

XML and XSL from servers to cell-phones

a new Internet content model

Ramin Firoozye

Activare Software, San Francisco, USA

ramin@activare.com

<http://www.activare.com>

Ranbir Chawla

EarthLink, Inc., Pasadena, USA

chawla@corp.earthlink.net

<http://www.earthlink.net>

Abstract:

XML and XSL provide a powerful metaphor for separating content from presentation. Content can be generated, assembled, and personalized from a variety of sources and media. Using XSL stylesheets matched to the end-user's environment, the content can be formatted and rendered to match the delivery platform, program, and connection.

Using next-generation technologies, the rendering can also be deferred to the individual devices, freeing servers to concentrate on generating highly targeted and personalized content.

This paper presents an application architecture that can be used to implement a "generate once, display anywhere" scheme for Web-based content delivery.

The underlying technology is in use by EarthLink Networks Inc., the largest independent Internet Service Provider in the U.S. to reach 3+ million users, from desktop browsers, to Internet-enabled cell phones.

Background

Today, the majority of content on the web is coded in HTML (HyperText Markup Language) , a markup language that combines presentation tags with the content (i.e. ****, ****, ...) Mixing the display code with the content makes it difficult to show the material on browsers that do not support the complete HTML standard.

XML allows content to be tagged based on the type of content itself, for example:

```
<USER> <NAME> <FIRST>John</FIRST>  
<LAST>Doe</LAST> </NAME> </USER>
```

The new eXtensible Stylesheet Language Transform (XSLT) specification released in 1999 by W3C provides a language for transforming XML data into HTML (or other XML flavors). By choosing the appropriate XSLT stylesheet to transform the XML, the content can be "rendered" into the appropriate display markup flavor. For example, one stylesheet can translate the content to HDML or WML (flavors of HTML for cell-phones), while another can generate DHTML with animation and links to streaming video.

A web-server can defer the rendering decision until the very last minute, so the same content can be experienced regardless of the mode of browsing. The server chooses the best stylesheet to match a user's immediate needs and renders the content to match it.

Although very flexible, this puts a large processing burden on the server. In this model, ALL web-pages are dynamically processed (either assembled, or rendered, or both). Performance will be key to the user experience. Through advanced XSLT tools, and intelligent caching techniques, the processing time can be reduced to a minimum. Further gains can be derived from browsers that are capable of performing the XSLT transformation themselves.

Web application architectures

Until now, web-based applications either involved sending static HTML files directly to the browser (Figure 1) or HTML code dynamically generated via an application server (Figure 2).

