# XML in action: A data driven web site with XML

**Michel Jouvernaux**
Department of Information Technology
Waikato Institute Of Technology
Hamilton, NZ
Michel.Jouvernaux@wintec.ac.nz

XML was carved out of SGML with the goal of being a basic building block of the "Semantic Web" of tomorrow. But before this "brave new Web" can become a reality, there needs to be a radical change in the underlying techniques used in building web sites.
While some debate whether to allow XML in their databases or not, this paper attempts to demonstrate that the future may lie in walking back from the now common N-tier architecture used in building e-commerce site in favour of a simpler 2 tier architecture based on XML and XSL. The basis for this idea is that after all, XML documents are data sources.
The need for interactive, dynamic, searchable e-Commerce web sites is established, and why this is being referred to as 'data-driven'. The hierarchical nature of e-Commerce data is also examined.
An explanation is given of some building techniques showing how these features can be achieved by the XML/XSL combination. Pivotal to this concept is the separation of contents and presentation made possible with XML/XSL and a debate on the innate advantages of a '2-tier' architecture for a web site built with XML/XSL, with emphasis on simplicity, cost, reliability, and the efficiency gains of distributed processing.
An example prototype web site demonstrates these techniques and the technology requirements.

## Keywords

XML, XSL, Semantic Web, e-Commerce, W3C.

## 1. INTRODUCTION

Firms have for a long time used computer systems to attempt gaining a competitive advantage in their respective markets (Gates, 1999. Porter, 1985). Recently, most companies have spent resources creating a presence on the Internet, often as a response to their competitors' moves in cyberspace; very few companies have actually gained a decisive advantage from this; analysis of the "dot com bubble" of a few years ago showed that resources were often squandered on e-Commerce projects that had no real chance of ever showing any benefits.

A basic rule is that a competitive advantage is not gained because a particular system is beautiful, or expensive, or cutting-edge, or built with fashionable technology, but simply because it is more efficient. Efficiency can be measured from the resources necessary to accomplish the task, the effort involved for the result, sometimes popularly expressed as "value for money". This simple tenet lay behind the assertion that open standard technologies, essentially free for all, can be used to create an efficient and effective e-Commerce architecture.

## 2. XML AND THE SEMANTIC WEB

There is no need to introduce XML here. It is sufficient to remind everyone that, like HTML, it is a subset of SGML and the W3C is the body that controls the XML recommendation. Often forgotten is the long term vision of the creators of XML, which has always been the Semantic Web (Berners-Lee). XML is but one of the set of technologies required to achieve this vision. RDF, Ontologies and Agent technologies are equally important.

> *"The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation."*
>
> Tim Berners-Lee, James Hendler, Ora Lassila (2001)

For the Semantic Web to work at all, web sites will have to have their contents based on XML documents. A current trend is to create XML as required from data, or even store XML itself in a database. In many ways this kind of architecture will not de-

liver the Semantic Web of tomorrow. Agent based components of the Semantic Web architecture will not be able to access the database-stored data and so will get an incomplete picture of what a page is about. Furthermore, the complex technology of a backend database and the processing required to transform raw data into "browser friendly" documents have a profound impact on reliability and performance.

The Semantic Web may require a web site's contents to be created entirely in XML and stored as such on a Web server (where any Intelligent Agent will be able to access it). A presentation layer can be added using either server processing technology (SUN-Apache Xalan, Apache Cocoon) or even better processed on the client itself using XSLT. Separation of contents and presentation is a holy grail of Web Design (Goldfarb & Prescod, 2002), and a necessity for the Semantic Web to operate. The Web Browser of the future will only need to understand XML and XSL, as HTML will be rendered on the fly (Moller, A, & Shwartzbach, M., 2003). A web site operating along those principles will be in step with the evolution of the HTML based Web of today into the XML based Semantic Web of tomorrow.

# 3. A HIERARCHY OF DATA

A lot has been written about the capability or otherwise of XML as a database; certainly, comparing XML to a full fledged RDBMS, the shortcomings become glaringly obvious (Bourret, 2003), even if XML is undoubtedly data storage of some kind. The contention expressed here is that if the "shape" of the data is right, then a RDBMS - and all the associated complexity - is simply not needed. Our goal is a web site where changes in the underlying data - *the contents* - will automatically be reflected in what is *presented* to the viewer. So to create a web site from an XML data source, the "shape" of the data has first to be considered.

Depending on its source and usage, data can be highly relational and normalised, or alternatively of a hierarchical nature. Most e-Commerce retailers use data that can be arranged as a 'tree' structure: e.g. categories, sub-categories and finally detailed product information. XML documents too have a tree structure, so lend themselves well to storing this kind of data. Data analysis will lead to the optimum tag arrangement for our XML database, according to best practice and modelling ideas, (Kennedy, 2003).

