera (e.g. TIFF).

Drawing Issues

Application Domain

Drawing editors are usually targeted to a *domain*: oriented towards producing certain types of documents (e.g. computer aided design or CAD). An application's emphasis on allowing the user to express some operations rather than others has an impact on what the datastream is capable of expressing. For instance, a CAD program can store three-dimensional shapes, while a graphic design program cannot. On the other hand, the graphic design program might provide an artist the ability to manipulate font outlines and save that information in a datastream which the CAD program could not read because it does not know how to manipulate fonts in such a manner.

---

[1] The term *monochrome* is sometimes used, but monochrome more precisely refers to a continuous tone image made up of shades of one hue.

Dimensionality

One of the basic differences among drawing editors is the number of *dimensions* they will handle. Graphic designers need only the two dimensions of a sheet of paper. Semiconductor and circuit board layout needs several co-planar and inter-connected 2-D surfaces. CAM systems need to be able to specify parts in three dimensions. The impact is that 2-D systems know nothing about wire frames, surfaces, and solids, and 2-D datastreams cannot represent these forms.

Richness of Primitives

Simple drawing programs provide only lines and polygons as *primitives*, while sophisticated editors allow curves, surfaces, and solids. Some editors have libraries of predefined shapes like logic symbols or gears. Simple editors can deal with complex shapes only when they are broken down into their simplest elements, losing the semantics of the object.

Continuous Time Media Issues

Because of their novelty, most attention has been focused on the storage of continuous time media for straightforward delivery alone. Unfortunately little has been done to enable users to structure, annotate, synthesize, and compose and synchronize these media. In any case, the real-time component of continuous time media presentation and perception presents the data format designer with challenges in bandwidth, latency, sampling rates, and compression and synchronization.

Sampling Rates and Depth

Continuous time media all start and end as analog phenomena. The process of capturing the analog signal and turning it into a digital stream is called sampling. The frequency at which the analog stream is sampled is the *sample rate*. The precision to which it is sampled is called the *sample depth*. Both affect fidelity: the sample rate limits the highest frequencies which can be reproduced, and the sample depth limits the dynamic response, or the ability to capture subtle changes in the signal.

''Lossy'' Compression

There is a class of compression techniques which result in a loss of information on decompression. These lossy compression techniques can be used effectively when the human perceptual system is relatively insensitive to the information lost. For instance, NTSC video encodes the signal into two components: a high-frequency luminance signal for creating high-resolution black and white images, and a low-frequency chroma signal, for adding color washes to the luminance information [Lippman89, Alexander90].

Bandwidth from Storage to Display

Bandwidth is the volume of data (or data rate) required for presentation of the media. Audio needs around one and a half million bits per second (1.5 Mbits/second) for high quality stereo (CD data rate). Television-grade video needs 30 Mbits/second (NTSC data rate), and high definition television boosts that requirement to over 100 Mbits/second. A serious problem develops if the storage device or operating system overhead puts a ceiling on this bandwidth; the only solution is to use compression to lower the bandwidth through the system and decompression at the output device.

Latency in Delivery

Because it is *continuous* time media, it is important to estimate the latency, or delay time, of a data format. A data format can contribute to latency through the use of encodings (e.g. compression) and poor organization. Encodings which are difficult to code or decode can cause data to be delayed, resulting in choppy performance. Poor organization may cause excessive seeking on the hardware device in order to collect all the required data, and the cumulative effect of all the seek times may also cause the data to be delayed. A bad format can also prevent quick error recovery or amelioration.

Compositional Media Issues

Compositional media systems attempt to bring different kinds of media together in a single document. The measure of how flexible a compositional system is in incorporating new media types is called its *extensibility*. *Reusablity* is the ability to encode and reuse common information.

Extensibility

A compositional system which defines support for only a few media types will soon become obsolete as new media types are invented. A better approach is to allow the datastream (and the application) to support an extension system whereby new media types can be introduced and incorporated into the document. Some systems are extremely flexible and have a well-defined architecture for supporting new media, while other systems provide only a trap-door approach of adding private data. The latter approach is often cumbersome and usually lacks the appropriate semantic support for the media.

