

2005

Efficient Delivery within the MPEG-21 Framework

S. J. Davis

University of Wollongong

I. Burnett

University of Wollongong, ianb@uow.edu.au

Publication Details

This article was originally published as: Davis, SJ & Burnett, IS, Efficient Delivery within the MPEG-21 Framework, First International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS 2005), 30 October-02 November 2005, 205-208. Copyright IEEE 2005.

Efficient Delivery within the MPEG-21 Framework

Abstract

MPEG-21, the new Multimedia Framework standard is nearing completion. It is centred around the concept of a Digital Item which is exchanged in the form of an XML declaration. This paper initially considers the challenge of compressing these Digital Item Declarations and presents conclusive results showing that schema based compression is the best solution. A new technique is then presented whereby the one-way delivery framework is expanded to allow client selection and control of Digital Item fragment transmission. This uses a new Remote XML Exchange Protocol (RXEP) which significantly reduces storage and time delays in mobile environments. Finally, the new approach is combined with schema-based compression to offer maximum efficiency.

Disciplines

Physical Sciences and Mathematics

Publication Details

This article was originally published as: Davis, SJ & Burnett, IS, Efficient Delivery within the MPEG-21 Framework, First International Conference on Automated Production of Cross Media Content for Multi-Channel Distribution (AXMEDIS 2005), 30 October-02 November 2005, 205-208. Copyright IEEE 2005.

Efficient Delivery within the MPEG-21 Framework

Stephen J Davis¹ and Ian S Burnett

*School of Electrical, Computer and Telecommunications Engineering
University of Wollongong
stephen@titr.uow.edu.au and i.burnett@elec.uow.edu.au*

Abstract

MPEG-21, the new Multimedia Framework standard is nearing completion. It is centred around the concept of a Digital Item which is exchanged in the form of an XML declaration. This paper initially considers the challenge of compressing these Digital Item Declarations and presents conclusive results showing that schema based compression is the best solution. A new technique is then presented whereby the one-way delivery framework is expanded to allow client selection and control of Digital Item fragment transmission. This uses a new Remote XML Exchange Protocol (RXEP) which significantly reduces storage and time delays in mobile environments. Finally, the new approach is combined with schema-based compression to offer maximum efficiency.

1. Introduction

MPEG-21 [1] is a framework standard aimed at making the consumption of multimedia content both simpler and more secure. A central aim of the framework is to ensure that multimedia content can be delivered to a wide range of devices; these are intended to range from the simplest wireless devices through to high bandwidth cinema quality video and audio systems. While there are challenges across the spectrum of devices, of particular interest are increasingly versatile mobile devices.

Many parts of the MPEG-21 standard are described though the use of XML [2] and in particular, the more structured XML schema [3]. Within MPEG-21, exchange of information is via the use of Digital Items (DI) which are structured collections of multimedia resources (content) held together according to relationships determined by the author of the DI. For any given DI, the constituent resources may have

accompanying rights expressions and descriptive metadata. These descriptors have a broad remit ranging from the terminal capabilities and multimedia content itself to the underlying network conditions. In terms of implementation, the DI is declared using a Digital Item Declaration (DID) validated on the basis of an XML Schema known as the Digital Item Declaration Language (DIDL) [4]. The XML DID provides a versatile method of describing the connected multimedia content, different choices of content, and the ability to configure delivery to better match terminal capabilities. However, while the DID is a versatile container, the consequence of its versatility will likely be long and potentially complex XML DIDs. One way of avoiding long XML declaration is to link to external XML descriptor files, but this just transfers the length problem to a further file retrieval.

In the case of mobile users, many of the mobile devices operate within bandwidth limited environments where the channel is shared amongst many users, and charging may be on a data volume basis (e.g. GPRS). For the MPEG-21 framework to be successfully mapped to mobile systems, it is important that mobile users are able to receive MPEG-21 Digital Item Declarations in a timely manner and be able to process them efficiently on their low power devices. Thus, it is vital that there are mechanisms to reduce the load associated with the storage, transmission and processing of the XML DID.

