GXF — the General eXchange Format

Bob Edge
Grass Valley Group

The General eXchange Format was originally conceived by Grass Valley Group for the interchange of simple camera shots over data networks, and for archival storage on data tape. Over the years it has evolved to include additional compression types, and support for compositions with video cuts, audio fades and metadata. It is now an SMPTE standard.

About five years ago, Grass Valley Group (GVG) introduced a file format for the interchange of simple camera shots on data networks, and for archival storage on data tape. This format is known as the General eXchange Format (GXF). As GVG's products evolved, new capabilities and features were needed. These included additional compression types, support for compositions with video cuts, audio fades and user data (metadata).

GXF now enjoys widespread acceptance. At the request of several customers and vendors, GVG submitted a technical specification for this format to the SMPTE, and it is now a standard (SMPTE 360M [1]).

Other proprietary exchange formats are in use and new formats are emerging. Some of these formats have capabilities that are critical in specific applications. An example is the Advanced Authoring Format (AAF) – a format designed for editing and post-production applications [2]. AAF describes a composition that includes content, transitions, effects and metadata that go into making a finished product. Work is underway in trade associations and standards bodies on additional formats.

How current formats and new formats will interoperate is an interesting topic. New formats may offer advantages. However, the value of new capabilities must be weighed against the cost of change. It is uncertain if the new formats will offer all of the capabilities and features found in existing formats. How all of these formats will interoperate with existing devices is another critical concern for end users. File-format conversion devices, and servers that support multiple interchange formats, are potential solutions.

Abbreviations

AAF	Advanced Authoring Format
API	Application Programming Interface
ASCII	American Standard Code for Information Interchange
ATM	Asynchronous Transfer Mode
EDL	Edit Decision List
FTP	File Transfer Protocol
GoP	Group of Pictures
GXF	General eXchange Format
HD	High-Definition
JPEG	Joint Photographic Experts Group
KLV	(SMPTE) Key Length Value
MPEG	Moving Picture Experts Group
MXF	(Pro-MPEG) Material eXchange Format
SDI	Serial Digital Interface
SMPTE	Society of Motion Picture and Television Engineers (USA)
XML	Extensible Markup Language

File formats and data networks will not replace a facility's analogue or SDI plant, as they do not offer real-time low-latency operation, nor do they allow seamless switching in the stream. These are key features for live production and some parts of the on-air material chain.

The diversity of end-user requirements will result in multiple formats in many facilities.

The development and evolution of GXF (SMPTE 360M)

The General eXchange Format (GXF or SMPTE 360M [1][3]) was originally developed to transport compressed video on Fibre Channel local-area networks and for archival storage on data tape. The advantages of data networks include:

- transfers that are faster than real-time;
- transfers that are less than real-time, utilizing low-cost networks;
- no generation quality loss from decompression-compression cycles.

These transfers are accomplished with standard protocols. The File Transfer Protocol (FTP) [4] guarantees bit-perfect exchanges of material. This reduces the need for repeated Quality Assurance checks in a facility's operation. These capabilities can improve a facility's productivity.

