XSLT/XPath Conformance Procedures and Deliverables

Many people will depend on conformant XSLT processors. Among them are writers of XSLT stylesheets, developers of XSLT processors, test labs, developers of tools with embedded XSLT processors, and developers of tools that generate XSLT stylesheets. OASIS is developing a vendor-independent test suite and the progress can be monitored at the XSLT/XPath committee's website at http://www.oasis-open.org/committees/xslt/.

Those who write XSLT stylesheets are concerned about conformance and the portability of their stylesheets across different XSLT processors. Conformant processors help stylesheet writers test whether their code is portable. A stylesheet writer who knows how to write portable code has more portability in the job market, too. While the tests issued by OASIS as a result of the XSLT Conformance project are not intended for stylesheet writers, they will represent examples of stylesheet code having well-defined behavior. The actual audience for the OASIS tests will be "test labs" that evaluate processors.

A developer of an XSLT processor should have a test lab, so developers are among the target audience. If they aspire to develop a fully-conformant processor, they can use the OASIS tests as a comprehensive battery of XSLT code that will give their processor a good workout. Independent test labs and the press may use the OASIS tests on several processors to compare them.

Developers of tools that embed XSLT processors want to choose a processor that meets their needs, and conformance is likely to be a high priority. It is especially important for a product that wants to claim that the user can supply any arbitrary stylesheet and obtain the correct transformation of their XML. Next-generation browsers may fall into this category.

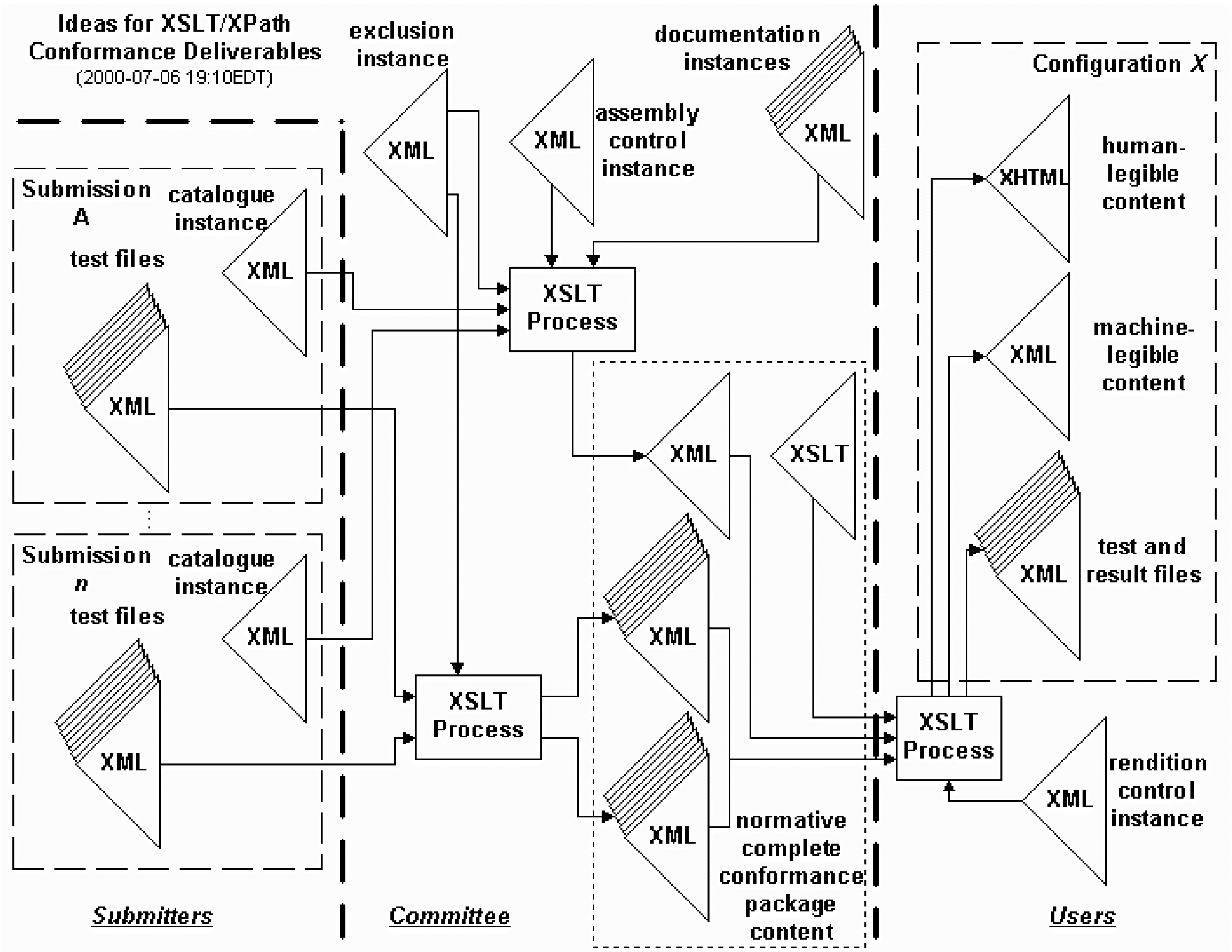
Developers of tools that generate XSLT stylesheets are very concerned about portability of the XSLT code generated by their software. While the OASIS tests do not directly provide a way to scan the generated code, an XSLT processor would perform that function. The tool developer is then left with the dilemma that their evaluation of how well their generated code conforms is limited by the conformance of the processor(s) in which they test it. By applying OASIS tests to those same processors, they can determine whether a processor has weaknesses that limit its ability to serve as a testbed for XSLT code. Also, the overall design of the OASIS test package may provide ideas about a flexible test platform for XML tools.

