Xpriori
Technical
Paper

An Alternate Method of Indexing XML Documents

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

This paper is an adjunct to the paper “Xpriori XML Document Storage and Indexing Engine Architecture” (Brandin, Chris, Xpriori, 2000).

An additional method for indexing XML documents is disclosed that offers increased performance for many applications. This method can be used in conjunction with other indexing modes previously disclosed, or it can be used by itself.

Using the New Indexing Mode

In order to use this indexing method a new piece of information must be provided when XML documents are posted – the “convergence point”. In many applications, XML documents represent information that can be logically grouped into “records”. In the Phonebook document file, for example, a record would be an XML fragment that starts with the Tag “Listing”, and the convergence point would be “Phonebook\Listing\”. Each XML document can have its own convergence point, so the XML document repository server can contain heterogeneous documents.

Convergence is not accomplished through navigation, rather convergences are pre-calculated based on the supplied “convergence point”. Index entries do not point to their own lines in the Information Couplet document, rather they point to Tag at the beginning of a “record”. Attribute values for Tags that are higher in the hierarchy than the convergence Tag are “pushed” (replicated) at the same level as the convergence Tag (and point to it).

Changes in the Files

The same example XML document will be used as in the previous paper:

```
1     <Phonebook country=USA>
2         <Listing category=Residential>
3             <Name>
4                 <Last> Brandin </Last>
5                 <First> Chris </First>
6             </Name>
7             <Address>
8                 <Number> 1234 </Number>
9                 <Street> Main Street </Street>
10                <City> Colorado Springs </City>
11                <State> CO </State>
12                <Zip> 80909 </Zip>
13            </Address>
14            <Telephone>
15                <Areacode> 719 </Areacode>
16                <Number> 555-1206 </Number>
17            </Telephone>
18        </Listing>
19        <Listing category=Residential>
20            <Name>
21                <Last> Brandin </Last>
22                <First> Alice </First>
```

```
                </Name>
                <Address>
                        <Number> 1234 </Number>
                        <Street> Main Street </Street>
                        <City> Colorado Springs </City>
                        <State> CO </State>
                        <Zip> 80909 </Zip>
                </Address>
                <Telephone>
                        <Areacode> 719 </Areacode>
                        <Number> 555-1061 </Number>
                </Telephone>
        </Listing>
        <Listing category=Business>
                <Name> Xpriori </Name>
                <Address>
                        <Number> 2864 </Number>
                        <Street> South Circle Drive </Street>
                        <Suite> 1200 </Suite>
                        <City> Colorado Springs </City>
                        <State> CO </State>
                        <Zip> 80906 </Zip>
                </Address>
                <Telephone>
                        <Areacode> 719 </Areacode>
                        <Number> 576-9780 </Number>
                </Telephone>
        </Listing>
</Phonebook>
```

The resulting flattened document is similar to the previous paper’s, only three new lines have been added (they are underlined below) – these are “pushed” Attributes (other elements of the flattened document have been omitted for clarity):

```
Phonebook>@country>USA
&Phonebook>@country>USA
Phonebook>Listing>@category>Residential
Phonebook>Listing>Name>Last/>Brandin
Phonebook>Listing>Name/>First/>Chris
Phonebook>Listing>Address>Number/>1234
Phonebook>Listing>Address>Street/>Main Street
Phonebook>Listing>Address>City/>Colorado Springs
Phonebook>Listing>Address>State/>CO
Phonebook>Listing>Address/>Zip/>80909
Phonebook>Listing>Telephone>Areacode/>719
Phonebook>Listing>Telephone/>Number/>555-1206
&Phonebook>@country>USA
Phonebook>Listing>@category>Residential
Phonebook>Listing>Name>Last/>Brandin
Phonebook>Listing>Name/>First/>Alice
Phonebook>Listing>Address>Number/>1234
```

