University of Wollongong
Research Online

Faculty of Informatics - Papers (Archive)

Faculty of Engineering and Information Sciences

2005

Bitstream Binding Language – Mapping XML Multimedia Containers into Streams

J. Thomas-Kerr
University of Wollongong

I. Burnett
University of Wollongong, ianb@uow.edu.au

Philip Ciufo
University of Wollongong, philip_ciufo@uow.edu.au

Publication Details

This paper originally appeared as: Thomas-Kerr, J, Burnett, I and Ciufo, P, Bitstream Binding Language – Mapping XML Multimedia Containers into Streams, IEEE International Conference on Multimedia and Expo, 6 July 2005, 626-629. Copyright IEEE 2005.

Research Online is the open access institutional repository for the University of Wollongong. For further information contact the UOW Library: research-pubs@uow.edu.au

Bitstream Binding Language – Mapping XML Multimedia Containers into Streams

Abstract
Bitstream Binding Language (BBL) provides an abstraction layer between XML multimedia containers and the way their resources and metadata are published in a bitstream. It allows multiple bindings from a single source document to facilitate interoperability and applicability of the multimedia content to a wide range of terminals and users. BBL introduces a number of features not found in other XML fragmentation techniques – such as support for random-access – and provides a highly general framework for combining metadata with multimedia content in a transport format. This paper presents an overview of BBL and provides details of a Java implementation using the MPEG-21 Multimedia Framework as a sample application.

Disciplines
Physical Sciences and Mathematics

Publication Details
This paper originally appeared as: Thomas-Kerr, J, Burnett, I and Ciufo, P, Bitstream Binding Language – Mapping XML Multimedia Containers into Streams, IEEE International Conference on Multimedia and Expo, 6 July 2005, 626-629. Copyright IEEE 2005.

This conference paper is available at Research Online: http://ro.uow.edu.au/infopapers/120

BITSTREAM BINDING LANGUAGE – MAPPING XML MULTIMEDIA CONTAINERS INTO STREAMS

Joseph Thomas-Kerr, Ian Burnett, Philip Ciufo

University of Wollongong, CRC SIT (Smart Internet Technology)
joetk@elec.uow.edu.au, i.burnett@elec.uow.edu.au, ciufo@uow.edu.au

ABSTRACT

Bitstream Binding Language (BBL) provides an abstraction layer between XML multimedia containers and the way their resources and metadata are published in a bitstream. It allows multiple bindings from a single source document to facilitate interoperability and applicability of the multimedia content to a wide range of terminals and users. BBL introduces a number of features not found in other XML fragmentation techniques – such as support for random-access – and provides a highly general framework for combining metadata with multimedia content in a transport format. This paper presents an overview of BBL and provides details of a Java implementation using the MPEG-21 Multimedia Framework as a sample application.

1. INTRODUCTION

The volume of multimedia published on the internet continues to expand at an exponential rate. It is widely acknowledged, however, that the utility of this vast store of content is diminished without machine-understandable metadata; this is needed to facilitate operations such as retrieval, categorization and filtering. Consequently, numerous standards have been developed to provide vocabularies which represent this metadata in a semantically meaningful way (for example, the Dublin Core and MPEG-7). Other standards are also in development to assist the transaction and processing of content and associated metadata (MPEG-21, TVAnytime, SCORM). Significantly, these standards all use XML to structure their content.

The complete set of metadata relevant to a piece of multimedia content in any context is potentially very large. This highlights two things:

- metadata relevant to one user in a certain context may not be relevant to a different user or in a different context, and
- bandwidth constraints may compel the use of streaming mechanisms to transmit metadata incrementally.

This work was partially funded by the Smart Internet Technology CRC.

Further, some metadata has temporal relevance. It may make sense to transmit this information at the time at which it is relevant. There are also numerous scenarios – for example in a broadcast context – in which it is desirable to repeat metadata and/or structural information at regular intervals to enable random access into a multimedia stream.

