The L-3 Communications, Integrated Systems position regarding

Binary Interchange of XML Information

11 August 2003

Bill Eller
L-3 Communications
Integrated Systems
CBN 095
10001 Jack Finney Blvd.
Greenville, TX 75402
903.457.6301
bill.eller@L-3com.com

Contents

1 L-3 Communications and Binary XML

Design and implementation of communication structures entails a constant and continual battle or tradeoff between flexibility and bandwidth usage. This is especially the case for low bandwidth mediums where the priority typically must lean toward the minimization of data size at the expense of flexibility. L-3 Communications, Integrated Systems (L-3/IS) in Greenville, Texas has since the 1970s developed numerous communications systems and frequently dealt with this dilemma.

In early 2000, L-3/IS was again faced with such a new development in support of a contract with the U.S. Air Force Detachment 2 of the 645th Material Squadron. L-3/IS immediately identified XML with its superb flexibility and extensibility as a nearly ideal solution save one major limitation. The document size associated with the fact that XML is character-based would totally preclude its use, and thus its many benefits, if a method for reducing the size could not be identified. A search for existing binary XML solutions with flexibility, extensibility, and vocabulary independence was not fruitful.

Consequently, L-3/IS set forth and developed a binary representation of the XML language. L-3/IS and Det. 2 645th MATS believe that this binary representation may be an ideal foundation for the definition of a W3C binary XML standard and would likewise be interested in aiding its pursuit and potentially coordination with ongoing government efforts.

2 Two Representations from One DTD

Utilizing the standard structure and capabilities of the XML DTD, a set of pre-defined attributes, rules, and a common parsing methodology have been developed which provide for a single DTD to support both a binary representation and a fully equivalent XML representation of any single data vocabulary. Both representations contain the same information with the same attributes.

The current contractual vocabulary implementation is called Common Message Format or CMF. The binary representation of the vocabulary is called CMF-B and the XML character-based representation is called CMF-X. The CMF-X representation is fully operable as a subset of the complete XML capabilities. The critical value of CMF-B as compared to other potential binary structures is that a CMF-X representation can be rendered into the CMF-B representation and vice-versa utilizing a common parsing methodology without any loss of information, structure, and without data “translation”.

The CMF-X representation, being a subset of the full XML standard, conforms to the same definition and use for the terms “well-formed” and “valid” as XML. Since CMF implements only a subset of XML (e.g. no mixed content models), some XML does not conform to CMF rules. On the other hand, all CMF-X data does fully conform to XML rules and therefore CMF-X data can be utilized with and operated upon by standard XML and Web Service tools and technologies such as XSLT, browsers, schemas, etc.

As the CMF-B representation relies upon the use of a DTD for parsing and/or rendering to CMF-X, the CMF-B data must always be “validated”. Additionally since CMF includes data typing, an optional additional level of conformance checking for both CMF-X and CMF-B called “verification” is possible against the data type ranges, units, repetitions, etc. As CMF-B is not actually pre-parsed, the *View Source Principle* is still supportable via a binary display.

3 L-3/IS Requirements, Activities, and Intentions

CMF is developed for and is being implemented on a Department of Defense (DoD) program called the Interactive Broadcast Service (IBS). The data on IBS is tactical situational awareness information (i.e. both geographic and identification) which is exchanged interactively among multiple network participants and provided in near-realtime directly to the warfighter. CMF is being deployed by IBS for use within tactical military radios, desktop personal computers, workstations and servers.

The CMF implementation including the language and an IBS-specific vocabulary has been assigned a namespace within the DoD namespace registry and is anticipated to be approved as a military standard (MIL-STD). Efforts are also underway to support multiple vocabulary interoperability within various DoD communities of interest via vocabulary harmonization and/or use of namespace features. Although IBS has a specific vocabulary defined, the design and structure behind CMF is interoperable between vocabularies as well as independent of any single application or hardware device.

