The Use of Binary Representations of XML Information Sets in Digital Broadcasting Systems

Nigel Dallard (ndallard@ndsuk.com)
Advanced Technologies Group, NDS Ltd.

Introduction

Alongside the compressed audio and video components of digital TV channels, a wide variety of other digital data and metadata is conveyed, in various compact binary formats to minimise bandwidth requirements:

- Data necessary for the correct basic operation of the receiver – for example, basic tuning information. This is usually a few hundred kbps.
- Data to populate the electronic programme guide with details of all the available channels and their schedules. With systems offering access to several hundred channels, and providing details of all programmes for several days, this is a massive amount of data – today, broadcasters often dedicate between 5Mbps and 15Mbps to this data.
- Data to drive interactive TV applications. Interactive versions of TV programmes (for example play-along versions of quiz-show "Mastermind" and comedy show "Banzai") rely on downloaded applications, running on the receiver whilst the viewer is watching TV, being supplied with timely data and triggers. Stand-alone interactive applications can also be deployed (for example shopping catalogues such as the "QVC Directory" and information services such as "Teletext"). These interactive TV applications can consume anything from a few hundred kbps up to 5Mbps – at least as much bandwidth as, if not more than, a standard-definition TV channel would consume on average.

Further examples of the kind of applications available on TV systems around the world today are available at:
http://www.nds.com/applications_showcase/applications_showcase.html.

NDS provides interactive TV infrastructure – head-end servers and receiver software – to digital broadcasters and content providers around the world. In this paper, I will provide some background information about the environment in which we deploy our interactive TV solutions, and some of the technical challenges that we have overcome. I will explain our solutions to these challenges, which are based on the use of XML and the subsequent translation of this data to a compact binary form. Finally I will present some figures which show the reduction in the broadcast bandwidth required that can be obtained by the appropriate use of binary information sets.

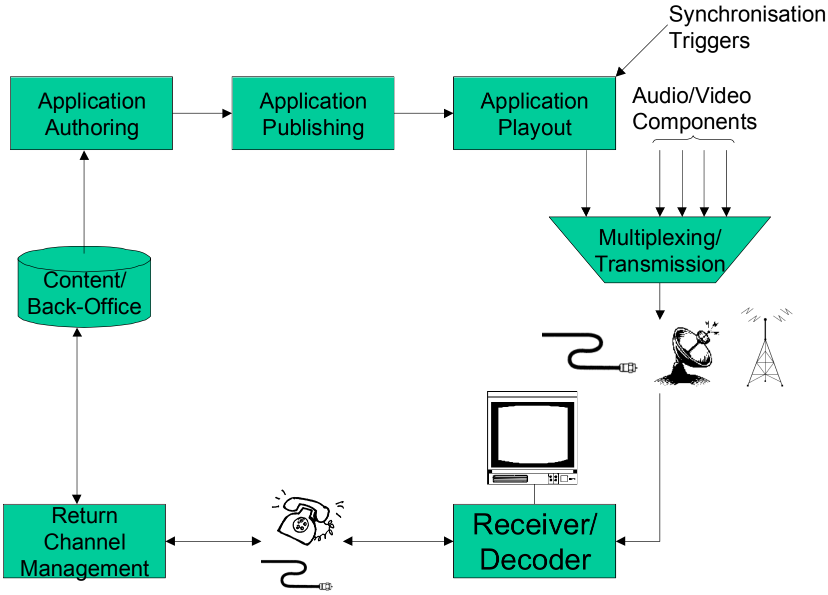
The Digital Broadcast Environment

The diagram below illustrates a typical interactive TV system.

The application authoring and back-office functions are standard computing platforms. NDS uses XML as a means for storing and conveying information between servers and applications at this point, enabling us to benefit from the wide industry support for this technology.

At the publishing stage, the various files and data-sets that make up the application are pulled-together and "packaged" for broadcast. It is at this point that NDS translates between textual XML and a binary representation of the data.

The playout function inserts the data into the digital multiplex for transmission. Depending on the actual application, the data may be streamed or carouselled (repeatedly cycled) into the multiplexer, and may be synchronised with the audio-visual content.



Satellite, terrestrial and some cable broadcasting networks do not have any sort of "always on" bi-directional interaction channel available to them. The applications therefore rely on the "head-end" servers pushing data into the transmission in a timely fashion, using combinations of streams of time-sensitive data and carousels of repeating data as appropriate. Applications contact the broadcaster via dial-up PSTN only when absolutely necessary, either to send data to the broadcaster, or to receive data that it is either not appropriate to broadcast (e.g. online banking), or simply not cost efficient to do so (e.g. personalised pages of information).

Some cable networks do have the luxury of an "always-on" interaction channel. Even here though, it is often more bandwidth-efficient to broadcast the majority of the downstream data rather than supply it separately on request from each receiver. If each receiver requests the data separately, the interaction channel and the data server can become overwhelmed with requests for the data at certain times. There is, however, obviously more flexibility available to the application designer as to when to switch from the broadcast delivery of data to the unicast, on-demand delivery method.

Typical receiver/decoders have restricted processing power and limited memory. This means that the transmitted application data needs to be small in size and easily-parsed with minimal processing. Often the complete data set is larger than can be held in the receiver/decoder's memory. The data must be structured in such a way that the application can work with only certain fragments of the data in memory at any one time. As there is no ability to cache data at the receiver, it has to be transmitted on a carousel, with delays occurring whenever the receiver has to await the next transmission of the data it requires. The data must also be able to be incrementally updated – simply marking the whole data set as out-of-date each time a minor change is made to one part of the data, requiring the applications to re-acquire and parse the whole data set again is not acceptable.