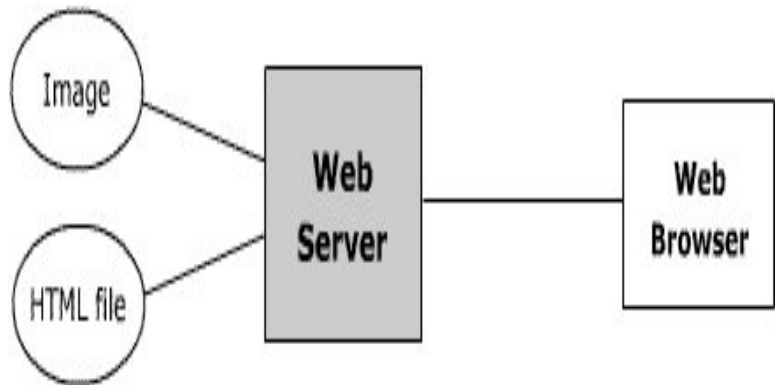


Figure 1

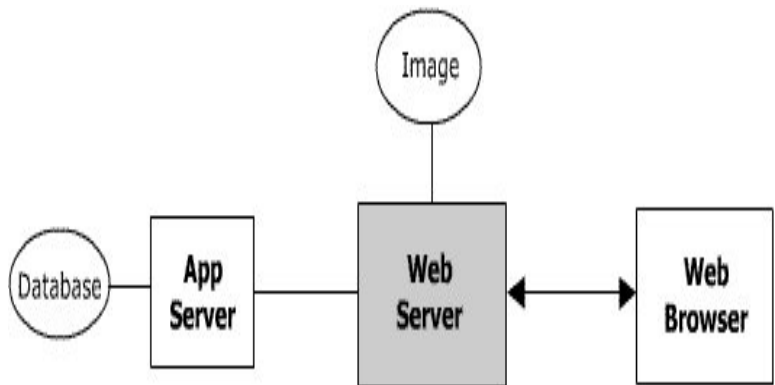


Figure 2

Today, through XML and XSL technologies, the HTML can be generated on-the-fly, with the added benefit that the flavor of display markup can be chosen at runtime (Figure 3).

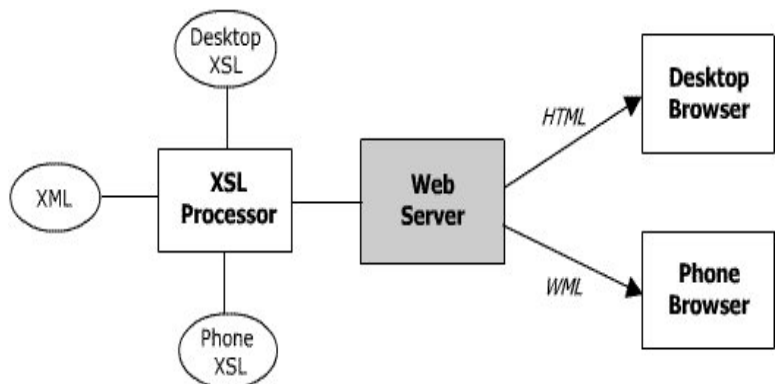


Figure 3

In the case of cell-phones, WML (a markup language for Wireless applications) has to be translated into binary form by intermediate servers and sent via WAP (Wireless Application Protocol) (Figure 4).

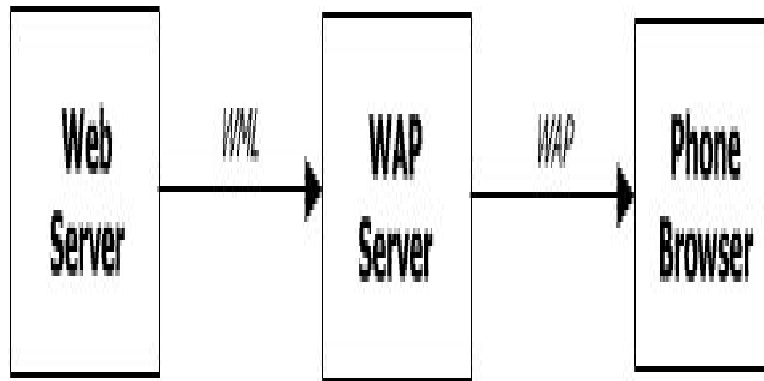


Figure 4

The advent of browsers with built-in XML and XSL processing technologies allows XML content to be directly sent to the browser. The browser can then format the content to best match its own capabilities (Figure 5).

