

Processing XML With Fun

- By Eric van der Vlist (vdv@dyomedea.com)
- XML Prague 2007
- June 2007

I am a big fan of DOM, and you, what are you using to process your XML?



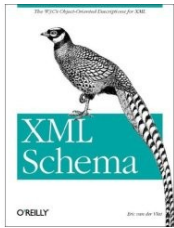
I process my XML with fun

And why should I trust you to give this talk?

About the author



- XML & Web expert and trainer, XML Guild member
- Author (or co-author) of:



– XML Schema (O'Reilly, 2002)

– RELAX NG (O'Reilly, 2003)



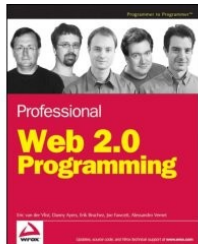
– Schematron (O'Reilly shortcut, 2006)

– XML Advanced applications (Thompson, 2006)



– Professional Web 2.0 Programming (WROX, 2006)

– Beginning XML 4th edition (WROX, 2007)



XML is too complex

XML processing is too complex, everyone says so.



- James Clark's five challenges for XML (2001):
 - Make progress yet keep XML simple
 - Don't neglect the foundations of XML.
 - **Control the processing pipeline.**
 - **Improve XML processing models.**
 - Avoid premature standardization.

Solutions are available

You see, even James Clark says that XML processing is too complex!



- Solutions have been worked out and are now available
 - To define pipelines
 - To provide simpler processing models
- This year, most of the XML Prague presentations are related to one of these topics.

XML Pipelines

Pipelines? Are you working for the oil industry or what?



- XML pipelines are declarative descriptions of sequences of actions to be carried on XML documents.
 - See the presentations by Norm Walsh, Erik Bruchez and Geert Bormans for more information

XML Processing

OK, but what really bugs me is that I have to use the DOM. What can you do for me?



- Using DOM or SAX manually to serialize and deserialize objects is just a waste of time.
 - Solutions are available for all the programming languages to do that for you.
 - You have no excuse for continuing to use DOM or SAX for this purpose!

Different solutions for different languages

Solutions available for all the programming languages?
Does that mean that they're all equal?



- The type of solution depends on the language you are using:
 - Native support for a few ones
 - Compile time binding only for some
 - Compile or run time binding for others

Native XML support

Native XML support? looks like a dream to me!



- A few programming language support XML fragments as native objects:
 - JavaScript (E4X)
 - C# (X.Linq project, under development, see Štěpán Bechynský's presentation)
- This provides native read/write access to XML

Compile time binding

What if I am using a strongly typed language such as Java, C# or C++ and I can't or don't want to use a native XML support?



- Compile time data binding libraries associate XML fragments to classes and properties at compile time.
 - Many (JAX-B, .Net, ...) can generate classes or interfaces from XML schemas.
 - Others (TreeBind, ...) use reflexion.

Run time binding

And if I am a dynamically typed language geek, don't you have something more lightweight for me?



- Run time data binding libraries associate XML fragments to classes and properties at run time.
 - They exist for most scripting languages.
 - They are usually able to create classes on the fly when they don't exist.
 - See Uche Ogbuji's presentation for a very good example of such a library.

Some use cases

Looks fine but that remains abstract to me. Can you give me a couple of use cases?



- Any application using XML as a data format such as:
 - Exchanging objects between a Web server in PHP and an Ajax Web client (XML becomes as simple to use a JSON) – See Professional Web 2.0 Programming.
 - Using Orbeon Forms to manipulate Java objects with XForms (demonstrated at Xtech 2006).

Thanks

XML processing doesn't have to be dull and hard then?



- No, process XML with fun and enjoy the rest of the conference!
- Questions/suggestions are welcome face to face or at vdv@dyomedea.com.