Research White Paper

WHP 148

April 2007

Business-to-business metadata interchange: Requirements for transport and packaging

R Bradbury

Future Media & Technology
BRITISH BROADCASTING CORPORATION

BBC Research
White Paper WHP 148

Business-to-business metadata interchange: Requirements for transport and packaging

Richard Bradbury

Abstract

The BBC has, for the past several years, been actively researching the applications of metadata in broadcasting. As well as contributing to a number of different standardisation groups it has considerable experience of implementing metadata systems in real content production and distribution chains.

This paper analyses some of the problems encountered when interchanging consumer and business metadata between multiple business systems. The exploitation of existing open standards to define a common metadata description profile and interchange interface are advocated, and the characteristics of such an interface are further elaborated.

Additional key words: TV–Anytime

White Papers are distributed freely on request.

Authorisation of the Head of Research is required for publication.

BBC Research
White Paper WHP 148

Business-to-business metadata interchange: Requirements for transport and packaging

Richard Bradbury

1 Introduction

Operators of digital television and on-demand delivery platforms use promotional information about the content available as a means of attracting consumers. This information is called **metadata** to distinguish it from the essence content and to emphasise the fact that it is information about content. Once populated with suitable consumer-facing metadata, a platform's content guide (or a suitable intelligent agent) becomes the means by which users navigate the sea of available content effectively. Provided the metadata is sufficiently rich, of good quality, and accurate, it will help consumers to assess the suitability of content for consumption.

Work done in this area by the UK Digital Television Group has identified a number of different actors in the metadata value chain. The model includes content creators and distributors from whom the source information originates and content aggregators who embellish this with promotional information. Metadata aggregators collate the information from a number of sources and pass it as formatted metadata to the parties responsible for delivery to set-top boxes and both electronic and print media outlets. The final party in the value chain is the consumer.

In such a diverse ecosystem, the natural outcome is for each metadata recipient (*e.g.* delivery platform operator) to define or adopt a proprietary metadata ingest interface and data format, leaving the suppliers with the unpalatable (and costly) prospect of developing a different publication interface for each target platform. If there is a particular dominant supplier, the alternative scenario is for this party to dictate an interface and to force this on all consumers of its metadata. This is equally undesirable because it forces operators of delivery platforms to implement multiple metadata ingest interfaces, especially if the dominant metadata supplier is unwilling to open the interface to other parties.

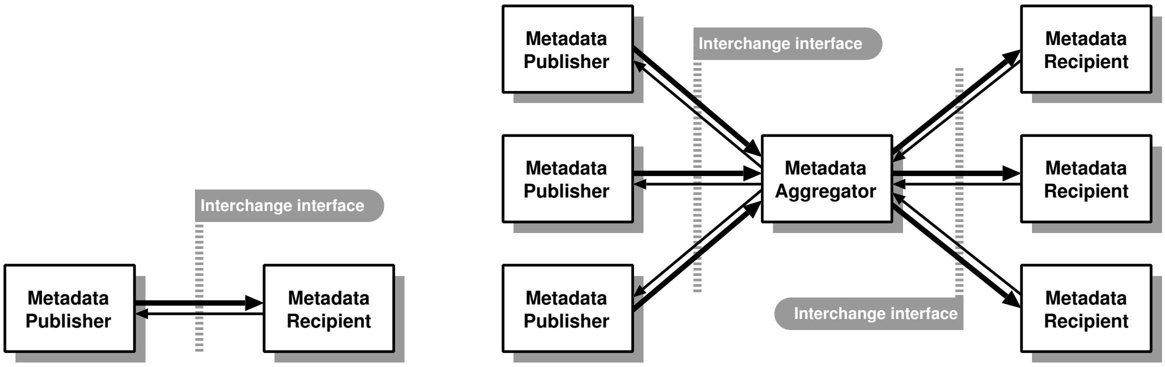
The present document describes a third approach: to define a single, standardised interface for business-to-business metadata interchange between metadata contributing parties and metadata consuming parties. To achieve this objective, the data format must be sufficiently descriptive to populate the set of target content guides. The profiling of TV–Anytime, an international standard for representing metadata, is advocated. Secondly, the interface itself must be fit for purpose. The abstract requirements for such an interface are therefore also discussed.

1.1 Metadata interchange scenarios

Figure 1 shows two different scenarios of interchanging metadata between parties. In 1(a) a single *publishing* party wishes to interchange metadata with a single *recipient* party. The interchange interface is simply shown between the two parties.

In 1(b), a more complex scenario is apparent. A central party takes on the role of *aggregating* the metadata received from a number of publishing parties. Having integrated these contributions, the aggregator then passes them on to a number of different recipients.

It can be seen that the adoption of a single interchange interface by publishers, aggregator and recipients simplifies the integration of this complex deployment considerably. The use of a different proprietary interface on each link causes a combinatorial explosion of implementations for the aggregator to deal with.



(a) Simple bilateral interchange. (b) Aggregation scenario.

Figure 1. Metadata interchange scenarios.

Note that the arrows in Figure 1 do not imply any particular mode of metadata transfer: both pull and push modes may be appropriate in real deployments.

2 Overview of TV–Anytime

As part of a work programme stretching back over a number of years, the TV–Anytime Forum has done some useful work analysing best practice in the area of metadata provision. Founded on real world use cases, the family of specifications published as a result of this work [ETSI TS 102 822] standardises the metadata required for the search, selection and rightful use of content on personal storage systems such as digital television recorders. In fact, the scope of TV–Anytime extends beyond this stated aim to provide the necessary data constructs for populating both linear and on-demand content guides. By adopting the TV–Anytime Metadata Description schema [ETSI TS 102 822–3–1], the parties exchanging metadata (a delivery platform operator and an end consumer in the TV–Anytime world view) have a standard means of describing the customer-facing fields associated with content assets and the structures underpinning the content.