This paper examines in detail the possible approaches to compressing Digital Item Declarations and shows conclusively that one approach is consistently more effective. However, it is evident that compression alone does not exploit the two-way communication environment that is the norm in modern mobile networks. Thus, we propose a new extension to MPEG-21 in which a client can control the delivery of compressed fragments using a protocol.

¹ This work was partially funded by the Smart Internet Technology CRC.

Table 1 - Comparison of sample results (all units in Bytes)

DID	Initial	BiM	Bzip2	Xmill
T1	330,705	175,833	177,217	176,767
T2	29,478	2,045	2,305	2,142
T3	21,040	1,143	1,376	1,110
T4	15,300	10,926	11,452	11,097
T5	10,755	1,314	1,702	1,526

2. XML Compression Techniques

XML compression techniques may be applied to DIDs to decrease the overall file length and consequently, the effective size of DIs when transmitted to a user. XML compression techniques are primarily aimed at reducing the transmission cost of the structure but it may also be possible to combine this with compression of the actual data. In particular, some XML compression techniques allow the incorporation of other standard compression algorithms (e.g. gzip) to reduce significant portions of textual material.

To date, many lossless methods of compressing XML files have been proposed and demonstrated [5]. Generally, XML compression is regarded as being classified into three main techniques [5]:

- Redundancy methods such as WinZip and gzip.
- XML-Conscious/Schema based Methods such as Millau, XML-XPRESS, XMLPPM and XBIS.
- Hybrid methods (combination of schema and redundancy) such as XMill and BiM.

One motivation for performing the analysis was that MPEG has previously standardised a compression technique for MPEG-7, known as BiM [6]. This is a Hybrid method, tree-based encoder which uses a finite state automata approach to compression. While it was previously demonstrated to be highly effective for the complex, hierarchical MPEG-7 metadata, DIDs are quite distinct in that the simple 15-element schema allows the definition of a structure.

2.1. Experimental Results

Using the compression techniques listed in section 2.1 we evaluated each on its compression of some sample MPEG-21 DIDs. Table 1 illustrates just a small sample of the results obtained; we have only shown the techniques and test files which reflect informative results. The set of DIDs were chosen/generated to

represent typical XML descriptors which could be used in future MPEG-21 systems. Examples of test files include: a CD album with choices of bitrates (T1), university subject information dataset (T2), Tourist Photo album (T3), Simple picture embedded in DID (T4) and Movie trailers with linked pictures (T5).

From the results of Table 1, BiM performed better than the other compression programs resulting in the smallest files. This is reasonable as the majority of the files are defining mostly structure (in the DID) with a smaller proportion of data. The only exception to this result was for T3 (photo album) in which XMill generated the smallest file. T3 also contains mostly structure but has large repetitive sequences which XMill is able to exploit (XMill performs better than gzip since it is also XML aware).

Overall, schema and hybrid techniques performed better than redundancy techniques. This is not surprising since these DIDs contain a significant amount of structure information, the compression of which is the strength of these methods. The main disadvantage of redundancy techniques for XML is that they perform compression at the character level and do not consider the overall structure of the document. From the overall results, we conclude that despite the quite distinct nature of the DIDL XML schema, BiM performs best amongst current candidates for compression of DIDs.

2.2. Discussion

To provide interoperability, DIs often contain superfluous metadata, which is more than is required for each individual client. For example, consider a Digital Item containing video and audio. If a client chooses to only receive audio; all the video metadata is irrelevant. This is a problem with a general container such as a Digital item and while compression can help reduce the file size, it is still inefficient to deliver this redundant data. BiM currently provides a method to deliver compressed fragments (small sections of the XML document), however, there are no dynamic mechanisms to allow only selected fragments to be sent. This is vital, as clients may need to make changes to selections mid-stream in the consumption of a Digital Item. For example, a client may acquire access to an external screen and thus need video data to be sent as well as the corresponding video metadata.

Thus, while compression offers some advantages, the fundamental issue with the MPEG_21 framework is that it is locked into one-way delivery of content. The next section proposes a bidirectional communication protocol which provides client-side selection and control of Digital Item delivery.