Reusability

The ability to record common information once and reuse it later results in smaller, more manageable documents. Multimedia implies different kinds of information, but it is also convenient to support multiple views on a common set of information, e.g.: two different projections of a 3-D drawing, or a table of numbers and a graph of that dataset. Functionally, there is one set of data supporting the multiple views, so that any change to the data from one view will cause all of the views to be updated. A compositional data format for such a system would need a way of encoding a view's reuse of data from another source in the document (or even another document).

Survey of Data Formats

There exist far too many formats for any one paper to document. What follows is a brief summary of some of the more common data formats: how they stand on the technical issues and any additional facts about the format related to the issues.

Static Media Formats

Formatted Text

Scribe and LaTeX

SCRIBE and LaTeX are two non-interactive document preparation systems. Both require that the user employ the text editor of his choice to create the source document, that is, the datastream for the document formatter program itself. Thus, it is a storage format. The text is marked up semantically, but the structure is mostly shallow (SCRIBE does allow the markup forms to create an environment, and environments can be nested). The advanced user can alter the layout rules, but not the casual user. While both systems require a tiny amount of document overhead, there is no limit to the length of a document. There are no provisions for referencing pieces of another document, but there is support for including outside files as subdocuments, and also for dealing with an external bibliographic database. [Reid80, Lamport86]

ATK's Ez

The Andrew Toolkit (ATK) is an interactive multimedia system, which users know through its formatted text application ''ez'' [Palay88]. The datastream was designed for storage, not interchange. However, by convention the datastream is written using only printable ASCII characters and line lengths less than 70 columns [Borenstein90]. Because of these conventions, ATK datastreams can be passed through nearly all electronic mailers anywhere on the Internet, Bitnet, or Usenet, allowing ATK documents to be mailed from one site to another. Ez can manipulate plain text documents, but styled documents have a small amount of overhead and a few encoding rules. Documents can be arbitrarily long. The page layout is accomplished through rules, which can be altered by casual users. More complicated layout changes (e.g. 2-column text) can be achieved by expert users through backdoor hooks into troff and PostScript (which together form the underlying printing model). However, the markup semantics are purely shallow, and there is no provision to reference pieces of an ATK document. It is possible to reference another ATK document in its entirety, but only by expert users.

RTF

Microsoft's Rich Text Format is an industry standard format for the interchange of text with typesetter information [Microsoft90]. The format is grammar-based and oriented towards page layout, although there is limited support for style sheets. The semantics are shallow, but assumes a section and paragraph organization to the document. There are no means for referencing other documents, but there are facilities for associating pieces of text (e.g. footnotes and annotations). There is no limit to the length of the text, but the style and font tables are limited in their size. There is a facility for including bi-level rasters and for organizing text into rows and columns.

ODA

The Office Document Architecture (ODA) is an international standard for the interchange of formatted and processable documents [ISO-8613, Rosenberg91, Brown89]. The file format is specified by a grammar, and there are currently two defined encodings: a binary ASN.1 [Rose90, ISO-8824, ISO-8825] encoding called ODIF, and an SGML encoding called ODL/SDIF.[2] Because of the encoding, it is tedious to represent small documents; on the other hand, documents may have unlimited length. ODA documents can have multiple page layouts, and can be laid-out with rules as well. The structure of a document is deep. It is not possible for one ODA document to include another, but it possible to reference a portion of another document.

SGML & DSSSL

The Standard Generalized Markup Language (SGML) is an international standard for the markup of documents [ISO-8879, ISO-8879/A1, Brown89]. It defines the syntax to be used for declaring and adding structural and other types of markup to a document. It does not specify how to layout the document. A companion standard, which is in progress, Document Style and Semantics Specification Language (DSSSL), does provide a standardized way to add layout rules to an SGML document [ISO-10179]. SGML documents can be encoded with very little overhead, and imposes no restriction on the length of a document. SGML supports the inclusion of whole documents (called subdocuments) as its only form of external reference. However, SGML does provide the ability to specify multiple versions of a document in a single file/encoded form.

Rasters

PBM

PBM is a collection of utilities for interchanging and manipulating rasters. It is available on the MIT X windows distribution or from the Usenet's comp.sources.unix archives. Early versions supported only bi-level rasters and stored the data with a brief header followed by the bits of the image, each bit encoded as an ASCII one or zero character. Newer versions of the PBM package support grayscale (8 bit deep) and color (up to 24 bits) images. The encoding remains ASCII-based, but a more compact non-compressed binary encoding also exists. While the ASCII encoding is highly transportable, facilitating interchange, it is very bulky, and because arbitrary white space can be present in the file, references into the file are difficult. However, the binary encoding is reasonably space efficient, and offsets into the file can be calculated to select individual pixels (or a range of pixels). The encoding has almost no overhead, and permits arbitrarily large rasters. There are no provisions for a CLUT nor alpha-channel. There are however, numerous filters for converting to and from many different formats and display devices.

