

# XML and the College Website

## A Practical Look at the Use of XML and XSL

By *Shahab Lashkari, Product Manager and Max Kaufmann, Technical Product Specialist, OmniUpdate*

### What are XML and XSL, and Why Should You Care?

In simple terms, XML is a way to structure data, and XSL is a process used to select, transform, and format data. XML has a non-rigid and non-proprietary document structure, allowing developers to structure data however they see fit. XSL uses a transformation language, XSLT, to transform XML content.

XML is a significant technology in that it provides a universal format for structuring and storing content, allowing that content to be more easily worked with and transformed to fit a variety of outputs, not least of which is a college or university website. XSL is the stylesheet that processes the XML and allows for the selective control of its output as, for example, a website, effectively controlling the website's format and display (look and feel).

The relationship between XML and XSL can be conceived of as being similar to the relationship between HTML and CSS, but only to a point. Modern XSL allows for greater transformational options than even the most modern CSS, giving complete transformation control over the output content, design, format, and even the file type. XSLT is also a rich scripting language that allows a user to filter, sort, and process content—and data of all types—prior to generating and outputting a file. These output files can be web pages or virtually any type of file.

It should be noted that XML files can be transformed from any addressable location on the web. A website might use pertinent content from XML files from within its own domain or from XML files located elsewhere on the web. For example, an XML file containing faculty information could be output at the district level, allowing subsets to be displayed on web pages at each individual college.

XSL is the template that contains the XSLT scripting code and formatting information used to transform XML content. Think of an XSL stylesheet as having a one-to-one correlation between the output format and the stylesheet used to create a design. XSL stylesheets can also reference and inherit design and transformation instructions from multiple XSL stylesheets.

The combination of XML and XSL has many uses for the college website. The remainder of this white paper will discuss two major applications for XML/XSL:

1. **Separation of content from design and format.** XML/XSL is more effective at isolating content from structure than are HTML and CSS alone. CSS remains an important factor in website design strategy, but its power is increased when used in tandem with XML/XSL.

2. **Content repurposing.** XML/XSL allows output from the same initial content to multiple types of industry-standard files for a variety of purposes. This is very important for data manipulation, transformation, and cross-site repurposing. As more institutions utilize XML as the preferred means for structuring data and content, the more efficiently they will be able to share data, transform content, and repurpose it when needed.

## Separating Content from Design with XML/XSL

Inherent in XML/XSL architecture is its unparalleled ability to separate content from design. As previously explained, XML is used to structure and describe the content, while the design is left up to the XSL stylesheet. This separation allows a design team to create a web layout, design, and information architecture to fit today's needs, then later modify all these elements as needed by simply changing the XSL stylesheet or stylesheets (several stylesheets may control the web design of a site: one might control the look and feel of top-level pages, while another controls the look and feel of second-level pages, and so on).

The use of CSS is encouraged in the XML/XSL model. XSL gives greater control than CSS alone, but allows the inclusion of CSS files so that web-oriented output (HTML output) can easily be edited if needed through the use of capable WYSIWYG editors.

The following is a simple example of how XSL can be used to change an entire website's overall design. This process, when coupled with a web content management system (CMS), is best described as Template-Based Publishing.

*Assume there is one XSL stylesheet controlling an entire site design (there may indeed be multiple XSL stylesheets—one for each particular layout used on a site—but the same principle applies). The web CMS processes the XML files (containing the web content) using the instructions in the XSL stylesheet, then generates and publishes all the web pages and associated supporting files.*



To propose a new website design, the web administrator works with designers and the campus community to change the XSL stylesheet. Upon agreement and acceptance, the administrator republishes the site with one click. The entire look and feel of the website is automatically updated while the content remains exactly the same. It's that simple. The use of XML/XSL to control the layout and design of a website ensures site-wide consistency of the approved look and feel. It also affords smooth transitions to new site designs whenever they are needed.



## Content Repurposing Using XML/XSL

Content repurposing is the practice of using XML/XSL to create content components, or “chunks,” that can then be reused, reformatted, and distributed to other parts of your site for different purposes with relative ease.