18	Phonebook>Listing>Address>Street/>Main Street
19	Phonebook>Listing>Address>City/>Colorado Springs
20	Phonebook>Listing>Address>State/>CO
21	Phonebook>Listing>Address/>Zip/>80909
22	Phonebook>Listing>Telephone>Areacode/>719
23	Phonebook>Listing>Telephone/>Number/>555-1061
24	&Phonebook>@country>USA
25	Phonebook>Listing>@category>Business
26	Phonebook>Listing>Name/>Xpriori
27	Phonebook>Listing>Address>Number/>2864
28	Phonebook>Listing>Address>Street/>South Circle Drive
29	Phonebook>Listing>Address>Suite/>1200
30	Phonebook>Listing>Address>City/>Colorado Springs
31	Phonebook>Listing>Address>State/>CO
32	Phonebook>Listing>Address/>Zip/>80906
33	Phonebook>Listing>Telephone>Areacode/>719
34	Phonebook/>Listing/>Telephone/>Number/>576-9780

An information Couplet document is produced in the same way as described in the previous paper. As it maps directly to the flattened XML representation, it will not be shown here.

A method-2 index is built, except this time the association points to the Couplet that represents the beginning of the XML fragment that has been pre-defined as the convergence point:

Index Entry	Association	
Phonebook>@country>USA		1
&Phonebook>@country>USA	2	
Phonebook>Listing>@category>Residential		2
Phonebook>Listing>Name>Last>Brandin		2
Phonebook>Listing>Name>First>Chris	2	
Phonebook>Listing>Address>Number>1234	2	
Phonebook>Listing>Address>Street>Main Street		2
Phonebook>Listing>Address>City>Colorado Springs	2	
Phonebook>Listing>Address>State>CO	2	
Phonebook>Listing>Address>Zip>80909		2
Phonebook>Listing>Telephone>Areacode>719	2	
Phonebook>Listing>Telephone>Number>555-1206		2
Phonebook>@country>USA		13
Phonebook>Listing>@category>Residential		13
Phonebook>Listing>Name>Last>Brandin		13
Phonebook>Listing>Name>First>Alice	13	
Phonebook>Listing>Address>Number>1234	13	
Phonebook>Listing>Address>Street>Main Street		13
Phonebook>Listing>Address>City>Colorado Springs	13	
Phonebook>Listing>Address>State>CO	13	
Phonebook>Listing>Address>Zip>80909		13
Phonebook>Listing>Telephone>Areacode>719	13	
Phonebook>Listing>Telephone>Number>555-1061		13

Phonebook>@country>USA	24
Phonebook>Listing>@category>Business	24
Phonebook>Listing>Name>Xpriori	24
Phonebook>Listing>Address>Number>2864	24
Phonebook>Listing>Address>Street>South Circle Drive	24
Phonebook>Listing>Address>Suite>1200	24
Phonebook>Listing>Address>City>Colorado Springs	24
Phonebook>Listing>Address>State>CO	24
Phonebook>Listing>Address>Zip>80906	24
Phonebook>Listing>Telephone>Areacode>719	24
Phonebook>Listing>Telephone>Number>576-9780	24

Another index is built to support method-3 duplicate resolution (see the paper: "Management of Duplicate Data Elements in DPP™ Virtual Associative Memories"). Note that the convergence point number has been pre-pended to the lookup index entry value. This index has no associations because it is only necessary to indicate whether an entry exists or not:

