Xpriori Technical Paper

Xpriori Knowledge Building Server Architecture

by Chris Brandin

Release 1.1

Xpriori, LLC
2864 S. Circle Dr.
Ste. 401
Colorado Springs, CO 80906
(719) 425-9840
www.xpriori.com

Version 1.1

Xpriori 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 Xpriori, LLC. To the best knowledge of Xpriori, it contains information concerning the current state of information processing technology. Xpriori, 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

Information is managed - knowledge is built. Information is a vehicle to express knowledge. People develop knowledge through an iterative, layered process where new information is based on conclusions drawn from old information. This poses special challenges for information processing systems. This paper describes an information system architecture that supports a unified framework for building knowledge in a natural, iterative manner.

The Challenges

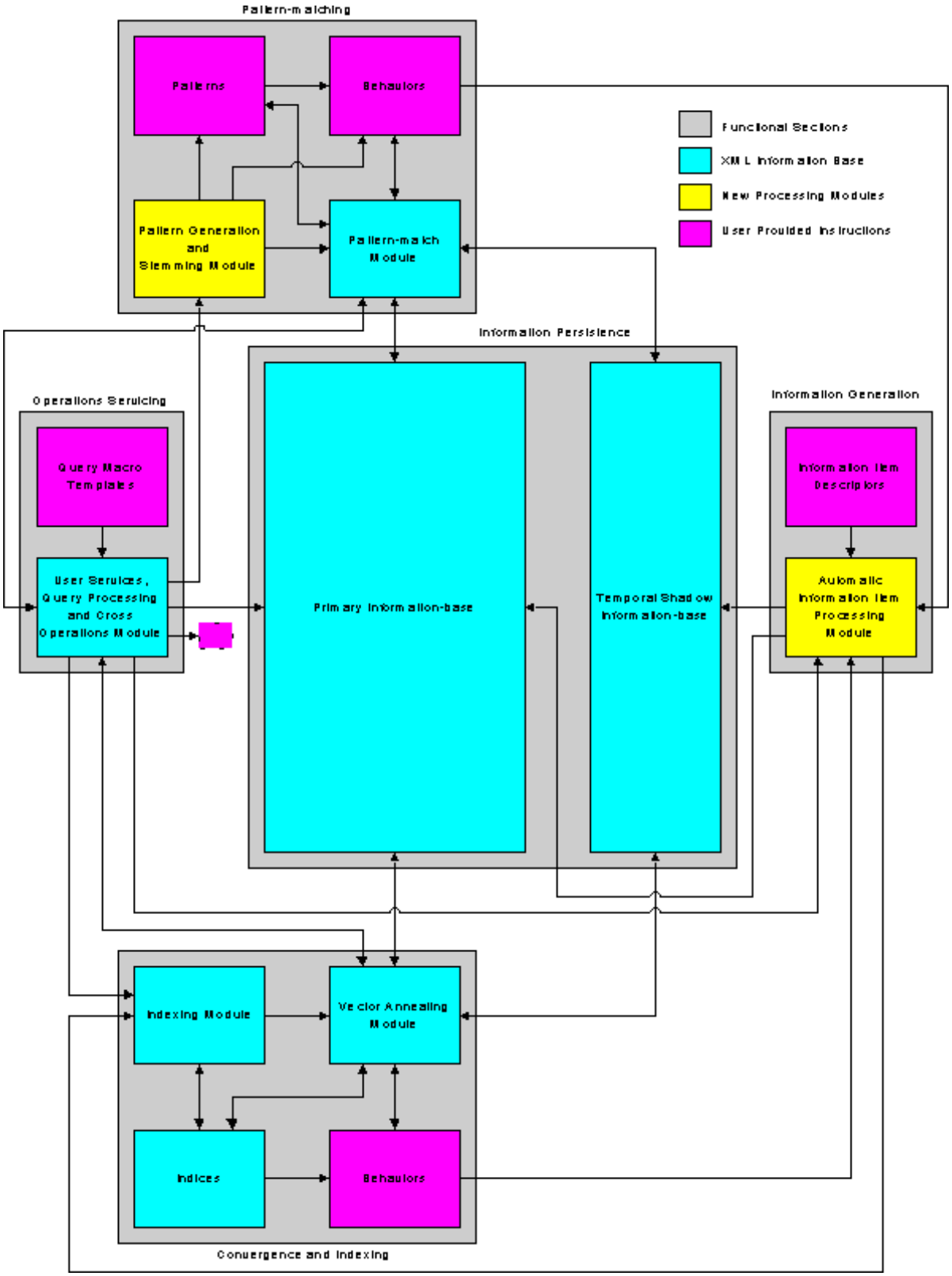
Knowledge building requires new information management and processing capabilities that are not provided by traditional systems. There are three fundamental limitations imposed by conventional data management systems that inhibit knowledge building:

- New data types must be defined for the data management system before they can used. This can be a very time, and resource consuming process. Preferably, data types should be handled in exactly the same dynamic manner as data elements, allowing metadata to be created and destroyed at will with no global allocations.
- Indices must be pre-defined. This can have a devastating effect on knowledge building because it implies that the user must define all query elements in advance, or endure time-consuming database reconfiguration. Preferably, an efficient means to service all queries, whether anticipated or not, should be available that does not require reprocessing database contents or indices.
- Database processes tend to be monolithic; that is, they must complete a process in one pass. Preferably, a means to execute multi-pass operations should be provided that can accommodate layered searches alternating between data-filtering and data-matching functions at the server.
- There is no efficient way to retrieve context based on data. Preferably, a means to efficiently service any type of query should be available, whether retrieving metadata based on data, data based on metadata, or associated information.

Document management systems can achieve a few of the search functions necessary for knowledge building applications, but are entirely lacking information management capabilities. Knowledge building requires that information can be manipulated at an arbitrary granularity and inserted or deleted without document-level re-processing.

Architectural Overview

Following is a diagram of the Xpriori Knowledge Building Server.



At the heart of the Knowledge Building Server is the Xpriori XML Information Server (the components shown in cyan). Other documents are available that characterize all features of the XML Information Server, so discussions will me limited here to those features that make it uniquely suited to knowledge building applications:

- XML is the only widely adopted standard for expressing extensible information. Knowledge building is a highly dynamic and heterogeneous process. The Xpriori XML Information Server fully supports the *extensibility* of XML. Data Elements and metadata (tags) are managed in exactly the same dynamic manner.

- The Xpriori XML Information Server is *information-centric.* Traditional DBMS's offer a *data-centric* view of information; that is, they do not dynamically manage metadata. Document management systems impose a *document-centric* view of information; that is, they deal with XML documents in their totality, making sub-document manipulations extremely inefficient. These traditional models impose serious obstacles to managing heterogeneous information in a dynamic way. The Xpriori XML Information Server treats XML documents as aggregations of information. No restrictions on how much information is returned or modified are imposed. New information can be inserted or deleted at will without any re-processing of documents. Knowledge building requires that information can be retrieved and manipulated with variable granularity and that data and metadata be treated agnostically.

- The Xpriori XML Information Server is completely schema independent. This means that new data types can be created or destroyed at will. Knowledge building requires freedom in creating new data types, both persistently and temporally, without pre-definition.

- The Xpriori XML Information Server effectively always indexes everything. This means that no prior knowledge of subsequent query parameters is necessary. In other words, queries can be ad-hoc. Knowledge building requires that queries can be supported even when they cannot be anticipated at the time information is posted. The Xpriori Server accomplishes this with no re-processing of Information Management System contents.

- The Xpriori XML Information Server locates information by combining pattern-matching and set-convergence operations. This *pattern-centric* method of managing information (as opposed to the conventional structure-centric method) makes it possible to leverage the extensibility features of the server in order to achieve multi-pass query operations.

Expansions to the XML Information Server involve five categories of functionality:

- Pattern-matching. A behavioral-set driven three-dimensional pattern-matching module has been added. This supports fuzzy-matching, user-defined pattern-matching methods, wild cards, and multiple simultaneous pattern-matching operations. Pattern-matching operations are typically preceded by a convergence operation in order to reduce the number of data elements being scanned. Behaviors are used to facilitate pattern-matching and trigger the Information Generation component to automatically post new information elements to the server as a result of pattern-matches.

- Convergence and Indexing. Custom indexing, and the ability to delete temporal index entries have been added. Aliasing has been added to facilitate partially or un-qualified queries. Behaviors have been added to convergence operations so that information

fragments can be added to or deleted from the server as a result of convergence events. Behaviors are also used to support hierarchal proximity detection functions.

