THE QUEEN'S AWARDS FOR ENTERPRISE 2001

BBC

R&D White Paper

WHP 097

September 2004

Broadcast media exchange for B2B applications

P.J. Brightwell

Research & Development
BRITISH BROADCASTING CORPORATION

BBC Research & Development
White Paper WHP 097

Broadcast media exchange for B2B applications

P.J. Brightwell

Abstract

Business-to-business (B2B) exchange of broadcast media files will become increasingly important in the future. This paper outlines requirements for such exchanges, describes a flexible architectural model and proposes the use of an XML document—the "Manifest"—as a means of encapsulating the information involved in an exchange transaction. Examples of the use of the Manifest and potential applications are described, together with possibilities for future work and standardisation.

This document was originally published in the Proceedings of the International Broadcasting Convention, September 2004.

White Papers are distributed freely on request.

Authorisation of the Chief Scientist is required for publication.

BROADCAST MEDIA EXCHANGE FOR B2B APPLICATIONS

P. J. Brightwell

BBC, UK

ABSTRACT

Business-to-business (B2B) exchange of broadcast media files will become increasingly important in the future. This paper outlines requirements for such exchanges, describes a flexible architectural model and proposes the use of an XML document—the "Manifest"—as a means of encapsulating the information involved in an exchange transaction. Examples of the use of the Manifest and potential applications are described, together with possibilities for future work and standardisation.

INTRODUCTION

In recent years, handling of broadcast media as computer files has become commonplace in all stages of the content lifecycle. Content is regularly transferred as files between tightly-coupled systems, in which content formats and transfer protocols are carefully controlled. However in the future, there will be a growing requirement to exchange files between disparate systems, between different organisations and over a range of types of network, including the Internet. Examples include delivery of production content between facilities houses and broadcasters, and provision of agency news feeds.

For widespread adoption of file exchange between organisations, some degree of standardisation is essential. There have been various initiatives in this area, for example by the EBU/SMPTE task force (1) and in the later work of these two bodies. The acceptance of open formats such as MXF and AAF provides part of the solution, as does the use of standard transfer protocols such as FTP and enhancements. But there is also a need to agree on standard mechanisms for organisations and systems to find out each others' capabilities, to be able to agree on the details of how and when exchanges are to occur, and to manage the lifecycle of exchanges. Such an approach will enable organisations to more easily set up a business-to-business (B2B) relationship for complex projects.

The Pro-MPEG Forum (2) has set up a working group on Media Dispatch to identify requirements for B2B exchange, and has proposed a model and protocol that could contribute towards the "missing standards". This document explains the proposals, gives examples of how they can be used in practice, and outlines they can be extended.

CONTEXT

Figure 1 presents a layered view of B2B exchange of media:

- The **Business Application Layer** represents the business systems, processes, rules and logic within the organisations that wish to exchange content. These vary widely between businesses and are thus less suitable for standardisation.
- The **Exchange Transaction Layer** offers a standard mechanism for events in the Business Application Layer to initiate an exchange, find out how the exchange is going and respond to its outcome. It decouples the layer above from dependencies of how the exchange is implemented.