Index Entry

#000000002Phonebook>@country>USA
#000000002Phonebook>Listing>@category>Residential
#000000002Phonebook>Listing>Name>Last>Brandin
#000000002Phonebook>Listing>Name>First>Chris
#000000002Phonebook>Listing>Address>Number>1502
#000000002Phonebook>Listing>Address>Street>East Pikes Peak Avenue
#000000002Phonebook>Listing>Address>City>Colorado Springs
#000000002Phonebook>Listing>Address>State>CO
#000000002Phonebook>Listing>Address>Zip>80909
#000000002Phonebook>Listing>Telephone>Areacode>719
#000000002Phonebook>Listing>Telephone>Number>630-1206
#000000013Phonebook>@country>USA
#000000013Phonebook>Listing>@category>Residential
#000000013Phonebook>Listing>Name>Last>Brandin
#000000013Phonebook>Listing>Name>First>Alice
#000000013Phonebook>Listing>Address>Number>1502
#000000013Phonebook>Listing>Address>Street>East Pikes Peak Avenue
#000000013Phonebook>Listing>Address>City>Colorado Springs
#000000013Phonebook>Listing>Address>State>CO
#000000013Phonebook>Listing>Address>Zip>80909
#000000013Phonebook>Listing>Telephone>Areacode>719
#000000013Phonebook>Listing>Telephone>Number>578-1061
#000000024Phonebook>@country>USA
#000000024Phonebook>Listing>@category>Business
#000000024Phonebook>Listing>Name>Xpriori
#000000024Phonebook>Listing>Address>Number>2864
#000000024Phonebook>Listing>Address>Street>South Circle Drive
#000000024Phonebook>Listing>Address>Suite>1200
#000000024Phonebook>Listing>Address>City>Colorado Springs
#000000024Phonebook>Listing>Address>State>CO
#000000024Phonebook>Listing>Address>Zip>80906
#000000024Phonebook>Listing>Telephone>Areacode>719

#000000024Phonebook>Listing>Telephone>Number>576-9780

The Query Convergence Procedure

This indexing method makes no use of PLevel or the Parent pointer to achieve convergence; rather the equivalent is accomplished by matching association values. Unless explicitly specified otherwise, all query parameters must converge at the pre-defined convergence point for there to be a match. The reason some Attributes are “pushed” is so that they can converge at a hierarchal level lower than the Tag’s to which they belong. Basically, there is a match if there is an entry for each query parameter that has the same association value. For example, a query for “Phonebook>Listing>Name>Last>Brandin” and “Phonebook>Listing>Address>State>CO” would return two listings, one starting at line 2 and the second at line 13 of the flattened XML (lines 2 and 19 of the original XML document).

The method-3 index is used to optimize searches when one, or more, large sets are converged against a smaller one. Suppose we want to find listings that contain the following elements:

- Phonebook>Listing>Telephone>Number>576-9780
- Phonebook>Listing>Name>Xpriori
- Phonebook>Listing>Address>State>CO

We can locate the listings in an optimized way by executing the following steps:

1 Lookup the number of entries for each query element (method-2 indices maintain an instance count for each entry).

2 Select the query element with the lowest instance count (the telephone number in this example), and look it up in the method-2 index. The association returned is 24.

3 Determine if the other query elements have en entry in the method-3 index with the appropriate key value. In this example, “#000000024Phonebook>Listing>Name>Xpriori” and “#000000024Phonebook>Listing>Address>State>CO” both have entries, resulting in a match

Appendix: Related Papers

Brandin, Chris, “A Definition of Digital pattern Processing™”

Brandin, Chris, “DPP™ Memory Management”

Brandin, Chris, “Three-Dimensional Content Scanning Using DPP™”

Brandin, Chris, “Optimized Coding Methods for Icon Generation and Manipulation In DPP™”

Brandin, Chris, “Management of Duplicate Data Elements in DPP™ Virtual Associative Memories”

Brandin, Chris, “Behavioral Set and Field Descriptor Implementations in DPP™”

Brandin, Chris, “Xpriori XML Document Storage and Indexing Engine Architecture”

Direen, Harry and Phillips, Keith, “Finite Fields and Properties of the Xpriori Icon Generator, Associative Processing Unit, and Associative Memory Controller used in Digital Pattern Processing™”

Direen, Harry, “Duplicate Tree Structures in DPP™ Virtual Associative Memories”

Direen, Harry, “The application of DPP to Storage and Retrieval of XML Data”

# # #