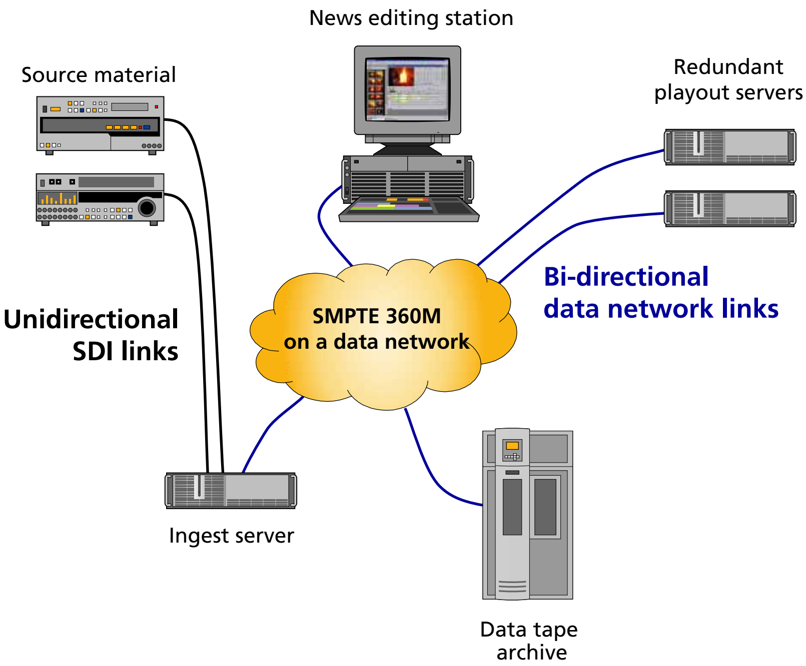

Figure 1
Data networks and SMPTE 360M material movements

First implementation

The first implementation of GXF supported JPEG video and uncompressed audio. The material was limited to simple shots with provisions for transporting "heads" and "tails". A key design objective was to enable viewing of the content as the file was received. This has been accomplished with a multiplex of the audio and video packets. However, the designers chose to keep the audio and video packets as logically separate streams – a choice which has several advantages. The streams are multiplexed at the network packet level. An advantage of network-packet multiplexing is the ease of incorporating multiple video and audio tracks into a single composition. This has been used when multiple-language tracks are used with a single video track. It is also used to combine low-resolution and high-quality video in a single file.

This composite stream is the core of a GXF file. In addition, the audio/video multiplex is preceded by a collection of packets containing the information needed to describe and reconstruct a server's internal files. Examples of this information are file names and mark in/out points for the content. Between the basic file description information and the audio/video multiplex is a coarse-grained frame lookup table. This table was designed to locate content on data tape during partial file-retrieve operations. An end-of-stream packet terminates the file.

These basic components are the foundation of the current GXF format. Enhancements have been made since the original design; however, all of these have been made as extensions, not modifications. The format has stood the test of time as technology has evolved.

Enhancements and extensions

A later release of GXF included MPEG compression and a rich composition capability. The MPEG video was transported as MPEG Elementary Stream (ES) packets with long GoP or I-frame-only coding. A compound composition includes cuts for edits with audio fades to eliminate pops and clicks at the transitions. The additional information needed to support an EDL is sent as a collection of packets, early in the file.

Another interesting issue is managing the inactive material in a compound composition. An application may or may not need this material. In a GXF compound composition, the heads and tails are sent after the active content. This has preserved the goal of previewing a stream that is being transferred, and maintains the integrity of the original collection of tracks. The originating device sends "handles" if they are requested.

Later additions to GXF include DV-based video, compressed audio, HD frame rates and other features.

Acceptance and standardization

Several vendors currently support SMPTE 360M (GXF). Applications include the transfer of material from News systems for on-air playout ... staging material between servers and operational archives ... spot distribution ... and stream/file format converters.

In 1999, several users and vendors asked GVG to disclose the technical details of GXF. The GXF design was then submitted to the SMPTE for standardization. The resulting standard included the functionality that was available at the time of the first letter ballot. A revision is now under way which describes all of the current functionality.

Concepts that worked well

As with all projects, some ideas and concepts were more successful than others. The next few sections describe some of the design concepts.

An interchange format

GXF was designed as an interchange format for data networks and for archival storage. A GXF file image does not exist on a Profile video file server. The sending server constructs a GXF file during a transfer. The receiving server ingests the GXF file and converts it to the appropriate internal format.

