

BetterXML

(an introduction)

1) A story of ill treatment

(how xml is being mis-used and abused)

In the beginning there was purpose

The original intent of XML, as found at www.w3.org/xml/ :

"Originally designed to meet the challenges of large scale electronic publishing"

But is the purpose clear?

- [What is meant by large scale?

- Sheer size of content?

- ex: 20 page book is small; a 300 page book is large scale.

- The size of the audience?

- ex: 100 people is small; 350 million is large scale

- The number of people participating in the creation

- ex: 1 guy from concept to delivery is small; multiple writers, a cadre of editors, and marketing company for delivery is large scale

— [What is meant by publishing?

— Content format and presentation (i.e. how it all looks)?

— Creation of the product for delivery

— print a book / displaying on a web site / sky writing

— Distribution

— selling a book / loading via ftp to a server / flying the plane

So clarification is needed

Disclaimer: These definitions come from no verifiable source; they are postulations of the author.

— [Publishing

- The gathering and formatting of information so that it can be made readily available to an audience

— [Large scale

- Let's just drop the "large scale". It is too general to be refined and no use of XML that is limited to large scale readily comes to mind
- The author is aware that this is a weasily avoidance of a difficult task

XML: The right standard for the job

— [In order to be of use to the limitless variety of data/info that can be published, you must be incredibly flexible.

— [XML is incredibly flexible. Just as the variety of data that you need to define is limitless, so is the amount of distinct XML tags

— [Every type of new data you encounter needs its own descriptor and with XML you can create the descriptors as you needed.

— [With XML Schema you also have a good deal of flexibility when it comes to describing your descriptors.

So the problem is not with XML

— [XML is intended to have a user defined structure, so the user who is formatting the data should be able to use XML to define/describe/tag the data as he sees fit.

— if the user wants to tag the data associated with a person's phone number in the following way, he can.

```
<PhoneNumber>  
  <AreaCode>773</AreaCode>  
  <Trunk>555</Trunk>  
  <DistinctTail>1234</DistinctTail>  
</PhoneNumber>
```


The problem is with the programs

- [The ability to create the format for the data is NOT the problem
- [The problem is the programmatic interaction with data, or in other words:

The problem is with the programs that view, edit, populate, and manipulate the data found within the structure of an XML document.

The problem further explained

(or, "beating you over the head with the point")

— [It is very easy to create the following XML:

```
<PhoneNumber>  
  <AreaCode>773</AreaCode>  
  <Trunk>555</Trunk>  
  <DistinctTail>1234</DistinctTail>  
</PhoneNumber>
```

— [But the data structure is useless if no program can provide the functionality required to edit, view, and otherwise manipulate the data.

— [If the structure of the data cannot be understood then having the phone number as presented above is equivalent to not having the phone number at all.

The end result of XML abuse

- [The form no longer follows the function of describing the data.
- [Instead, the form with which the data is described follows the function of the application(s) that use the document.
- [The flexibility of form, which is the driving force behind XML, is meaningless if nothing can make proper use of the resulting document.

2) Patching up our differences

(how XElement enables the proper and affective use of XML)

Making a BetterXML

— [BetterXML is an initiative aimed at allowing for a better overall use of XML

— [The target audience is the programmer. We want to make it easier and quicker to create programs that use XML data sources and have a flexibility that can match that which is inherent in XML.

— [XElement is only part of BetterXML (there is also NaturalXML) but I am focusing on XElement because it is what I work with/on

XElement: the basics

— [XElement parses an XML document and creates an intuitive and flexible data tree

— [Once the data is in the tree, the data is easier to get at, change, remove, and add to than is normally the case using normal SAX or DOM implementations

— [In the resulting structure, every XML Element is an instance of XElement and the attributes become XAttributes

An instance of XElement

— [Keeps the following data

— the parent element

— element name

— element attributes

— children elements

— PCData

For example

— [This XML

```
<PhoneNumber firstName="Joe" lastName="Schmoe">  
  <AreaCode>773</AreaCode>  
  <Trunk>555</Trunk>  
  <DistinctTail>1234</DistinctTail>  
</PhoneNumber>
```


For example

— [This XML

```
<PhoneNumber firstName="Joe" lastName="Schmoe">  
  <AreaCode>773</AreaCode>  
  <Trunk>555</Trunk>  
  <DistinctTail>1234</DistinctTail>  
</PhoneNumber>
```

becomes

For example

This XML

```
<PhoneNumber firstName="Joe" lastName="Schmoe">  
  <AreaCode>773</AreaCode>  
  <Trunk>555</Trunk>  
  <DistinctTail>1234</DistinctTail>  
</PhoneNumber>
```

becomes

```
XElement  
  Attributes{  
    firstName:Joe  
    lastName: Schmoe }  
  Name{PhoneNumber}  
  Children{  
    XElement  
      Name{AreaCode}  
      PCData{773}  
    XElement  
      Name{Trunk}  
      PCData{555}  
    XElement  
      Name{DistinctTail}  
      PCData{1234}  
  }  
}
```

Going Beyond the Data Tree

- [XElement does more than provide a better data tree

- [XElement aides in allowing function to follow form

- by allowing the programmer to extend the XElement object it becomes trivial to associate the desired behavior with an element of data

- [This means that the data inside of the XML Document becomes actionable. The programming (and program) is driven by the data instead of it being the other way around.