Given these considerations, an approach is required which facilitates packetization of the metadata in semantically meaningful units, and provides functionality to schedule and repeat each packet as required. While there are a number of methods which enable fragmentation of XML documents (see Section 2) – some targeted specifically at multimedia metadata – none address all of the issues discussed above. Consequently, we present here a mechanism to enable the mapping of XML-based metadata containers, along with their associated content, into various bitstream transports: a Bitstream Binding Language (BBL).

A BBL instance will describe a particular binding between a multimedia metadata container, its associated resources, and an output bitstream. This facilitates multiple presentations of the same content by providing a level of indirection between metadata authoring and storage, and its presentation to a user. Different bindings may thus be optimized to provide different views/subsets of the data, different renderings of the content, or different bitstream transports. The latter allows the framework to be interoperable with the wide range of terminals, networks and users that wish to consume the media.

2. RELATED WORK

A number of approaches exist for streaming XML data. These are presented briefly here, with an analysis of the further work needed to enable the above applications.

2.1. XStream

XStream[1] is a mechanism developed for packetization of XML documents in wireless environments. It specifies a simple technique of assembling packets node by node, until a predefined Maximum Transfer Unit size is reached. There is no control over the inclusion of specific fragments in semantically related units.

0-7803-9332-5/05/$20.00 ©2005 IEEE

2.2. XML Binarization

BiM[2] and Millau[3] are XML binary encoding formats, both of which address the fragmentation of XML trees for individual transport. BiM also defines Action Units which can add, replace, delete or reset fragments within the destination tree - allowing an XML document at the receiver to become dynamic. In [3], the authors of Millau discuss the possibility of linking XML metadata with specific fragments of a multimedia resource; however [3] does not detail the mechanism by which this is to be achieved.

2.3. Annodex

The Annodex[4] format has been developed with the aim of providing hyperlink and search functionality to multimedia data. An Annodex stream consists of a media resource interleaved with XML markup containing annotation and index information (*anno-dex*). A small XML grammar is specified to provide very simple structured and unstructured metadata elements.

2.4. XFlavor and BSDL

XFlavor[5] and BSDL[6] are both approaches for describing bitstreams in XML. Their aim is to allow standardized XML tools such as XSLT to perform operations on the bitstream for content adaptation, repurposing and the like. BSDL also provides an XML-to-Binary parser, to convert a (possibly transformed) XML description back to its original binary syntax for transmission or storage. Neither work specifically addresses the inclusion of metadata within the bitstream.

2.5. Discussion

A number of these related works discuss decomposition of XML into structural fragments, with varying methods of specifying the original context of each fragment. BiM, in particular, provides a dynamic framework whereby fragments may be updated and deleted, enabling a metadata document to change over the course of its lifetime.

Annodex allows simple metadata to be attached to temporal intervals of a single resource, annotating resource fragments, and allowing the consumer to hyperlink into and out of the content at defined points.

However, none of these approaches provide a general framework to attach metadata to media content in a streaming format. Specifically, the requirements identified are:

- Support for the association of metadata and multiple resources, to allow specification of the relationships between various elements,
- Fragment repetition and support for random-access, and
- The ability to specify multiple bindings from a single source document, in order to allow the adaptation of the content to a variety of users, terminals and networks.

3. BBL TEST CASE: STREAMING MPEG-21 DIGITAL ITEMS

MPEG-21[8] is one possible application of BBL. It was chosen as an initial test case because of its wide applicability and its content format-independent approach. Other domain specific applications are TVAnytime and SCORM.

MPEG-21 is designed as a multimedia framework, providing metadata to facilitate the production, exchange, and consumption of multimedia content. Its major functionalities include Content Identification and Description, Intellectual Property Management and Protection, Digital Rights Management, and Content Adaptation.