Evaluation of XML-derived Transmission Formats

NDS has investigated the current "state-of-the-art" in binary representations of XML information sets. This investigation highlighted a number of potential formats for our application data, and technical papers describing comparisons of their performance. Using the available information, we eliminated a number of the possible contenders as not being appropriate for our particular scenario, before proceeding with some comparative tests of our own.

The potential application data formats include:

- **Compressed XML** – The use of a generic data compression algorithm such as Lempel-Ziv, Huffman or Burrows-Wheeler, as used in utilities like WinZip, gzip and bzip2. This technique was rejected as being unsuitable for our purposes. It removes our ability to split the document up into smaller fragments, and it is not very efficient at compressing small fragments individually. It also requires that the binary document be converted to its original textual form before use, requiring both additional processing power and memory.
- **Millau**[1],[2] – An extension of the WAP Forum's WBXML to generic XML documents with the addition of content compression. Millau allows the binary document to be processed via both the standard DOM and SAX APIs without having to expand it back into its original textual format, however the reported compression ratio (XML source to binary format) was 4:1 or less for small documents. This format was therefore rejected.
- **XMill**[3] – This tool uses a generalisation of the column-wise compression used in relational databases. Content-specific encoders are applied per content-grouping, with gzip applied as a secondary and default compression algorithm. Various papers showed that XMill was not particularly efficient on small documents, so it was rejected.
- **MPEG-7 BiM**[4],[5] –As the technology specified by the standards body responsible for the MPEG-2 standard used by digital TV broadcasting systems, this was the "obvious" format for us to utilise. It is aimed directly at our application, it is very efficient at coding, and the client is able to directly parse the binary document.
- **ASN.1 PER**[6],[7],[8] – The XML Schema is converted to ASN.1 according to the draft ITU-T Rec. X.694. The data is subsequently encoded according to ITU-T Rec. 691. ASN.1 has been around for many years, and it is used in interactive TV middleware based on the MHEG-5 (ISO/IEC 13522-5) standard. The current work on a standardised mapping from XML schema to ASN.1, together with the very efficient Packed Encoding Rules make this a viable technique for our application.

Evaluation Results

In this section a summary of the results of our investigations are presented. The size of the data in XML format was compared with the size of the same data transformed into MPEG-7 BiM and ASN.1 PER.

Data Set #1: Programme Schedule Information

This data set consists of small fragments of programme schedule information (information about 1, 4 and 8 programmes) such as would be used to populate an electronic programme guide. Typical EPG data transmission formats split the multi-channel, multi-day listings data into segments containing the programme listings for a three-hour period on a single channel, hence the representative numbers of programmes coded. For the purposes of comparison, the data was also encoded into a DVB-SI (ETSI EN 300 468) Event Information Table segment, as used by many digital TV broadcasting systems today.

	8 Programmes	4 Programmes	1 Programme
Source XML	9317 bytes	4868 bytes	1350 bytes
DVB SI EIT	1050 bytes	531 bytes	154 bytes
MPEG-7 BiM	991 bytes	497 bytes	148 bytes
ASN.1 PER	1049 bytes	525 bytes	153 bytes

Data Set #2: Stock-Price Data

This data set consists of stock-price information from a "stock-ticker" application. The application uses three different sets of data which are transmitted concurrently – real-time stock price changes, intra-day data for each stock and 60-week historical data for each stock. For comparison, the data was also manually encoded into a compact binary form.

	Real-Time	**Intra-Day**	**60 Week**
Source XML	640 bytes	16518 bytes	10421 bytes
Manual Encoding	33 bytes	1620 bytes	1441 bytes
MPEG-7 BiM	34 bytes	1622 bytes	1443 bytes
ASN.1 PER	32 bytes	1620 bytes	1441 bytes

Conclusion

The results above show that a compact binary representation of interactive TV application data can be produced that is between 5% and 15% of the size of the same data presented as textual XML. Without this saving in bandwidth, it would be uneconomical to broadcast many of the interactive applications that are deployed today, and future developments in this area would be severely constrained. In addition, the compact binary representation allows the deployment of these applications to low-end receiver/decoders which are constrained in both memory and processor power.

References

1. Marc Girardot, Neel Sundaresan. "Millau: an encoding format for efficient representation and exchange of XML over the Web", *http://www9.org/w9cdrom/154/154.html*
2. Neel Sundaresan, Reshad Moussa. "Algorithms and Programming Models for Efficient Representation of XML for Internet Applications", *http://www10.org/cdrom/papers/542*
3. "XMill – An Efficient Compressor for XML", *http://www.research.att.com/sw/tools/xmill*
4. ISO/IEC 15938-1:2002 "Information Technology – Multimedia Content Description Interface – Part 1: Systems"
5. ISO/IEC JTC 1/SC29 WG11 Moving Picture Experts Group. "MPEG-7 Overview" Section 3.5, *http://www.mpeg-industry.com/mp7a/w4980_mp7_Overview1.html*
6. ITU-T Rec. 681 (2002) | ISO/IEC 8824-2:2002 "Information Technology – Abstract Syntax Notation One (ASN.1): Specification of Basic Notation", *http://www.itu.int/ITU-T/studygroups/com17/languages/X.680-0207.pdf*
7. ITU-T Rec. 691 (2002) | ISO/IEC 8825-2:2002 "Information Technology – ASN.1 encoding rules: Specification of Packed Encoding Rules (PER)", *http://www.itu.int/ITU-T/studygroups/com17/languages/X.691-0207.pdf*
8. Draft ITU-T Rec. X.694 | ISO/IEC 8825-5 "Encoding XML-Defined Data Using ASN.1", *http://asn1.elibel.tm.fr/xml/#schema-mapping*