Any file transferred to the client browser should be an optimum size (no more than 60k, or a 10 second download) (Stockley, 2004). Data can be arranged in multiple XML documents for the needs of the system, broken according to the product hierarchy. There are no limitations in the practical range of products that can be displayed this way, but admittedly it will complicate document management. This problem is tempered by the observation that most e-Commerce sites have a limited range of products on offer.

This hierarchy is also useful as the navigational path to the product. This approach facilitates presenting other products in the virtual store along the way. The design should be balanced so that the user can get to what he wants quickly enough, without frustrations, but still using the opportunity for marketing. The user will not likely know everything that the site is selling… so "search" in the conventional web sense - on string match - can be detrimental to sales in these situations; leading the user in a straight path to one and only one product is wasting the marketing opportunity.

# 4. THE BATTLE OF THE TIERS

The most common design for e-Commerce websites today is based on N-tier architecture. This consists as a minimum of a client, a web server, some application logic and a database. Web pages are created dynamically from data stored in the database, in response to preset queries or user requests. This architecture has many advantages like easy management of site contents, different levels of security applied to different contents and the possibility of full text search on all the contents (although as we've seen before that may be of doubtful usefulness to the e-Retailer). There are also many inconveniences to the design:

- The necessity of communicating with the database in the first place means an extra sizeable overhead is placed on the response time to an HTTP request
- Maintenance/Synchronisation of the site data with a production database, which often necessitates downtime to be accomplished.

■ Scalability: the requirements for hardware and/or network bandwidth between the web server and the database increase relative to the number of requests to the site. This can be very hard to get right.

■ Reliability is a major consideration, as statistically the more components to a system the greater the risks of failure. If the data source fails, or the application layer fails, the whole system fails.

■ Cost: the extra hardware and software licensing (RDBMS, middleware, etc…) and the extra technical maintenance required can become an obstacle to the small, even medium-sized web entrepreneur.

■ Another potential inconvenience of this approach is the loss of control that occurs when smaller firms have to develop a site based on whatever RDBMS and development technology their access supplier (ISP) supports.

The cost of hardware and the complexity of the 3 tier model are surely not necessary for every web site. As we have seen previously, a set of XML documents can be a database, and thus a data source to a web site; but these documents are merely text files and are easily transferred to a client from any web server. The processing required for the presentation of the XML contents can be done by the client web browser, using a style sheet, and this includes sorting and query operations. The XML/XSL combination allows a data driven web site at a fraction of the cost (no licensing required, ISP needs not be involved), delivering great performance (not need to connect to the DB server every time) and reliability (less components). What we have is not only a simpler 2 tier architecture, but a great example of distributed processing.

# 5. AN XML BASED E-COMMERCE WEB SITE

Our example is based on a real-life company and its e-Commerce requirements. Le Gastronome Ltd specialises in the sale of French delicacy products. They sell several range of products across 3 separate channels: wholesalers (supermarkets), specialty shops (delicatessen) and direct to the public. They want a web site that can help present their products to their buyers, and also accept orders and payments online. Users can log on to the site and will view different products and most importantly different prices depending on their channel status. A prototype XML based site was created; below is a run down of its components. Refer to accompanying site map (Fig. 1.0) for clarity.

**5.1 The server**: run of the mill Apache. Due to the nature of the site, merely transferring text files, any web server software would do, making the site highly portable. PERL was chosen for CGI requirements so as not to compromise this portability.

**5.2 The client browser**: Internet Explorer 6.0, Netscape 7.0 and Mozilla 1.6 all support XML and XSLT in a reasonable sort of way. The site being after all a prototype it was felt good enough that it worked 100% with Internet Explorer version 6.0 (IE 6.0). That browser has 68% of the browser market to date (Onestat, 2004), with all versions of IE accounting for nearly 95% of browser use on the Internet. Note that known compatibility issues between the site and Netscape or Mozilla hinge on the scripting used (JavaScript), not XML and XSLT transformation.

**5.3 User login:** for the necessary security around this process, user information is contained in a secure XML document on the server (in fact outside of CGI-scripts folder and outside the root of the web server, thus inaccessible to clients). The process uses PERL (CGI). The XML document can be efficiently 'slurped' into a PERL hash structure (associative array). On validation of the user credentials, a client-side cookie is written for keeping track of the session. Once validation is done, the frame in the browser is redirected to another XML document specific to that user (*username.xml*). That document is created the first time the user logs in and is afterwards updated with details of the previous orders, giving quick and direct links to those products.