CCITT G3/G4 Fax

The CCITT Group 3 [CCITT-88-VII.3, Warner90] one-dimensional encoding for bi-level images is a scan-line compression scheme which uses a predefined Huffman table to encode runs of black and white pixels. Group 3 two-dimensional coding (which is also used by Group 4) allows the compression of a scan line by encoding its delta from the previous line, again by using a predefined delta coding table. Both coding schemes take one page at a time and produce a bit string. The bit string makes arbitrary reference into the file meaningless, and the two-dimensional coding requires the decoding of all the scan lines back to the reference scan line (which in Group 4 is at the top of the page!).

[2] Currently, all implementations of ODA use the ASN.1 encoding.

CCITT Group 3 (one-dimensional) compression is used primarily by telefacsimile (fax) machines [Glass91]. In addition to the raster encoding, however, fax machines also exchange other information about the transmission (number of pages, identity of the caller) using additional fax protocols. CCITT Group 4 (two-dimensional) encoding is becoming increasingly popular for use as the digital storage format for page images (scanned microfilm, electronic paper archive).

Because the implicit tables were designed to support pages of text and simple line drawings, half-toned images compress poorly. However, on pages of text, a tenfold compression is common.

JPEG

The Joint Photographic Experts Group [JPEG90, Wallace91, Baran90] has defined a lossy compression scheme based on the Discrete Cosine Transform (DCT). The DCT is simply the cosine component of the Discrete Fourier Transform. Performing a DCT on an image produces an array, which when quantized yields many zero elements. This transformed array can be compressed using well-known sparse matrix compression techniques. However, the quantization which leads to the sparse matrix also causes the decompressed image to differ from the original image. Due to the nature of the compression, small amounts of quantization produce negligible loss of quality. Because the quantization factor can be increased, the amount of compression (and also the degradation in image quality) can be increased. Color images up to 24 bits deep can be compressed. There are no facilities for a CLUT nor alpha-channel. Because of the compression, it is only possible to refer to the whole frame and there is no facility to reference other images in a JPEG file.

The JPEG standard defines a file format which is self-describing, and thus is suitable for interchange. The high degree of compression makes the format suitable for storage as well. However, as the processing is tedious and geared towards larger rasters, small rasters are unlikely to be encoded with JPEG. Several vendors will license software to perform the compression and have chip-level hardware which will perform the compression at high speed--fast enough to compress NTSC video frames to fast hard drives [Webster90, Rosenthal90].

TIFF

The Tagged Image File Format (TIFF) is an industry standard promoted by both Aldus Corp. and Microsoft Corp. [TIFF5.0, Graef89] It is a standard file format which allows for the encoding of bi-level, grayscale, CLUT, and full-color RGB images, with various depths, and optional compression. It was designed for interchange among desktop publishing (DTP) applications, but is suitable for storage as well. For each image in the file (there can be more than one), there is a header for the image which is a set of named parameters. The parameters are named, or ''tagged'', so that they can be omitted (defaulted) or the list of tags extended. Pointers are used heavily to point to variable sized blocks of data within the file, and images can be broken into horizontal strips of arbitrary height and order. This means that the pixels which make up the image can be scattered in blocks throughout the file; however, it is also possible to lay out the file in a straightforward fashion, with all of the pixels of the image in one block. TIFF is best used for small to large rasters, as there is some overhead and there are hard limits to the extent of a raster. There is no way to reference an external file from inside a TIFF file, but it is feasible for an application to refer to strips of an image within the file, although there is no provision for this referencing within the file itself. There are three kinds of optional compression available: CCITT G3, both one and two dimensional and G4 [see also TIFF-F]; Macintosh Pack Bits, a run length encoding; and Lempel-Ziv & Welch, an adaptive compression scheme. TIFF will probably be extended to incorporate JPEG compression in the near future. There is no provision for an alpha-channel as such, but one could be stored for private interchange as additional RGB planes or with a new private tag.