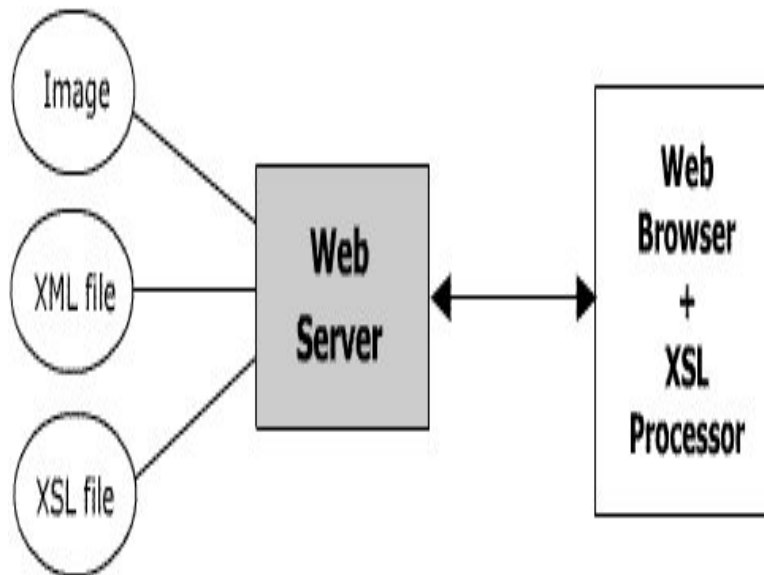


Figure 5

Through intelligent browser-side caching technologies, the XSL stylesheets can be pre-loaded into the browser and used to rapidly process incoming XML into visible form. Today, Microsoft's IE5 and Netscape 6.0 under Windows are desktop browsers with built-in XML/XSL processing technologies (Figure 6).

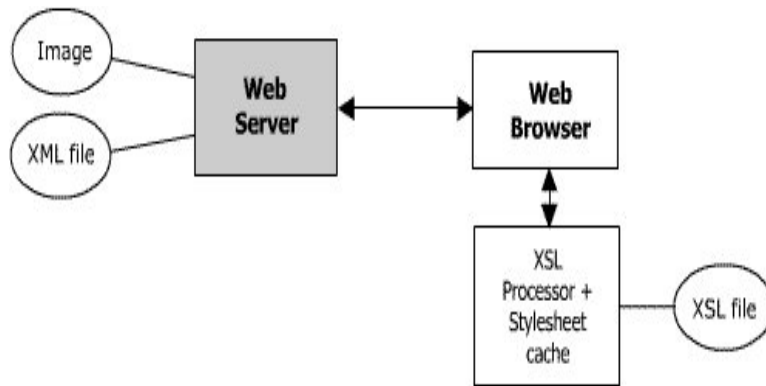


Figure 6

Next generation technologies

The next generation of web-based applications will have to provide support for more than just the desktop browser. To do this, they have to support content generation, rendering, and interactivity. XML and XSLT technologies are ideally suited for this. XML and XSLT applications in each area include:

Content personalization via assembly/generation

An XSLT processing engine with "plug-in extensions" can obtain data from remote sources and assemble the content into a personalized XML content file. The content may include third-party syndicated material (i.e. news, horoscopes, sports, etc.), direct database access, remote services (via ActiveX, RMI, or CORBA), application-generated data, and legacy HTML. The XSLT processor can use the user-preferences to assemble content specifically targeted to a single user. The XSLT stylesheet contains "rules" for obtaining and formatting each information source (Figure 7).

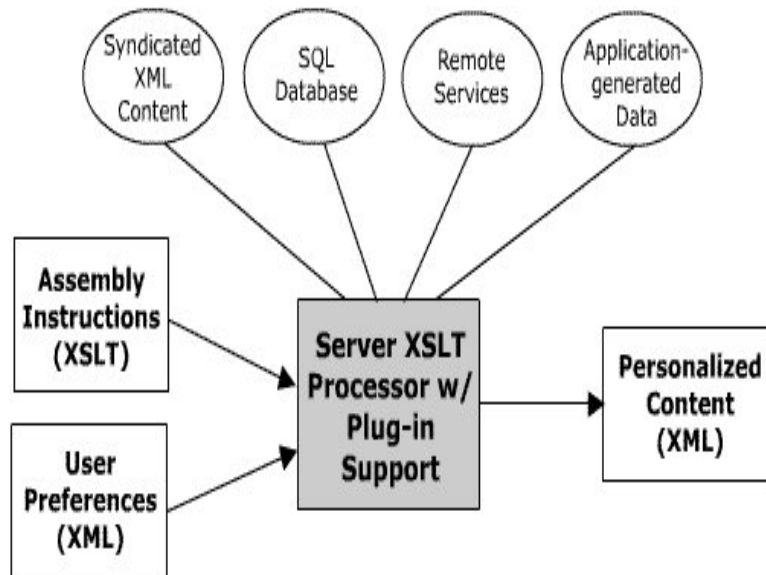


Figure 7

Rendering

The personalized content can be rendered to best match a user's preferences (i.e. themes) as well as browser-type, device-type, and line-speed. A variety of algorithms could be used within a content-matching engine (CME) to best match these input parameters to the optimal stylesheet for a given type of content (Figure 8).

