Improving Doxygen:

by Daniel Hilferty

Abstract:
Doxygen is a documentation tool used by many Open Source projects. It is an excellent documentation parser, but it supports limited output formats which are tough to customize. I propose reworking the Doxygen XML output system to create XML that can be more efficiently parsed and easier to configure via XSLT stylesheets.

What:
This project will be a cross-platform implementation of a code comment parser. It will extract inline code comments and create XML output from these comments. An XSLT stylesheet can then be used to configure the output into HTML which can be distributed for use as project documentation. The program will involve two steps: 1) Creating the XML output via Doxygen and 2) Transforming that output into XHTML via Xalan.

Why:
Doxygen currently provides an XML output system, but the resulting XML is inefficient to process. Doxygen developers have shelved the XML output because of the increased processing time. I believe using XML as an intermediate processing language is the optimal implementation for Doxygen; the current XML DTD is just inefficient to process.

I believe that modifying Doxygen to produce only XML will have the following advantages:
1. Simplify Doxygen's code. Separating the parsing tools from output formatting will allow one set of programmers to concentrate on parsing new programming languages, while an entirely different set can create output extensions.
2. Improve code quality. Creating small tools that do one thing well is a core principle to Unix design. This system has worked well for decades and I whole heartedly believe in it.

How:
The first step in this project is to create an XML Schema that can be used efficiently to produce a variety of output formats. It should be tailored to support a large set of multiple XML files. I will need to determine common operations which will be performed on the data and optimize the Schema for these operations.

The second step is to actual modify the internal workings of Doxygen's XML output classes. These classes will be modified to produce the developed Schema.

Finally an XSLT stylesheet will be created to test the efficiency of the new Schema. It will be desirable to keep additional processing time to under 25%. This seems like a fair trade-off for the advantages that XML will bring to the table.
Questions

Does a Schema or DTD already exist that I could use for this purpose?