The fundamental unit of transaction in MPEG-21 is a Digital Item (DI) – a digital container for related resources and metadata that defines the relationships between each element. An example DI describing an Internet TV channel is shown in Figure 1 (some sections replaced by commenting for brevity). The advantage of the BBL approach is that a DI may collect a large quantity of related resources and metadata, only the relevant parts of which are selected by BBL to send to a particular user.

```
<DIDL>
  <Item>
    <!--global metadata-->
    <Item id="SMIL">
      <!-- SMIL presentation for DI -->
    </Item>
    <Choice>
      <!-- Selections for audio & subtitles-->
    </Choice>
    <Item id="PROG_A">
      <!-- Metadata for Program A -->
      <Component>
        <!-- video stream metadata -->
        <Resource mimeType="video/mp4"
             ref="progA/video.mp4"/>
      </Component>
      <Component>
        <Condition require="AUDIO_EN"/>
        <Resource mimeType="audio/mpeg"
             ref="progA/english.mp3"/>
      </Component>
      <!-- other audio components -->
      <Annotation target="#PROG_A">
        <Condition require="SUBTL_EN"/>
        <Anchor>
          <Fragment fragmentID="time(1.0,2.5)"/>
          <!-- Subtitle -->
        </Anchor>
        <!-- Subtitles continue -->
      </Annotation>
      <!-- other language subtitles -->
    </Item>
    <Item id="PROG_B">
      <!-- Program B -->
    </Item>
  </Item>
</DIDL>
```

Fig. 1: Example Digital Item – Internet TV

Element	Belongs to	Use
`<BBL>`		The BBL root element. Activates the BBL parser.
`<file>`	`BBL`	Selects an XML container file to operate on. If absent, file containing `<BBL>` is used. Multiple `file` elements are permissible.
`<AU>`	`file` or `BBL`	Begins a new Access Unit.
`<include>`	`AU`	Selects a fragment of the active `file` to include in the current AU. This is specified via an xPath statement and a depth parameter to indicate the number of levels of children to include.
`<resAU>`	`file` or `BBL`	Begins a stream of AUs, each containing one or more resource fragments.
`<resInclude>`	`resAU`	Identifies a resource to be included in the current resAU stream. Specified either via an xPath pointer to the resource URI within the container file, or directly via a URI.
`<metaInclude>`	`resAU`	Allows temporal metadata to be included at the appropriate point(s) in a resAU stream.

Table 1. Basic BBL Elements[7]