**5.4 The XML database**: Le Gastronome sells several 100 products, but the data that this represents fall neatly within the hierarchy of categories and subcategories mentioned previously. As the range grows, it can even be broken down further (e.g. white wines could be categorised further into sweet and dry whites). No XML files on the site are bigger than 10Kb for efficient download. The file at the top of the hierarchy (MENU.XML Fig. 1) contains a list of the different product categories available on the site, with reference to the actual XML files where category information is stored. A Document Type Definition (DTD) exists for the prod-

uct files, this to ensure that the files are valid, i.e. they conform to the required structure for the system. This is used mostly by the maintenance utility.

**5.5 Presentation layer:** this is accomplished of course with XSL style sheets. In fact there are only 3 XSL sheets for the whole site (Fig. 1.0). The first one renders the categories XML file in a top frame, formatted as a menu bar and also a drop down box from which users can use to navigate to the various product categories. A design goal is that no product listing should be further than 3 clicks of the mouse. The second style sheet is applied to each and every product XML file, since they share the same structure. Any functionality required (e.g. displaying a photograph, sorting, order of columns, etc…) can be added to the XSL document and becomes available to all the product pages; this greatly simplifies the maintenance and update of the presentation logic. The 3rd sheet is used to display the user specific home page upon login. The combined style sheets weigh 8Kb in total, so the total documents for the site loaded at any one time is no more than 19Kb, on top of which some product photos will be loaded as per design. This makes the whole site remarkably fast loading, efficient, and enjoyable to browse even over a slow modem line.

**5.6 Secure contents:** Some of the site contents depend on the channel status of the logged user. The logic in the products style sheet will read the user status from the cookie set at login and render the product information accordingly. Pricing information is available via a CGI script which responds to channel and product id parameters. This was done so that pricing information will never be downloaded as a file to the client browser (a requirement set by the company manager).

**5.7 Maintenance:** The advantage to a company of having a data driven web-site would be naught if there was no capability of updating the data in the XML database. Although one can open any XML document in Notepad, there is a need for more complete tools to assist in that task (Sharpe, 1999). Currently, the data is created and updated in MS Excel, then saved as comma separated files and converted to XML using an ad-hoc C++ conversion program. The site is then updated via FTP. The future plan is to create an online administration utility, possibly using the same PERL technology as the secure components of the site.

# 6. CONCLUSION

It is possible to create a fully functional e-Commerce Web site using nothing else than XML and related technologies. The advantage stems form the open standard nature of XML and XSL, meaning that the licence cost of such a site can be kept to a minimum. This also simplifies the life of the budding e-Commerce entrepreneur by giving near total portability to the web site, freeing him/her from having to consort with the access provider (ISP) and being dictated what database is made available and at what cost.

Additionally, it only requires a 2 tier architecture, and gains another great boost to performance by effectively distributing the processing requirements. This type of architecture, although not suitable for each and every e-Commerce venture, possibly due to security and data structure consideration, does make a viable alternative to proprietary technologies like PHP, ASP, and other N-tier architectures commonly in use today. It is especially suited for the budding e-Commerce company who wants to retain a high level of control and portability on their web contents.

# REFERENCES


Bourret, R., 2003. *XML and Databases* [ONLINE] Available: http://www.rpbourret.com/xml/XMLAndDatabases.htm Accessed 28/01/2004

Moller, A, & Shwartzbach, M., 2003 *The XML Revolution: Technologies for the future Web* [ONLINE] Available: http://www.brics.dk/~amoeller/XML/ Accessed 19/01/2004

Onestat January 2004 *Browser Market Share* [ONLINE] Available: http://www.onestat.com/html/aboutus_pressbox26.html Accessed19/01/2004

Sharpe, B. 1999 *Authoring Tools and the Expanding Radius of Deployment* [ONLINE] Available: http://www.infoloom.com/gcaconfs/WEB/granada99/shab.HTM Accessed: 5/02/2004

Webber D. & Kotok Alan, 2001 *ebXML: The New Global Standard for Doing Business on the Internet* 1st Ed.

