Desirable Properties for XML Update Mechanisms

Martin F. O'Connor

Interoperable Systems Group
Dublin City University
Ireland

March 22, 2010

Introduction

Labelling Schemes for XML may be broadly categorised into three categories:

- Containment Schemes.
- Prefix Schemes.
- Prime Number Schemes.

Our focus is on Dynamic Labelling Schemes (DLS) supporting XML updates

- Leaf node insertions.
- Internal node insertions.
- Subtree insertions.
- Node deletions.

Motivation

Existing DLS have different characteristics with respect to one another in terms of:

- The types of queries supported.
- The size of node labels.
- The update cost.

To date, evaluations of DLS have been limited to an analysis of:

- The average size and growth rate of node labels.
- A computational complexity analysis of the update cost.
- A comparative performance analysis with other DLS.

There is a bigger question to be asked:

- What constitutes a good DLS for XML?

Contribution

In our paper, we provide:

- A comprehensive survey and review of the principle DLS for XML proposed to date.
- We analysed and identified the core properties of each individual DLS.
- We specified a template of properties that we consider to be representative of the characteristics of a *good* DLS.
- We constructed an evaluation framework based on our template of properties to assess and critique each of the labelling scheme in our survey.
- Our evaluation framework should assist in the specification and analysis of new DLS.
- The evaluation framework is intended to complement existing analysis techniques and not replace them.

XML Tree

- The basic structure underlying XML is a rooted ordered tree.

A tree is an abstract datatype

- A tree has no defined Application Programming Interface (API).
- A tree has no defined object model.
- A tree is only a conceptual model.

XPath defines its operations in terms of a tree representation of the XML document.

- This motivates the need for a labelling scheme and an encoding scheme for the XML tree.
- However, the terms *labelling scheme* and *encoding scheme* are often used interchangeably, and thus require clarification.

Labelling Scheme

- An XPath processor must be capable of distinguishing between nodes in an XML tree.
- The purpose of a labelling scheme is to assign unique labels to each node in the XML tree.
- The node labels should also facilitate the evaluation of node order.

```
<book>
  <title genre="Fantasy"> Wayfarer </title>
  <author> Matthew Dickens </author>
  <publisher>
    <editor>
      <name> Destiny Image </name>
      <address> USA </address>
    </editor>
    <edition year="2004"> 1.0 </edition>
  </publisher>
</book>
```

Figure: A Sample XML file.

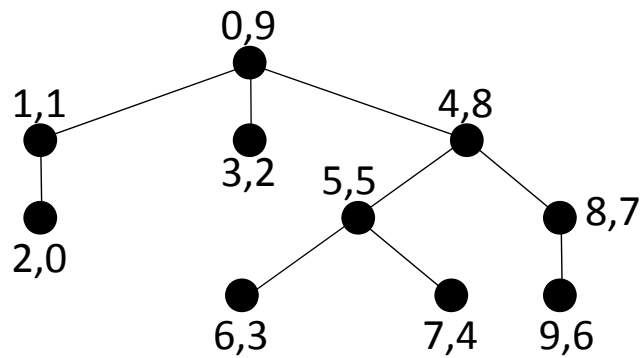

Figure: Preorder / Postorder Labelled XML Tree.

Encoding Scheme

No labelling scheme captures:

- The node type.
- The element or attribute names.
- The data contents of the XML elements.

```
<book>
  <title genre="Fantasy"> Wayfarer </title>
  <author> Matthew Dickens </author>
  <publisher>
    <editor>
      <name> Destiny Image </name>
      <address> USA </address>
    </editor>
    <edition year="2004"> 1.0 </edition>
  </publisher>
</book>
```

Figure: Sample XML file.

Pre	Post	Node Type	Parent (Pre)	Name	Value
0	9	Element		book	
1	1	Element	0	title	Wayfarer
2	0	Attribute	1	genre	Fantasy
3	2	Element	0	author	Matthew Dickens
4	8	Element	0	publisher	
5	5	Element	4	editor	
6	3	Element	5	name	Destiny Image
7	4	Element	5	address	USA
8	7	Element	4	edition	1.0
9	6	Attribute	8	year	2004

Figure: Sample Encoding of XML file.

