Elementary Truths about XML

Toward a Set of Building Blocks about XML

In ancient Greece a mathematician named Euclid wrote a book now famously known as *Euclid's Elements*. It summarized all that was known about geometry. It is a fascinating book. It starts with extremely simple concepts and from them it derives an incredibly rich body of knowledge.

The attempt here is to modestly proceed in a similar fashion: identify a set of extremely simple concepts (building blocks) about XML from which increasingly rich concepts may be rigorously derived.

Background

Inside computers there are no characters, strings, Booleans, integers, or URLs. There's only sequences of zeros and ones called bits. An octet consists of 8 bits. (Commonly people refer to this as a byte and that is the terminology used here). So, inside computers are sequences of bytes.

Software applications may be written to read the bytes inside a computer.

Here is an example of a byte:

```
00110001
```

Different software applications may interpret that byte in different ways. For example, an application may interpret it as:

- corresponding to an integer in base two. In base 10 it represents the integer 49.
- corresponding to a character. In the ASCII character encoding scheme it represents the character 1.

There are various character encoding schemes such as ASCII and UTF-8. Some character encoding schemes require more than one byte to encode a character.

When a "text editor" reads a sequence of bytes it always interprets them as characters. When a text editor writes characters it writes them encoded to a character encoding scheme. Some text editors can be configured to a particular character encoding scheme.

A "hex editor" (or *binary file editor* or *byte editor*) is a type of computer program that allows a user to manipulate the fundamental binary (0 / 1, zero / one) data that makes up computer files. [Wikipedia]

Elementary Truths about XML

1. XML is a human-readable language as well as a machine-readable language.

2. XML documents can exist inside and outside of computers. An example of the latter is an XML document written on a piece of paper. This article will focus on XML inside computers.

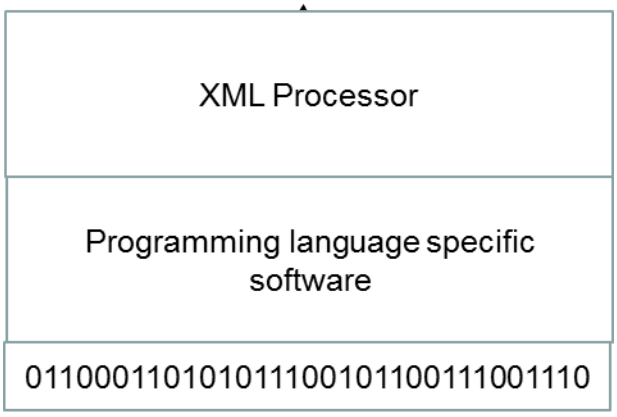
3. An XML document is a sequence of characters. That is, XML is text.

4. As noted above there are no characters in a computer, only bytes. Thus, "An XML document is a sequence of characters" actually means that an XML document is an abstraction of the underlying sequence of bytes.

5. A text editor may be used to create and edit XML documents.

6. An XML processor is software that reads and processes the characters in an XML document. Colloquially, XML processors are known as XML parsers.

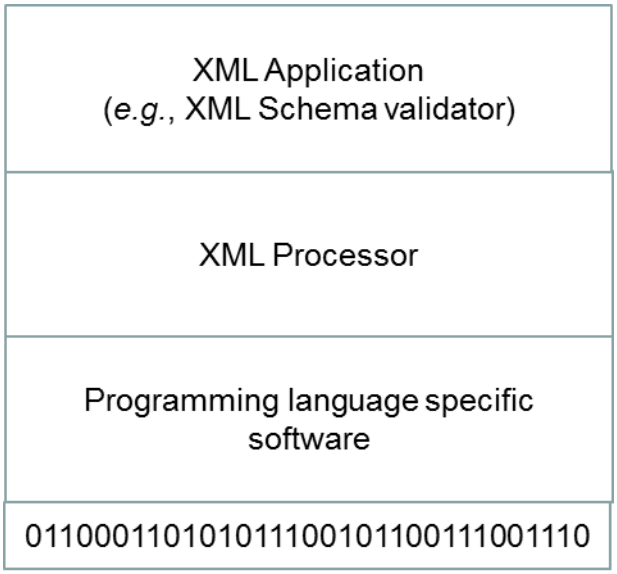
7. There exists software that can read bytes, interpret them as characters, and output a character abstraction. The software that does the bytes-to-character-abstraction is programming language specific. An XML processor uses this programming language specific software to read the sequence of characters. Metaphorically, an XML processor is a layer of software on top of programming language specific software.



8. An XML processor processes the characters in XML documents and makes the results available to XML applications.

9. An XML application is software that processes the output of an XML processor. Metaphorically, an XML application is a layer of software on top of an XML processor.

10. An XML Schema validator is an XML application.



11. XML applications may interpret the characters in XML documents as other than characters.

12. For example, consider the XML Schema that declares an element `A` with a Boolean data type:

```
<element name="A" type="boolean" />
```

Suppose the content of <A> is `1`. The element declaration informs the XML Schema validator and the XML Schema validator interprets the `1` as the Boolean value `true`.

13. Thus, an XML processor interprets the `1` as representing the character `1` whereas an XML Schema validator interprets the same character as representing the Boolean value `true`.

Summary

An XML document is a character abstraction of the sequence of bytes that actually exists inside the computer. Programming language specific software is used to read the sequence of bytes and generate a character abstraction of them (*i.e.*, generate characters). An XML processor reads the characters, processes them, and makes the results available to XML applications. XML applications may interpret the characters as strings, Booleans, integers, or URLs.

Here is the layering and processing:

a. In the computer is a sequence of bytes

b. Programming language specific software interprets the bytes as characters and output a sequence of characters

c. An XML processor reads the sequence of characters, processes them, and outputs the results

d. An XML application reads the XML processor's output and interprets the characters as strings, Booleans, integers, and URLs.

Acknowledgements

Thanks to the following people who contributed to this article:

- David Allen
- Norm Birkett
- Len Bullard
- Chris Byrnes
- David Carlisle
- Roger Costello
- Toby Costidine
- John Cowan
- Jonathan Doughty
- Craig Garrett
- Michael Glavassevich
- Bjoern Hoerhrmann
- Peter Hunsberger
- Michael Kay
- David Lee
- Norman Ma
- Timothy Miller
- Jason Peterson
- Jens Ostergaard Peterson
- Liam Quinn
- Douglas Rand
- Richard Salz

- Rob Simmons
- Andrew Welch