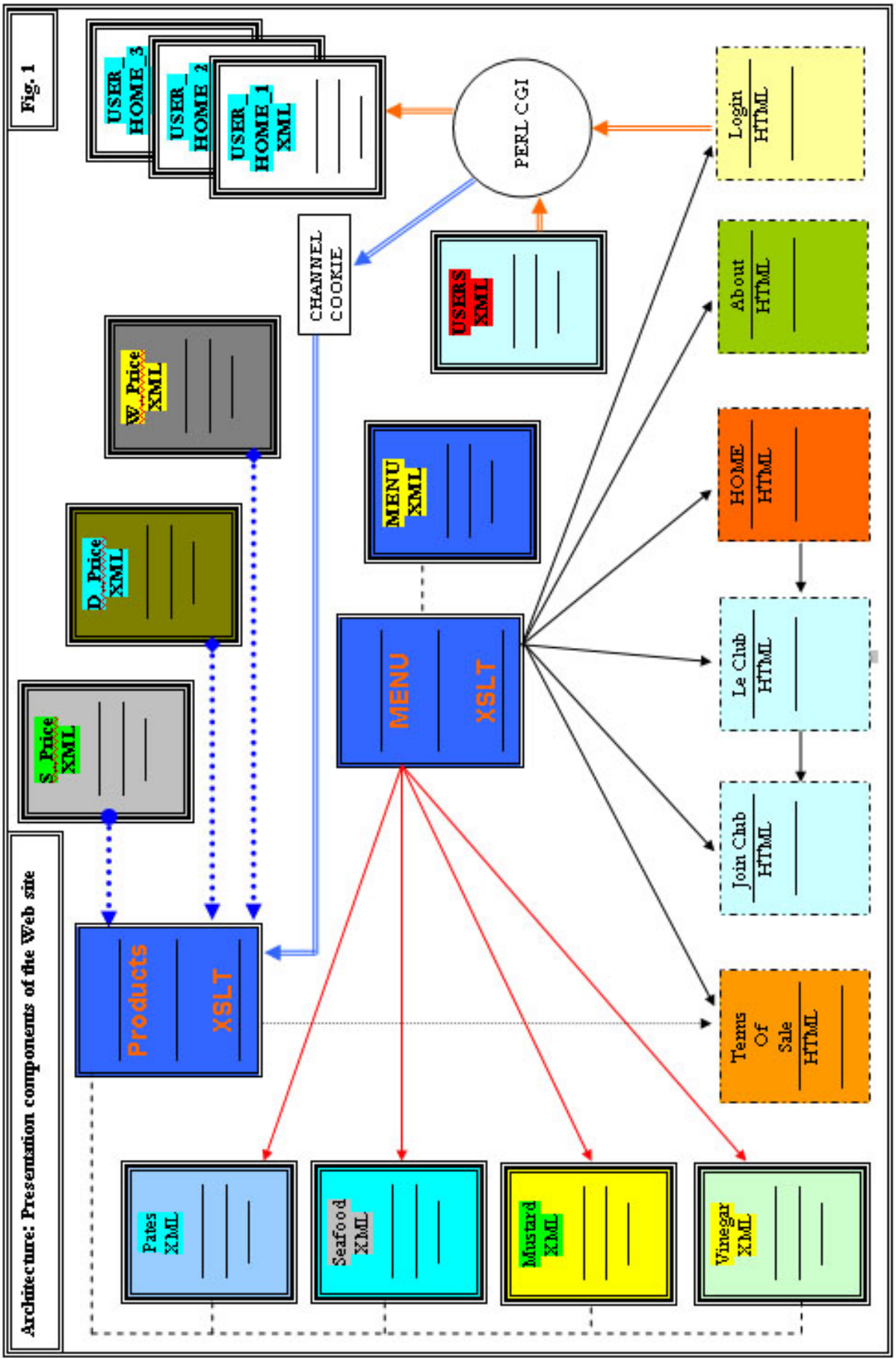


**Fig. 1 Architecture: Presentation Components of the Web Site**

*[Chapter 4: The Promise of XML]* Berkeley, CA: SAMS / New Riders

W3C 2001 *Recommendation: Extensible Style Sheet Language Version 1.0* [ONLINE] Available: http://www.w3c.org/Style/XSL/ Accessed: 01/02/04

Stockley, D. 2004 *Good Web Design Approaches - Web Pages and Websites* [ONLINE] Available: http://derekstockley.com.au/ej1e-good-web-design.html Accessed: 21/02/04

Berners-Lee, T. ,Hendler J. & Lassila O., 2001 *Scientific American: The Semantic Web* [ONLINE] Available: http://www.sciam.com/print_version.cfm?articleID=00048144-10D2-1C70-84A9809EC588EF21 Accessed:12/06/01

Apache Cocoon *Apache Cocoon Project* [ONLINE] Available: http://cocoon.apache.org/ Accessed: 09/02/04

SUN Xalan *Apache Xalan XSLT Compiler* [ONLINE] Available: http://wwws.sun.com/software/xml/developers/xsltc/xsltc_webpack.html Accessed: 09/02/04

Porter, M., 1985. *Competitive Advantage*. New York: The Free press McMillan

Gates, W. with Hemingway, C., 1999. *Business @ the speed of Thought.* New York: Penguin Books

Goldfarb, C & Prescod, P. 2002 *XML Handbook 4th Edition* Upper Saddle River NJ, Prentice Hall PTR

Kennedy, D. 2003*Relaxng with XML data structures In "Proceedings of the 16th annual conference of NACCQ"*