TV–Anytime has a clear *business-to-consumer* focus. Note, however, that the ultimate purpose of *business-to-business* metadata interchange is to drive business-to-consumer applications and all interchange solutions must build on the needs of end consumers if they are to be successful.

TV–Anytime metadata is defined in terms of two principal database schemas: the **Metadata Description schema** and the **Location Resolution schema**. The characteristics of the two schemas and the dynamic behaviour of the data sets they describe are quite different and so any interchange interface based on TV–Anytime will most likely need to treat the two as different logical services.

2.1 Descriptive Metadata service

The Metadata Description schema (Figure 2) describes a database of descriptive metadata relating to programmes and their groupings (series, seasons, brands and so on). The schema allows *linear schedules* to be described as well as *on-demand propositions*, their non-linear counterpart. The schema is *normalised* so that a single programme record may correspond to several instances in a linear schedule (*e.g.* narrative repeats) or in an on-demand proposition (*e.g.* alternative downloads), while sharing the same title, synopsis and so on. Individual fields may also be overridden on an instance-by-instance basis where necessary.

Programmes and Groups are assigned a globally unique **Content Referencing Identifier** and this CRID is then used as a cross-reference in many of the other database tables. Global uniqueness

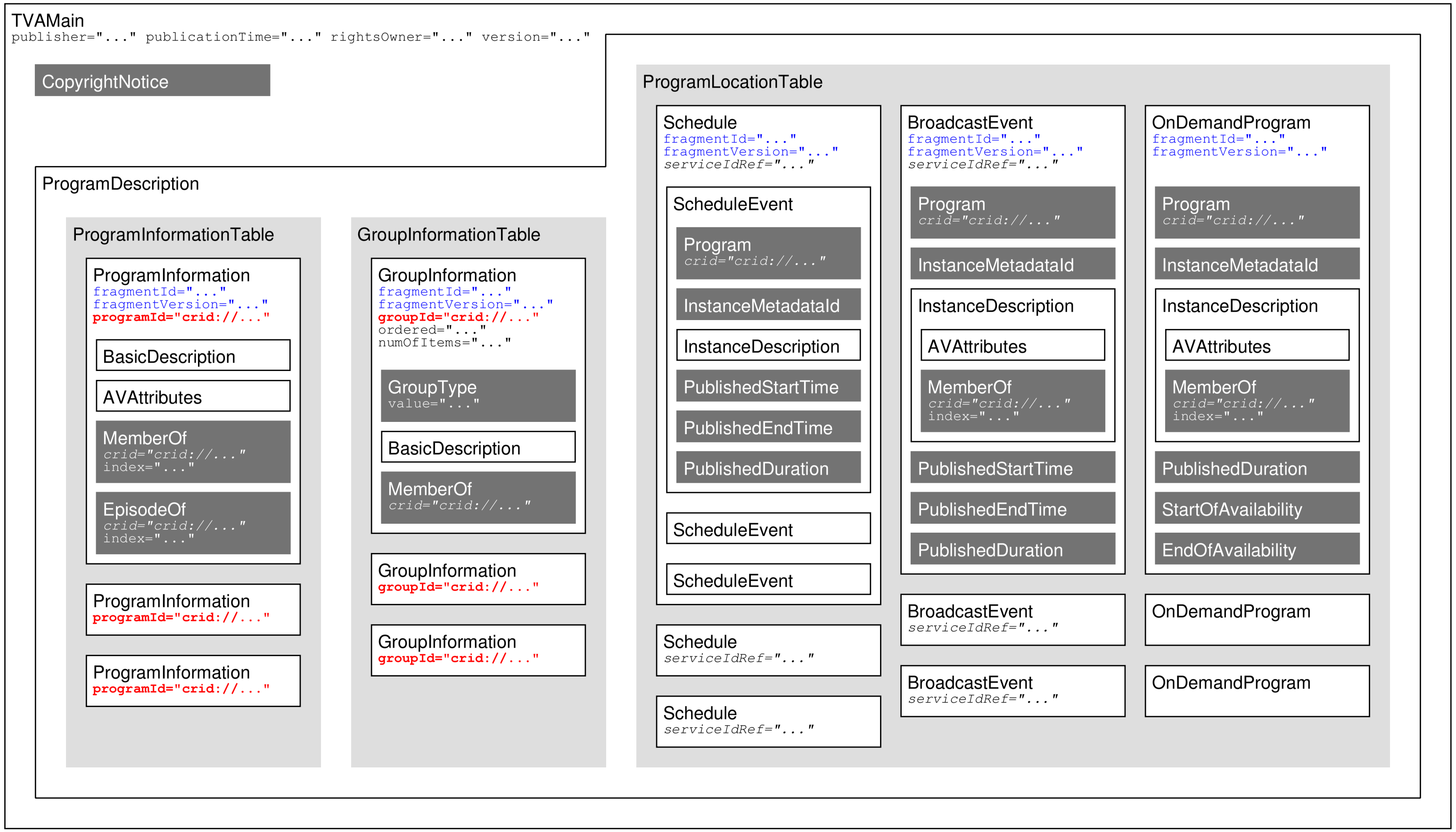


Figure 2. Schematic of a TV–Anytime Metadata Description (not all tables depicted). The root element is called <TVAMain> although the bulk of the descriptive Fragments live in the <ProgramDescription> structure. Individual programmes are described by ProgramInformation Fragments in the <ProgramInformationTable> but information about their availability is stored separately in the <ProgramLocationTable> since a particular programme may have a number of linear repeat showings or may be available on demand in more than one format.
The BroadcastEvent and OnDemandProgram Fragments convey instance availability information for these two cases. The CRID is used to to cross-reference common metadata in other tables and an Instance Metadata Identifier disambiguates different instances.
For efficiency, a sequence of linear instances can be collected into a single Schedule Fragment. The individual <ScheduleEvent> elements representing a period of a linear channel's schedule are then transmitted as an indivisible Fragment.
GroupInformation Fragments are used to represent higher level structures such as series, seasons, programme brands and so on. ProgramInformation Fragments declare their membership of a particular Group by means of the <MemberOf> or <EpisodeOf> linkage, citing the CRID of the parent Group. Similarly, Groups can themselves declare membership of higher level parent Groups, and so on.

