A NATIVE XML FRAMEWORK FOR WEB APPLICATIONS

O. Burlaca
Institute of Mathematics and Computer Science, ASM

INTRODUCTION

XML now is more than just the best way to exchange information between systems. XML is infiltrating into every aspect of software development. But the tools behind these processes are traditional applications with a relational database at its roots. With the advent of native XML databases, the number of software «layers» for processing the information is decreasing. The problem is that all server-side scripting languages (Perl, ASP, JSP, PHP...) view XML as something of an afterthought. With the XQuery W3C Recommendation, a scripting language that natively supports XML is becoming a standard. From the other side, the XForms W3C Recommendation allows us to create a native XML communication channel with the user. Theoretically, it means that we are in a position of creating web applications that are built exclusively on XML technologies. In this article we try to practically prove this affirmation and outline future work in order to bring this approach of developing web applications to the enterprise level.

1. XFORMS/REST/XQUERY

In this section we describe the advantages of XQuery and how XForms can be seamlessly connected to the server using XQuery via REST interface. Real world scenarios demonstrate that developers prefer REST over HTTP: *"Amazon has both SOAP and REST interfaces to their web services, and 85% of their usage is of the REST interface." (Tim O'Reilly)*

1.1 XQuery

In February 2007, the XQuery specification became a formal W3C Recommendation, after nearly six years of development: the official press release (http://www.w3.org/2007/01/qt-pressrelease) is entitled "W3C XQuery 1.0 and XSLT 2.0 Become Standards: Tools to Query, Transform, and Access XML and Relational Data".

Where XML is the powerful model for content from meta-data to books, XQuery is the application language that unlocks the potential of this content to build content applications to deliver content to multiple formats. This lets you build full applications on one platform.

XQuery advantages:

- A standard, high-level, powerful programming language. Standard means risk reduction and freedom of choice. High-level and powerful means greater development and maintenance productivity;
- XQuery works with XML as its default native data manipulation type. Provides a seamless XML intake, manipulation and storage mechanism that dramatically reduces complexity of the middle tier. Now you're working with XML end-to-end. More precisely, it means elimination of tiers;
- Elimination of tiers: Programming Languages(PL) & XML, XML & Relational, PL & Relational. PL are object-oriented, XML is hierarchical, and relational databases are tabular. The mapping between these three different data models generates a lot of zero-value-added work in developing an application. When you're XML top-to-bottom, that work is all gone.

Eliminating tiers and building web applications on one platform is a Huge saving. Figure 1 illustrates how most of XML web applications are built today. Figure 2 removes unnecessary application logic. To see how XQuery can replace server-side scripting languages (PHP, ASP.NET, JSP) we recommend you to read «XQuery, the Server Language» [2].

In an XQuery Industry Study [3] performed back in 2005 when XQuery wasn't a standard yet, we see that the Government sector has a 8% share after IT Consulting, Software Development and Academic sectors, Figure 3. It means that in big, scattered software systems where vendor lock-in can't be accepted, the open standards are the key. In our opinion, the fact that XQuery became a standard in 2007 will significantly increase the adoption of XQuery in sectors shaped by communication and collaboration (Government, Financial Services, Telecoms).

1.2 XForms

We think that with the advent of native XML databases and the development of REST interfaces to the repository, XForms will get into the industrial

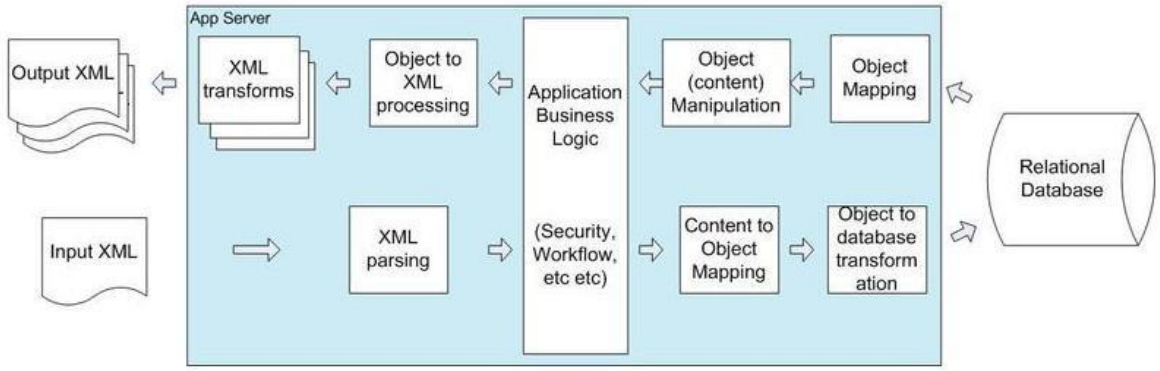

Figure 1. Traditional meta-data system [1].