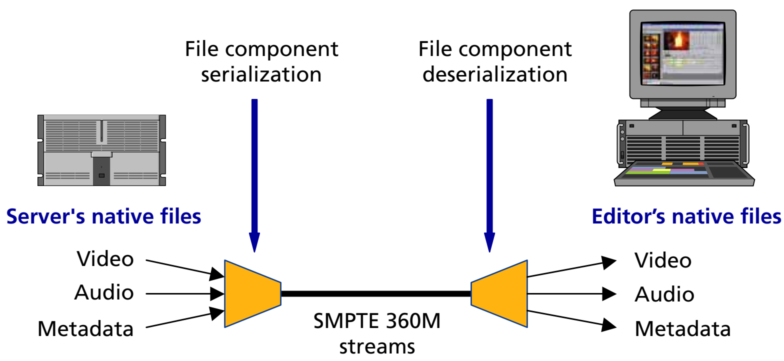

Figure 2
An interchange format

Freeing the interchange format from the internal design requirements of servers, editing systems and archives has proved to be a very good choice. Had we gone for a complicated relationship between the interchange format and the internal architectural requirements of various products, we would have lessened the chances of the format being accepted – and made the implementation more complicated for manufacturers.

Design trade-offs were made to meet each application's requirements. In some cases, clever solutions could solve the needs of multiple applications. In other cases, unique solutions were required.

Multiplexing and latency

Editing, voice-overs and other operations that are very useful in the production environment are facilitated by keeping the audio and video as separated track files on a server. This is not ideal for an interchange format. The GXF designers recognized this fact and chose to use a coarse-grained audio/video multiplex to simplify GXF and to support playout during transfer. This also enables transfers while a feed is being recorded.

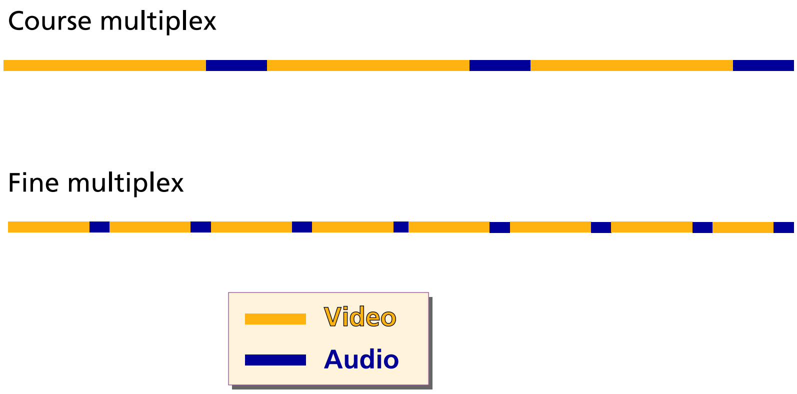

Figure 3
Multiplexing and latency

GXF was not intended to replace analogue or SDI video interfaces. Analogue – and some SDI – systems transfer the audio and video with separate concurrent streams. Even SDI with embedded audio presents the video and audio with only a few video lines of latency between the video and audio. This is a critical feature for live programmes and on-air production. The multiplex for GXF files is approximately one second long. With buffering, the practical latency for a multiplex is several seconds. This is too long for live production but it works well for the intended application.

Standard data networking protocols

During the early GXF effort, specialized protocols were considered to optimize the network's effective payload bandwidth. It was soon recognized that using standard FTP would greatly enhance the interoperability. Several factors drove this conclusion. One was the availability of FTP on PCs and workstations. Another was the availability of IP-based local and wide-area networks.

An interesting issue was the support for partial file retrieval, using a standard file transfer protocol. The in/out marks for the partial retrieval should be sent from the requesting device to an archive server. It is desirable to cut the material on the archive server so the unwanted content does not cross the network. In addition, the in/out points can be used to minimize tape motion. Data tape drives can usually search at sixty times their normal read speeds.

To implement these optimizations, the requesting device must share in/out marks with the archive server. The implementers of GXF chose to pass this information as part of the FTP file-name string. This enables an efficient partial retrieval without requiring any new protocol-level commands.

Interoperability and compression types

GXF supports multiple compression types. Video material can be compressed as a stream of JPEG fields, DV-based frames or MPEG frames (including long GoP and I-frame-only). The basic goal of the format is to interchange material between devices. One implication is that each compression family should use a single stream type.