of CRID values is ensured by delegating the assignment to self-regulating **CRID issuing Authorities**. This Authority may be the original creator of the content, the metadata publishing party or even a downstream aggregator. The identity of the Authority (its registered Internet Domain Name) is then combined with a locally unique identifier to generate a Uniform Resource Identifier of the following general form:

`crid://`*<Domain Name>*`/`*<Identifier>*

Segmentation information may optionally be bolted on to permit finer-grained navigation within a given programme or to form a composite programme from segments of others. Segmentation metadata is supplied as a separate database table (not shown in the Figure) so that a single programme may be segmented in several different ways.

Cast and crew lists are also supported, either inlined in the Programme or Group record, or by reference into a separate database table called the `<CreditsInformationTable>`.

The database of descriptive metadata records is collectively known as the Metadata Description, but it is acknowledged that interchanging the entire database on every update would be grossly inefficient. Instead, the Metadata Description schema defines the smallest unit of interchange: Programme and Group records, Schedules and so on. Provision is made in the schema for these so-called Fragments to carry version control information: a fragment identifier and a fragment version number. The fragmentation and version control features together enable a recipient to process updates in an efficient manner.

The Metadata Description schema describes the minimal information set required to render a linear or non-linear content guide, fulfilling TV–Anytime's search and selection criteria. It is designed to embody the promotional "billing" for a piece of content rather than accurate acquisition information. Descriptive metadata is assumed to be semi-static: once published, a record rarely changes, except perhaps to correct a mistake. CRIDs are designed not only to be globally unique but long-lived.

2.2 Location Resolution service

The Location Resolution schema (Figure 3) deals with the logically separate issue of content acquisition [ETSI TS 102 822–4]. Essentially, it provides the means to resolve an abstract content reference (the CRID) into one or more **acquisition Locators**. A Locator may point to a particular event in a linear schedule (to support real-time capture from a broadcast stream) or to some network resource (*e.g.* for download acquisition by means of non-real-time file transfer).

A programme that is repeated in the linear schedule may resolve to several linear Locators, giving the recipient some flexibility in scheduling recordings to avoid clashes (where the number of recordings requested exceeds the number of simultaneous record channels available). Similarly, for non-linear instances a number of different download locations may be advertised, either for reasons of load balancing or to provide the recipient with a choice of alternative acquisition methods. A particular CRID resolution may contain any combination of linear and non-linear Locators. For example:

Resolve: `crid://broadcaster.org/12345678` Complete: `true` Acquire: `any`
→ `dvb://233a..1044;39f2@2007-03-17T17:29:23Z--PT0H28M14S` (Premier on Saturday circa 5.30pm)
→ `dvb://233a..1044;463a@2007-03-18T20:00:24Z--PT0H28M14S` (Repeat screening on Sunday circa 8pm)
→ `ftp://downloads.broadcaster.org/programmes/12345678.m2t` (Download instance)

In addition, so-called **structural CRIDs** – those corresponding to higher order Groups in the descriptive metadata – do not resolve directly to a set of Locators, but instead resolve to a set of CRIDs corresponding to other records in the Metadata Description. For example, a "branch" CRID representing a series Group may resolve to a number of "leaf" CRIDs representing the individual episodes of that series. These "leaf" CRIDs in turn resolve to Locators representing individual events in the broadcast schedule. For example:

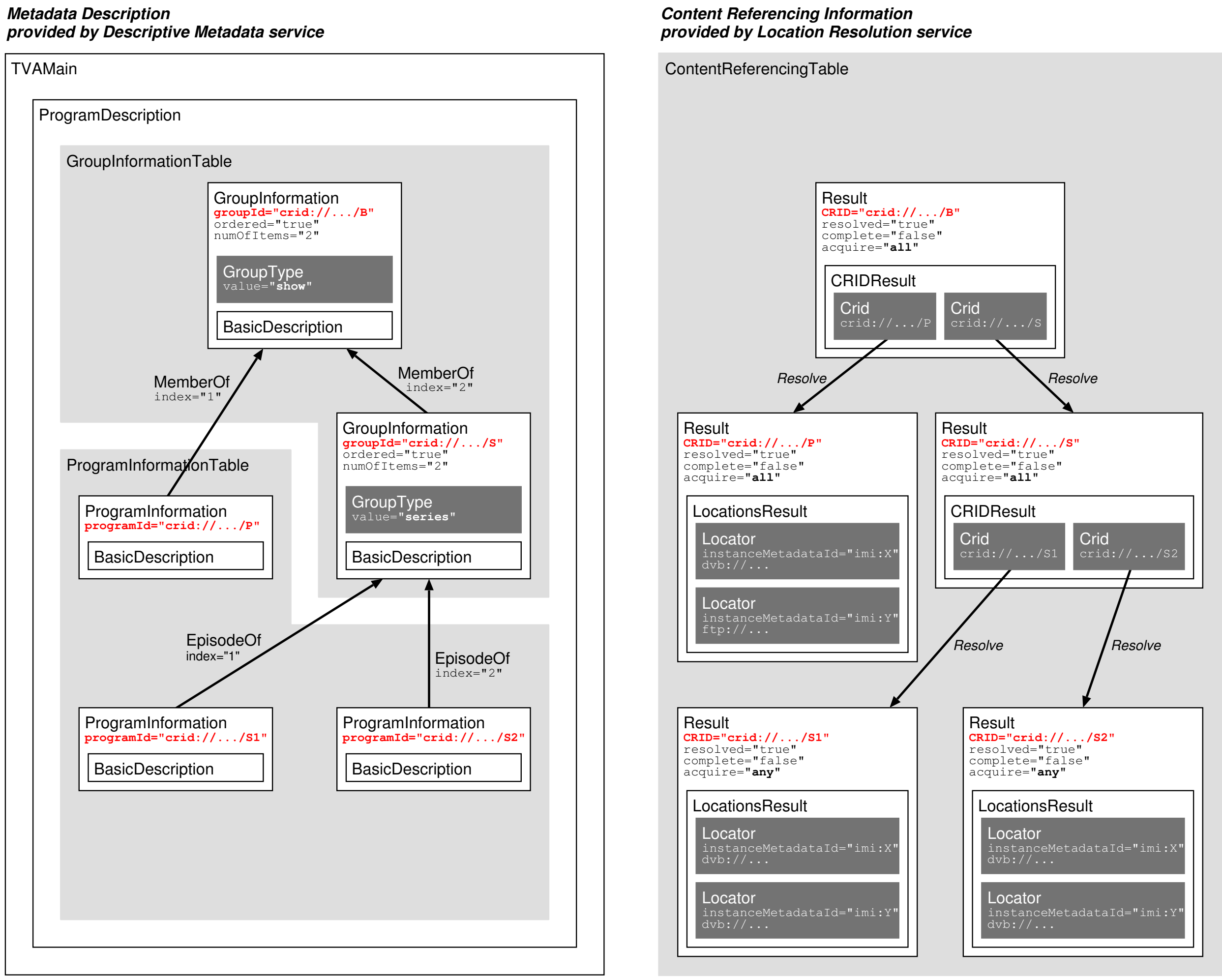


Figure 3. Schematic of a sample TV–Anytime Location Resolution service. The Metadata Description on the left shows a programme brand B, represented by a GroupInformation Fragment, which comprises a pilot episode P and a series S of two further episodes S1 and S2. Only the leaf programmes are represented by ProgramInformation Fragments. Notice that the group membership linkages are all upwards from the child to the parent.

Conversely, the <ContentReferencingTable> on the right hand side provides a downward means of resolving CRIDs to assist with the acquisition of content items. A user wishing to acquire all items relating to the brand in the example would attempt to resolve CRID B.

The CRID corresponding to a GroupInformation Fragment resolves into a further set of CRIDs via a <CRIDResult>, in this case accompanied by a directive to acquire all of the children.

The CRID corresponding to a leaf ProgramInformation resolves into a set of <Locator>s with an acquisition directive of any. These Locators correspond to narrative repeats of the same content item in the linear schedule, or perhaps to alternative downloads.

Note the use of the Instance Metadata Identifier, *e.g.* imi:X, to distinguish unambiguously between the possible acquisitions of the same content item. Each acquisition instance corresponds to an entry in the <ProgramLocationTable> (Figure 2). Instance Metadata Identifiers need be unique only within the scope of their enclosing CRID.

Resolve: `crid://broadcaster.org/series6789` Complete: `false` Acquire: `all`
→ `crid://broadcaster.org/prog2345` (First episode)
→ `crid://broadcaster.org/prog3456` (Second episode)
→ `crid://broadcaster.org/prog4567` (Third episode)

At each stage of resolution, the exact semantics of acquisition can be specified. For example, if a booking is made for a series CRID, *all* episodes in that series need to be acquired. Conversely, when a programme CRID is resolved to a number of Locators, *any* of these can be used to acquire the content asset itself and the recipient must choose between them.

The data set represented using the Location Resolution schema is referred to as **Content Referencing Information** (CRI) and is assumed to be highly dynamic. For example, the descriptive metadata for a forthcoming season of programmes may be published months in advance of linear transmission, but the scheduled time for each programme may be finally decided only ten days before transmission. In this case, the CRID published in the descriptive metadata will remain unresolved until nearer the time of transmission, but the end user can still make a booking to acquire the content using the published CRID (*c.f.* reserving a book by its ISBN before the publisher's release date). Similarly, the scheduled transmission of a programme may change at the last minute: it may be postponed to a later time or a different day, switched to a different channel or cancelled altogether. The consumer of a Location Resolution service cannot necessarily assume that a CRID has been resolved once and for all. Instead, it may need to adopt a strategy of re-resolving right up to the point of acquisition.

Furthermore, the set of CRIDs or Locators in a particular resolution may expand or contract over time as metadata is added or is removed. To assist with this, the completeness (or otherwise) of each resolution is signalled in the Content Referencing Information.

Finally, note that the TV–Anytime Locator is not by itself intended as a means of supporting accurate capture from linear streams. A linear Locator may contain a scheduled start time and duration, but this is not typically the most accurate means of acquisition. The Locator may optionally contain an event identifier, such as the DVB `event_id`. This can then be cross-referenced with an accurate real-time event transition signalling channel, such as the DVB Event Information Table[1] [ETSI EN 300 468] or Synchronised Auxiliary Data [ETSI TS 102 823].

3 Transport and packaging of TV–Anytime

The scope of the TV–Anytime standardisation effort extends to both broadcast and online delivery of content, but focuses on the business-to-consumer requirements. This includes a generic framework for the transfer of a fragmented Metadata Description over both unidirectional and bidirectional delivery networks [ETSI TS 102 822–3–2]. It also includes a normative Transport schema for query-based retrieval of metadata Fragments from a server and for packaging Fragments in the server's response [ETSI TS 102 822–6–1]. A mapping of this schema into SOAP Envelopes is also defined for transport over a bidirectional HTTP connection.

The technical module of the Digital Video Broadcasting (DVB) project has profiled the use of TV–Anytime metadata [ETSI TS 102 323] for use in broadcast networks. The fragmented XML Metadata Description is tokenised into a bit rate efficient form called BiM that has been standardised in the MPEG–7 Systems norm [ISO/IEC 15938–1]. DVB further specifies the transport of these BiM-tokenised Fragments in a DSM–CC Object Carousel [ISO/IEC 13818–6].