- Information Generation. The ability to add and delete information fragments to documents as a result of pattern-matching or convergence operations has been added.

- Operations Servicing. This module services all user interactions. Several capabilities have been added. Queries are typically serviced by a combination of indexing, convergence (if there is a multi-term query), and data pattern-matching (if the query is not fully qualified). Knowledge building applications often involve queries that can return very big result sets. Furthermore, these result sets may serve no purpose except to feed a process that uses the information temporarily in order achieve a subsequent processing step. In order to minimize traffic between the server and a client, the Operations Servicing Module acts as an agent for the client, managing the direct exchange of information between the Convergence/Indexing and the Pattern-Matching modules. The Query Macro Template feature has been added to manage iterative searches, involving multiple passes through these modules.

- Persistence. Provisions for a "shadow" information store have been added. The shadow store typically contains temporal information items that "belong" to corresponding documents in the permanent store. This way, temporal information items only exist for the information building application, and do not appear for other applications. The assumption is that once an information building application has finished its task, the result will be posted to the permanent store.

The Knowledge Building Process

Knowledge building is by nature an incremental process. This is true both from a computational and a human standpoint. In order to illustrate this, a hypothetical (albeit probably nonsensical) example in genetic research will be used:

Assume that gene sequence information has been posted to the XML Information Server consisting of base-pair sequences and additional information such as researcher, species, chromosome, etc. A single pass through the Convergence/Indexing and Pattern-matching operations can be used to create amino acid sequences from base-pairs. The Convergence/Indexing module is used to provide a list of base-pair sequence locations for subsequent pattern-matching operations. The Pattern-matching module is used to actually derive the amino acid sequences from the base-pairs. The Information Generation module creates and posts the new information fragments (amino acid sequences) in response to triggers (behaviors) from the Pattern-matching module, thus avoiding transferring sequence data to and from the client. So far, the process has been fairly straightforward and monolithic. It is well known that amino acid sequences are valuable for a wide variety of research tasks, so adding the results to the permanent store is justified.

Now, let's examine a circumstance where the meaning and value of information is not clear in advance. Suppose a researcher has discovered that the presence of a particular amino acid sequence at a particular location is responsible for the susceptibility to a certain type of cancer. Furthermore, she has developed a virus that can change a base-pair at that location, thus eliminating the susceptibility to cancer. This virus achieves this by bonding to the sequence and then changing it. At this point the researcher has a number of questions (assume, for this

example, that there are sequences available for many people, not just a few, and that there are a considerable number of annotations):

- Who is susceptible?
- What effect will changing like sequences at different locations have (the virus is not selective)?
- Can variations of the virus be used to achieve other things?

Answering the first question is simple. The Convergence/Indexing module is used to limit searches to the appropriate location based on non-sequence data. The Pattern-matching module is used to search through the selected sequence data.

The second question requires first searching through all sequence data (at least for humans) to locate like sequences in other locations. If any exist, a temporal shadow document is created containing sequences as they would be changed by the virus. The researcher can then analyze the resulting sequences in order to identify known correlations. The results of the analysis can be added to the shadow document as new annotations. Now the researcher has new information that can optionally be made a part of a permanent document.

The third question probably can't be answered immediately, but much can be done to facilitate future research along these lines. Base-pairs map directly into amino acids, and the function of amino acids is well known. Sequences of amino acids map to other, not so well known, things. There is mounting evidence that genetic sequences are fractal, which would imply that there are a variety of sequence granularities that correspond to a variety of functions. It may follow, then, if an amino acid sequence of a particular length corresponds to a susceptibility to a particular form of cancer that other sequences of the same length correspond to other things. Also, if variations of the virus attach themselves to different sequences of the same length, representing sequence data at this granularity will facilitate future research. Thus, it makes sense to post sequence data based on this granularity as a permanent part of the primary genetic information store.

Conclusion

There is nothing unusual about this process of building knowledge (it's how we, as humans, do it) except for one thing – it is utterly foreign to the way computerized information management systems have worked. The Xpriori Knowledge Building Server has been designed to operate in direct support of this process, minimizing custom programming and eliminating database programming altogether while significantly reducing network traffic. As a result, knowledge building applications ranging from research tools to data mining systems can be built exhibiting unprecedented performance and ease of use.

# # #