



Xen – Unifying XML, Databases and OO

Swaroop C H

www.g2swaroop.net

1PI00CS090

8th Sem CSE

Guide

Mrs. Shrividya



“Xen is a proposed extension to popular object oriented programming languages such as C# and Java with native support for XML and Databases.”

- Xen is an extension to existing OO languages like C# and Java.
- XML documents and fragments become first-class citizens of the language.
- Database access is transparent.



Xen Research

The authors of the Xen research paper are

Erik Meijer
Microsoft Research

Wolfram Schulte
Microsoft Research

Gavin Bierman
University of Cambridge Computer
Laboratory



- ◆ Our discussion assumes a working knowledge of C#. However, Xen can be applied to Java or even VB as well.
- ◆ Since C# is used, Xen can be built upon the .NET and Mono platforms.
- ◆ A working knowledge of XML, XSD (XML Schema), XPath, XQuery and SQL is required.



Today, we access XML using various models and APIs:

- ◆ DOM Strategy
 - ➔+ Simple and fast
 - ➔- Entire XML document has to be loaded into memory
- ◆ SAX Strategy
 - ➔+ Can handle large documents
 - ➔+ Good for parsing documents
 - ➔- Very difficult and cumbersome
 - ➔- Non-intuitive

- ◆ The XmlTextReader class in .NET
 - + Innovative API
 - + Intuitive
 - Logic encoded in foreach loops

The gist of the present day scenario is that although XML itself is simple, *programming XML is **hard**.*

- ◆ We access databases using various APIs such as ADO.NET, JDBC, Perl DBI, Python DB PEP, etc.
- ◆ All these assume an **I/O model** which is non-intuitive and cumbersome.
- ◆ Lot of text manipulation and parsing is required
- ◆ Invalid SQL has to be detected by the database server



- ◆ Xen combines the three in a simple and elegant way:
 - ➔ *Circles* (Object oriented programming)
 - ➔ *Triangles* (XML)
 - ➔ *Rectangles* (Databases)
- ◆ XML and Database access become part of the language itself.
- ◆ The XML and database usage are **validated** by the Xen compiler!



Where is the problem?

- ♦ The problem is the Impedance Mismatch - *circles can't easily be made into triangles!*
- ♦ Current “databinding” approaches such as JAXB and the .NET xsd.exe hide the XML instead of embracing it.

Mismatch between XML and

- ◆ Probably the deepest and most fundamental difference between the XML and the object data-model is that
 - XML assumes *node-labelled* trees/graphs
 - OO assumes *edge-labelled* trees/graphs
- ◆ XML has different element and attribute axis. There is no equivalent of this axis in OO.
- ◆ Multiple occurrences of the same child element occur in XML but OO languages do not allow duplicated fields.



- ♦ In XSD, occurrence constraints are part of the container instead of the type.
 - ➔ For example, element “A” has at most 10 children named “B”. Whereas in OO, we have an array of type “B”. Also, the size of the array is not constrained directly.
- ♦ XML has mixed content such as text nodes and element nodes. Since objects represent data, not documents, there is no natural interpretation for mixed content in the object world.



- ♦ At a foundational level, there is a significant gulf between the XML and object data-models. This impedance mismatch is too big to attempt a complete integration.
- ♦ How to proceed? We take the object data-model as the starting point. This is the model that programmers are familiar with.
- ♦ A careful integration is made. It may not support every possible situation but must be rich enough to support many potential scenarios.



- ◆ “We consider XML 1.0 as simply syntax for serialized object instances”.
 - ➔ A simple intuitive approach.
 - ➔ Programmer sees only instances of objects
 - ➔ The Xen compiler sees XML
- ◆ To support rich scenarios, there are new type constructors such as unions, sequences and streams. Interestingly, these types look very similar to Relax NG compact notation.



