The W3C Workshop on Binary Interchange of XML Information Item Sets

Position Paper
Systematic Software Engineering

Prepared for

W3C (Liam Quin)

by

Systematic Software Engineering Ltd.
The Coliseum
Riverside Way
Camberley
Surrey GU15 3YL

Revision

Rev 1.1 06 Aug 2003 ADG
Initial revision.

Rev 1.2 06 Aug 2003 ADG
Updated after internal review comments from Bejorn & ADG.

Table of Contents

1 Introduction

This paper has been prepared in response to the W3C call for participation in 'The W3C Workshop on Binary Interchange of XML Information Item Sets', 24th – 26th of September 2003, Santa Clara, Calafornia, USA.

Systematic Software Engineering has been providing a range of XML related information exchange capabilities to the Defence and Healthcare sectors for many years. Due to the limitations imposed by the relatively low bandwidth radio environments that many of our defence customers must use to exchange structured information, Systematic has already invested significant resources in developing XML compression software, utilising knowledge of the XML structure and data content provided by W3C XML Schema, to produce the most bandwidth efficient binary representation. Therefore, Systematic has a natural interest in the proposed workshop.

Systematic believe that we can provide a number of useful inputs to the workshop and any subsequent working group activities, such as:

- example compression ratios
- results of tests already conducted using different compression techniques, e.g. gzip versus our own software implementation on different types and size of schema and content
- suggestions for other knowledge based XML compression techniques that we have not yet implemented but may be useful research candidates
- R&D effort to implement and evaluate further suggested concepts produced by the group

Systematic will help ensure that the work of the group is utilised to the full by implementing any resulting specifications in our products and ensuring standards compliance where appropriate.

1.1 Document Structure

The remainder of the position paper is structured as follows:

- **Systematic Background,** introduction to Systematic and further detail on our experience working in the XML arena
- **The Systematic XML Compression Software Implementation,** provides an overview of our current work, the approach adopted and sample results
- **Priority Goals for XML Compression,** describes some of the background behind our priorities for a bandwidth efficient, binary representation

2 Systematic Background

2.1 Company Background

Systematic Software Engineering Ltd has been developing software products and services for the exchange of structured information for over 15 years. Our products have been used in numerous different business sectors and more recently we have focused on the Defence and Healthcare sectors. Over the years we have built up a technology and knowledge base to support a number of different sector specific information exchange standards, each with their own peculiar attributes designed to suit specific needs. In commercial sectors we have developed EDIFACT and ANSII X.12 based implementations. In defence we have implemented a range of standards, both character based and bit oriented, where the design goals for the type of encoding used vary between the man readability of the information exchanged to using the minimum amount of bandwidth possible to exchange the information.

2.2 Relevant Experience

Over recent years, XML has been become the defacto standard representation throughout the commercial sectors. It is also been widely adopted throughout defence organisations to improve the potential for using a wider range of COTS products for exchanging, processing and presenting structured information. Systematic is a leading software vendor in this field, with a range of software products that help XML enable a number of defence legacy standards:

- W3C XML Schema generators to convert the existing standards definitions and rule sets into a standard set of W3C compliant XML Schemas
- XML converters that translate documents encoded using the legacy formats to/from XML for easier processing using the wide range of XML technologies now available
- Presentation tools that simplify the use of HTML and XSL-T to produce highly customised user friendly presentation forms for information that must be encoded according to legacy rule sets
- Sophisticated information mapping tools that support the automatic code generation (.NET & Java) for mapping between different structured document standards and databases. The tool supports a range of character and bit oriented defence standards, XML and databases such as Oracle, Sybase and SQL Server.

2.3 The Need for a More Efficient Representation

It is well understood now that there are numerous interoperability gains to be achieved by standardising on an XML data encoding scheme for a variety of information exchange needs. However, one of the major disadvantages XML has compared with many legacy/domain specific standards, especially in the defence arena where there is often a need to exchange information over low bandwidth radios, is the rather verbose nature of the mark-up compared to the amount of actual data content being transmitted. In certain areas of the Defence arena, this drawback is one of the factors behind domain specific standards, with their own encoding rules, still being preferred to XML making interoperability across all parts of the defence arena more difficult and expensive.