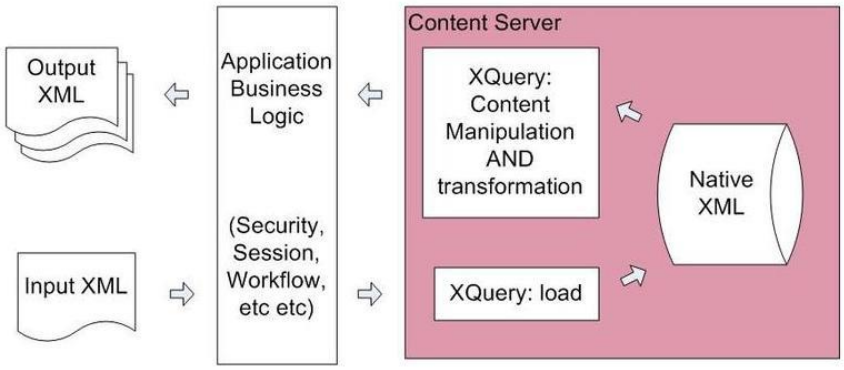

Figure 2. XML Content Server Solution [1].

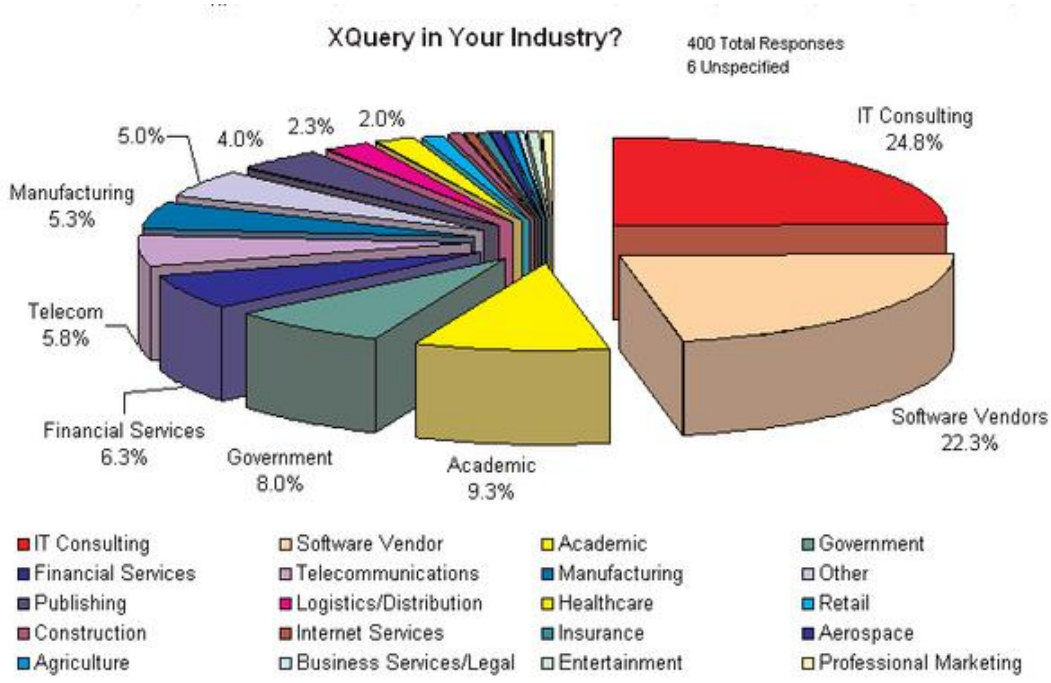

Figure 3. XQuery usage by industry [3].

mainstream. With XForms and eXist [4], you don't need a middle tier for a wide category of applications: XForms can directly talk to the persistence layer, with xforms:submission and REST, while providing all the CRUD (Create Read Update Delete) and search operations (through XQuery). This topic is addressed by Erik Bruchez[5], the creator of Orbeon Forms [6]. The fact that the popularity of XForms is rising is a special «XForms evening» at the XML2007 [7].