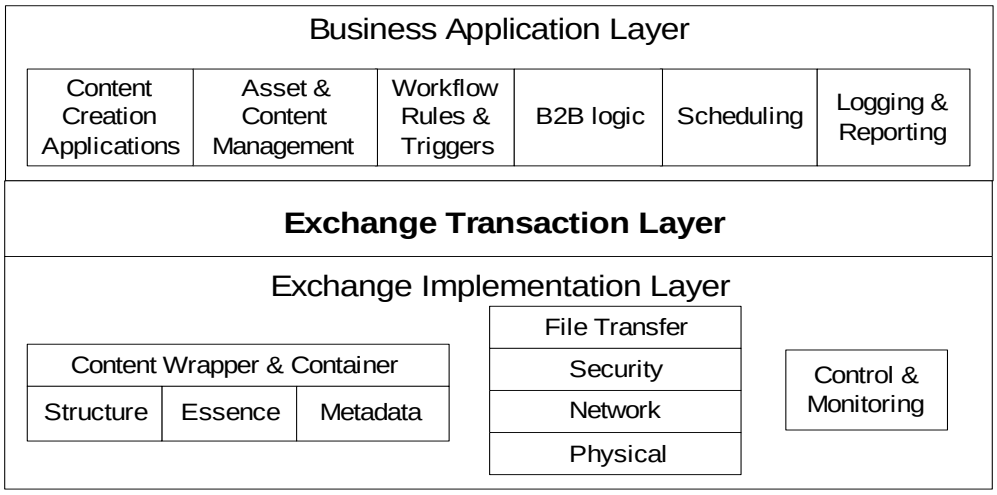

Figure 1 – B2B exchange layered architecture

- The **Exchange Implementation Layer** represents the details of the contents of the files that are transferred, and the protocols and control mechanisms used to implement the transfer. The Exchange Transaction Layer initiates and controls activities in this layer. There are a wide range of well-established and upcoming standards in this layer, for example MXF, MPEG, FTP, TCP/IP, and SNMP MIB.

This paper is concerned primarily with the function and implementation of the Exchange Transaction Layer, but also on the requirements that are made on the other layers.

REQUIREMENTS

Significant background work has already been carried out in SMPTE, EBU and Pro-MPEG on gathering requirements for networked exchange of broadcast content. The requirements relevant to the Exchange Transaction Layer can be summarised as follows:

- Usable by a wide range of applications, ranging from simple "email-like" transfer user interfaces to complex content and asset management systems with sophisticated workflow engines.
- Allows organisations to start exchanging material with minimal effort.
- Provides a standard mechanism for organisations to come to an agreement on what is going to be exchanged, when, by what mechanism, and who is responsible for managing the transfer.
- Allows different versions of media to be exchanged, for example broadcast quality and browse quality versions.
- Supports the exchange of business-specific metadata, for example news agency "dope sheets".
- Provides a standard mechanism for informing the Business Application Layer of the status of an exchange.
- Specification must be simple to understand.
- The Exchange Transaction Layer should support whatever protocols, formats and unique identification schemes that are required in the Exchange Implementation Layer to meet the functional and technical requirements for exchange.

- Specification should be extensible to support a range of different models for exchange, The initial requirements are for push-model and pull-model exchange, with synchronisation, third-party initiated and multicast as possible enhancements.
- Implementation should use standards in widespread use wherever possible, but allow extension to adopt new standards as appropriate, e.g. web services protocols.
- When used over public networks, implementation must adopt best security practice.
- Implementation should be usable with typical corporate intranet and internet architectures, e.g. where network address translation and firewalls are employed.

The Exchange Transaction Layer should also allow exchange to occur using "enhanced" transfer features, where supported by the protocols and formats used in the Exchange Implementation Layer:

- Use of adaptive protocols that aim to overcome problems with TCP/IP congestion (3,4).
- Suspending and restarting transfers, where the protocol permits.
- Partial transfer specified as a byte range, a time-code range.
- Transfer of individual audio, video or metadata components of a multi-component file.
- Exchange of large media files by "chunking" into a number of small file transfers.

TRANSACTIONS AND MANIFESTS

A **Transaction** is a formalisation of a B2B media exchange in which the Exchange Transaction Layer "wraps up" the details of the transfer of individual files to provide a loosely-coupled interface with Business Application Layer. As shown in Figure 2, this interface can be modelled as a number of services, which operate on Transactions. These allow organisations to negotiate the details of the exchange, to initiate transfers, to query the other organisation about the state of the transfers and how much of the files have been received, and to notify that the exchange has completed. In addition a Discovery service provides the Business Application Layer with information on how to access the other services.