In summary, the various components required for a business-to-business interchange interface are all in place, but not necessarily in the required configuration. Figure 4 shows one way in which the various tools already described could be stacked together to meet the objectives of the open standards-based business-to-business interchange interface outlined in the introduction.

- The TV–Anytime Metadata Description schema and the corresponding Fragmentation specification combine to form a **Descriptive Metadata service**.

[1] DVB EIT Present/Following.

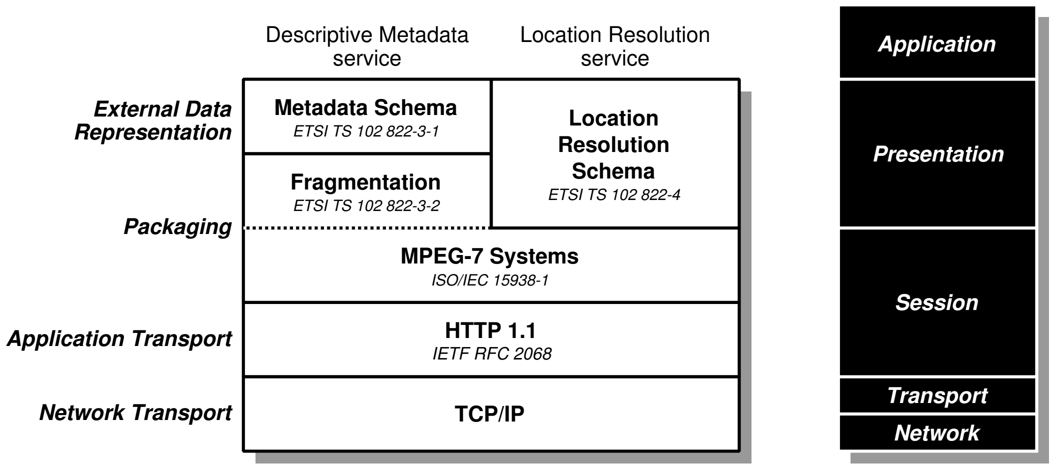


Figure 4. Proposed protocol stack for metadata interchange with OSI reference model shown alongside.

- In parallel with this, the Location Resolution schema provides the means of delivering a **Location Resolution service**.
- Hypertext Transfer Protocol [IETF RFC 2068] layered on top of TCP/IP is commonly used as the underlying data transport mechanism between clients and so-called Web Services in enterprise systems.
- Sandwiched between the Presentation and Transport lies a layer of *adaptation* responsible for packaging up the XML-based representations into transactions. The MPEG–7 Systems specification has a crucial role to play in this layer, although this is likely to be combined with other Session layer technologies such as SOAP in the Web Services context.

The advantage of a layered approach is that different protocols can quite straightforwardly be inserted or substituted into the stack to accommodate new requirements. For example, the MPEG–7 Systems layer shown in Figure 4 could be mapped to an Application Transport other than HTTP if a more suitable candidate emerges. Likewise, Transport Layer Security can be inserted between the Network Transport and the Application Transport if encryption across unsecured networks is required.

The remainder of the paper discusses the abstract requirements for the upper layers in this protocol stack with a particular emphasis on what is required for practical implementation.

4 Abstract requirements

The top-level goal of a business-to-business metadata interchange interface is to synchronise a number of Metadata Descriptions and Content Referencing Information data sets. The operating model assumed is one in which a metadata publishing party (the **Publisher**) wishes to update one or more metadata consumer parties (the **Recipients**), as shown in Figure 1(a). However, the concept is easily extrapolated to more complex scenarios, such as that shown in 1(b).

The object of the exercise is to maintain consistency between a logical Metadata Description at the Publisher and that at each of the Recipients. (Because of various filtering rules at source, the logical Metadata Description seen by each Recipient may actually be a different subset of the complete Metadata Description known to the Publisher, but the Publisher is responsible for constructing and maintaining the logical view onto that data set for the benefit of each Recipient.)

At an abstract level, the requirements for a business-to-business metadata interchange interface can be summarised as shown in the list below. Each requirement impinges upon one or more different layers in the protocol stack. The subsections that follow below describe the requirements in more detail and suggest how they may (or may not) be satisfied by the TV–Anytime standards.

- Commonly understood descriptive language.
- Working set.
- Distributed allocation of unique identifiers.
- Referential integrity.
- Low latency publication mode.
- Lightweight, stateless transaction processing by recipient.
- Robust recorded delivery.
- Exception reporting.
- Secure delivery.
- Means of resynchronisation after failure of the recipient.
- Means of resolving content references into Locators.

4.1 Commonly understood descriptive language

It is important that all parties involved in the metadata interchange share a common understanding not only of the data representation (*syntax*), but also the meaning (*semantics*) of the data set that forms the Metadata Description. The TV–Anytime Metadata Description schema provides a syntactic framework that deliberately avoids being overly prescriptive about the exact semantics. Moreover, the syntax is designed to be reasonably flexible with the obvious consequence that there is often more than one obvious way of representing a particular concept or relationship.

Clearly, some additional profiling of the Metadata Description schema is required in order to arrive at a semantics that can be understood by all parties. This is called an **Interchange Profile**. This profiling activity would typically involve:

- superimposing a semantic representational model on the syntax. For example, defining what constitutes a series of programmes; or deciding whether different transmission variants of a programme should be represented by different instance Fragments or by different ProgramInformation Fragments with a common parent Group.
- fixing the syntax to reduce the number of representations for each concept to (ideally) one. For example, the availability of subtitles should be signalled in just one place.

The Interchange Profile should take the form of a descriptive document with clear definitions of the semantic model and illustrative examples of usage.

Experience of implementing metadata systems suggests that finding a universal Interchange Profile is an unattainable goal in practice because of the fine level of detail involved at the semantic level. Instead, it is accepted that an Interchange Profile is likely to be specific to a particular application domain, although a good profile will be applicable to a wide range of domains.