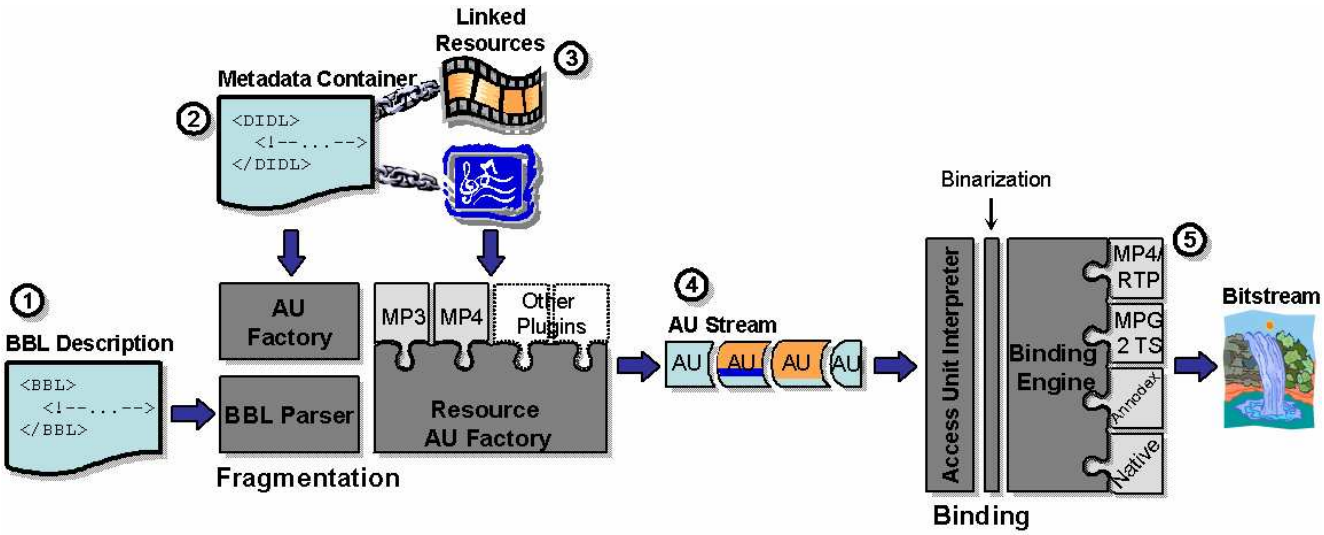

Fig. 2: BBL Reference Implementation

4. BITSTREAM BINDING LANGUAGE

The Bitstream Binding Language allows a content author to easily assemble multiple bitstreams from an XML metadata container and its associated resources. A BBL instance describes one particular view of the content, and multiple views may be created to suit the various contexts in which the content is to be presented.

The basic elements of the Bitstream Binding Language are shown in Table 1. For a complete specification, the reader is referred to [7].

A reference implementation has been developed to demonstrate the use of BBL in metadata streaming, using Java and the Apache SAX/DOM parsers. This system, shown in Figure 2, operates as follows:

A particular bitstream binding is described in a BBL instance document ①, which may be incorporated in the metadata container, or as a stand-alone XML file. The BBL instance uses xPath to locate or identify *fragments* of the metadata container ② and attached resources ③ which are inserted into Access Units (AUs) ④. The AUs are then bound into a specific transport format ⑤.

The way in which resources are inserted into the final bitstream varies according to the stream format. Where a binding exists for the particular resource type in-band within the stream (eg MPEG-4 or MPEG-2 video within MP4/RTP), this method is used. At other times, resource fragments can be included as binary data within the metadata AU. Such AUs are interpreted at the client via the metadata processor – in this case an MPEG-21 client. Handlers for individual resource types ③ and transport formats ⑤ are specified via a standard API to allow additional formats to be plugged in as required.

While textual XML may be appropriate in a (limited) set of circumstances, it is fundamentally verbose, and binarization is often preferable. BiM[2] provides a 90 - 95% reduction in the required XML bit rate, and also allows parsing and filtering directly at the bit stream level. This is attractive because it combines the advantages of XML processing with an efficient syntax. Other encoding schemes offer greater compression but sacrifice flexibility.

```
<?xml version="1.0" encoding="UTF-8"?>
<BBL xmlns="urn:mpeg:mpeg21:2004:01-BBL-NS">
  <file path="DID_InternetTV.xml">
    <AU id="RA" cmd="update" ver="1.0" rpt="0.5">
      <include xPath="/" depth="2"/>                    A
      <include xPath="//Item[@id='SMIL']"               B
               depth="-1"/>
    </AU>
    ...
    <resAU id="PROG_A" cmd="res" ver="1.0">
      <metaInclude depth="-1"                           C
      xPath="//Item[@id='PROG_A']//Annotation"/>
      <resInclude maxTime="1"                           D
        xPath="//Item[@id='PROG_A']//Resource"/>
    </resAU>
    ...
  </file>
</BBL>
```

Fig. 3: BBL Instance Document - Internet TV

5. BBL EXAMPLE – INTERNET TV

The Digital Item shown in Figure 1 will help to illustrate how a BBL Instance (Figure 3) may be used to generate a structured bitstream containing metadata and resources.

In this example, the first requirement is an AU which provides all of the information needed to interpret the stream. This AU must be regularly repeated to provide random-access, and should expose the overall structure of the DI Ⓐ, as well as the presentation configurationdata Ⓑ.