- Class

$C ::= \text{class } I : I \dots I \{ T; A^* M^* \}$

- Attribute

$A ::= \text{attribute } T I;$

- Legend: C – class declarations, I – identifiers, T – types, A – attribute declarations, M – method declarations

- ♦ *sequence* type corresponds to
 - XSD <sequence> particle
 - the DTD (...,...)
- ♦ *choice* type corresponds to
 - C-style union
 - XSD <choice> particle
 - DTD (..|..) construct
- ♦ *all* type corresponds to
 - XSD <all> particle
- ♦ *possibly empty stream* type corresponds to
 - XSD minOccurs="1"
maxOccurs="unbounded"
 - DTD * construct



Xen type system (continued)

- ♦ nonempty stream type corresponds to
 - XSD minOccurs="1"
maxOccurs="unbounded" occurrence constraint
 - DTD + construct
- ♦ option type corresponds to
 - XSD minOccurs="0" maxOccurs="1"
 - DTD ? construct

```
T ::= sequence { ... T I? ; ... }
      | choice { ... T I? ; ... }
      | all { ... T I? ; ... }
      | T* | T+ | T?
      | I
      | Xen
```

- A bibliography of books is given by the following DTDs:

```
<!ELEMENT bib (book* )>  
<!ELEMENT book (title, (author+  
| editor+),  
publisher, price)>  
<!ATTLIST book year CDATA  
#REQUIRED>
```



```
public class book {  
    sequence {  
        string title;  
        choice {  
            sequence{ editor editor; }  
+;  
            sequence{ author author; }  
+;  
        }  
        string publisher;  
        int price;  
    }  
    attribute int year;  
}
```



- Xen internalizes XML serialized objects into the language, allowing programmers to use XML fragments as object literals.

```
book b = <book year="2004">
  <title>A Byte of Python</title>
  <author>
    <first>Swaroop</first><last>C
    H</last>
    <publisher>APress</publisher>
    <price>100.00</price>
  </book>;
```



- ♦ Xen allows arbitrary embedded code, using curly braces as escape syntax. As Xen is statically typed, the type of the embedded expression must be of an “allowable” type.

```
book b = <book year="2004">
  <title>A Byte of Python</title>
  <author>
    <first>Swaroop</first><last>C
    H</last>
    <publisher>APress</publisher>
    <price>{LookupPrice("Byte")}</price>
  </book>;
```



More Xen Concepts

- ◆ Generators

```
author* authors = new (author1,  
    author2);
```

- ◆ Iterators

```
book b = ...  
author* authors = b.author;  
foreach(author a in authors)  
    Console.WriteLine(a);
```

- ◆ Filtering

```
book* Abooks =  
    bib.book[it.publisher == "APress" &&  
    it.year > 2003];
```



- ◆ Lifting

```
public class bibliography {  
    book* books;  
}
```

```
string* titles = bib.books.title;
```

- ◆ Apply-to-all

```
→ sequence{string; int;}* bs =  
    Abooks.{return new(it.title,  
        it.year);};
```

```
→ Abooks.{Console.WriteLine(it);}
```



Where's the database?

- ♦ Databases are integrated in the same way.

```
CREATE TABLE Customer  
( name string NULL,  
  custid int);
```

becomes

```
sequence{ string? name; int custid;}*  
Customer;
```

```
void Render(HTMLTextWriter output)
{
    output.add(
        <table>
            <tr><th>Product</th>
                <th>Quantity</th>
                <th>Price</th></tr>
            {select <tr><td>{p.ProductName}</td>
                <td>{o.Quantity}</td>
                <td>{o.Price}</td>
                </tr>
                from o in db.OrderDetails inner join
                    p in db.Products
                    on p.ProductID == o.ProductID
            }
        </table>);
```



- ♦ It is possible for a modern object-oriented (circles) language to provide first-class support for manipulating both relational (rectangles) and hierarchical data (triangles) in a sound and statically typed manner. This is demonstrated by Xen, the hypothetical extension of C#.
- ♦ Xen unifies CLR, XML and SQL.



Bibliography

- ♦ Erik Meijer, Wolfram Schulte, Gavin Bierman - “Programming with Circles, Triangles and Rectangles”
 - <http://www.cl.cam.ac.uk/~gmb/Papers/vanilla-xml2003.html>
- ♦ Extremetech.com - “Microsoft expands .NET with Xen”
 - <http://www.extremetech.com/article2/0,3973,1441099,00.asp>
- ♦ Williamson - The Complete Reference, XML
- ♦ Jesse Liberty – Programming C#
- ♦ The MSDN .NET Documentation
- ♦ The ECMA C# and CLR Standards



Thank you!

Thank you!

Contact Details

Email - xen@g2swaroop.net

Website - www.g2swaroop.net



Swaroop C H
Feb '04 – Jun '04

Xen