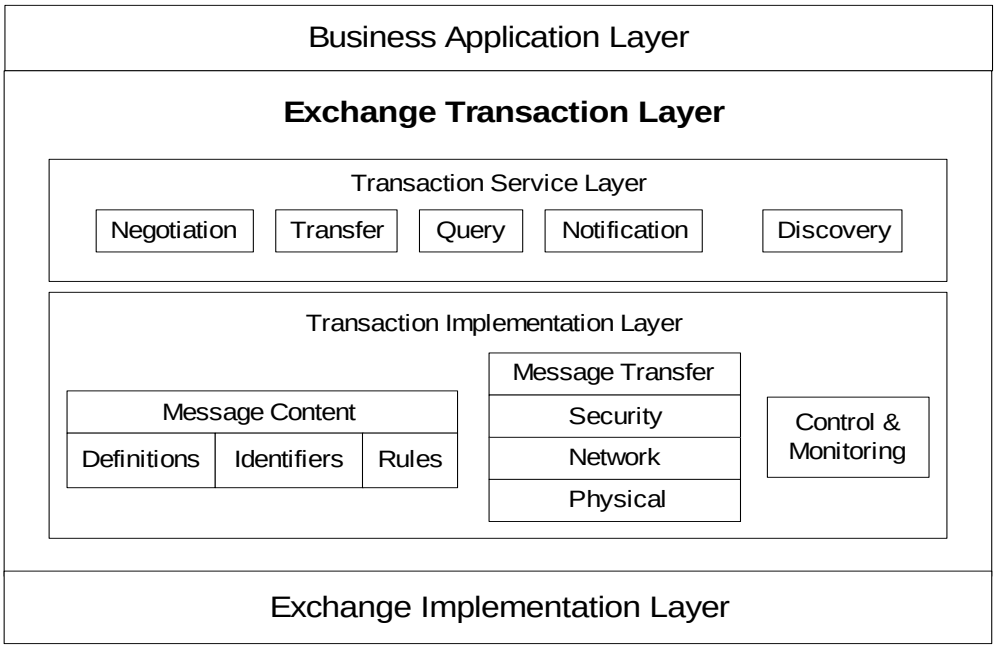

Figure 2 – Transaction services and implementation

Manifest Document

These services could be implemented in various ways, but a relatively simple mechanism has been chosen. An XML document called the **Manifest** encapsulates the data of the Transaction. This is passed between the organisations' systems using HTTP GET and POST commands, and the service implementation can be as simple as a CGI script.

The Manifest contains:

- Information about the Transaction itself – who initiated it, the date and time, etc.
- A list of potential options for transferring files, giving the protocol, available times, URIs to be used, who is responsible for initiating it, etc.
- A list of media and other files that potentially could be transferred, giving the path-name, size, when it must be transferred by, etc.
- Information about which transfer options could be used for which files.
- Unique identifiers for the Transaction, transfer options and files.
- State information – see below.
- Modification history for the document

Figure 3 shows a simplified example of a Manifest document with just one file and one transfer option.

```
<?xml version="1.0" encoding="UTF-8"?>
<manifest>
	<transactionid>123abc789</transactionid>
	<initiator>facility.com</initiator>
	<transferoption>
		<troptid>HTTPS GET from dispatch1.facility.com</troptid>
		<protocol>HTTPS GET</protocol>
		<sender>https://dispatch1.facility.com/</sender>
		<receiver>https://ingest1.broadcaster.co.uk/</receiver>
		<controller>broadcaster.co.uk</controller>
	</transferoption>
	<file>
		<fileid>20031120140000001001</fileid>
		<pathname>P0001257/prime/programme.mxf</pathname>
		<size>123456789</size>
		<mimetype>video/x-mxf</mimetype>
		<finishbefore>2004-03-15T:13:00:00</finishbefore>
		<comment>Star Wars Episode 7, 16:9, MXF OP1a</comment>
		<transferwith>
			<troptref>HTTPS GET from dispatch1.facility.com</troptref>
			<status>offered</status>
		</transferwith>
	</file>
</manifest>
```

Figure 3 – Simplified Manifest document

State information

The `<status>` fields in the Manifest carry information giving the state of the each media file in the Transaction. The possible states are: `offered`, `accepted`, `rejected`, [transfer] `in progress`, `succeeded`, `failed`. This allows the Business Application Layer to make appropriate actions. For example if a file is rejected by the other organisation because it is in an unacceptable format, a transcoding process might be initiated. Once all the files in the Manifest have reached the state `rejected`, `succeeded`, or `failed`, the Transaction is

completed.

Figure 4 shows an example of a Transaction in which the Manifest contents are represented graphically. `facility.com` offers `broadcaster.co.uk` a broadcast quality MXF file and Quick Time proxy version by means of secure FTP or HTTP. The broadcaster already has the proxy version and prefers not to use FTPS so rejects three of the four offers and accepts the HTTPS transfer of `programme.mxf`. At the agreed time, the broadcaster initiates the transfer and once it is completed notifies the other party of the outcome by sending a final version of the Manifest.