The `depth` of -1 (Ⓑ) indicates that the entire subtree beneath the node (selected by the `xPath` statement) is to be included in the AU. When all `include` elements have been processed, fragments are reassembled into document order, so that the semantic relationships between elements are maintained (Figure 4). The `cmd` and `ver`(sion) attributes are attached to the AU to indicate the action to take on receipt, and `rpt` indicates the interval (in seconds) between subsequent transmissions of the AU.

In order to create a stream of AUs which combine resource content with temporal metadata, a `resAU` element is used. Metadata is selected via the `metaInclude` command Ⓒ. The temporal parameters specified in the container file (in this case the `fragmentID` attributes) are used to indicate when to insert each fragment into the resource stream. In this way, each subtitle is received at the correct time for presentation. The resource location is usually pointed to by an xPath statement into the the metadata container Ⓓ, although it may also be directly specified in the BBL description. Resources are fragmented according to the framing of the underlying format; fragments may also be aggregated to a specifiedmaximum timespan and/or size (in bytes).

6. CONCLUSION

In this paper, we have proposed a Bitstream Binding Language, which provides a level of indirection between a multimedia metadata container and its transport as a bitstream.

This approach provides a number of advantages over existing techniques, the firstof which is the ability to provide different bindings of the same content – providing interoperability with the widest range of networks, terminals and users. BBL also provides the flexibility to repeat and/or omit fragments so as to easily allow authoring of bitstreams suitable for broadcast applications that provide random-access. Another advantage is the wide applicability of BBL – as an XML-based language it is easily integrated with the many metadata containers that exist in this format.

In future work, we intend to test the framework with further container and bitstream formats, as well as investigate performance gains from other XML encoding schemes.

```
<DIDL id="root" bbl:cmd="update"
      bbl:ver="1.0" bbl:context="/DIDL[1]">
  <Item>
    <!-- Global metadata -->
    <Item id="SMIL">
      <!-- SMIL presentation for DI -->
    </Item>
    <Choice/>
    <Item id="PROG_A"/>
    <Item id="PROG_B"/>
  </Item>
</DIDL>
```

Fig. 4: Random Access AU

7. REFERENCES

[1] E.Y.C. Wong, A.T.S. Chan, and H.V. Leong, "Semantic-based approach to streaming xml contents using xstream," in *Computer S/W and Apps Conf., 2003. (COMPSAC 2003). Proc. 27th Annual Intl*, 2003, pp. 91–96.

[2] U. Niedermeier, J. Heuer, A. Hutter, W. Stechele, and A. Kaup, "An mpeg-7 tool for compression and streaming of xml data," in *MM & Expo, 2002 (ICME '02). IEEE Intl. Conf. on*, 2002, vol. 1, pp. 521–524.

[3] M. Girardot and N. Sundaresan, "Efficient representation and streaming of xml content over the internet medium," in *MM & Expo, 2000 (ICME 2000). IEEE Intl. Conf. on*, 2000, vol. 1, pp. 67–70.

[4] S. Pfeiffer, C. Parker, and A. Pang, "Open standards for annotating and indexing networked media," http://www.annodex.net, Dec. 2003.

[5] D. Hong and A. Eleftheriadis, "Xflavor: bridging bits and objects in media representation," in *MM & Expo, 2002 (ICME '02). IEEE Intl. Conf. on*, 2002, vol. 1, pp. 773–776.

[6] M. Amielh and S. Devillers, "Bitstream syntax description language: Application of xml-schema to multimedia content adaptation," in *11th Intl. WWW Conf.* 2002, W3C.

[7] J. Thomas-Kerr and I. Burnett, "Bitstream binding language," http://www.whisper.elec.uow.edu.au/people/jtkerr/BBL.html, Dec. 2004.

[8] I. Burnett, R. Van de Walle, K. Hill, J. Bormans, and F. Pereira, "Mpeg-21: goals and achievements," *Multimedia, IEEE*, vol. 10, no. 4, pp. 60–70, 2003.