To understand how this works, take the following example:

*Assume there are several XML documents available as components of a university site's overall web content (this content may be internal to the website, from external sources, or even an aggregate of multiple sources). One of these XML documents consists of contact information for the entire faculty.*

*XSL stylesheets contain the instructions that transform the XML “contact info” content into the desired output—in this case, an HTML file (web page). There's typically a one-to-one relationship between the XSL stylesheet and the output file, so to further illustrate this practical application, let's introduce a second XSL stylesheet. This second stylesheet contains the instructions that describe a completely different output file—a CSV file for use in a spreadsheet.*

*When the XSLT transformation engine (supplied by the web CMS) processes the XML content (faculty contact info) using these two very different stylesheets, the resulting output is an HTML web page nicely formatted with names, phone numbers, and photographs of each staff member*

and a CSV file that opens in a spreadsheet and contains the same basic contact information, but without all the formatting. The XSL stylesheet could also be designed to output PDF files, raw or transformed XML content, plain text, and script (PHP, ASP, .NET, ColdFusion)—virtually any file type imaginable.



Many institutions have a lot of component-based, distributed content that is ripe for repurposing using XML/XSL such as faculty information, course catalogs, calendars, and events. In some cases, the data may even be stored in XML already, but has yet to be transformed into various practical outputs. There's no time like the present to begin identifying applications, databases, and systems that can output content to XML. Small, individual XML initiatives at a staff or departmental level, when combined over time, can lead to enhanced overall capabilities over the long run.

When identifying sources of XML content, it's also important to look beyond the college or university. There is an abundance of XML content available to the public in the form of RSS news feeds, blogs, and a wide variety of external topic-based and niche information that can be used for various purposes, including mapping and researching. All of these resources can be filtered, sorted, combined, and repurposed on any institution's website.

Implementing XML/XSL in an intelligent and appropriate manner is worth the time and investment. It increases efficiency in managing content by allowing internal web content to be shared and integrated across various web applications (as well as sections of a website or across multiple sites), while ensuring content consistency.

## Web Content Management Systems and XML/XSL

The use of XML by web content management systems is not new. The benefits mentioned in this paper have been well understood by eCommerce media publishing and transaction-oriented websites for many years. For this reason, many web CMS solutions that cater to these markets are designed completely around the XML/XSL model. However, these systems often impose the XML/XSL model across an entire website, requiring a major effort to launch the site.

OU Campus™, the leading web CMS for higher education developed by OmniUpdate, has a unique approach to XML in that it allows for the management of multiple content types within a single environment. This includes content stored in structured XML, as well as the thousands of pages of legacy HTML content most colleges and universities continue to maintain on their websites. This approach allows institutions to take advantage of the benefits of XML without having to convert or force migrate any legacy HTML. This is a substantial benefit for institutions that wish to begin incorporating XML/XSL into their existing web architecture, but at a pace that's right for them.

OmniUpdate also sets itself apart from other CMS providers in its use of XSLT 3.0, which is maintained by the W3C and is the most powerful transformation engine ever created. It's become a standard many programmers use today because of its ability to output nearly anything and everything. XSLT 3.0 accepts any code and outputs anything specified by the user, transforming line by line the instructions given in the XSL. OU Campus supplies its users with all the standard XSL templates required for outputting HTML4, HTML5, PDF, and other commonly used web files. But if something else is required, the XSL can simply be written by a developer to perform the required output. Because it's so powerful, the use of XSLT 3.0 eliminates the need for a college or university to use multiple transformation systems, as is necessary with some CMS solutions.

It's also important to note that XSLT 3.0 is defined and recommended by the W3C for use as a templating language. Content management systems that don't use a standard transformation engine such as XSLT 3.0 often force their users to use proprietary tags in HTML templates. While it might at first sound simple to create templates in HTML, users of these systems must learn a unique syntax and tagging vocabulary that is not transferable to other systems. Moreover, users of these systems are completely reliant on the CMS vendor to develop engines for each type of output. As the world evolves from HTML4 to HTML5 to something else, these users must wait for the CMS vendor to catch up. If a user needs a unique output for something they're working on, they are completely at the mercy of the vendor to agree to build a unique output engine. OU Campus customers never have this problem, thanks to XSLT 3.0.

### Additional XML/XSL Resources

1. [W3C.org – XML Tutorial](#)
2. [XSL Transformation on Wikipedia](#)
3. [List of XML and HTML character entity reference](#)

The information contained in this document represents the current view of OmniUpdate, Inc. on the issues discussed as of the date of publication. This white paper is for informational purposes only. OmniUpdate, Inc. makes no warranties, express or implied, in this document. Copyright OmniUpdate, Inc. All rights reserved.