Document Order

Document order is defined for all nodes in the XML tree.

- Document order corresponds to the order in which the first character of each element occurs in the XML document.
- XPath requires node sequences in the result set to be returned in document order.

There are three generic approaches to capturing document order.

- Global Order.
  - The node label contains an identifier representing its absolute position in the XML tree.
- Local Order.
  - The node label contains an identifier representing its position relative to its siblings nodes.
- Hybrid Order.
  - The node label permits the evaluation of both the absolute and relative positions of the node.

Document Order - Comparison

Each of the approaches have advantages and limitations.

- Global Order.
  - Efficient for query processing over static data.
  - Very high update cost.
- Local Order.
  - More update friendly.
  - Difficult to evaluate following and proceding axes.
- Hybrid Order.
  - Strikes a balance between Global and Local order.
  - Most DLS follow the hybrid approach to document order.

Containment Schemes

- Also known as Interval based labelling schemes and Region Encoded labelling schemes.
- They exploit the properties of tree traversal to determine the structural relationships between nodes.
- Most containment schemes adopt the global order approach to document order and thus are unsuitable as DLS.

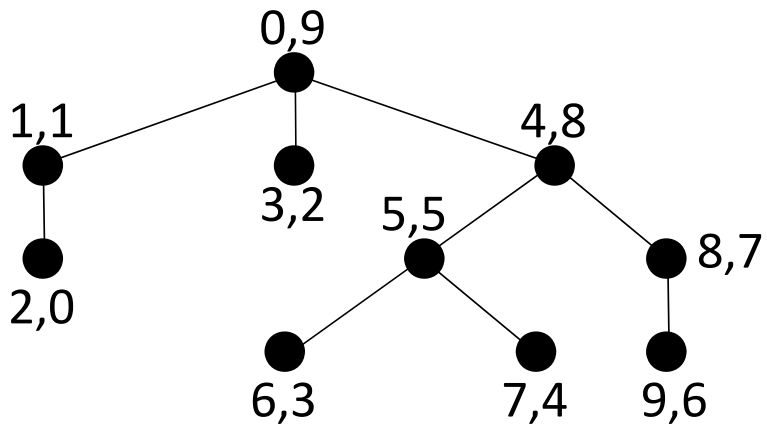

Figure: Preorder / Postorder Labelled XML Tree.

Prefix Schemes

In a prefix scheme, the node label consists of a concatenation of:

- The parent's label.
- A delimiter.
- A positional identifier of the node itself.

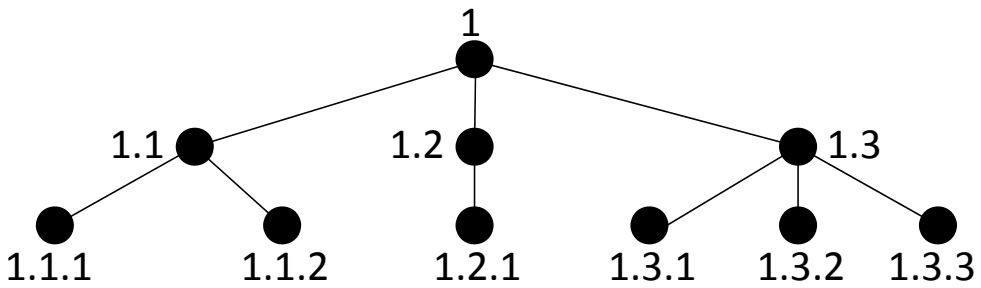

Figure: DeweyID Labelled XML tree.

- DeweyID is a prefix labelling scheme adapted from the Dewey Decimal Classification for the organisation of library collections.