Drawings

IGES

The Initial Graphics Exchange Specification (IGES) is a U.S. file standard for the interchange of 2-D and 3-D CAD/CAM documents [Smith88, Mayer87]. The format is record-oriented, with three encodings: fixed card-image, free-form ASCII, and binary. The file is broken down into several sections, the two significant parts being the entity[3] declaration and entity definition sections. Thus, there is overhead, which works against small drawings, and limits to the number of elements in a drawing. There is no mechanism for referencing external files, but there are several ways of referring to and reusing internal entities. The list of primitives and their attributes is very rich--so rich in fact, that interchange is difficult because few CAD/CAM systems understand all primitives. Work is progressing on a new standard to replace IGES, called Product Data Exchange Standard/Standard for the Exchange of Product Model Data (PDES/STEP) [Wilson89, Owen87, Wilson87].

CGM

Computer Graphics Metafile (CGM) is an international standard for the interchange of drawings and images [Arnold88]. It supports the typical range of 2-D drawing primitives and supports color, multi-fonted text, and rasters. Each file may encode multiple pictures, each of which is independent from the others, permitting easy access to each of the images. References to sub-images is not possible, nor are references to other CGM files. The file format is based on an encoded representation of a sequence of drawing operators. The drawing operators are clustered into pictures, and the pictures into metafiles. There are three defined encodings: binary, which is the easiest to parse and most compact; clear text, which is human readable, self-delimiting, and highly transportable; and ASCII, which is hard to decode, hard to read, but is self-delimiting and reasonably compact. It is also acceptable to define private encodings for use among applications. Additionally, there is a mechanism for private application data, user messages, and an escape mechanism for adding functionality. The addition of 3-D primitives is being considered.

[3] An entity is a geometric shape such as a line or surface, an annotation, a structure, or a macro.

NAPLPS

The North American Presentation Level Protocol Syntax was designed to support low-bandwidth picture information for use in Videotex and Teletext applications [Ninke85]. Its data format was optimized to transmit picture elements (polygons, lines, text) in as few bytes as possible and as a superset of the ASCII standard. This means plain text ASCII files are NAPLPS datastreams, but not vice-versa. Nevertheless, it is a reasonable encoding of a 2-D color drawing. Its main limitation is that information providers are supposed to be aware of the ''minimum configuration'' (called the Service Reference Model, or SRM) which restricts the number of points in the image and the number of colors in the palette. Thus, while the format is capable of fairly rich drawings, most NAPLPS files have been constructed to rely on only the subset of facilities available on the minimum configuration. It is possible to download character sets, patterns, and macros (sequences of drawing commands) for reuse in the picture. There is also no high-level information about the image being stored, nor are there provisions for private extensions, references to other files, or references to parts of the drawing being defined. There is a provision for transmitting rasters, however.

PostScript

PostScript[4] is a page description language developed by Adobe Systems Incorporated [Adobe90]. It is a stack-based language [Loeliger81] that has powerful two-dimensional page markup primitives and excellent font support. Beyond the language itself, Adobe has established the PostScript Document Structuring Conventions (DSC) and the Encapsulated PostScript File Format (EPSF). The DSC organizes the file in terms of its prolog, page descriptions, and trailer information which is used by a document manager. It also explicitly states any resources needed by the document and allows for intelligent spool management (skipping downloading of resources if the document manager knows that they already exist in the printer). The DSC also allows other files and documents to be imported, and makes it possible to refer to pages, objects, and resources in the document (although DSC files cannot refer to these objects themselves). However, much of the overhead is required, making it difficult to encode small documents, but there is no limit to the length of a document. EPSF documents are subsets of DSC documents that define a single page image, intended to be imported into another document for reuse. PostScript is an output option only for most applications, although some applications (Interview's idraw application from the MIT X Windows distribution and Adobe Illustrator 3.0) use it as a storage medium as well. Of course, most page layout systems allow the inclusion of EPSF documents, but for printing only.

Work is progressing on the definition of an Editable PostScript; that is, an extension to the DSC that organizes the PostScript page descriptions themselves into nested objects which can be intelligently edited. These ''out of band'' comments are necessary because it is mathematically impossible to predict what a PostScript program will do. The additional comments form a ''contract'' which specifies what the enclosed PostScript code accomplishes.

[4] And Display PostScript, which is an extension to PostScript to support user interaction.