For MPEG material, Elementary Streams (ES) was chosen as the common format. It is relatively simple to convert MPEG Programme Streams and Transport Streams to Elementary Streams. In addition, most of the professional-quality MPEG codecs use MPEG-ES. Finally, the editing of MPEG-ES is relatively simple. Having only one format for each compression type, rather than trying to support all formats (for example MPEG ES, PES, PS and TS) increases the likelihood of interoperability. Selecting ES as the lowest common denominator was a good choice.

Frame lookup tables for archives

Support for data tape-based archives is one application for GXF files. In order to implement partial file retrieval efficiently, we need information that describes what part of a file can be skipped. To meet this requirement, GXF files have a coarse-grained frame lookup table. Each lookup table entry represents $^{1}/_{1000}$ th of the file's playout duration. Normally the frame lookup table is at the beginning of the file. An exception occurs when a file is being transferred before the original recording is complete. In this case, the frame lookup table is at the end of the file.

The frame lookup table is of a fixed size, which simplifies its construction and management. When a partial file retrieval from an archive occurs, it is undesirable to force the construction of a new frame lookup table as this increases the complexity and latency of that operation. In this case, a frame lookup table is omitted. Since partial retrievals almost never send material to another archive, this is not a limitation.

Frame lookup tables for servers

Within a server or an editing system, a complete frame lookup table is required for effective operations. These are used to perform the cuts and transitions specified by an EDL. When a device receives a GXF file, an internal frame lookup table is constructed. This technique has several advantages.

The formats for internal frame lookup tables vary from server to server. The internal lookup table is usually optimized for each architecture and application. Editing devices usually manage the tracks as individual continuous files. Store-and-forward devices may keep the content in a multiplex. Archives usually store the material without any alterations; in fact, many archive systems make a point of not looking inside a file. There is no index table design that will meet all of these goals.

By encoding information about each frame in its packet header, the difficulty of building a server-specific frame lookup table was minimized. The frame type for MPEG frames (I, B or P), the frame's packet size, its track identifier and its timeline location are all part of a packet header. The receiver only needs to examine packet headers to construct an index table. This can be accomplished during the transfer process, which is usually network bandwidth limited.

Lessons learnt

As with all established systems, the GXF developers have learnt some lessons. None of these has seriously impacted on the effectiveness or usage of the format.

Complexity usually grows exponentially with the number of features, although some features only add linearly to the overall design cost. In many cases, the features interact – sometimes in unexpected ways. In general, more features mean more complexity, increased implementation costs, increased product prices and other difficulties – frequently reducing the possibility of interoperability.

Simplicity is good; less is more!

Stability for archives

Whenever material goes into an archive, it must be retrievable. If the archive contains files that do not meet the current specifications or standards, they establish a new de facto standard. All of these must be supported. The potential for a proliferation of format variants could become a serious issue.

Great care must be taken to verify the quality and robustness of these designs.

Encoding schemes

The encoding of packet headers, frame lookup tables and other file description information in a binary format can improve the encoding and decoding efficiency. Binary and ASCII schemes can be used to encode descriptive metadata, EDLs and other information. Both have advantages:

- **Binary encoding** offers some efficiency advantages. Scalability and extensibility are sometimes difficult with binary coding. Binary formats require carefully drawn-up specifications, and registration of all the encoding details. SMPTE KLV is an example of a standardized binary encoding scheme.
- **ASCII encoding**, including XML, is usually easy to extend and is human-readable. This is not to say that everyone will understand exactly what the information means or whether it is correct or in error. It is simpler to deal with ASCII than binary encoding, but it is not always as simple as one would hope.

It is the author's opinion that ASCII or XML metadata and control information has more advantages than disadvantages. However, this is not always a first-order issue for a file format design. It is also important to note that binary metadata can be easily transported in the SMPTE's KLV format. Binary information requires recoding for XML-based designs.

The basic file structure is usually best coded in a binary format.

Sectoring and file interchange

The GXF designers padded all JPEG packets to 4096 byte boundaries. The idea was to simplify the construction and reception of stream packets. It also adds to the file size. GXF MPEG packets do not have sector size padding. We found that implementing variable-length packets was not a significant effort and the smaller file size is important.