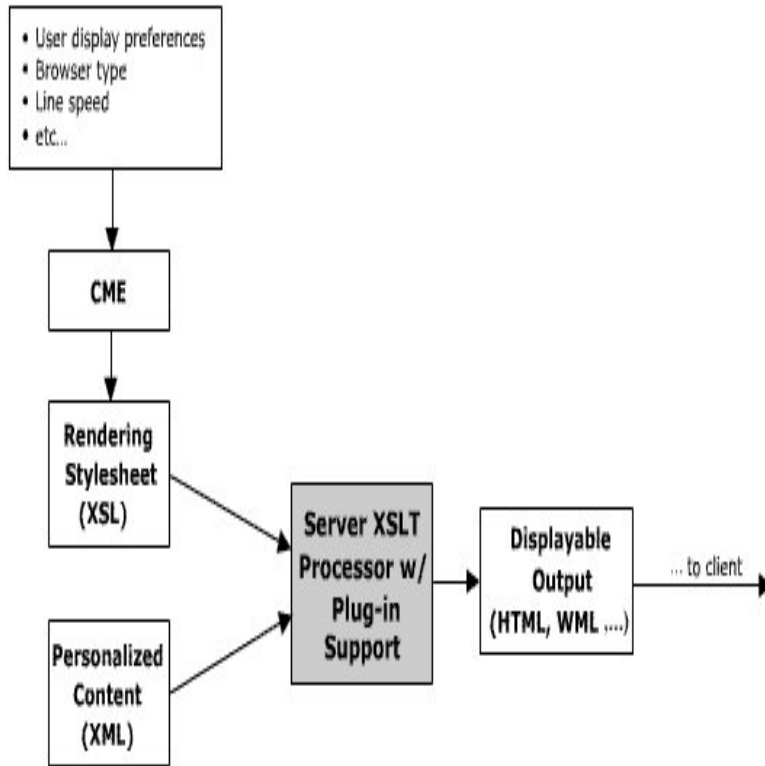


Figure 8

Interactivity

When a user selects a link or fills out a form, the request is transmitted to the server. The server converts the HTTP request into an XML request, processes and generates a response back to the user. Using XML and XSL, user requests can be mapped onto any custom application code (Figure 9).