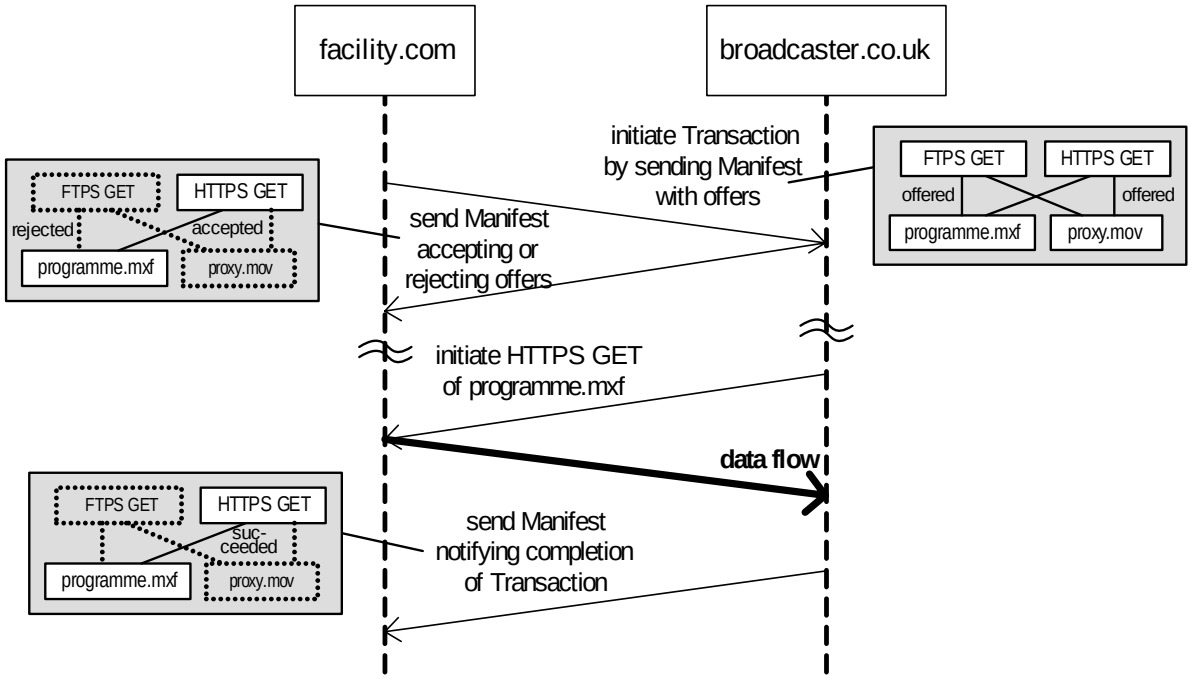

Figure 4 - A simple Transaction

In this example the Transaction is initiated by the organisation sending the media files, while the transfer itself is initiated by the receiver. In other words the Transaction is push model and the transfer pull model. The Manifest can also be used for pull model Transactions and for push model transfers.

Note that in this proposal fine-grained state information is carried within a single document representing the entire transaction. Other arrangements, such as communicating state information in separate documents are under consideration.

Metadata and unique identifiers

Several different types of metadata are likely to be encountered in media exchange:

- Metadata closely related to the essence may be wrapped with it (e.g. in an MXF file).
- Metadata may be in separate files (e.g. XML documents), often subject to a domain-specific standard or schema. Examples include rights management metadata, and metadata related to home storage.
- Metadata may be provided from an external source (e.g. a Uniform Resource Identifier (URI) might specify a web page that provides access to a database).

In the latter two cases, the Manifest can include filenames or URIs that *refer* to the metadata. It will not be used to *contain* any metadata other than in the form of comments to aid readability. The external metadata file or source may make reference to the unique identifiers for files that appear in the Manifest. The Business Application Layer will determine

the syntax and semantics for unique identifiers, so certain applications may require the use of UMIDs, while in other cases perhaps a TV-anytime CRID (5) may be more appropriate.

Discovery

Organisations cannot initiate Transactions unless they know how to contact each other at the Exchange Transaction Layer. Therefore, each organisation will provide a Discovery service that describes how and where to send Manifests. A simple approach is to provide this information via a web page at a 'standard' location within the organisation's domain (e.g. `www.broadcaster.co.uk/mediadispatch`). A more flexible and robust approach could be based on that used for web services, using Universal Description Discovery and Integration (UDDI) (6).