The most obvious initial answer is to use GZIP to compress XML for use in such bandwidth limited environments. For some types of document the results can be quite impressive, e.g., where the documents are quite large, there is a large amount of free text or there is a high degree of repetition in the tags. However, due to the very general nature in which GZIP works, in many circumstances the results are far less impressive - sometimes resulting in no gain at all, e.g. when the documents are relatively small and/or there is little tag repetition. Thus, when compared against encoding schemes that have been designed with minimum bandwidth as a priority, a GZIPPED XML equivalent is less efficient.

3 The Systematic XML Compression Software Implementation

3.1 Overview of the Systematic Approach

In order to address the low bandwidth problem and attempt to provide the most efficient data transfer mechanism, Systematic have worked with Qinetiq (UK based research establishment) to build a knowledge based compression engine with the aim of producing consistent compression results regardless of the size of message being compressed. Where GZIP is a general-purpose compressor that uses a combination of the LZ77 algorithm and Huffman coding [Deutsch, 1996], our approach relies on the knowledge about the legal structure and content of the XML documents being compressed, inherent in the associated W3C compliant XML Schema. A local copy of the schema is required for both the compression and decompression algorithms to ensure that no information that can be deduced from the schemas is transmitted. The XML document must be schema valid for the compression to work.

Our complete approach has proved superior to using GZIP, with more efficient compression in all cases and consistent results regardless of how small the document to be compressed.

3.2 Compressing Structure

The structure of legacy defence formats is characterised by the fact that elements, or groups of elements, are sequentially ordered. For more generic unordered XML Schemas, the XML document can reordered into schema sequence so the same encoding method can be adopted. Knowledge of the order eliminates the need for mark-up to identify the individual elements. Only indications of the existence of optional and repeatable elements are needed.

Elements that have alternative representations, e.g. a position may be specified in either WGS-84 (World Geodetic System of 1984) or UTM (Universal Transverse Mecator), are preceded by a code indicating the alternative used.

Complex types, such as time and position, are also sequentially ordered, which eliminates the need for markup to distinguish the individual parts. Consider a simple time element, which expressed in XML may look like this:

```
<simple_time>
  <hour>08</hour>
  <minute>15</minute>
</simple_time>
```

The legacy defence equivalent of this is:

```
TIME/0815//
```

In our compression approach we only need to keep the data content, i.e. the “0815” value, as it easily can be parsed into the appropriate parts.

3.3 Compressing Data Content

The data content of legacy defence formats is restricted to numbers, strings, or enumerations. Data content may be further restricted by numerical ranges or regular expressions.

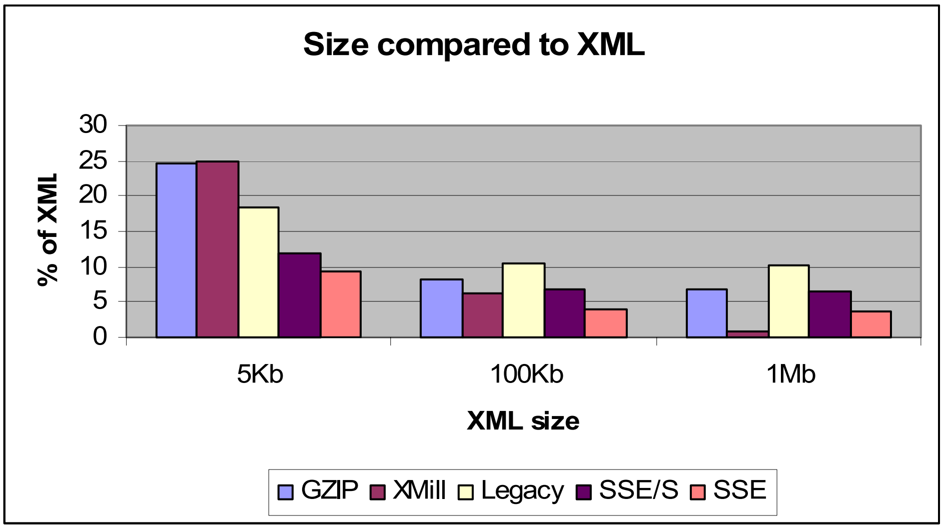
Different coding techniques are used depending on the nature of the individual data elements. Numerical data content is coded with the minimal number of bits for the individual number determined by the numerical range. Textual data content are typically Huffman [Huffman, 1952] or Arithmetic [Witten el al., 1987] coded. Statistics can improve the compression ratio, although the sender and receiver must agree on the statistics used. Enumerations, which typically are textual descriptions such as month names, are coded as numbers.