Continuous Time Formats

Audio

CD & DAT

Compact Disc digital audio and Digital Audio Tape are interchange standards for the transmission of audio samples. The samples are organized into tracks, with indices to points within a track. There is no way for a sample to reference another sample, either within the file or in another file. The audio information is sampled at 44.125 KHz (48 KHz optionally for DAT), 16 bits/sample, left and right channels, for a bandwidth of about 150 Kbytes/second, and no latency (the data must arrive on time). There is no compression. However, there is room for additional data to be stored along with the audio samples. These so-called sub-codes have been put to use in two ways: as storage for MIDI note information; and as crude, limited color graphics and text (CD+GRAPHICS).

Speech

There is no ''standard'' format for speech information but informally, systems use the following parameters: a sample rate of 8KHz, 8 bits/sample, single channel or *monophonic*, for a bandwidth of about 8Kbytes/second, which is what a voice-grade telephone line provides [Cater83]. A 250 millisecond delay is tolerable as long as it is not frequent, and occasional missed sample segments can be replaced with white noise as long as they are not too long or too frequent. Also, it is possible to perform various kinds of compression, due to the fact that the waveform signal changes relatively slowly. For example, a 4-bit delta encoding is usually sufficient.

Music

MIDI

The Musical Instrument Digital Interface (MIDI) is a one-way serial protocol for sending a musician's gestures, such as key presses and breath control, from one device (e.g., a keyboard) to another (e.g., a synthesizer) [MIDI1.0]. The serial data is sent 8 bits at a time at a data rate of 31.25 kbits/second. The format of the data is a command word followed by a varying number of operands. There is a mechanism which allows the command word to be elided if it would be repeated, for rapid continuous changes. The data is sent in real time, allowing a bandwidth of up to 3k gestures/second. A latency of more than 2 msec is unacceptable to some musicians, and MIDI is not capable of phase-level synchronization [Loy85]. To convert MIDI to a data format, it is customary to transcribe each command into a file, along with the delta in time between two commands inserted between the two commands. This allows faithful playback of the MIDI datastream. It is not possible for the MIDI datastream to refer to itself nor to another datastream, but it is possible to merge two datastreams in time. There are systems which extract a score from the MIDI datastream, once the keys, tempos, and time signatures have been specified. This score can either be linked to the datastream, or can be used to resynthesize a new datastream.

MIDI does not have enough bandwidth for some applications--for example those that amplify and compound the gestural input of the user, so there is work on a superset of MIDI with a much wider bandwidth called MIDI-2.

Motion Pictures

DVI

Digital Video Interactive (DVI) is a proprietary video and still compression medium available from Intel Inc. and IBM Corp. [Ripley89, Glass89, Loveria90] It was designed to allow the storage of low-quality video, high-resolution stills, and variable quality audio onto CD-ROMs. Proprietary lossy three-dimensional (intraframe and interframe) compression of full-motion video (30 frames/second, 256x240 pixel frames, 8 bits of luminance and 1 bit of sub-sampled chroma information) reduces the data rate to 1.5Mbits/second, which is the data rate of CD-ROM drives. There is no latency in CD-ROM drives, and DVI does not allow any latency. Because a DVI file can be broken up on a magnetic hard drive, the seek time between blocks can cause the image to pause noticeably unless care is taken to properly buffer the data. The DVI format cannot reference other files nor itself, but an application can jump to any point in a video sequence where an absolute frame (a frame coded two-dimensionally, or intraframe only) exists--it is the DVI file author's responsibility to ensure that adequate entry points exist (although the DVI algorithm will automatically insert absolute frames when it would be more space efficient than the interframe coding). Interestingly, the compression is asymmetric: it requires more time to compress the image than to decompress it. This allows the ''player'' machines to be inexpensive. There are centralized facilities with compression engines which use parallel processing to compress source video for the application developer [Tinker89].

MPEG

The Motion Picture Experts Group (MPEG) motion picture compression scheme is being developed as an international standard. Like JPEG, it will also use DCT compression, but will add interframe compression to the intraframe compression of JPEG. The committee is aiming for a compression ratio of 100 to 1, which will allow full-motion TV-quality video (720x576 pixels/frame, 30fps) and CD-quality audio to be packed into a 1.5Mbits/second bandwidth channel (e.g. CD-ROM). Intel is involved in the MPEG committee and promises to add the MPEG algorithms to its DVI system [LeGall91, Baran90].

