

# *Computer Writing and Research Lab*

White Paper Series: #040512-1

## *XML as a Computational and Rhetorical Technology*

Jason Craft  
jcraft@mail.utexas.edu  
University of Texas at Austin

12 May 2004

*Keywords:* XML, XSLT, XML Schemas, Markup, Rhetoric

*Abstract:* This paper provides a brief description of the Extensible Markup Language (XML), the Extensible Stylesheet Language for Transformations (XSLT), and XML Schemas. It delineates some of the design principles that inform XML as a method of data communication, and suggests some affinities between those design principles and concepts taught in rhetoric and composition classes.

---

### *What is XML?*

The Extensible Markup Language (XML) is a language created to facilitate the organization and sharing of structured data. First presented as a World Wide Web Consortium (W3C) Recommendation in 1998, XML has enjoyed a fairly radical rate of adoption,<sup>1</sup> both within the Web and outside of it:

- The computer I use stores everything from my user preference data, to core application services data, to my music library metadata and usage data in XML.<sup>2</sup>
- Microsoft, an early adopter of XML, incorporated XML as a core technology in its .NET platform; XML is considered lingua franca for Web services.
- The XML syndication services built into most popular weblog applications allow dynamic sharing of log data among bloggers, forming one of the connective matrices that constitute the “blogsphere.”<sup>3</sup>
- As a standard of data exchange, XML is widely used not only by devices (cell phones, notebook computers, Web servers) but by organizations worldwide (libraries,<sup>4</sup> universities,<sup>5</sup> businesses,<sup>6</sup> governments<sup>7</sup>).

Because of the rapid and widespread dispersal of XML in popular computing technologies – a testament to its logic and utility – we

<sup>1</sup> See <<http://xml.coverpages.org/xml.html#applications>> for a sample.

<sup>2</sup> See <<http://developer.apple.com/documentation/Cocoa/Conceptual/PropertyLists/Tasks/XMLPListsTask.html>> for details.

<sup>3</sup> See <<http://www.xml.com/pub/a/2002/12/18/dive-into-xml.html>> for an overview.

<sup>4</sup> Discussion of the role of XML in library science can be found at <<http://sunsite.berkeley.edu/XML4Lib/>>.

<sup>5</sup> <<http://www.softwareag.com/xml/applications/freiburg.htm>> provides an example.

<sup>6</sup> RosettaNet (<<http://www.rosettanet.org/>>) is a non-profit consortium promoting XML standards for business-to-business data exchange.

<sup>7</sup> <<http://www.xml.com/pub/a/2003/09/24/government.html>> provides a good summary of XML adoption in the US Government.

can look forward to its increasing presence in the CWRL. However, I'd like to discuss the Extensible Markup Language not only as a computational tool but as a system with rhetorical implications – as a technology that reflects specific points of view on the effective organization, presentation, and exchange of information.

### *The Extensible Markup Language (XML)*

As a markup language, XML identifies the structure and significance of data through the use of semantic markers (tags). Anyone familiar with HTML coding should find this XML document fairly familiar:

```
<?xml version="1.0" encoding="utf-8"?>
  <class>
    <semester>Fall</semester>
    <year>2003</year>
    <name>Rhetoric and Composition</name>
    <member type="instructor">
      <firstName>Mitchell</firstName>
      <lastName>Jobs</lastName>
    </member>
    <member type="student">
      <firstName>Marianne</firstName>
      <lastName>Patton</lastName>
    </member>
    <member type="student">
      <firstName>Sam</firstName>
      <lastName>Perlmutter</lastName>
    </member>
  </class>
```

This document contains elements (tagged units of data, such as “member” or “firstName” above), attributes (such as “type” above), and character data (“2003”, or “Patton”). As you can see from the first tag in the document (which follows a document declaration, a sign that this document is written in XML), this represents a class, from Fall 2003, which contains three members: an instructor and two students. Notice that this document has containment or parent-child relationships: a class possesses individual members, who have first and last names (as well as types). By wrapping tags within tags, I have signified these relationships in the body of the document.

When a document such as this is parsed or processed, all these components are “read”: the data, and the structural information about that data, are understood and manipulated. This parsing need not be done by a computer; anyone with fairly good literacy in

English should find the content and intent of the above document at least approachable, even if she or he has no experience with markup languages. This reflects one of the design goals of the W3C<sup>8</sup> in their planning of XML: human-readability.

Though XML has some minimal rules for syntax – a well-formed XML document must contain a document declaration, must use consistent text string formatting and capitalization for its tags, and must always close its tags (that is, follow <tag> with </tag>) – there are no universal rules for vocabulary or semiosis. My impromptu document structure for a class followed no template; I know of no standard for structuring class data in XML at the University of Texas at Austin (though there undoubtedly is one somewhere in the institution), and, even if I knew of one, I would be free to ignore it and define XML vocabularies and structures locally. This XML document is as well-formed as the first, and has a functional internal logic of its own:

```
<?xml version="1.0" encoding="utf-8"?>
  <group type="class">
    <metadata>
      <term year="2003">Fall</term>
      <name>Rhetoric and Composition</name>
    </metadata>
    <members>
      <member>
        <firstName>Mitchell</firstName>
        <lastName>Jobs</lastName>
        <type>1</type>
      </member>
      <member>
        <firstName>Marianne</firstName>
        <lastName>Patton</lastName>
        <type>2</type>
      </member>
      <member>
        <firstName>Sam</firstName>
        <lastName>Perlmutter</lastName>
        <type>2</type>
      </member>
    </members>
    <memberTypes>
      <type number="1">instructor</type>
      <type number="2">student</type>
    </memberTypes>
  </group>
```

<sup>8</sup> The full list of design goals for XML are available at <http://www.w3.org/TR/REC-xml#sec-origin-goals>.