It is believed that the combination of XForms/REST/XQuery (XRX) to be a magical fit

[8]. It allows to quickly build web applications with XML data without ever having to use JavaScript or shred inserts into multiple tables of an RDBMS.

XRX approach changes the way web applications are built, and it requires time to become a trustworthy paradigm:

Andrzej Jan Taramina (Chaeron Corporation, CEO and Chief Architect): *But I don't believe XRX has "become" that stack yet. Yes...it's there today, and usable, but not many folks know about XRX, nor it's power, flexibility, agility and productivity. In that sense, I think XRX has the potential to become that stack, if we can get the word out and show some mission critical, complex apps (like ours) in the wild.*

For a real story about the use of the XRX approach you may want to read [9], it describes how using three technologies (XForms, REST and XQuery) you can dramatically transform your organizations development methodology. You can move true model-driven architecture (MDA) that reduces the temptation to duplicate code.

Web browsers can support XForms in two ways:

a) Natively: IE has a formsPlayer plugin, FireFox has the Mozilla XForms extension

b) JavaScript/Ajax/HTML approach: Chiba, Orbeon Forms, FormFaces.

The simplest way is to use FormFaces [10], because it's s a pure JavaScript solution that utilizes AJAX techniques. You only need to add a single line of code to your HTML with XForms: <script type="text/javascript" src="formfaces.js"></script>

and XForms are translated to JavaScript&HTML at runtime on the clientside.

2. A NATIVE XML FRAMEWORK

In previous section we've tried to show the importance of these emerging technologies (flexibility, standardization, reduced costs, etc). This section has a practical aspect. We created a proof of concept application, trying to encompass the smallest set of open source products for creating a native XML framework to build web applications. Demonstrating the feasibility of our approach, we outline what have to be done to apply this approach at an enterprise level. For our small application we used the eXist [4] database. Entire web applications can be written in XQuery, using XSLT, XHTML, CSS and maybe Javascript (for AJAX functionality). XQuery server pages can be executed from the filesystem or stored in the database. eXist features a webDav interface to the repository. It allowed us to directly manage the repository using a XML editor! It means no layers between the database and the person who needs to directly manage XML documents.

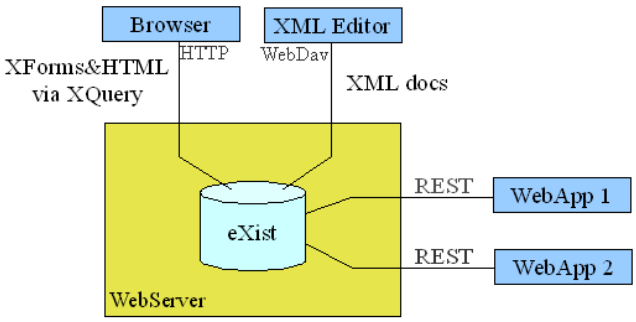

Figure 4. The architecture of a simple XRX application.

The application we created is a simple news reader that stores acquired items from RSS feeds into eXist database. It allows you to edit news items through XForms. The whole application is written in XQuery. Here is the link to the app: http://www.zenorg.md:8085/exist/news/list.xql. The main features of the application:

- XQuery server pages are used like page templates for XML data
- Relevant news items are fetched in XQuery and saved to DB using Xupdate
- user can edit a news item using an XForm that is generated an XQuery.
- The action of the XForm is an XQuery that replaces the edited news item using XUpdate.
- FormFaces [10] is used to enable XForms in browsers.

2.1 The Choice of a Native XML DB

Having proved the viability of the XRX approach and before applying it at an industrial level in enterprise applications, we should find such software pieces of the XRX puzzle that will face enterprise requirements. The fact that eXist integrates with Cocoon is a good thing, allowing developers to create complex web applications entirely based on XML. But eXist[4] is not suited for the enterprise because it misses important things like transactions, recovery, security.