The XSLT/XPath Conformance Technical Committee (TC) was established by OASIS in April of 2000. The purpose of the committee is to assemble, document, and develop a suite of test files and an associated XSLT/XPath Conformance Test Report. The report will help users measure how well a processor adheres to the W3C XSLT and XPath Recommendations. The deliverables will include the material with which to produce both human- and machine-legible renditions of the report and a package of associated test files.

The committee deliverables include:

- A comprehensive test suite addressing the grammar productions, the prose, and the character set issues described in the XSLT 1.0 and XPath 1.0 recommendations. The test files that are correct XSLT/XPath must be accepted by an XSLT/XPath processor, and the test files that are incorrect XSLT/XPath must be rejected by the processor.
- A human- and machine-readable report (XML) describing all the tests, contributor, Recommendation sections tested and expected results.
- Normative conformance package content. A description of:
    - Alternative behavior as described by the specifications.
    - Ambiguous situations that may affect a test outcome.
    - Error handling issues

The following Diagram illustrates the planned XSLT/XPath Conformance Deliverables.



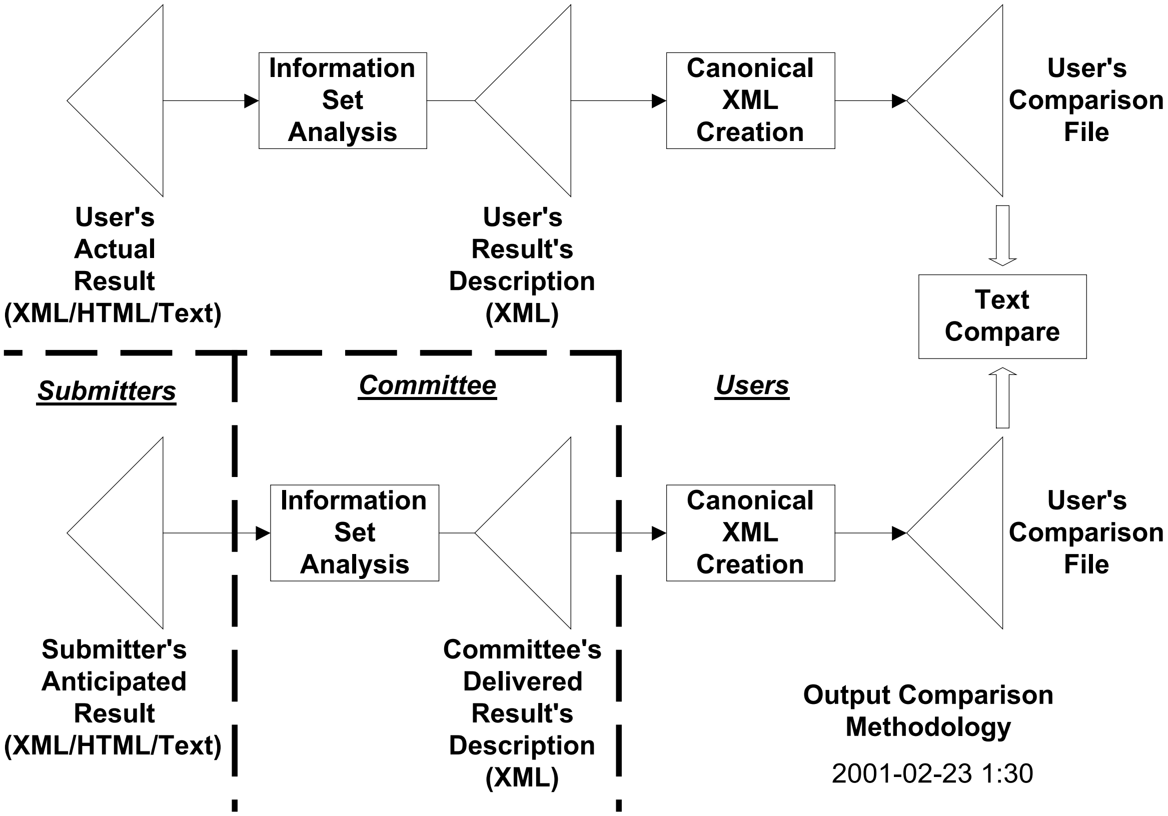
Submitters will provide test files, result files, and a catalog in XML. Lotus Development Corporation, NIST, and Michael Kay have contributed the tests to date. The committee will work with other interested submitters who want to contribute tests to the conformance suite.

