University of Waterloo

Faculty of Mathematics

Writing documents in XML

IBM
XML Engineer
Cupertino, USA

prepared by

Jordan Naftolin
98096908
2A Computer Science
August 20, 2000

Table of Contents

Executive Summary

This report will look at the advantages of writing a document in XML. Its purpose is to demonstrate how writing a document in XML can help manage the document and improve the document's usefulness.

Three main advantages of writing a document in XML are discussed. First, it is discovered that writing a document in XML allows the document to be transformed into many other formats, thus facilitating the requirements of different viewers. Second, it is discovered that it is easy and efficient to modify the content of a document written in XML. Third, it is revealed that a document's programmatic manipulation and searching capabilities are improved when the document is written in XML.

It is concluded that documents written in XML have much to offer and it is recommended that companies begin writing their documents in XML in order to experience these benefits.

Introduction

This document was written in eXtensible Markup Language (XML). However, to the reader, this document should not appear any differently from other document formats that they have read. This is because this document has been transformed from its original XML state into a more reader friendly format. You may be wondering why this document was written in XML in the first place, considering the fact that it was transformed into a different format before being presented. This report will answer this question by exploring three advantages of writing a document in XML:

1) The ability to transform the document into multiple formats

2) the ease of changing the content of the document

3) the capability of using computers to manipulate and search through the document

A conclusion will summarize these main advantages and a recommendation concerning this technology will be made to companies as to how to manage its documents.

Analysis

What is XML?

Before delving into the benefits of XML, it is important to understand what XML is and how it can be transformed into other formats. XML is a markup language that is used to store structured information. Therefore, documents that contain some form of logical structure (such as sections, paragraphs, and arguments) make excellent candidates for being written in XML. XML is actually nothing more than a human readable text file that encapsulates its contents in what are known as tags. These tags surround parts of the underlying text and give meaning to what the surrounded text represents. Thus XML documents contain not only the underlying content, but also information about this content. As a result, computer programs can read through XML documents and comprehend what different sections of the document represent. The extra information can be used to transform, modify, and search through the document's content.

Though XML documents can be read and understood by humans, they are primarily used for storing information and are not well suited for presenting it. It is for this purpose that XML documents are often transformed into other more aesthetically pleasing formats. This transformation is done with the use of XSL stylesheets and an XSL processor. Stylesheets are documents which dictate the actions required to transform an XML document from one state to another. Each generated stylesheet specifies a different style for the XML content. For example, one stylesheet may specify that certain text of an XML document be bold and blue, while another stylesheet may specify that text be green and underlined. Yet another stylesheet may omit the particular text completely in the case where it is not needed for the specific style. XSL stylesheets are applied to XML documents using an XSL processor, which is an application that outputs the result of the transformation as a new document.

Generate Multiple Formats

There are many different document formats currently in existence, and each one offers its own benefits and drawbacks. HTML, for example, is an excellent format for presenting content on the Internet. However, there is no guarantee that HTML will look the same on every system. PDF, on the other hand, will always appear the same on every system, but its file size can grow to be quite large making it more troublesome to download. Rather than generating a document in only one particular format, it is clearly advantageous to offer a document in multiple formats thus allowing the viewer to choose the format which is most convenient for him or her.

Documents written in XML have the advantage that they can be transformed into multiple formats quickly and easily. With the use of XSL stylesheets and XML format processors, XML documents can be rendered to many popular formats such as HTML, PDF, Flash, and WAP. In addition to being able to produce documents in multiple formats, XSL stylesheets allow each of these formats to be created in many different styles. Thus it is possible, for example, to create a classic style as well as a modern style of a document in both PDF and HTML formats.

By writing the document's contents in XML and using XSL stylesheets for the presentation, the content is kept completely separate from the presentation. This can be extremely advantageous in managing the document since the writers responsible for the document's content can work independently from the designers responsible for the document's presentation. This separation also permits having one XSL stylesheet defining a particular format and style that can be used for several different XML documents. Thus, if a company changes its letterhead, the change can be made to the single stylesheet, and this stylesheet can then be applied to all of the company's XML documents.

Easily modify document content

One of the common problems with creating a document in multiple formats is that whenever the content of the document needs to be modified, the change needs to be made for every produced format. It therefore becomes necessary to ensure that all copies of a document contain the same content, otherwise the document copies will become out of synch. For example, if a change is made to an HTML copy, but is not made to a PDF copy, then there would be two different versions of the content.

Documents written in XML avoid this problem completely since there is only one source for the content and all desired formats are created from this single source. Thus, when a change is made to the XML source, it is easily replicated to all desired formats by simply regenerating them using the modified XML source. To make things even easier, there are currently applications available, such as Apache's Cocoon project, which dynamically create the desired format and style when a user requests a document via a web server. This means that when a request is made for a document via a Cocoon enabled web server, Cocoon will check to see if the XML content of that document has been modified. If it has, then Cocoon will automatically regenerate the desired format and style using the new content and then deliver it to the user. If the content has not changed since the document was last requested, then Cocoon will simply deliver the transformed document that it had generated the last time the document was requested.

Manipulate and search documents programmatically

It is very convenient to be able to search and modify a document via computer software. However,

searching is often difficult to do with ordinary documents since the software sees only the underlying text of the document and has no way of knowing what this text represents. Thus it would be hard for a program to find, for example, the third main argument in a document since the program would have no idea what text represented the third main argument. Modifying the document is equally as difficult. Consider attempting to insert into a document an additional argument that follows the same text structure as the existing arguments. This would be troublesome for a program, as the text structure of the document would be unknown.

Since documents written in XML give context to the underlying text, advanced searches and modifications become possible. Searches for particular contexts within a document, rather than specific text, can now be easily retrieved. Manipulations to the document are also easier because of the way XML is written and understood. In addition, information about the document can be easily extracted and used by the program. For example, it would be easy to extract from the document such things as table of contents and summaries. It could also be possible to determine how many arguments a document contained, and how many points supported each of these arguments. Comparing specific sections of documents is also made easier, thus enabling, for example, the comparison of two documents' conclusions.

Conclusion

The benefits of writing a document in XML are numerous. Documents written in XML can easily be transformed into many other formats such as HTML or PDF in order to accommodate the preferences of different viewers. As well, each of the generated formats can in turn be created in several styles to allow the same document to be viewed differently depending on who is reading it. Changes to the content of the document need only be made once to the XML source, and yet are easily replicated across all generated styles and formats, thus making it easy to modify the document's content. Writing a document in XML also gives computer programs a way to understand the content of the document, and thus allows applications to programmatically search through and manipulate the content of the document.

Recommendation

It is recommended that companies take advantage of XML technology by writing their client documentation in XML. By doing so, companies will be able to easily generate their client documentation in multiple formats and thus provide clients with a choice of formats to best suit their preferences. Documents written in XML will also be easier for a company to manage since changes will only have to be made once and the document content can be modified independently of its presentation. Though converting existing documentation to XML may require some time and effort, the benefits that come from making this change are clearly worthwhile.