Compacting XML Documents

Miklós Kálmán and Ferenc Havasi

These days it seems that XML documents are becoming ever more important. The number of applications capable of storing things in XML format is growing quite rapidly. The applicability of the XML format spans medical science (human genome mapping), database storage , military use, component modeling. If the growth continues at this rate, XML documents will span every area in computing.

XML documents can be quite large, but many systems can only handle smaller files (e.g: embedded systems). The size factor is also important when an XML document has to be transferred via a network. One solution to overcome this problem is to compress the documents using a general (e.g. zip) or XML compressor (XMill). Unfortunately the compressed size of the files may still be too large.

Compressors are the most effective when they can find the most dependencies in a set of data and can utilize these dependencies to store the data in a smaller form. XML documents may of course contain dependencies which are not discoverable by the above-mentioned compressors. One of these dependencies could be a relationship between two attributes, where it might be possible to calculate one from the other. Our method offers a solution to this problem, employing a special (SRML: Semantic Rule Meta Language) file format for storing the rules. These SRML rules describe how the value of an attribute can be calculated from the values of other attributes. These rules are quite similar to those of the semantic rules of Attribute Grammars, and can be used to compact the XML document by removing computable attributes.

The generation of these SRML files can be done manually (if the relationship between attributes is known) or via machine learning methods. The method examines the relationship between the attributes and looks for patterns in them using specific rules.

We have implemented our algorithm in JAVA in order to make the modules more portable and platform independent. The whole implementation is based on a framework system (every algorithm is considered as a plug-in).

During the testing of the implementation, the input XML files were compacted to 70-80% of their original size, maintaining further compressibility (e.g. the XMill XML compressor could compress the file after first being compacted making it even smaller). The increased compressibility of XML files is the main advantage of our method, apart from gaining a general understanding of the relationships between attributes.

For testing our approach we used XML documents generated from large C++ programs. XML can be considered as a common format for information exchange between software development tools (e.g: XMI in case of UML documents). This trend can be discovered in the field of reverse engineering, where it is important for tools (e.g: source analyzers, visual modelers, metric calculators, program analyzers) to communicate with each other during the analysis of large "legacy" systems. One of these is the Columbus system, which is a widely used tool for the analysis of C++ programs. This system offers the opportunity of storing derived information in an XML format. The output of this system is an XML based file called CPPML, which contains detailed information about the C++ code that was analyzed and aids developers in the reverse engineering process. The size of CPPML documents can be quite large on real systems. This is why applying the technique mentioned in this article is very important, since compacting CPPML documents using this technique, followed by XMill, the compressibility ratio increased by 10% (of the original compressed size). This could make the new method a useful partner in future XML compressors. The method operates using SRML rules. These rules can be generated by hand or by machine learning methods. The effectiveness of an SRML file created via machine learning can attain that of manual SRML generation. It is also possible to combine the two, making the compaction more effective.