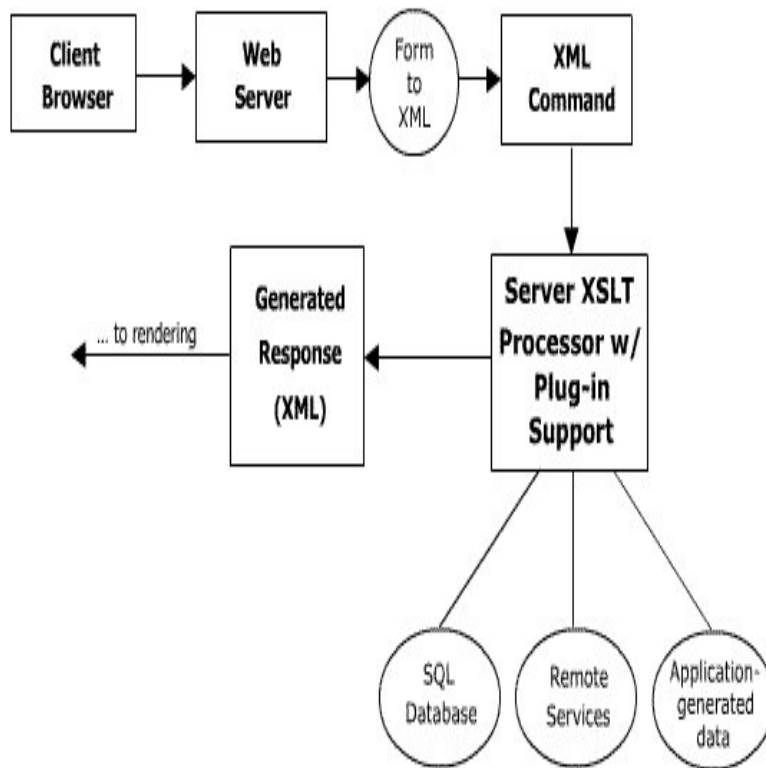


Figure 9

Performance tuning

To be able to maximize performance, some techniques can be employed throughout the application flow:

Caching

Server-side (XML, XSL, legacy HTML) - Content that has already been processed can be cached and reused until the source changes. This can be used to avoid re-assembly and/or re-rendering.

Client-side (XSL, HTML/WML) - Request for content that has not changed on the server may be serviced through the browser-cache. Programmable caches allow relatively static files (such as stylesheets and images) to be stored in the client.

Binary compilation

Stylesheets: An XSLT compiler can translate XSL source into compressed binary form, therefore avoiding a time-consuming reparsing in subsequent times.

Content: An XML compiler can translate the XML source into compressed binary data leading to faster load times and smaller file sizes.

Advantages: Smaller file-size, faster processing due to removal of parsing stage, smaller XSLT engine size since parser will no longer be necessary.

Optimization

Distributed networking: content sources can be distributed across multiple machines. XSLT aggregation and rendering may be performed on separate servers.

XSLT analysis: XSL source may be analyzed for optimal instruction processing and XPath optimization.

Multithreading: A multithreaded XSLT engine can process multiple requests simultaneously.

Case study: EarthLink Networks

Company background

EarthLink is the largest independent Internet Service Provider in the United States second only to America Online in total customers. In 1996 EarthLink developed the first user-personalizable start page, PSP 1.0, for use by its access customers. Since that time the product has gone through five iterations leading up to the state-of-the-art portal it has become today. The members-only version of this product is on-track to generate US\$50 Million Dollars in revenue for the year 2000, with less than US\$1.5 Million in capital investment in hardware and a development team of 5 Java/C++ engineers and 10 XSL/Markup Engineers.

Development goals

The current version of the EarthLink portal was built with the following goals in mind:

1. **Open Access:** It must support all forms of content access; PC browsers, Internet Appliances, PDAs, and wireless phones, with minimal changes required to support new devices, and no changes to the content harvesting process with content partners.
2. **Flexible:** Allow for elegant and simple user-customization of content and interface and "consistency of identity" across multiple devices.
3. **Profitable:** It must scale across multiple versions of the portal, customized for affinity marketing partners and key OEM relationships (EarthLink currently supports over 95 separate versions of this portal).
4. **Scalability:** It must scale to allow for open access to the portal for key strategic partners and everyday Internet users, with a target traffic level of over 15 Million requests per day, using the current hardware and network infrastructure: a four times increase in capacity over the previous system.
5. **Maintainable:** It must scale internally across the development organization, and require no new personnel to support these additional devices.

How it was done?

In order to support these conflicting goals, EarthLink worked in conjunction with Activare to develop a pure XML/XSL solution to this problem.

1. **Open Access:** Content is assembled from third-party sources, using XML, XSL, and XSL plug-ins, then automatically rendered in real-time using XSL stylesheets for any number of devices.
2. **Flexible:** User-preferences are stored directly in XML files and drive the assembly and rendering process. New types of content and services can be added quickly, unlike the widely used application-server/database model.
3. **Profitable:** The portal can be cloned very quickly with a new user-interface, look-and-feel, and branding with simple changes to core XSL stylesheets. XML/XSL architecture allows integration of the portal with third-party advertising, E-commerce, and co-branding opportunities.
4. **Scalability:** Through the use of compiled binary representations of all XML and XSL objects within the portal engine EarthLink was able to use its existing hardware and disk infrastructure while allowing a four-times increase in user-base and traffic.