Compositional Formats

IFF

Electronic Arts' Interchange File Format (IFF) is an interchange format in use on Commodore Amiga and Apple Macintosh personal computers.[5] IFF files are hierarchically structured, with tagged blocks of data. There are blocks which can be used to concatenate, sequence, define shared properties, and structure other blocks. There are also media-specific blocks (text, rasters, drawings, scores, samples, etc.) which contain

[5] For economy of code, many programs also use IFF as their storage format.

an encoded representation of the media [EA-IFF85]. IFF, like TIFF, has hard limits to the maximum size of a block, and there is a small amount of required overhead. It is not possible for the file to refer to or reuse any part of itself, nor is it possible to point to another file. However, the structure was designed to be parsable without knowledge of the media types, so it is possible to manipulate and point to parts of the file without corrupting the unknown media types. There is a central registry for adding new media types, which helps prevent name space collision.

ATK

The Andrew Toolkit (ATK), as previously mentioned, is a storage format with a grammar structure. However, in addition to formatted text, container objects (like the text, spreadsheet, drawing, and layout objects) allow the placement of *arbitrary* media types within their datastream, since the protocol for dealing with nested objects (*insets* in ATK parlance) is well defined. This nesting can be arbitrarily deep, as long as ''container'' objects are used [Palay88]. Some objects, like rasters and buttons, do not allow insets. Adding new media types occurs in an *ad hoc* fashion, with no mechanism for preventing name collisions. It is possible to refer to other objects within the file (which allows for multiple views on shared data), but currently it is not possible to refer to objects in another file (only to the whole file).

ODA

In addition to providing for the organization and layout of text, ODA allows other media (currently CCITT Group 4 compressed bi-level rasters and CGM drawings) to be placed at the paragraph level. There is no provision for arbitrary nesting of media types [Brown89]. New media types can be added either via private extensions, or through the international standardization process (tables and color rasters are proceeding through the latter process). It seems possible to cause an embedded media object to appear in multiple locations in a document, but not to change the view of the underlying data.

SGML

Once again, Standard Generalized Markup Language (SGML) follows the ODA lead of allowing CGM datastreams to be included in an SGML document, but also adds support for spreadsheets and mathematical notations. SGML also permits the inclusion of tagged binary data, where the tag would be interpreted privately among applications, but allow the SGML document to be correctly scanned. SGML does provide the ability to identify elements for reuse, but permits only whole file reference [ISO-8879, ISO-8879/A1].

Conclusions

Any multimedia system needs the ability to interchange and store different media types. Many of the data formats surveyed were designed for, or are suitable for interchange. Some of the formats, particularly the compressed formats such as JPEG and DVI, are also suitable for storage and may be an application's *only* storage option, due to space and bandwidth limitations. Unfortunately, very few formats provide access to portions of a document or substructure. Also, without a uniform representation, maintaining a mixed-media collection can be difficult.

Access to Substructure

In order to encourage reuse, particularly of copyrighted and voluminous data, it is important for authors to be able to select exactly what they need, and nothing more. Since many systems allow a document to be broken up into several files, one poor substitute would be to fragment all datastreams into ''interesting'' components, and reassemble them via file inclusion to resemble the original document. This would allow authors to reference the ''interesting'' pieces only. This scheme is insufficient for two reasons: the notion of what is interesting cannot be precomputed; and the ability to recover the context of the referenced piece is lost, since the piece has no knowledge of the larger whole of which it is a part. Further research is needed into reliable and uniform methods for accessing fragments of a medium.

Mixing Media

Using standard data formats for storage requires that the application know how to parse, manage, and rewrite many formats. Due to the diversity of the formats, the software overhead could prove unwieldy. It may be best to synthesize a new extensible format which has the power to express the data and constructs of the other formats, and uses filter functions to preserve the difficult-to-replicate data formats (such as JPEG compression). PostScript Level 2 has the beginnings of such a scheme.

Use in an Information Management System

Ultimately, all multimedia documents should be available under a single information management system (two such systems are Alexandria [Palay90] and IDEX [Ritchie89]). Such a system would have powerful database technology to support the structuring and compositing of arbitrary media. Standards like ODA, which define the structure of the information, and then an encoding, will migrate easily to such a system, since the database would then be just an alternate encoding. However, it will still be necessary to provide interchange functionality with older systems.