Security

Where exchange occurs over public networks, in particular the Internet, it is essential that all stages of the Transaction are secure, and best practice should be adopted, for instance the HTTP transfers that are used to send Manifests should employ the Transport Layer Security (TLS) protocol (7), and the Discovery service should include full end-to-end authentication with public-key certificates. Unless the media files are already encrypted, a secure transfer protocol should be used.

IMPLEMENTATIONS AND APPLICATIONS

The Manifest-based Transaction described above can be used in a variety of different situations. Figure 5 shows a simple implementation which might be used to provide a broadcaster with a media upload/download facility via its website. In this case, no assumptions can be made about the client capabilities other than that it has a web browser, and so the code to implement the Transaction services and file transfer is downloaded as an applet or browser script. Further information is available at (8).

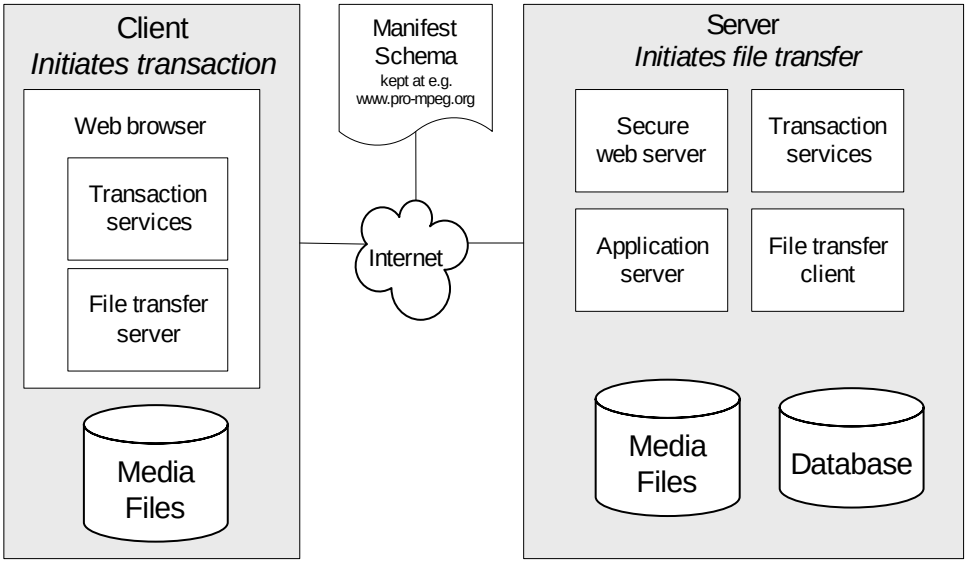

Figure 5 - Simple implementation with browser-based client

Figure 6 shows an application where two broadcasters working on a co-production exchange content. Here each organisation is likely to have its own local workflow rules implemented by an asset management system, which requests Transaction services.