An ideal sector size is surprisingly difficult to select. It changes depending on the network and disk array performance characteristics and other design issues. The result is that any specific sector size may be wrong. Because of this, systems may be optimized to values that, in fact, are not useful and, at worst, are counter-productive.

If the design goal is to build an interchange format, this is one "feature" that should be eliminated.

Compatibility between established and emerging formats

File interchange formats will evolve over time. Today, GXF enjoys widespread use by several vendors and many end-users. Literally thousands of GXF-capable machine are used every day to move content across IP-based networks (Ethernet, Fibre Channel and ATM). In addition, several large-scale GXF-based archives are in use.

As GXF evolves, changes must be made in an upward-compatible manner. This should protect the end-user's investment. As new formats emerge, linking the existing systems with new technologies will be a challenge for everyone.

GXF and AAF

The AAF Association provides a file exchange format that is intended for post-production and rich editing applications. The AAF [5][6] offering includes a reference implementation and documentation. The documentation describes an Application Programming Interface (API) that is used by developers of editing systems and other devices. The API allows one to create and manipulate AAF compositions.

AAF is designed for editing and post-production systems. The architecture is optimized to allow rich editing capabilities and features that are needed for non-linear editing. AAF is not an ideal format for the storing of finished material or for use in TV operations. Conversion of AAF compositions to an operational format is key. This conversion is the compilation step that results in a finished programme. This programme may be written to videotape or converted to a file format.

GXF was designed for, and is heavily used in, on-air operational environments. Using GXF as a source for AAF-based systems, or for compiling completed AAF composition into GXF files, is a reasonable approach. The technical challenges are not significant if both formats are being used as intended.

AAF and GXF complement each other in this environment.

GXF and MXF

MXF [7][8] is an interchange format that is under development. It is being standardized by the SMPTE with work being contributed by the EBU, Pro-MPEG, G-FORS and other organizations.

MXF is being designed to cover a broad range of material interchange applications, which is both its strength and its challenge. As the breadth of applications being addressed increases, the complexity in the format grows. This makes designing and standardizing the format more difficult.

GXF was designed to address many, but not all, of the same applications. In an ideal world, an end-user could simply purchase equipment with any desired format, and move forward. However, many users have conflicting interests, so choosing a single format may be difficult.

This leads to the idea of converting GXF to MXF files and vice versa. For simple shots with limited metadata, this is a reasonable approach. For compound compositions (ones with cuts in the audio or video stream), this may be more difficult.

For most real applications, conversion between these two formats does not appear to be an issue. For a few applications, conversion efforts may be more costly.

Format converters

A few companies are making products that are second-generation format converters. They do much more than the conversion from 525- to 625-line signals. These devices can read basic streams (MPEG Transport Streams or DV streams), standard file formats such as GXF and some of the proprietary file formats – and convert them to other formats.

As the number of file formats and compression stream types grow, these format converters will be a good solution for many real applications.

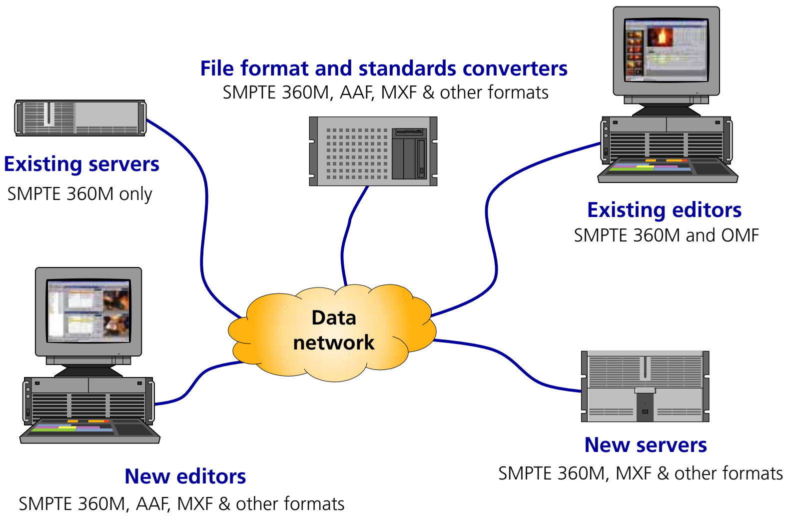

Figure 4
A multi-format facility

The future

When one embarks on a new design, a set of requirements and assumptions is established. These are frequently not correct, and they evolve over time. The most common surprise is a user's ability to use technologies in creative ways that were never envisaged by the original designers.