The use of TV–Anytime as a common syntactic representation of descriptive metadata means that Interchange Profiles *can* be independent of the underlying mechanism employed to package up transactions and transport them between parties. This means that designing a universal Transport and Packaging protocol stack for conveying all TV–Anytime Metadata Descriptions *is* a realistic objective. Building real world deployments on a common technology platform has a number of practical benefits, not least the cost savings made when integrating heterogeneous systems and the simplification of testing.

Because a number of different Interchange Profiles could be mapped onto a standard protocol stack, an important concrete requirement is therefore to be able to discriminate between them. This signalling rightly belongs in the Session layer: compatibility between Publisher and Recipient should be negotiated before the exchange of any transactions.

The TV–Anytime Metadata schema offers a rich set of tools for describing customer-facing metadata that serves most needs. When agreeing on an Interchange Profile for a business-to-business application domain, however, the question inevitably arises as to how non-customer-facing concepts can be shoehorned into the metadata representation. Three main approaches are possible:

1. Redefine the semantics of an existing TV–Anytime data type. This is generally undesirable because of the opportunity for confusion when descriptive metadata is exchanged with other parties.

2. Modify the Metadata schema to incorporate a new data type. A small number of abstract types are defined with the Metadata schema and these can straightforwardly be extended with custom concrete types in a discrete schema namespace. Modifying the schema directly is discouraged because this hinders interoperability.

3. Exploit TV–Anytime's ability to reference `<RelatedMaterial>` and convey the opaque data either inline (using Hexadecimal or Base64 encoding) or externally (by means of a URI). The referencing mechanism uses controlled vocabularies (MPEG–7 Classification Scheme terms) to unambiguously label the type of the object being pointed at. The MPEG–7 Classification Scheme tool itself is designed with extensibility in mind.

Changes to the TV–Anytime Metadata schema specification are, of course, under the control of the publishing standards body (ETSI in this case). This limited scope for extension by implementors could be viewed either as a strength or as a weakness of the standard depending on the point of view!

In many use cases, however, it may prove architecturally cleaner to define a separate interface on the Recipient system to deal with non-TV–Anytime concepts and cross-reference with the Metadata Description where necessary using CRIDs or TVAIDs.

4.2 Working Set

The Metadata Description is a dynamically changing data set with new information constantly being added and old information falling dormant or becoming irrelevant. It is unreasonable to require all Recipients to accommodate an ever-expanding database of metadata. For practical implementation of a business-to-business metadata interchange the concept of a Working Set therefore needs to be agreed. This could, for example, be defined as the minimum set of metadata required by a Recipient to deal with some current editorial content proposition.

In practical terms, it would be relatively straightforward to agree that a Schedule Fragment may be dropped from the Working Set once its end date/time has passed. Similarly, an OnDemandProgram Fragment can safely be discarded after its `<EndOfAvailability>` has expired. Implicit expiry of other Fragment types is not so obvious, however.

For ProgramInformation and GroupInformation Fragments, the situation is slightly more complex. Membership of the Working Set might naïvely be defined in terms of whether the CRID is referenced by other Fragments in the Metadata Description, but this would preclude the (sometimes legitimate) advance publication of "placeholders". For example, a broadcaster may wish to promote a forthcoming series before production is complete. This might involve early publication of a suitable GroupInformation Fragment that will only subsequently be referenced by individual ProgramInformation Fragments.[2]

The removal of Fragments that are no longer part of the Working Set could be achieved by means of an explicit deletion by the original Publisher. However, this approach has the unattractive side-effect of placing storage management at the Recipient under the control of the Publisher. In practice, this may be unacceptable to some Recipients, each of which may have its own different caching policy. A more flexible technical approach is therefore to place responsibility for maintaining the Working Set with the metadata Recipient. One of several possible Recipient-based solutions to this problem is the use of automated reference-based garbage collection. Another is the inclusion of explicit expiry information embedded in the Fragment itself.

Note that this definition of the Working Set does not preclude a particular party from archiving metadata for longer periods if it so chooses. A metadata aggregator may, for example, choose to retain information on a long-term basis. Conversely, a consumer-facing platform that carousels Fragments on a limited-capacity broadcast channel might take a more aggressive stance on expunging out-of-date metadata.

4.3 Distributed allocation of unique identifiers

In the general case, where a Metadata Description is sourced and aggregated from multiple Publisher parties, it is essential that unique identifiers appearing in the Metadata Description received from one party do not clash with those received from another party unless, of course, they are intentionally referring to the same entity in the combined Metadata Description.

The allocation of CRID values for ProgramInformation and GroupInformation Fragments is already distributed by virtue of the **CRID issuing Authority** portion – the registered Domain Name at the start of the string.

The other main class of identifier in the Metadata Description is the TVAID schema type, used as the primary key for almost all other Fragment types, including the `fragmentId` itself. These identifiers are required to be unique only within the scope of a particular Metadata Description. However, when aggregated from multiple sources, additional constraints need to be placed on these identifiers to ensure uniqueness at the Recipient.

An alternative approach is for the Recipient party to maintain logically separate Metadata Descriptions for each Publisher party it is aggregating. This has an obvious performance penalty when there is an overlap of metadata provision.

Where several parties are providing alternative descriptive metadata to a Recipient, the same CRID may quite legitimately appear in different contributing Metadata Descriptions, provided it is describing the same piece of content. For example, there may exist more than one ProgramInformation Fragment with the same CRID. In such cases, a globally unique `fragmentId` is an obvious means of disambiguating. The business rules used to select the most appropriate set of descriptive metadata are application-dependent.

4.4 Referential integrity

The term **Referential Integrity** is used here to mean that there are no unresolved cross-references within the Metadata Description. In other words, CRID and TVAID values must be declared in a Fragment before they are referenced by another Fragment.