A good alternative to eXist might be Sedna XML DB [11], it features:

- Support for ACID transactions
- Support for fine-grained XML triggers
- Database security (users, roles and privileges)
- SQL connection from XQuery

SQL Connection allows access to relational databases from XQuery using SQL. The resulting

relations are represented on-the-fly as sequences of XML-elements representing rows. These elements have sub-elements corresponding with the columns returned by the SQL query. and thus can be easy processed in XQuery. This feature can be extremely useful in complex scenarios where an application is scattered on multiple databases.

Unfortunately, Sedna doesn't have WebDav and REST interfaces to the repository. Moreover, for XQuery server pages support, it has to be tested how the integration with Apache HTTP server (via Apache module) works. Both Sedna and eXist supports vendor independent XML::DB API and are trying to achieve a 100% validation of the W3C XML Query Test Suite [12]. It would make sense to test/identify other XML DB for production-grade applications that conforms to these requirements.

2.2 XForms Challenge

XForms is not supported natively by IE and FireFox. The dynamics of the development of plugins for both of these browsers is low. Moreover, the plugins do not fully supports the W3C XForms recommendation. For our test application we've used FormFaces[10] that is a pure JavaScript implementation and doesn't need a plugin, but once again: it doesn't implement all XForms specifications and the last version was released back in January 2007.

The most solid XForms processor/product seems to be Orbeon Forms[6]. It is similar to FormFaces but runs on the server: the XForms engine translates your XForms pages to HTML, CSS and JavaScript that your browser understands.

Running XForms on the server has even several benefits over running it in the client:

– Users don't have to upgrade their web browser or install plugins or extensions. You can use and develop with XForms today and target over 90% of the users on the web.

– The data used by your XForms application remains safely on the server. With client-side implementations, on the other hand, you have to be very careful not to send confidential information to the XForms engine, as that information may be seen by the user.

– You can upgrade your XForms implementation at a single location on your server instead of waiting for users to upgrade their browsers when bugs are fixed or new features are implemented.

Orbeon uses eXist internally, it should be checked how Senda can be coupled with Orbeon Forms.

3. CONCLUSIONS

At the XML 2007 conference there were many people that seemed to have independently discovered that if you combine XForms/REST and XQuery you can create a software development environment that circumvents the need for middle tier objects and conversion to and from relational databases.

The XRX leads to development architecture based on international standards that is designed to minimize the probability of vendor-lockin. Portability on both the client and the server. XRX avoids costly shredding (and reconstitution) of complex XML documents into RDBMS tables.

We are in a position to develop complex web applications that stands on two solid pylons: XForms and XQuery. What is needed is a good XForms processor (Orbeon Forms[6]) and a solid native XML database that have support for XQuery server pages and a REST interface (Sedna[11] is the candidate).

References

1. *Matt Turner, «Metadata, XML and XQuery», http://xquery.typepad.com/xquery/2006/06/metadata_xml_an.html*

2. *Kurt Cagle, «XQuery, the Server Language», http://www.xml.com/pub/a/2007/06/01/xquery-the-server-language.html*

3. *DataDirect 2005 XQuery Industry Study, http://www.datadirect.com/web/global/xquerystudy/docs/datadirect_xquery_survey.pdf*

4. *eXist - Open Source Native XML Database, http://exist.sourceforge.net/*

5. *Erik Bruchez, «XForms and the eXist XML database: a perfect couple» http://2007.xmlconference.org/public/schedule/detail/426*

6. *Orbeon Forms, http://www.orbeon.com/*

7. *XForms evening sessions at XML2007 conf. http://2007.xmlconference.org/public/schedule/topic/15*

8. *Introducing the XRX Architecture: XForms/REST/XQuery,http://datadictionary.blogspot.com/2007/12/introducing-xrx-architecture.html*

9. *XRX and MDA,http://datadictionary.blogspot.com/2008/01/xrx-and-mda.html*

10.*FormFaces: an XForms pure JavaScript solution, http://www.formfaces.com/*

11.*Sedna XML Database, http://modis.ispras.ru/sedna/*

12. W3C XML Query Test Suite, http://www.w3.org/XML/Query/test-suite/

Recomandat spre publicare: 15.10.2008.