References

[Adobe90] Adobe Systems Inc. *PostScript Language Reference Manual, 2^{e}.* Addison-Wesley Publishing Company, Inc., New York, 1985.

[Arnold88] Arnold, David B. and Bono, Peter R. *CGM and CGI: Metafile and Interface Standards for Computer Graphics.* Springer-Verlag, New York, 1988.

[Alexander90] Alexander, Michael. "Data storage: grace under pressure," *Computerworld* vol.24 no.21; May 21, 1990; p17.

[Baran90] Baran, Nick. "Putting the Squeeze on Graphics," *BYTE*, Dec. 1990, pp. 289-294.

[Borenstein90] Borenstein, Nathaniel S. *Multimedia Applications Development with the Andrew Toolkit.* Prentice Hall, Englewood Cliffs., New Jersey, 1990.

[Brown89] Brown, H. "Standards for structured documents," *Computer Journal*; vol.32, no.6; December 1989; pp. 505-514

[Cater83] Cater, John P. *Electronically Speaking: Computer Speech Generation.* Howard W. Sams & Co., Inc., Indianapolis, Indiana, 1983.

[CCITT-88-VII.3]
CCITT. *Blue Book*, vol. VII.3, recommendations T.4 (Group 3) and T.6 (Group 4), 1988. ISBN 92-61-03611-2.

[Chen84] Chen, Wen-Hsiung and Pratt, William K. "Scene Adaptive Coder," in *Visual Communications Systems*, eds. Netravali and Prasada, IEEE Press, Inc. New York, 1989. pp. 463-478.

[EA-IFF85] Electronic Arts. *"EA IFF 85" Standard for Interchange Format Files.* Electronic Arts, January, 1985.

[Glass91] Glass, L.B. "Fax Facts," *BYTE*; vol.16, no.2; February 1991; pp. 301-308.

[Glass89] Glass, L.B. "Digital Video Interactive," *BYTE*; vol.14, no.5; May 1989; pp. 283-289

[Graef89] Graef, G.L. "Graphics Formats," *BYTE*; vol.14, no.9; Sept. 1989; pp. 305-306, 308-310

[ISO-8613] *Information Processing Systems - Text and Office Systems - Office Document Architecture (ODA).* International Organization for Standardization, ISO 1988.

[ISO-8824] ISO/TC 97. *Information Processing Systems - Open Systems Interconnection - Specification of Abstract Syntax Notation One (ASN.1).* International Organization for Standardization, ISO 8824:1987(E).

[ISO-8825] ISO/TC 97. *Information Processing Systems - Open Systems Interconnection - Specification of Basic Encoding Rules for Abstract Syntax Notation One (ASN.1).* International Organization for Standardization, ISO 8825:1987(E).

[ISO-8879] ISO/TC 97. *Information Processing Systems - Text and Office Systems - Standard Generalized Markup Language (SGML).* International Organization for Standardization, ISO 8879:1986(E).

[ISO-8879/A1] ISO/TC 97. *Information Processing Systems - Text and Office Systems - Standard Generalized Markup Language (SGML): Amendment 1.* International Organization for Standardization, ISO 8879:1986/A1:1988(E).

[ISO-10179] *Information Processing Systems - Text Composition - Document Style, Semantics and Specification Language.* International Organization for Standardization, ISO 10179:1989.

[JPEG90] Joint Photographic Expert Group (JPEG). *Revision 8 of the JPEG Technical Specification*, ISO/IEC JTC1/SC2/WG8, CCITT SGVIII, August 14, 1990.

[Lamport86] Lamport, Leslie. *LATEX: A Document Preparation System.* Addison-Wesley Publishing Company, Inc., Reading, MA, 1986.

[LeGall91] Le Gall, Didier. "MPEG: A Video Compression Standard for Multimedia Applications," *Communications of the ACM*; vol.34, no.4; April 1991; pp. 46-58.

[Lippman89] Lippman, Andrew and Butera, William. "Coding Image Sequences for Interactive Retrieval," *Communications of the ACM*, vol. 32, no. 7, July 1989; pp. 852-860.

[Loeliger81] Loeliger, R. G. *Threaded Interpretive Languages - Their Design and Implementation,* Byte Books, Peterborough, NH, 1981.

[Loveria90] Loveria, Greg and Kinstler, Don. "Multimedia: DVI Arrives," *BYTE IBM Special Edition*; Fall 1990; pp. 105-108.