Applications which are applicable to the IBS mission regarding use of CMF include:

1) data dissemination,
2) interactive data exchange,
3) database storage and retrieval,
4) dynamic geographical plot/display,
5) textual/tabular information display,
6) translation to/from non-XML data formats (preferably via standards such as XSLT),
7) automated production of test data and procedures based upon vocabulary definition (e.g. ranges, etc.)

and likely others.

The primary requirements (i.e. goals) of the IBS program which prompted the development of CMF are two-fold. First, a language is required which is both extensive and flexible enough to support the merge of four separate legacy formats/vocabularies. Second, the combined vocabulary must be supported over existing mediums with extremely narrow bandwidth capabilities (i.e. maximum of 9600 bps) with documents of around 1 Kbytes. Processing time is also a concern but is not currently restrictive given the limited bandwidth versus available processor speeds.

XML is an obvious solution to the first requirement, but the second requirement, as mentioned, necessitated the identification of a binary implementation. Development of a language which supports both the character-based and binary representations from a single DTD-based vocabulary enables the use of standard XML tools and applications while also supporting a standardized and lossless compression into the binary representation during any narrowband transmissions.

4 Answers to Additional W3C Concerns

The CMF binary representation differs from gzip on raw XML in that knowledge of the XML structure is used as a form of domain compression rather than either data domain knowledge or generic compression. By taking into account that all of the data to be rendered into binary is in the form of the XML structure, a reduced-size binary structure was able to be identified which leveraged a number of known XML features. The resulting binary representation is not actually a “pre-parsed” XML, a typical data domain knowledge compression, nor a generic compression. It would more accurately be termed a structure knowledge compression. Although it is not really “pre-parsed”, a structurally self-contained termination simplifies parsing (i.e. no end tags) and the uniquely byte-aligned data nearly eliminates bit level operations. Both facilitate a minimization of parse processing.

CMF supports data streaming, random access, and dynamic update. The generic parser software developed as part of the IBS program uses stream-based processing similar to SAX to decode documents into a DOM tree which is then available to applications for access and update. The IBS parser software also supports encoding the DOM tree structure into a document. These capabilities are critical to CMF use on the IBS program.

Measurements have indicated a reduction in document size of approximately 18 to 1 between the character-based CMF-X representation and the CMF-B binary representation of information. Where desirable, further zip-based compression of the CMF-B representation is possible and has been tested with results, as expected, directly dependent upon both document size and algorithm. Additional compression of 40% or more is achievable with document sizes of a minimum of 1K bytes.

It should be noted that, for the current L-3/IS implementation, the reduction in document size is not translated into a corresponding reduction in processing memory when using a DOM tree since the standard tree was enhanced to support both the binary and character-based representations. For the IBS program, memory size is a consideration, but not necessarily a critical criterion given the rapidly and continually shrinking form factor and cost. On the other hand, where only support for the binary representation is required, it may also be possible to devise a smaller DOM tree.

Although the CMF language provides support for data typing (via reserved attributes), all data type bit representations are not restricted to any range boundary and are in fact infinitely extensible (full CMF bit representation information is too detailed for inclusion herein). For example, the integer type is not bounded to a byte, double-byte, etc. and the floating point type is not restricted in either mantissa or exponent. Although not bounded by the design or structure, capability is provided to bound data types in range, unit, etc. for a given vocabulary when desired. String data is currently only supported with 7-bit ASCII, but it is conceivable that strings could be internationalized to other character sets. Likewise, accessibility within CMF-B has been neither directly addressed nor specifically precluded.

5 Overview of CMF Structure

There are six type attributes provided for representation of data values (i.e. #PCDATA items, aka “fields”) in CMF:

- STRING - Character values
- INTEGER - Positive natural numbers (including zero)
- FLOAT - Floating point and negative integer numbers
- ENUMERATED - Pre-definable character strings with assigned numbers substituted in CMF-B for efficient transmission
- PATTERN - Values which must contain only pre-definable combinations of characters and/or digits
- PACKED_COMPONENT - Boolean/two-state values

Instances of each of the binary value representations provide a self-contained end-of-field indication and are byte-aligned. Additionally, the majority of FLOAT element value instances are expressible in 16 to 24 bits and FLOAT binary values optionally have unit, accuracy, greater than/less than indication, and accuracy greater than/less than capabilities all as one binary unit.