3. Client Side Control

Since MPEG-21 is aiming to provide interoperability amongst many devices, it must be able to cater for all these devices in different scenarios. A clear example of this is facilitating reception of content on limited capability mobile devices. The problem with the standard MPEG-21 scenario is that the DID is pushed to a client device as a whole document (regardless of the ability of the device to consume that document).

In this section a new approach is proposed whereby a new XML protocol (Remote XML Exchange Protocol or RXEP) is employed to request small fragments of the DID on the basis of client need. This requires structured upstream messages to be standardised so as to communicate to the server peer the client's required fragments.

The proposed system is shown in Figure 1 and the following sections consider in detail the RXEP engine and the nature of the Fragment Request Units.

3.1. Remote XML Exchange Protocol (RXEP)

Remote XML Exchange Protocol (RXEP) is a method for requesting relevant data from an XML document located on a remote server (or peer). RXEP has the ability to create XML fragments building upon pull-parsing techniques (navigating on an element-by-element basis), or from queries (i.e. using XPath). RXEP queries can also receive multiple fragments at once, i.e. it is possible to use the XPath query //Item, to retrieve all <Item> elements in the remote XML document [7]. RXEP can operate on a request-response basis or in streaming mode. In the request-response scenario, commands are used to send instructions to the server [7], while in the streaming mode fragments are pushed to a client continuously on the basis of an initial request.

3.2. Fragment Request Units

To introduce the concept of client side control of XML DID delivery to the MPEG-21 framework, control/selection data must be sent to the sending peer(s). BiM [6] and the textual metadata mechanisms (TeM) [6] both define methods for delivery of a fragmented XML documents through Fragment Update Units (FUU). It is possible to use these mechanisms for delivery but it is necessary to create a new upstream messaging system. The new, proposed upstream fragment creation/selection data will be referred to as Fragment Request Units (FRU). In defining these, the following requirements were identified:

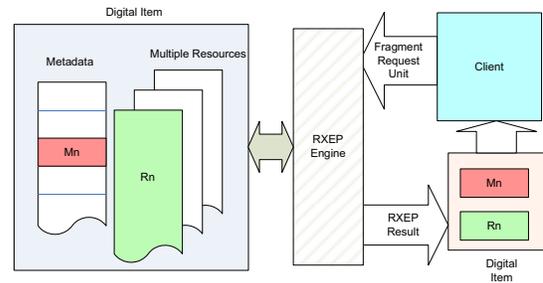


Figure 1 – Block Diagram of system

- Navigation through a remote DID – ability of a client to navigate through a remote DID such that branches of the XML ‘tree’ are selected for retrieval;
- Request of fragments – the client should be able to select fragment(s) to be delivered based on a query;
- Request of a stream of fragments – the client should be able to send a single request and the pre-defined fragments are ‘pushed’ or streamed to the client (using e.g. mechanisms supported in BiM and Tem).
- DID Choice Selection transmission - DIDs also contain <Choice> elements which allow DI authors to configure and tailor a DI to meet a user’s or terminal’s needs. As the client will not have access to the entire DID locally to completely resolve the <Choice>s, it is advantageous for the FRUs to deliver the <Choice> selections made by the User or terminal. These choices can then be used by the server peer to automatically select the correct parts of the DID, reducing the number of fragment requests

Since FRUs are capable of requesting any fragment (based on fragment size and location), clients have random access to the DID. This allows a client to jump into a DID (or other linked XML), or to simply, “navigate backwards” if previous fragments are not cached. For example, a client displaying anchored metadata from a DID as song lyrics may need to skip ahead 2 minutes in the song. In the new bi-directional framework, an FRU can be sent to immediately request the lyrics anchored to the new location.

The requirements for FRU messages are fulfilled using the RXEP mechanisms with minor extensions to allow extra information to be added (such as DID <Choice> selection). RXEP XML messages can also be compressed using tree-based compression [7].