Prefix Schemes - A survey and review

In our paper, we review and analyse the following prefix based labeling schemes:

- OrdPath labelling scheme.
- Dynamic Level Numbering labelling scheme (DLN).
- LSDX labelling scheme.
- Compressed Dynamic labelling scheme (Com-D).
- Quaternary Encoding Dynamic labelling scheme (QED).
- Compact Dynamic Binary String labelling scheme (CDBS).
- Compact Dynamic Quaternary String labelling scheme (CDQS).
- Vector dynamic labelling scheme (Vector).

Characteristics of DLS for XML

- Each of the dynamic labelling schemes reviewed have differing characteristics offering distinct advantages and limitations with respect to one another.
- From our analysis, we attempted to extract and identify the core properties that are characteristic of a *good* DLS for XML.
- These properties should contribute to a framework of metrics by which all new and existing DLS may be evaluated.

Template of Properties

Document Order

- Global, Local or Hybrid.

Encoding Representation

- The labelling scheme requires a fixed or variable length storage representation for node labels.

Persistent Labels

- The labelling scheme assigns node labels that are both unique and persistent. Persistent labels ensure all deletion and insertion operations on the XML tree do not effect existing nodes - and thus maintain node identify.

Template of Properties (2)

XPath Evaluations

- The value of the node label alone permits the evaluation of ancestor-descendant, parent-child and sibling-based relationships.

Level Encoding

- The node labels indicate the level or depth of the node in the XML tree.

Overflow Problem

- The labelling scheme is subject to the overflow problem. Most fixed-length and variable-length storage representation are subject to overflow after a large volume of insertions and thus do not scale.

Template of Properties (3)

Compact Encoding

- The labelling scheme supports a compact storage representation.

Growth Rate

- The labelling scheme maintains a reasonably constrained growth rate under various update scenarios such as: frequent random updates, frequent uniform updates, frequent skewed updates (frequent updates at a fixed position).

Recursive Labelling Algorithm

- The labelling scheme employs a recursive algorithm to compute and assign positional identifiers during the initial labelling of the XML tree. The use of a recursive algorithm is computationally more expensive as it requires multiple passes of the XML tree.

The Evaluation Framework

Labelling Schemes	Document Order	Encoding Rep.	Persistent Labels	Xpath Eval.	Level Enc.	Overflow Prob.	Orthogonal	Compact Enc.	Division Comp.	Recursion Alg.
XPath Accelerator [9]	Global	Fixed	N	P	F	N	N	F	F	F
XRel [30]	Global	Fixed	N	P	F	N	N	F	F	F
Sector [23]	Hybrid	Fixed	N	P	N	N	N	P	F	N
QRS [2]	Global	Fixed	N	P	N	N	N	P	F	F
DeweyID [22]	Hybrid	Variable	N	F	F	N	N	N	F	F
Ordpath [18]	Hybrid	Variable	F	F	F	N	N	N	N	F
DLN [3]	Hybrid	Fixed	N	F	F	N	N	N	F	F
LSDX [7]	Hybrid	Variable	N	F	F	N	N	N	F	F
ImprovedBinary [13]	Hybrid	Variable	F	F	F	N	N	N	N	N
QED [14]	Hybrid	Variable	F	F	F	F	F	N	N	N
CDQS [16]	Hybrid	Variable	F	F	F	F	F	F	N	N
Vector [27]	Hybrid	Variable	F	P	N	F	F	F	F	N

Analysis of Results

- No two labelling schemes share the exact same properties. This is a positive finding because there is no ONE-SIZE-FITS-ALL solution to the XML Update problem.

The evaluation framework can assist in the selection of a DLS for an XML Repository.

- It provides a template and metrics to enable the database designer or data modeller to select a DLS suitable for their requirements.
- E.g., An XML repository that requires document history to be recorded and version control to be enabled would select a DLS supporting persistent labels.
- Alternatively, an XML repository that will consume large documents on a regular basis should consider a labelling scheme not subject to the overflow problem.

Conclusion

In our paper:

- We sought to identify a set of desirable properties that would constitute a good dynamic labelling scheme for XML.
- We examined the principle schemes proposed to date and focused on the primary issue of structural updates to XML documents.
- We constructed an evaluation framework that will assist in the specification of new dynamic labelling schemes and in the evaluation of existing schemes.