XML is a language with a high degree of locality, flexibility, and adaptability: it can communicate information using whatever semantic structure the writer requires. This, undoubtedly, facilitated its rapid rate of adoption. However, it also raises the question: given this flexibility, how can this language facilitate successful data communication among disparate contexts? As readers, we have the cognitive skills required to understand the possible commonalities between the two XML documents above; we can process them and, with analysis, see that they describe the same thing. But how can a computer do this? How can automated cross-communication among local XML “idioms” happen? Data exchange through XML is enabled largely through formalized grammars – represented in XML schemas and namespaces – and XML translation, enabled by the Extensible Stylesheet Language for Transformations (XSLT).

### *Namespaces and Schemas*

XML documents can be parsed according to schemas.<sup>9</sup> A schema is a grammar for XML documents within a particular context; schemas define what a document within a given domain (or, in the terminology of XML, within a given namespace) may contain, and within which possible structures. Schemas do so by defining which tags and attributes can appear in a given XML document, and where.

To use our first example, a schema can decree that a document for a class may or must begin with <class>; that this class tag may or must contain, in order, semester, year, class name and members information; and, that members of a class have a “type” attribute and child elements “firstName” and “lastName.” A schema defines sequences of elements, numbers of elements, optional elements, parent/child relationships, and attributes within a document.

A document signifies the schema (or schemas – a heterogeneous document can invoke more than one) that it follows by declaring its schematic domains, or namespaces. By declaring itself subject to a namespace, an XML document invites the (human or mechanical) parser to refer to the schema for that namespace. Once schema and document are at the ready, the parser can compare the document and schema to verify the document’s conformance to the schema-defined grammar. This process is called validation, and a valid document in XML is specifically defined as a document that conforms to its declared schemas.

All schemas are written according to the XML Schema language defined by the W3C. They are themselves XML documents – they declare and are validated against the schema for XML Schemas provided by the W3C. Any group communicating with XML may

<sup>9</sup> XML Schemas have replaced an older technology, Document Type Definitions (DTDs), as the recommended method for defining valid XML documents. DTDs are not themselves XML documents; otherwise, the definitional functions met by Schemas are likewise met by DTDs.

create schemas and define namespaces for its documents. The rules of communication in XML are therefore locally defined; a school does not have to learn the “right” way to describe a class, but instead decides what mode of description is right for them.

Though schemas allow organizations to easily define and use grammars for XML data, we are left with the question of inter-organizational communication; what if I need to communicate my locally valid class data to a group that uses a schema for classes closer to our second example? The Extensible Stylesheet Language for Transformations addresses this need; in a system of locally-defined rules of communication, XSLT allows documents to travel from domain to domain.

### *Extensible Stylesheet Language for Transformations (XSLT)*

XSLT (which, like Schemas, is a valid subset of XML) shares the term “stylesheet” with the Cascading Style Sheets (CSS) language, and loosely shares some of the principles of reference and hierarchy found in CSS. However, while CSS serves a very specific purpose – the communication of presentational information for an HTML document to a browser – XSLT has a far broader range of applications. In addition, while CSS is a supplemental technology – CSS does not change an HTML document’s structure, and Web standards advocate an HTML document’s independence from its associated CSS – XSLT is, as its name suggests, a transformational technology, used to change the data structure of an XML document into a different structure as needed.

An XSLT processor, given an XML document or documents and an XSLT stylesheet, enacts a transformation and outputs a new document or documents according to the rules of transformation established in the stylesheet.

The proper stylesheet, then, can automate the transformation of our first class example into our second. A group with a locally valid document and a schema for another organization’s data can write their own stylesheet and use it to translate their documents for the new context.

### *The Rhetorical Principles of XML*

Together, these three technologies – XML, XML Schemas, and XSLT – reflect a unified and specific perspective on the organization and presentation of data: they reflect a particular series of rhetorical propositions.

- The global rules for valid communication of information are basic and minimal.
- The majority of rules and definitions for a successful exchange of information are defined within specific contexts.
- In this system of local sovereignty, there are no universally valid communications; instead, information is structured and communicated according to the needs of a very specific context, and is, if necessary, translated from context to context.

XML is a decentralized system, with a minimal set of central rules and a set of tools that facilitate the local definition and translation of structured data. Its widespread adoption suggests that these rules of design were both well-considered and well-executed; it also provides a strong example of the locality of audience and context in systems of communication, and of the benefits of structuring information for a specific context. The success of XML is a testament to the real-world efficacy of a local perspective on structuring and presenting information, and thus reflects key principles we teach in our Rhetoric and Composition classes.

### *Conclusions*

The XML document will not replace the short essay as the preeminent genre for teaching Rhetoric and Composition, nor should it. However, XML provides a useful example for considering rhetorical practices and choices, particularly for those students already familiar with computer science or Web technologies. The terminology of XML can be used to inform the practice of composition; writers “transform” arguments to appeal to particular value systems. “Validity” in this context refers less to the content of an argument and more to its appropriateness in relation to a particular “schema” or audience.

If nothing else, XML shows that the concepts we teach transcend the genres we commonly use to teach them, and that the perspectives we present are applicable – and, in the vernacular of information technology, mission-critical – within real-world contexts.