

SiXML Version 1.2

Succinct indexable XML (SiXML) version 1.2 provides an efficient in-memory representation of large XML documents, with stable and predictable memory usage. It provides a framework around which a variety of XML processing APIs can be implemented. SiXML is based on succinct data structures, which use an information-theoretically minimum amount of space to represent a given data type
<http://www.cs.le.ac.uk/SiXML/>

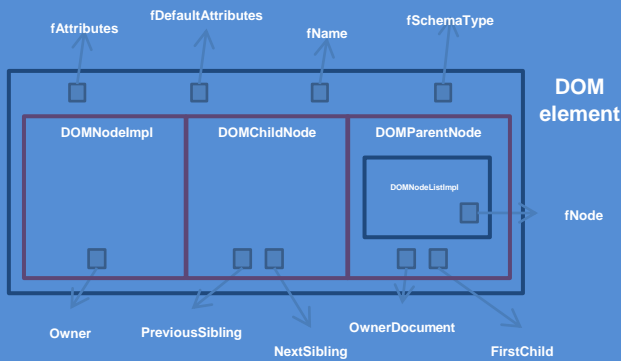
Case Study: Xerces-C 2.8

Represents XML documents in main memory as a tree

- Fast navigation
- Fast data access/modification.

A Xerces-C DOM element node is almost all pointers and requires 416 bits on a 32-bit machine

```
<book catalogue="java">
  <author>Selman</author>
  <title>Java3d programming</title>
  <year>2000</year>
</book>
```



SiXDOM Performance

In the table below, we show the space usage of SIXDOM-(CT) 1.1 compared to Xerces-C, Saxon's TinyTree and a state of the art XML compressor, XMILL.

Percentage given is proportional to the file size.

File	Size	SIXDOM 1.1	Xerces-C	Saxon	SIXDOM-CT	XMILL
Orders.xml	5MB	37%	451%	157%	17%	12%
Lineitem.xml	32MB	28%	399%	161%	13%	5%
XCDNA.xml	607MB	50%	491%	130%	14%	8%

Memory usage typically is less than 50% of the file size. SIXDOM-CT compresses the text so space usage is even better.

Running Times

Test: Full navigation of document, retrieving all text nodes

Result: SiXDOM 1.1 was ~1.8 times slower than Xerces-C DOM

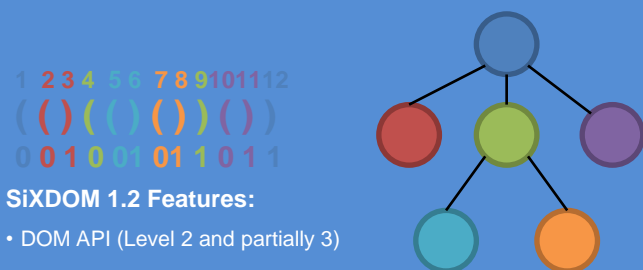
SiXDOM Benefits

- Very low memory footprint
- Fast processing
- Essentially a plug-in replacement for DOM

Pointerless Data Structure

We represent the structure of the tree as a parenthesis string

- Number nodes in document-order
- Efficient index for "parenthesis-matching"
- Uses ~4.88 bits/node
- Also using pointerless data structures for attributes, text etc.



SiXDOM 1.2 Features:

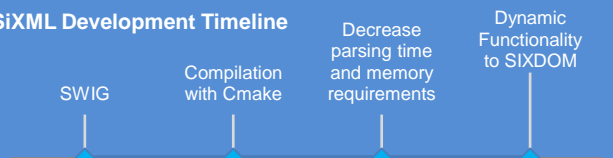
- DOM API (Level 2 and partially 3)
- Only contains the static DOM methods
- SWIG Bindings
- Ported to 64-bit systems

Future Developments

Recent developments on "dynamic" succinct data structures:

- support changes to parenthesis tree representation
- Can create 256 million node document tree by parsing XML document and inserting one node at a time:
 - 515 seconds (2µs per node)
 - Subsequent document-order traversal in 8.31 seconds
- Creating 32 million node tree ~40% faster than with Xerces

SiXML Development Timeline



SiXDOM 1.2 is out in Spring 2012

For more information visit <http://www.cs.le.ac.uk/SiXML/>