The Committee develops an exclusion instance, an assembly control instance, and documentation instances to process using an XSLT Stylesheet into the normative complete conformance package content. An exclusion instance is applied to the catalog instance for any tests the committee deems inappropriate for inclusion in the deliverable. Information in the exclusion instance will dictate how many of each submitter's files actually get used in the final product. The committee will create a number of XML files to document the conformance process and how people work with the files. Each of these individual files comprises the documentation instances. A stylesheet will create the documentation associated with the final package. The assembly control instance points to all of the documentation pieces, the exclusion instance, and the catalogue files feeding the XSLT stylesheet with the location of all the information to be pulled together into the end result. The stylesheet needs to know the list of resources to pull together and gets that from the assembly control instance.

By 'normative complete conformance package content,' the committee is referring to the entire set of test cases, result files, an XML catalogue document, and the XSLT stylesheet containing the locations of all the information pulled together into the end result. The set of test cases provide a complete set of tests for conformance testing processors according to the XSLT and XPath specification and errata for all possible configurations of conformant processors. The result files will be used to compare the processor being tested to the desired results using information set and Canonical syntax concepts and tools. Information in the exclusion instance will dictate how many of each submitters files actually get used in the final product. The stylesheet (or perhaps more than one) will go through all the information configuring the normative complete package from all the raw materials created and collected by the committee.

A rendition control instance is applied by the user of the package to the entire report package containing the entire suite of associated test files, result files, and catalogue in XML. The rendition control instance is a configuration document that qualifies all of the optional behaviors and features of an XSLT processor. The committee has catalogued a set of discretionary items defined by the specification. Processor designers should provide answers as to how the discretionary items are implemented in their processor. Running a projection of the report with the rendition control instance produces an informative (non-normative) rendition of the report tailored for a given XSLT processor. It is this tailored rendition that is then used for processor testing, with all the tests selected for the anticipated behavior as described by the rendition control instance.

In the February meeting, the committee decided on a mechanism to provide expected results for comparison (see the diagram that follows). The anticipated expected results from the submitters will be XML, HTML or text format as will the user's actual results. This result will be converted to an XML document with an indirect representation of the content in InfoSet-style markup. The XSLT/XPath Conformance Committee will supply the user with the anticipated results and the user will apply a committee-supplied Information Set Analysis mechanism to produce the Users Results Description of the actual test results run on a particular processor. The XML results can be canonicalized and the user can perform a byte-wise or text comparison. For example, if the result is an XML document, the user can apply an XSLT stylesheet supplied in the XSLT/XPath Conformance Testing package and use the processor being tested or another processor to produce an XML InfoSet representation of that result. Processing the information set result using Canonical XML with both the committee-supplied expected results and actual results allows for easy comparison.


Information Set Analysis
Canonical XML Creation
User's Comparison File
User's Actual Result (XML/HTML/Text)
User's Result's Description (XML)
Text Compare
Submitters
Committee
Users
Information Set Analysis
Canonical XML Creation
User's Comparison File
Submitter's Anticipated Result (XML/HTML/Text)
Committee's Delivered Result's Description (XML)
Output Comparison Methodology
2001-02-23 1:30

Conformance testing provides a means for determining if an implementation satisfies the requirements and behavior of a specification (i.e., Recommendation). Conformance testing is bound in its scope to the specification it is testing. Specifically, the XSLT/XPath Recommendations specifies the conformance criteria as requirements and behaviors that an implementation or instance must meet. If the Recommendation does not specify it, it cannot be tested for conformance. Within this framework, a user of XSLT/XPath conformance testing could focus on the XSLT/XPath processor or XSLT/XPath instances. That is,

- Confirm a processor returns information to an application about a putative XSLT/XPath instance as defined by the standard;
- Confirm an XSLT/XPath instance produced by an application conforms to the production of the standard.

The user may want to test one processor or many. The report generated by the user using their testing tool will specify which tests the processor failed to produce the desired output when compared to the result files.

The XSLT/XPath Conformance Committee maintains a public information web page at http://www.oasis-open.org/committees/xslt/ and an email archive of committee discussions at http://lists.oasis-open.org/archives/xslt-conformance/. The public information page contains information on the technical committee's policy on conformance issues, the guidelines for discretionary items, public sessions, documents of note for public review, and contributing members. The committee encourages comments and will monitor and respond to the public comment list at xslt-conformance-comment@lists.oasis-open.org. G. Ken Holman, Crane Softwrights Ltd, is chairperson of the committee. Voting members include Crisman Cooley, Overdomain; John Evdemon, XML Solutions; Dr. Kirill Gavrylyuk, Microsoft Corporation; Tony Graham, Sun Microsystems; Lofton Henderson; David Marston, Lotus Development Corporation; Carmelo Montanez, NIST; Lynda Van Vleet, Class Integrated Quality Inc.; and Norm Walsh, Sun Microsystems.