



# Xpiori and the XML Domain

by Chris Brandin

Release 1.1

Xpiori, LLC  
2864 S. Circle Dr.  
Ste. 1200  
Colorado Springs, CO 80906  
(719) 527-1315  
[www.xpiori.com](http://www.xpiori.com)

© 2007 by Xpiori, LLC. All rights reserved.

**Version 1.1**

**Copyright Xpiori, LLC All Rights Reserved**

**Xpiori technology is protected by the following patents:**

**US Patent #5,742,611 (21 Apr 98)**

**US Patent #5,942,002 (8 Aug 99)**

**US Patent #6,157,617 (5 Dec 00)**

**US Patent #6,167,400 (26 Dec 00)**

**US Patent #6,324,636 (27 Nov 01)**

**US Patent #6,493,813 (10 Dec 02)**

**US Patent #6,792,428 (14 Sept 04)**

**Other U.S. and international patents pending.**

**The information in this white paper has been provided by Xpiori, LLC. To the best knowledge of Xpiori, it contains information concerning the current state of information processing technology. Xpiori, LLC disclaims any and all liabilities for and makes no warranties, expressed or implied, with respect to products described in this paper, including, without limitation, the implied warranties of merchantability and fitness for a particular purpose. No specific reliance should be made on the material provided herein without thorough investigation of the technology and its proposed application to specific circumstances. Product and technology information is subject to change without notice.**

## Introduction

XML is fast becoming the lingua franca of information processing over the Internet. It has taken hold so fast, in fact, that the technology necessary to implement XML based systems is still in its infancy. Basically, there are two value propositions that XML brings to the marketplace: The first – the one everybody knows about – is that it is an excellent information transport standard. This is made possible by the self-describing nature of XML documents. The virtually universal acceptance of XML as an information transport mechanism has made it the “hot” technology of the year. The second, and possibly more interesting, value of XML is its extensibility. This feature of XML has not yet been leveraged into the marketplace, primarily because the technology to make use of XML’s extensibility has heretofore been unavailable. Xpiori is changing all that.

## What to do with all that XML data

Soon, nearly all data being exchanged between computers on the Internet will be expressed in XML. Where will all this data come from, and where will it be stored? How will it be processed? Currently, data typically only exists as XML while it is being transported; storage and processing takes place in other domains. This is unfortunate because XML can express information in a way that is very natural and useful, but at the same time generally un-accommodatable using current data processing technologies. XML allows for the dynamic creation of *new data types*, as well as new data elements. This allows users to create the data they need – when they need it - rather than having to anticipate the type of information required in advance. This capability alone may represent the biggest advancement in information processing to date. The challenge is to find a way to store and process XML data in a way that allows the user to take full advantage of its extensibility. Or, to put it another way – let XML data be stored, managed, and processed in its own domain.

Faced with this new form of information, the system designer basically has three alternatives for managing XML data: use a DBMS, use variations of DOMs (Document Object Model), or use Xpiori’s XML eCommerce Server. Following is a discussion of these alternatives:

## Managing XML Data with a DBMS

For all practical purposes, XML can’t really be managed by a DBMS at all. The information generally has to be converted to a format native to the DBMS first, and then it can be managed. There are two major drawbacks to this approach: First, there is no support for the extensibility feature of XML because DBMS’s require pre-defined data schemas. Second, performance is abysmal (typically tens of transactions per second, or worse).

Another approach is to use an object relational database and store XML documents as BLOB’s (Binary Large Objects). This results in even worse performance because relatively efficient database indexing schemes can no longer be used. Also, this method does not scale well, as the database gets bigger – performance degrades. Although extensibility is supported in the academic sense, performance is so bad that it essentially doesn’t matter.

Simply put, DBMS’s are ill suited to storing and managing XML data – and no “band aids” will render them up to the task. XML is a domain that is very different from the data management paradigms that DBMS’s were designed to address.

## Managing XML with a DOM

DOMs were never intended to store XML documents, so they do so very badly. The proper use for a DOM is to aid navigation through relatively small, temporal, XML documents. DOM's use a huge amount of memory - over ten times the memory required for the original XML. They also have no provisions for indexing – so queries across all but the smallest of DOM's are excruciating slow. The time it takes to service queries is directly proportional to the size of the DOM, so scaling is a problem as well.

There are a couple of products on the market for managing XML data that are essentially indexed DOM's. This method improves query performance, but is still very inefficient because there are now three components to be stored – the XML documents, the DOMs, and the indices.

A significant drawback to the DOM approach is that database management features necessary when data is stored are not part of the DOM paradigm. This results in solutions that lack adequate and robust data management features.

In short, these “pure” XML solutions (even the ones with indices) are slow, feature-poor, inefficient, and not scalable. Just as DBMS's were not designed with extensibility in mind, these solutions use components that were not designed with data storage and management in mind.

## Managing XML with the Xpiori eCommerce Server

Xpiori took a novel approach to solving this problem. Rather than applying existing information management schemes to XML, we re-examined the entire task of storing and managing XML data from the ground up in a cohesive manner. The result is not a RDBMS, nor an Object Database, nor a DOM; rather it is an entirely new method of managing and representing data – “Information Couplets”.

The Xpiori XML Server uses Information Couplets to internally store and manage XML documents. In slightly more space than it takes to just store the XML documents themselves, the functions of DOM's, document storage, and indexing are combined into one mechanism that outperforms competing methods of managing XML in every way. This results in vastly superior performance and a much smaller storage footprint. A method called “Vector Annealing” is employed to service queries. This results in direct query (retrieving a name based on a telephone number, for example) performance that is measured in hundreds-of-thousands of items per second (that's *second*, not minute). This means that the Xpiori XML Server can saturate any web service network mechanism used to connect to it - *for the foreseeable future*. For example, a fast Ethernet connection saturates with data at about one thousand transactions per second.

When XML is stored, there are really two sets of feature criteria that must be met – those imposed by the nature of XML, and those imposed by the *storing* of data. In other words, aside from the full support of XML and its extensibility, there must be traditional DBMS-like features as well. The Xpiori XML server supports transactions, element-level access control, SQL-like queries, aliasing, and many other features not directly related to XML, but necessary for “server” applications.

The Xpiori XML Server is massively scalable. Current versions can support billions of lines of heterogeneous XML documents per instance. The server is completely modular, so virtual databases of any desired size can be built.

## **Conclusion**

The Xpiori XML eCommerce Server removes the remaining barriers to XML-based system deployment by letting XML be its own domain. There is simply no comparison between Xpiori's XML Server performance and any other system we know of – both from a performance and functionality standpoint. The Xpiori XML eCommerce Server makes it possible to fully realize the true potential of XML.

**# # #**