5. **Maintainable:** Existing HTML markup personnel were retrained in XML and XSL to maintain the system, eliminating the need for scarce C++ and Java programming talent.

The solution involves a pure C++-based XSLT processing engine and XML/XSL compiler from Activare, with support for C++ and Java plugins. For optimal performance, the display rendering system for the EarthLink portal was written in C++. For maximum flexibility and time-to-market, the core personalization system was written in Java using a JNI version of the Activare XSLT system.

EarthLink has developed custom versions of the portal for Apple Computer, Sprint and Sprint PCS, Palm, USAA, and Sony, all of which are accessible from PC browsers, Sprint PCS hand-held phones, and Palm devices.

Conclusion

XML and XSL are highly flexible technologies for use in development of next-generation web-based applications. XSL is an ideal solution for deployment on both servers and clients, allowing the existing infrastructure to handle the demands of future content-distribution systems. Highly customizable content, delivered to any device, any place, is finally within reach.

Authors

Ramin Firoozye

CEO
Activare Software
Postal Address:
601 Minnesota St.
Suite 218
94107 San Francisco
CA
USA
Telephone: 415-826-3113
E-mail: ramin@activare.com
Web: www.activare.com

Ramin Firoozye — Ramin Firoozye has worked in high-technology in Silicon Valley since 1983, in various technical and management positions. Prior to Activare Software, he was involved in founding several San Francisco Bay Area software startups. He has also worked as technical consultant to numerous companies, including Apple, AutoDesk, EarthLink Networks, IBM, Oracle, and Sony.

In 1999 Activare developed a high-performance XSL processing engine for high-volume applications. In year 2000, Activare is releasing products to help move XML and XSL into the embedded systems market. Mr. Firoozye's last presentation was a presentation at XTech '99 in San Jose on DBML, an XML grammar for defining database schema and content, and DataXML, a Java-based tool for automatically converting DBMS content and schema to/from XML.

Ranbir Chawla

Dir. Web Development
EarthLink, Inc.
Postal Address:
3100 New York Dr.
91107 Pasadena
CA
USA
Telephone: +01 626.296.3027
Fax: +01 626.296.5624
E-mail: chawla@corp.earthlink.net
Web: www.earthlink.net

Ranbir Chawla — As Director of Web Apps Development Ranbir Chawla has overseen the design and development of the current and past architecture for our personalized Web engine, as well as our core usage tracking system, for the last three years, since version 1.0. This architecture has allowed EarthLink to scale the current portal system by a factor of three times without increasing the capital expenditure for hardware or software, helping to further drive up margins on incremental revenue. The system also allows for complete partner cobranding and customization, which further drives member acquisition. His team has also driven the PSP systems integration process with IT and MIS, allowing the personalized Web engine to tie into the Radius authentication systems and our CIS for seamless login and provisioning of all of our personalized content.

Ranbir began his career at EarthLink in November of 1996 as a manager of Web hosting products, with the goal of building a state-of-the-art Web hosting service. In January of 1997, he transferred to the fledgling Member Services division and took on of the construction of EarthLink's Personal Start Page. The current extends the core architecture with the use of XML and XSL style sheets, and a complete rewrite in C++ of the core display code. The goal here is a much more flexible system that can scale by a factor of 2 without additional hardware, and a state-of-the-art user interface with DHTML, etc.

In mid 1998, Ranbir created a Data Services group within Web Development to create a more scalable and accurate tracking system, complete with a real data warehouse of user activity/traffic patterns, user retention relationships, and partner tracking information. The last phase of this system was implemented December 1999, completing the project on time and in budget.

Before joining EarthLink, Ranbir was the founder of a successful consulting firm in Denver, Colorado, specializing in Internet E-Commerce, large-scale systems integration projects, and network technology applications. Prior to that he was a securities trader with a variety of top tier firms, specializing in NASDAQ securities and Derivatives.