Data content elements may contain unused values, which can be harnessed by combining elements during compression. As an example, an enumeration for a month element could have 12 legal values. To represent these you need a minimum of 4 bits, but these four bits have a potential to represent 16 different values, so there are 4 spare values that could be utilised for the values of another enumeration.

3.4 Measurements

The diagram below compares our approach to other approaches. The primary axis shows measurements of XML documents of varying size. All documents used are representative of realistic legacy documents. The secondary axis shows the size of the compressed document compared to the original XML document. Thus, the smaller the value, the better the compression.

GZIP and XMill should be familiar to most. Legacy is a character based legacy defence format, to which the XML document has been converted without the use of any compression techniques. Two sets of numbers are provided for our approach. SSE/S only consists of the structural compression, and SSE additionally uses Huffman coding for the free text data content.



The results for GZIP and XMill are similar to those found by other experiments [Cokus and Winkowski, 2002]. XMill does well on large documents. Surprisingly, Legacy does better than GZIP and XMill on small documents; recall that no compression was applied in the conversion from XML to the legacy format. This shows that legacy formats, designed with bandwidth efficiency as a major goal, are better suited than XML for their operating environment unless the overhead of the mark-up scheme can be completely removed through intelligent compression. SSE/S shows that structural compression is significant for small documents, and SSE shows that data content compression further reduces the size for larger documents. For the numbers presented in the table above, free text content was only compressed using plain Huffman coding so the compression ratio for the SSE approach can be further improved.

4 Priority Goals for XML Compression

4.1 The Low Bandwidth Environment

From a Systematic perspective, in the particular markets that we work and the types of problems our customers face, creating the most efficient bit oriented encoding scheme to enable the transfer of structured information using the minimum amount of available bandwidth is the key priority. Our customers are often faced with the problem of having to prioritise, or even not send information that might be useful, due to the very low bandwidth environments in which they are operating; sometimes providing as little as 2 KBit or lower throughput.

The types of document that are being transmitted can often be relatively small, sometimes as low as just a few Kilobits of data, but the overriding goal to reduce the use of bandwidth to a minimum remains just as important for the small documents due to the potential high number of documents required to be exchanged over the low bandwidth link in a short space of time. It is therefore vital that any XML compression technique used must operate efficiently on small documents as well as large ones.

4.2 Other Compression Goals

Relative to the low bandwidth problems, other issues such as storage space are much less of a problem, with higher capacity, faster, smaller storage devices appearing on the market all the time. So the need to provide a permanent bit oriented storage format with direct access facilities is much less of an issue.

5 Conclusion

The Systematic XML Compression software attempts to facilitate the use of XML standards in operating environments that would otherwise be completely inappropriate. The philosophy being that once exchanged, recipient systems will convert the received compressed bit oriented data into a standard character based XML representation for local storage and use simplifying the further integration of that data with the range of applications that might require it.

When assessing the needs for a compressed, bit oriented XML representation throughout the IT industry as a whole, there are probably a wide variety of priorities regarding the performance, efficiency and features that should be available. For a successful open specification to emerge, it will need sufficient flexibility to permit the use of the most appropriate compression technique for specific circumstances (in our case efficient compression of relatively small messages).

The availability of an open W3C endorsed specification would enable the potential inter organisational interoperability to occur actually over the low bandwidth link, potentially with the sender and receiver being free to adopt different vendor’s software. If such a specification is endorsed by the W3C, then Systematic wish to ensure that our XML Compression software product is compliant with it.

6 References

Cokus, M.; Winkowski, D. (2002) “XML Sizing and Compression Study For Military Wireless Data”. XML 2002 Proceedings.

Deutsch, L. P. (1996) “RFC 1951: DEFLATE Compressed Data Format Specification version 1.3”. IETF

Huffman, D.A. (1952) “A Method for the Construction of Minimum Redundancy Codes”. Proceedings of Institute of Radio Engineers, vol. 40 (9), pp. 1098-1101.

Witten, H.; Neal, R. M.; Cleary J. G. (1987) “Arithmetic Coding for Data Compression”. Communication of the ACM, vol 30 (6), pp. 520-540.