3.3 Overall System

This section considers the overall approach shown in Figure 1 in more detail. Clients first initiate a

```

(a) <FRU>
    <Get>/mp21/FavMP3list.xml</Get>
</FRU>
(b) <FRU>
    <XPath>//Item[id="mp3list"]</XPath >
</FRU>
(c) <FRU>
    <Stream>/DIDL/Item[3]</Stream>
</FRU>

```

Figure 2 –Simple FRUs

connection to the sending peer and request the DID they require (as shown in the example of Figure 2(a)); the received DID is then a ‘skeleton’ of the complete DID. This skeleton is valid according to the DIDL and gives the client enough starting information to begin consumption of the remainder of the DID. In some cases, the sending peer will include important <Choice> elements required to configure the DID, thus requiring that certain fragments be sent.

Using the skeleton DID, the client can construct FRUs to request the required parts of the DID. These FRUs may simply contain navigation commands, or in more ‘intelligent’ cases, XPATH queries to retrieve multiple fragments (for example, select all Items whose id is mp3list as shown in Figure 2(b)). A client may also request a stream of fragments e.g., stream all elements under the 3rd Item of the DID as illustrated in the FRU of Figure 2(c). The FRU may also contain any <Choice> selections made by the client in the consumption of the DID. This allows the server to correctly interpret FRU navigation of the DID.

The server peer decodes the FRUs and generates fragments customised to the individual client. The server peer may optionally package any requested resources into the new DI, which can be compressed using a tree-based compression technique.

3.4. Example Deployment of RXEP

RXEP needs to be supported by each client, however, each serving peer does not necessarily need to be RXEP enabled. The RXEP engine could be placed on the borders between fast and slow networks, as illustrated in Figure 3, or simply as a local content mirror. For example, a mobile provider could provide this service to all its clients, even though the locations from where the actual content being request does not support it. The DIs would be downloaded locally to the gateway and the gateway, supporting RXEP, can then deliver the DID in fragments to the client. Alternatively, if the content locations also support

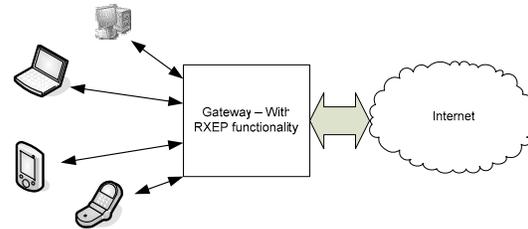


Figure 3 – Example of RXEP placement in a mobile Environment

RXEP, then the RXEP gateway would be able to simply cache requested fragments, only storing the most commonly used parts of DIs.

4. Conclusion

As MPEG-21 is centred around DIs, it is important that these be delivered efficiently. Due to the schema basis of DIDs and the large amounts metadata required for interoperability, tree based encoding (such as BiM) is a solution for compression of DIDs. However, compression is only one part of the solution, as it does not reduce irrelevant data transmission. This can only be achieved if the client communicates requests for fragments of the DID. This client side control approach, combined with compression, creates an efficient method of delivering custom DIs to clients. Further we have demonstrated that this bidirectional messaging mechanism can be easily incorporated into the MPEG-21 Multimedia framework.

5. References

- [1] Bormans and K. Hill, "MPEG-21 Overview v.5," ISO/IEC JTC1/SC29/WG11/N5231, October 2002.
- [2] W3C, "Extensible Markup Language (XML)," [online document] <http://www.w3.org/XML/>.
- [3] W3C, "XML Schema," [online document] <http://www.w3.org/XML/Schema>
- [4] Information Technology, "Multimedia Framework (MPEG-21)—Part 2: Digital Item Declaration," ISO/IEC 21000-2:2003, Mar. 2003.
- [5] M. Cokus and D. Winkowski, "XML Sizing and Compression Study For Military Wireless Data," *XML Conference and Exposition 2002*, Baltimore convention center, Baltimore, MS USA, December 8-13 2002.
- [6] Information technology, "Multimedia content description interface -- Part 1: Systems," ISO/IEC 15938-1:2002.
- [7] S. Davis and I. Burnett, "Exchanging XML Multimedia Containers using A Binary XML Protocol," *IEEE ICME 05*, 6 – 8 July, 2005