How is this achieved?

— [The programmer can extend the XElement object to contain the desired functionality.

— this functionality can recursively iterate through the children if the programmer decides to make it so.

— [This extension of the XElement object can be associated with one or more XML elements.

Creating the smart data tree

— [These custom extensions of XElement are instantiated upon parsing the XML Element

— [When parsing the document XElement adds a node to the tree for every element in the document

— [The type of the node is determined in the following order:

- 1) Looks for a specified association between an element and an extension of XElement
- 2) Uses reflection to look for an extension of XElement with the same name as the element
- 3) If all else fails it falls back to the default and uses XElement

So instead of this...

```
XElement
  Name{PhoneNumber}
  Children{
    XElement
      Name{AreaCode}
      PCData{773}
    XElement
      Name{Trunk}
      PCData{555}
    XElement
      Name{DistinctTail}
      PCData{1234}
  }
```

We can end up with this...

```
PhoneNumber (extends XElement)
  Name{PhoneNumber}
  Children{
    AreaCode (extends XElement implements PhoneNumPart)
      Name{AreaCode}
      PCDATA{773}
    Trunk (extends XElement implements PhoneNumPart)
      Name{Trunk}
      PCDATA{555}
    DistinctTail (extends XElement implements PhoneNumPart)
      Name{DistinctTail}
      PCDATA{1234}
  }
```

Add a dash of functionality

- [PhoneNumber has the function `getNumber`

- this function, returns all phone number parts separated by a dash

- [PhoneNumPart can require the function `getPartAsText`

- AreaCode, Trunk, and DistinctTail all implement `getPartAsText` by returning the PCData

- [The result of calling `getNumber` in the example would be:

773-555-1234

3) Beyond the Pie in the Sky

(the very real uses and possibilities of XElement)

Semantic Web Browser

— [Whoa! This is the big one! Let your imagination run wild...

— [The problem with the ideal of the semantic web and all current tools: anyone can create an data set in XML but how does my browser know how to handle it?

— [XElement can solve this by allowing people to write the XElement based plugins that are then accessible over the internet.

— [If a plugin does not exist, the user can specify a currently existing plugin to attempt to use and the program can always fall back to the default.

A True Content Management System

- [An ability to easily handle new data types using an XElement based plugin system

- [By implementing different content types we can create the ability to quickly switch from one format of the content to another

- ex: from website to print magazine to an anthology of magazine issues

- ex: from recipe submission to accepted submission to web page to a cookbook

- [Because of having default functionality for editing, a form for editing/creating data can be generated for any new data set the back-end comes across.

Rapid Development of Small Apps

— [A large amount of popular and often used applications are actually small applications with limited functionality sets that work with non-diverse data

— [Examples: email archiver, bookmark collections, recipe box, address books, LDAP manager, slideshow maker, mind map, outliner, play list creator

— [Because so many data sets fall easily in the data tree paradigm many of these data sets are expressed in XML. XElement allows for programs to handle these data sets to be created quickly.

4) Onward and Upward

(how dreaming of the possibilities is driving future development)

Future features

- [XML Schema support

- on the fly validation and generation of forms for data entry/edit/and display

- [XPath and/or XQuery

- to allow for even easier searching for and retrieval of information

- to allow the user to quickly jump to a certain level or “bookmarked location” within the XML document

- [Allow user to define a default element type

- this lets the user create a collection of default functionality that will apply to all processed elements

Freedom of Flexibility vs. Strictness of Structure

— [To stay as flexible as possible, many times it is good to not have a strict structure of data to follow

— [However, a strict structure, as can be defined in an XML Schema is often very helpful for data validation, the creation of forms for gathering/editing data, etc.

— [XElement needs to balance both of these needs and work well with and without the strict structure.

— [But what are the repercussions of this?

(Hint: they aren't fully evident yet, which is a fancy way of saying "I don't know")

There's a lot of programming to do

- [Continuing work on XElement library (such as XML Schema support, XPath support, and strengthening of the loading and using of extended XElements)
- [The creation of demo applications and full fledged applications that make use of extended XElements
- [Plugins for all purpose browsing and editing applications, such as a content management system.

For Further Information

Email: nabicht@gmail.com

Web Address: www.betterxml.org