In practical terms, the integrity rule is that the application of any transaction must result in the transition of the Metadata Description from one coherent state to another state that is also coherent. Where there are circular references to new CRID and TVAID values between Fragments, the affected Fragment updates should be delivered in a single logical transaction in order to satisfy the integrity rule.

[2] A more explicit means of supplying metadata expiration hints has been proposed as an amendment to the TV–Anytime Metadata Description schema and, if accepted, could be used to prevent premature garbage collection.

For example, if a SegmentGroupInformation Fragment is added to the Metadata Description, it can only refer to the `segmentId` of a SegmentInformation Fragment previously added to the Metadata Description or to the `segmentId` of a new SegmentInformation Fragment *delivered in the same logical transaction*.

Likewise, the SegmentInformation Fragment is only allowed to refer to the CRID of a ProgramInformation or GroupInformation Fragment that already exists in the Metadata Description, unless the relevant CRID is defined by another Fragment in the same logical transaction.[3]

A Publisher party should be able to "prepare the ground" by publishing Fragments in a logical sequence of transactions where appropriate, but it should also be possible to publish mutually referential Fragments in a single transaction where there is no other way of maintaining Referential Integrity. In other words, it is technically possible to deliver the best of both worlds. The TV–Anytime table structure and Fragmentation specification assist to a certain extent by defining Fragments at a sensible level of granularity.

The main concrete requirement to emerge from the need for integrity is that the Session layer should support a means of "blocking" a number of Fragment updates into a single logical transaction. A number of application transport protocols support the `multipart/related` MIME content type, for example, which could be used to attach multiple Fragments to a root message.

4.5 Low latency publication mode

An important requirement is for updates to the Metadata Description to be published with the minimum latency so that the change encapsulated in the transaction is reflected as quickly as possible to the Recipient(s). An update may be something as simple as correcting a spelling mistake, but could potentially involve the complete withdrawal of a piece of content for legal reasons.

In order to meet the objective of low latency a pushed mode of operation is preferable to a pulled mode. It also points towards the use of an online transport between active nodes rather than primitive file transfer. (It is recognised that certain legacy Recipient systems may enforce a polling mechanism in which new transactions are detected on a periodic basis. Although less desirable, pull modes cannot therefore be ruled out completely.) A number of push-based application transports are already in common use in TCP/IP internetworks:

- File transfer (*e.g.* File Transfer Protocol, FTP) is designed for asynchronous bulk transfer of files between a sender and a recipient. The major drawback of FTP is that a separate signalling mechanism is required to inform the recipient of the transfer. Alternatively, the recipient can poll a known drop-off area for new transactions pushed to it.

- Electronic mail (*e.g.* Simple Mail Transport Protocol, SMTP) employs a point-to-point push message delivery semantics between Message Transfer Agent nodes, the packaging of the message being based on the [RFC 822] message envelope.

- Usenet news (*e.g.* Network News Transport Protocol, NNTP) employs a point-to-multipoint message delivery semantics between news nodes, again using the standard [RFC 822] message envelope for packaging the message. NNTP effectively provides a subscription-based multicast service at the Application Transport level based on regular polling of the Publisher by the Recipient.

- Web services (*e.g.* Hypertext Transport Protocol, HTTP) employ a point-to-point remote invocation metaphor when combined with a suitable Packaging layer such as the W3C SOAP Envelope.

[3] In point of fact, `<ProgramRef>` is an optional child element of `<SegmentInformation>`, so the CRID reference can be deferred by omitting the corresponding child element if it is not known when the SegmentInformation Fragment is published.

4.6 Lightweight, stateless processing

In order to better accommodate the objective of low-latency transmission, the processing requirements placed on the recipient of the transaction should be minimised as much as possible. It should not, for example, be necessary for the recipient to parse and validate vast tracts of Metadata Description only to find that the majority is identical to previously received updates.

This objective can, to a certain extent, be met by complying with the TV–Anytime Fragmentation Specification [ETSI TS 102 822–3–2 Clause 4.3]. The recipient is then only required to parse and validate a small fragment of the overall Metadata Description at a time rather than trying to merge two potentially unwieldy XML trees, validate the result and roll back if validation fails.

TV–Anytime's fragmentation rules specify that only the Fragment itself should be transmitted. This includes only the root element of the Fragment and any child elements; enclosing parent elements above the level of the Fragment root are omitted for brevity. This ensures a reasonably compact XML instance document that is well-formed but not one that can be validated directly against the Metadata Description schema. [ISO/IEC 15938-1] specifies that it is the responsibility of the Packaging layer (the Systems layer in MPEG–7 parlance) to signal where in the Metadata Description tree the new Fragment is to be "grafted" – the so-called Fragment Update Context. MPEG–7 also specifies the use of an XPath expression as the means of encoding this information.

By combining the received Fragment with its Fragment Update Context, the recipient can reconstruct a minimal Metadata Description tree that enables the Fragment to be validated against the schema prior to integration into the current full Metadata Description. The process of validating the Fragment meets the required criteria by being both lightweight and stateless. Furthermore, the set of identifiers contained in the Fragment (CRID and TVAID values) can also be harvested as a by-product of validation, thereby providing a lightweight route to performing semantic checking of the received Fragment.

Finally, by making proper use of the `fragmentId` and `fragmentVersion` attributes the Recipient can better track changes to a particular Fragment and ignore fragment updates carrying the same version number as a previously received update. This requires careful management of the Metadata Description dynamics by the Publisher.

The thrust of these particular requirements puts the responsibility for the management of the Metadata Description squarely at the door of the Publisher with the Recipient relieved of the burden of calculating how one update differs from a previously received one. This is very much in keeping with the central philosophy of MPEG–7 Systems and TV–Anytime. Logically, it makes a great deal of sense: experience points to the fact that the sender always has the best idea of what is changing in a dynamic system. Architecturally, too, it is more appropriate for a single Publisher to do this work than for every Recipient to have to do it.