[Loy85] Loy, Gareth. "Musicians Make a Standard: The MIDI Phenomenon," *Computer Music Journal*, vol. 9, no. 4, Winter 1985.

[Mayer87] Mayer, Ralph J. "IGES." *Byte* vol.12 no.6; June 1987; pp. 209-214.

[MIDI1.0] MIDI Manufacturers Association & Japan MIDI Standards Committee. *MIDI 1.0 Detailed Specification (document version 4.1, Jan. 89).* The International MIDI Association, Los Angeles, 1989.

[Microsoft90] Microsoft Corporation. "Rich Text Format," *Microsoft Word Technical Reference: for Windows and OS/2.* Microsoft Press, Redmond, Washington. ch. 10, pp. 381-409.

[Ninke85] Ninke, William H. "Design Considerations of NAPLPS, the Data Syntax for VIDEOTEX and TELETEX in North America," in *Visual Communications Systems*, eds. Netravali and Prasada, IEEE Press, Inc. New York, 1989. pp. 408-421.

[Palay90] Palay, A., Anderson D., Horowitz, M., and McInerny, M. *The Alexandria Project: In Search of a Unified Environment for Information Access and Management.* Information Technology Center, Carnegie Mellon University, Pittsburgh, PA. In preparation.

[Palay88] Palay, A. J., et al. "The Andrew Toolkit - An Overview", *Proceedings of the USENIX Winter Conference.* Dallas, February, 1988. pp. 9-21.

[Reid80] Reid, Brian K. and Walker, Janet H. *SCRIBE - Introductory User's Manual, 3e.* Unilogic, Ltd., Pittsburgh, PA, 1980.

[Owen87] Owen, Jon and Bloor, M. Susan. "Neutral Formats for Product Data Exchange: the Current Situation," *Computer-Aided Design* vol.19 no.8; Oct. 1987; pp. 436-443.

[Ripley89] Ripley, G.D. "DVI--a Digital Multimedia Technology," *Communications of the ACM*; vol.32, no.7; July 1989; pp. 811-822.

[Ritchie89] Ritchie, I. "HYPERTEXT--Moving Towards Large Volumes," *The Computer Journal*, vol. 32, no. 6, 1989; pp. 516-523.

[Rosenberg91] Rosenberg, Jonathan, *et al. Multi-media Document Translation: ODA and the EXPRES Project.* Springer-Verlag, New York, 1991.

[Rose90] Rose, Marshall T. "Abstract Syntax," *The Open Book: A Practical Perspective on OSI.* Prentice Hall, Englewood Cliffs, New Jersey, 1990; ch. 8, pp. 225-335.

[Rosenthal90] Rosenthal, Steve. "JPEG emerging as standard for compressing image files; compression chip price decline predicted," *MacWEEK* vol.4 no.12; March 27, 1990; pp. 31-32.

[Smith88] Smith, Bradford, *et al. Initial graphics exchange specification (IGES), version 4.0* SP-767 (Society of Automotive Engineers) .

[TIFF5.0] *TIFF Revision 5.0 Specification.* Aldus Corporation, Seattle, Washington, 1988.

[TIFF-F] *TIFF Class F Specification.* Cygnet Technologies, Berkeley, California, 1990.

[Tinker89] Tinker, M. "DVI Parallel Image Compression," *Communications of the ACM*; vol.32, no.7; July 1989; pp. 844-851.

[Wallace91] Wallace, Gregory K. "The JPEG Still Picture Compression Standard," *Communications of the ACM*; vol.34, no.4; April 1991; pp. 30-44.

[Warner90] Warner, William C. "Compression Algorithms Reduce Digitized Images to Manageable Size," *EDN* June 21, 1990; pp. 203-212.

[Webster90] Webster, Bruce F. "The Color Squeeze: Image Processing Demands a Lot of Resources, but Help may be On the Way," *Macworld* vol.7 no.4; April 1990; pp. 81-86.

[Wilson89] Wilson, Peter R. "PDES STEPs forward," *IEEE Computer Graphics and Applications* vol.9 no.2; March 1989; pp. 79-80.

[Wilson87] Wilson, Peter R. "A Short History of CAD Data Transfer Standards," *IEEE Computer Graphics and Applications* vol.7 no.6; June 1987; pp. 64-67.