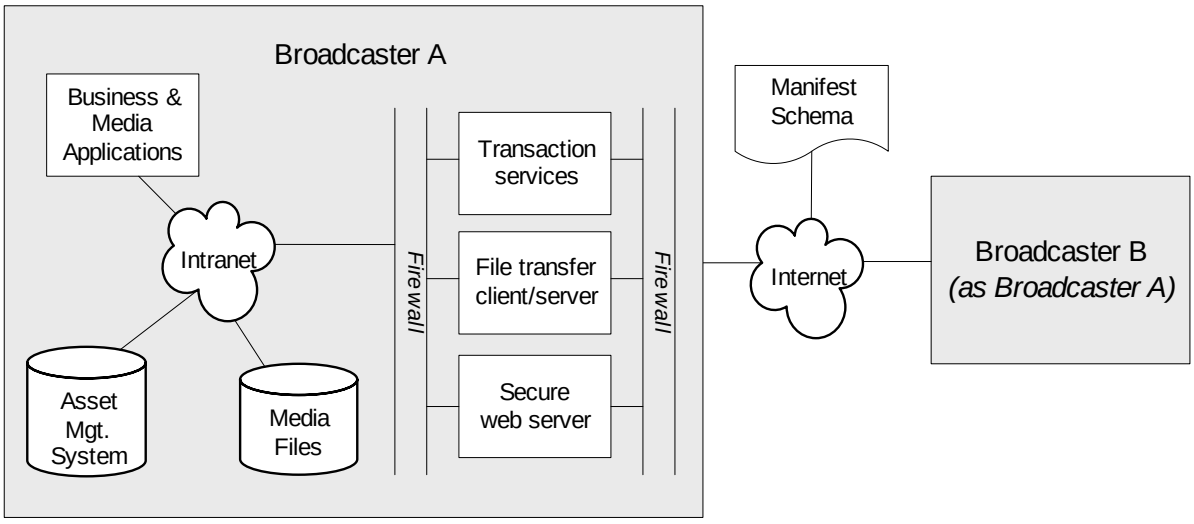

Figure 6 – Exchange between broadcasters

Variations of the above would be suitable for exchange with production facilities and news agencies. It can also be adapted to be used for exchange between different departments of an organisation over a corporate intranet, for example between a production workgroup and a playout area.

FUTURE WORK

The BBC is investigating the use of these proposals in trials of file transfer of media between its broadcast centres. Other trial implementations are also in discussion. This work will likely reveal additional requirements, which will lead to refinement of the proposals, and a more formal specification, which should include a W3C Schema for the Manifest (9). This will require that transfer protocols, file formats, unique identification schemes, and other items in the Manifest can be identified using registered values, e.g. from the SMPTE dictionary (10).

Further work will address the remaining requirements stated earlier, such as support for Transactions initiated by a third party, and extending the Manifest to include information about partial file transfer. This latter example raises the issue of whether a time-code range should be specified in a format-independent way.

In addition to specification work, easily available tools will be essential for the adoption of a protocol such as that outlined here, and development of an open-source SDK would be greatly beneficial. Such tools should be able to identify standard file formats such as AAF and MXF. They could provide a straightforward method of integrating the Exchange Transaction Layer with upcoming B2B standards, for example ebXML (11), which specifies XML-based web services for conducting electronic business over the internet. The BBC is also investigating how such tools could also integrate media exchange facilities into an organisation's internal enterprise integration software layers (often called 'middleware').

Of course, standardising the Exchange Transaction Layer does not answer the question of what file formats and transfer protocols *should* be used, and there is a danger that two organisations might not be able to negotiate an agreement. One way of tackling the problem is to restrict the sets of options in domain-specific recommended practice documents, e.g. for exchange with production facility houses and for exchange between news agencies.

CONCLUSIONS

The growing need to exchange broadcast media as files between organisations requires the

need to agree on certain common standards such as file formats and transfer protocols. But to enforce a rigid and minimal set of standards is unrealistic, so a means of negotiating an agreement on exactly how an exchange is to occur is desirable. The Pro-MPEG Forum has proposed a model in which an Exchange Transaction Layer decouples the organisations' business requirements from the details of the exchange. This layer is implemented by using an XML Manifest document to communicate transaction information. This straightforward approach is suited to a number of different applications and could be extended to interface with frameworks for B2B web services, with organisations' middleware architectures. The proposal will be refined and extended as a result of practical implementations, potentially leading to a protocol suitable for standardisation.

REFERENCES

1. EBU/SMPTE, July 1998, Taskforce for Harmonized Standards for the Exchange of Program Material as Bit-streams, Final Report.
2. Professional MPEG Forum, www.pro-mpeg.org.
3. GridFTP, www.globus.org/datagrid/gridftp.html
4. FASP, www.asperasoft.com/technology-fasp.html
5. TV-Anytime Forum, www.tv-anytime.org
6. UDDI Technical White Paper, September 2000, www.uddi.org/pubs/Iru_UDDI_Technical_White_Paper.pdf
7. IETF, January 1999, The TLS Protocol, www.ietf.org/rfc/rfc2246.txt
8. BBC R&D Desktop Production project, www.bbc.rd.co.uk/rd/projects/dp
9. W3C, May 2001, XML Schema version 1.0, www.w3.org/XML/Schema
10. SMPTE Recommended Practice 210, 2002, Metadata Dictionary Registry of Metadata Element Descriptions
11. Electronic Business using eXtensible Markup Language, www.ebxml.org