For both further size savings and to enable mapping to/from the binary representation, CMF also utilizes a set of five definable element types which identify the structure of CMF elements. The set of element types is a subset of the possible combinations of the XML content model structure. The use of these predefined element types permits the identification, and therefore the use, of pre-defined structures within the CMF-B. This thus avoids the inclusion of numerous tag bytes and significantly reduces document size. The element types are critical to the CMF-B structure and fundamental for rendering to and from standard XML structure (i.e. CMF-X).

The five element types include:

- FIELD - Elements with values (one of the six data representation types)
- GROUP - Provides organization of other elements some of which are not always sent (one or more elements in the content model may be OPTIONAL)
- COMPOSITE - Groups elements always sent together (all elements REQUIRED in the XML content model)
- REPETITIVE - Identifies repetitions of same type elements or same element groups (all elements REQUIRED in XML content model and entire content model identified and repeatable)
- PACKED - Groups multiple PACKED_COMPONENT (i.e. Boolean) elements

All elements may have a binary tag attribute assigned. In CMF-B, the binary tag replaces the normal character based start tag used by standard XML. The binary tag is represented using the same bit representation as the INTEGER type. No stop tags are used since lengths are known from a self-contained termination indicator in each byte. All components of the grouping structure are also byte-aligned.

CMF-B instances of COMPOSITE and REPETITIVE elements do not include intermediate tags for the required children. Likewise, the PACKED elements represent up to three boolean children values per byte without the tags of the child elements (note: this is the only bit level representation type, but the set of three is still on an overall byte boundary). Similarly, the REPETITIVE element is self-defining for the number of repetitions in the instance. Each of these structures is compliant with a form of the XML content model capability and each facilitates savings in CMF-B document size.

There are further characteristics of CMF which have significant contribution to the minimization of data size. Unlike most pre-defined data typing mechanisms, all instances of data values in CMF-B contain only the number of bytes necessary to provide sufficient bits to represent the measured (i.e. instance) values. Also, since CMF-B must by definition be validated, CMF-B provides for an optional default value capability which allows the DTD to indicate the most common value for any required field. For additional space savings, any such default value is not placed into the instance document as the default value is known and is inserted by the parser at receipt if not present.

Leveraging further upon the default value capability, CMF includes an optional generic path (i.e. medium) definition feature within a DTD. Utilization of defined path attributes in combination with identification of the transmission path (i.e. medium) of an instance document permits additional path-specific bandwidth savings. Path-specific default values are declared in the DTD using the value most often used on the particular path. Additionally, data can also be excluded from a specific path.

Given the dependence of the binary representation upon a DTD (note: schemas are not currently supported), CMF provides within its definition an optional mechanism for inclusion into an instance document of a hierarchical (major and minor) version number for both the parser API definition and the DTD which were used for the encode operation. Corresponding version number inclusion within the parser software and DTD permits version checking and subsequent notification of potential mismatches to the application.

6 Summary of Position

As can be seen, CMF maximizes the use of the flexible and extensible nature of XML, while including numerous approaches to minimize binary document size. Additionally, the CMF structure incorporates a number of features desirable for a consistent and capable vocabulary-independent language such as data typing, vocabulary-definable ranges, vocabulary-definable units, component version identification, etc. Finally, CMF retains considerable interoperability with the XML standard which enables significant leverage of the wide range of XML tools and existing vocabulary implementations.

In summary, it is believed that CMF or a variation of CMF can be utilized, in fact would be ideal, as the basis of a binary XML standard. L-3/IS and Det. 2 645th MATS therefore request that the Program Committee accept this position paper and register at least one and preferably two individuals for attendance at the W3C Workshop on Binary Interchange of XML Information Item Sets.