4.7 Robust recorded delivery

In a business-to-business interchange scenario the successful delivery of transactions is very important and the Application Transport should therefore be able either to guarantee delivery, or at least be able to report whether delivery was successful or not. Given that the former requirement is difficult to achieve in practice, the lesser requirement implies the need for some sort of recorded delivery régime to be provided by the interchange interface.

This sort of thing is conventionally implemented in the Session layer as some form of transaction acknowledgement with a retry mechanism. This further implies the need for the Packaging layer to carry a session-unique transaction identifier assigned by the Publisher.

At minimum, a positive acknowledgement message must also be defined so that the Recipient can signal successful receipt of each transaction. The Publisher's transaction identifier needs to be quoted in this response so that a log of successful receipts can be retained by the Publisher in case of dispute.

A recorded delivery mechanism is most straightforwardly provided by synchronous point-to-point transports because the Recipient can acknowledge receipt directly using the reverse communications path back to the Publisher inherent in most Transport Layers, including TCP. In a point-to-multipoint publishing scenario acknowledgements from the various Recipient parties must be provided over individual point-to-point links back to the Publisher.

4.8 Exception reporting

At a higher level than the acknowledged Session transaction protocol just described there are a number of exceptional conditions that can usefully be reported to the Publisher by the Recipient. These include:

- **Syntax error.** The transaction was received, but one of the metadata Fragments was malformed in some way. This class of exception can be further divided into problems with the well-formedness of the XML instance document(s) and errors arising from schema validation.

- **Referential Integrity exception.** The transaction is syntactically correct, but attempts to reference a non-existent identifier in the Metadata Description that is not declared elsewhere in the same transaction.

- **Semantic exception.** The transaction is syntactically correct and obeys the Referential Integrity rule, but breaks some other high-level rule of the Interchange Profile.

The Transport and Packaging protocols need to define a means of reporting exceptions like these, including sufficient contextual information to allow the Publisher to identify, correct and retransmit the data when appropriate.

Low-level exceptions

As with any layered protocol model, each layer needs to provide its own acknowledgement and error reporting mechanism. So, in addition to the above, there is a separate class of low-level system errors that may prevent the successful delivery of transactions. For example, a listening process on the Recipient system may fail for some reason. The underlying Application Transport mechanism is therefore expected to support its own low-level acknowledgement, timeout and retry mechanisms to deal with such occurrences and recover as best it can from them.

4.9 Secure delivery

The principal requirement here is not that transactions should be kept secret from potential eavesdroppers, but rather that they can be transmitted without interference or other manipulation by third parties ("man in the middle" attacks). The prevention of "spoofing" is particularly important where the business-to-business model is based on the generation of revenue from metadata interchange.

This security requirement could be met by the use of an encrypted Transport layer (*e.g.* SSL/TLS) where the underlying network is inherently insecure, or else by providing a secure Network layer (*e.g.* IPsec) beneath the Transport layer.

4.10 Resynchronisation after failure

The metadata Recipient is assumed to maintain a current Working Set of metadata in line with the requirements already described. A number of different implementation strategies may be employed by the Recipient. It may, for example, choose to store the Metadata Description in native TV–Anytime form as an XML-based database. On the other hand, it may prefer to store the metadata in a more optimised non-XML database format. Such implementation considerations are outside the scope of an interchange interface specification.

However, if the Recipient's database fails for some reason then it needs a means of re-acquiring a complete Working Set of the Metadata Description. This could be achieved simply by signalling upstream metadata Publishers with a request that the Working Set is republished using the conventional publication mechanism. On the other hand, the requirement could also be met by providing a suitable query interface at the metadata Publisher that enables individual Fragments to be retrieved [ETSI TS 102 822–6–1 Clause 5.1]. In the second scenario, however, it is unclear how the Recipient knows which Fragments make up the Working Set.

4.11 Location Resolution service

A means for transporting Content Referencing Information between a CRID Resolving Authority and a Recipient is also required.

TV–Anytime's Location Resolution service [ETSI TS 102 822–4] defines a simple schema for describing the resolutions of content references (CRIDs) into either Locators or into other CRIDs. However, unlike the Metadata Description schema, where specific rules for Fragmentation are defined, there is no specific Packaging layer defined for Location Resolution metadata: each resolution is simply presented as a separate Result element in a Content Referencing Table structure. This allows multiple resolutions to be conveyed in a single transaction.

The concrete requirement here is therefore to provide a means of mapping this service directly onto a suitably defined Session layer.

In some application domains, a means of signalling accurate linear event transitions is required. The TV–Anytime Location Resolution service is not intended to provide such a service directly, but does have a means of conveying an external event identifier in the Locator. However, the delivery of the event transitions themselves is outside the scope of the metadata interchange interface described here.

5 Conclusion

The TV–Anytime family of specifications has been described and advocated as the basis for a standardised business-to-business metadata interchange profile. A set of abstract requirements for packaging and transporting TV–Anytime Metadata Description Fragments and Content Referencing Information has been proposed and some concrete requirements have been distilled out from them.

While TV–Anytime provides a useful basis for fulfilling these requirements, further work will be required in three specific areas to reach the desired goal of interoperability:

- **Profiling the Metadata schema.** A domain-specific profile requires an unambiguous semantic definition for every entity in the TV–Anytime data model and constraints on the usage for each corresponding schema element.

- **Selection of appropriate Network and Application Transport protocols.** There are a number of obvious choices in this area.

- **Selection of suitable packaging technologies** to bridge the gap between the schemas and the underlying Application Transport. Again, there are some obvious candidates already included in the TV–Anytime and MPEG–7 toolkits, but some work will be needed to reconfigure individual tools to satisfy the requirements outlined in this White Paper.

The next obvious step is to find suitable partners to collaborate in reaching agreement on a technical solution that meets these requirements.