One format (AAF) offers a reference implementation. This implementation serves as a strong standard for implementers. The method of building a reference design, and proving it in the marketplace, is a sound way to establish solid solutions.

Some formats are in widespread use. Some of these are proprietary technologies and others are standardized. New formats will be introduced and new standards are being developed. End-users must carefully consider their own circumstances and requirements when making decisions about future facility architectures.

Conclusions

GXF is a well-established format. It is supported by more than ten vendors and is used today in many facilities for material transfers and archives. It is also used for spot distribution and is supported by file-level format converters. GXF's specification has been documented and standardized by the SMPTE.

AAF is well suited for post-production and high-end editing. GXF and AAF complement each other.

New formats, including MXF, are under development. These offer different feature sets from GXF and AAF and will be used in some applications and facilities. GXF and other established server formats will not be replaced in all facilities. Finding ways for these formats to interoperate will be critical for the broadcast industry.

The PC industry also offers technologies in this area. Apple Computer's® *QuickTime*® is an example of a format that is used by some broadcasters today. Microsoft® also offers technology in this area and other PC-based technologies will probably be available in the future.

Selecting the formats that match a facility's workflow and other requirements will be a real challenge in the future.

Acknowledgements

The author would like to thank Ray Baldock, of Grass Valley Group, for his valuable contribution to this work.

Bibliography

[1] SMPTE 360M-2001: **General Exchange Format (GXF)**
http://www.smpte.org/.

[2] Brad Gilmer: AAF – the Advanced Authoring Format
EBU Technical Review No. 291, July 2002.

[3] Ray Baldock: **Enabling Network Interoperability between Video File Servers, The SMPTE 360M General Exchange Format (GXF)**
Proceedings of the 143rd SMPTE Technical Conference, p 103-110, November 2001
http://www.smpte.org/smpte_store/books/.

Bob Edge is a Principal Engineer for Thomson Broadcast Solutions / Grass Valley Group (GVG) and, currently, is responsible for enhancements on the Emmy® award-winning Profile line. He has a bachelor's degree in Computer Science from Oregon State University and is the author of several technical papers.

During his career, Mr Edge has worked as an Engineer, Architect, Project Leader and Program Manager on several products. Some of his early efforts included desktop computing systems, peripherals and high-performance graphics terminals. He participated in a research project that used high-bandwidth data networks to transport video streams. Following this research project, he worked on additions to GVG's Profile product line.

Recently, he lead the effort to standardize the General Exchange Format (GXF), now known as SMPTE 360M.

Bob Edge is currently active in Television standards work, including participation in the Society of Motion Picture and Television Engineers (SMPTE), a few EBU projects and the Pro-MPEG Forum. Previously he participated in the EBU/SMPTE Joint Task Force.

[4] IETF RFC 959: **File Transfer Protocol (FTP)**
http://www.ietf.org/rfc.html.

[5] Brad Gilmer: **Enabling better media workflows – An overview of the Advanced Authoring Format**
http://www.aafassociation.org/html/techinfo/.

[6] AAF web site: http://www.AAFassociation.com/.

[7] James H. Wilkinson: **An Overview of the Material Exchange Format (MXF) Specification**
Proceedings of the 143rd SMPTE Technical Conference, p 118-126, November 2001
http://www.smpte.org/smpte_store/books/.

[8] Bruce Devlin: MXF – the Material eXchange Format
EBU Technical Review No. 291, July 2002.