Missing Pieces of the XML Jigsaw

Erik Wilde (net.dret@dret.net)
Computer Engineering and Networks Laboratory
ETH Zürich

September 23, 2002

Abstract

XML technologies provide an increasingly solid foundation for applications in very different problem areas. However, for some areas, there are still some gaps to fill in. In this talk, one such area will be presented, the area of hyper-links in XML. Even though the XLink standard defines how to encode hyperlinks in XML documents, it leaves potential users with a lot of questions to solve, such as how to access this information programatically, for example using the DOM API or XSLT. A possible solution is to extend the XML Information Set, the data model of XML, with hyper-link-specific information. A method how this could be done in a reusable way is presented and put into the context of existing XML specifications.

Outline

- Motivation
- Current State of XML Technologies
- Missing Pieces and Open Problems
- XML Information Set
- Case Study: Hyperlinks in XML
- Proposal for an Open Data Model
- Future Work and Conclusions
- Q&A

Motivation

- XML and hyperlinking
  - XLink/XPointer development by W3C
  - additional components such as API and protocol
- XLink adoption is slow
  - lack of support by software such as browsers
  - lack of support for software authors
    - how to program with XLink? how to style XLinks?
  - XLink integration in other languages is a problem
    - XHTML currently uses HLink rather than XLink
- XML hyperlinks need a better foundation
  - a well-defined data model endorsed by W3C
  - syntaxes (XLink/HLink) and interfaces (API/UI) as necessary

Current State of XML

- XML 1.0 Second Edition (W3C Spec)
  - defines XML based on EBNF syntax
- XML is the foundation for many other specs
  - some of them are *host languages*
    - containers for data (e.g., XHTML or SOAP)
  - some of them are *integration languages*
    - reusable components (e.g., MathML or XLink)
- interworking requires additional specs
  - XML Namespaces for recognizing names
  - schema languages for usage constraints
    - but not always: XLink does not have a schema
  - specific ways for dealing with extensions

Missing Pieces & Open Problems

- how to plug in additional semantics
  - in an extensible way
  - without the need to change core standards
- how to keep XML simple and flexible
  - SGML suffered from its complexity
  - OSI was too complicated for simple applications
- XML has been developed evolutionary
  - XML 1.0 for the syntax of documents and DTDs
    - XML Namespaces for globally unique naming
    - XML Infoset for the data model of XML
    - XML Base for the interpretation of relative URIs
  - what comes next?

XML Information Set

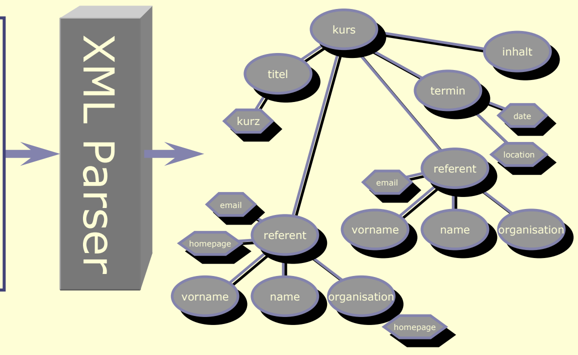
- several XML applications need a data model
  - APIs (DOM, SAX)
  - XML programming languages (XSLT)
  - XML query languages (XQuery, XQL, …)
  - XML formatting (CSS, XSL-FO)
  - XML fragment identifiers (XPointer)
- XML 1.0 does not define a data model
  - implicitly defined, but open to interpretation
    - does attribute order matter?
    - does whitespace in tags matter?
- XML Information Set (XML Infoset)
  - defines a set of *information items* with *properties*
  - an XML document is a set of such items

XML Infoset Essentials

- only Namespace-compliant XML allowed!
- what is in the Infoset?
  - elements (child sequences) and attributes (sets)
  - Namespace declarations and prefixes
  - comments and processing instructions
  - character data
  - some type information (attribute types)
- and what is not in the Infoset?
  - whitespace within tags
  - the order of attributes within element tags
  - most information from the DTD

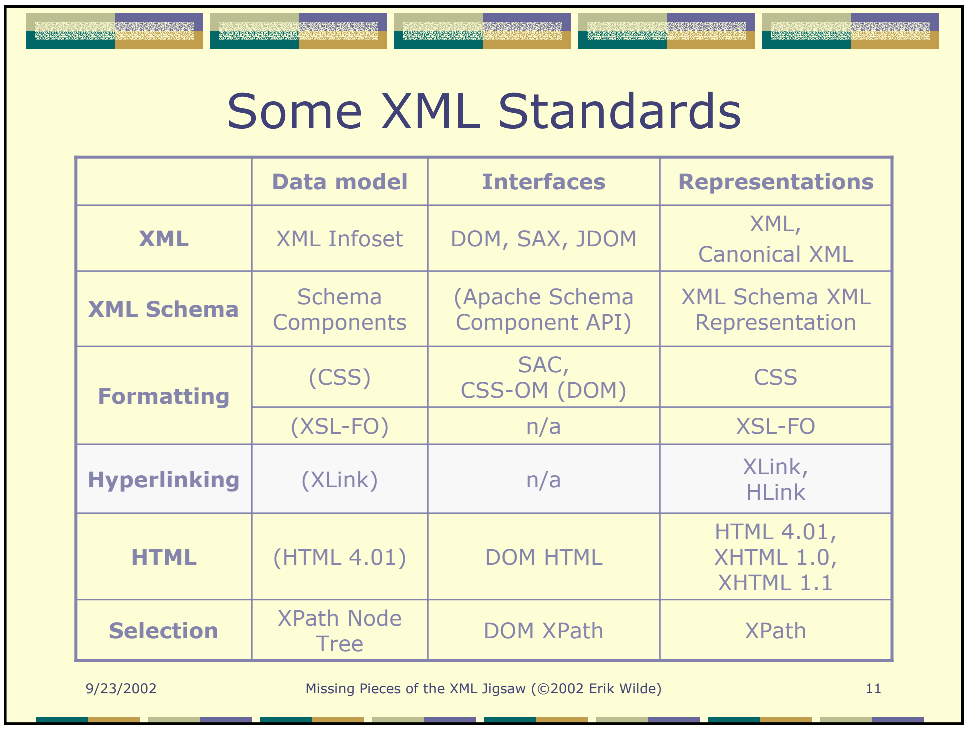
Where does it come from?

```
<?xml version="1.0" ?>
<!DOCTYPE kurs SYSTEM "kurs.dtd">

<kurs>
<titel kurz="XML">XML -
Grundlagen und Umfeld</titel>

<referent email="xml@dret.net"

homepage="http://dret.net/">
  <vorname>Erik</vorname>
  <name>Wilde</name>
  <organisation
homepage="...">ETH
  Zürich</organisation>
</referent>

<referent> ... </referent>
<inhalt> ... </inhalt>

</kurs>
```



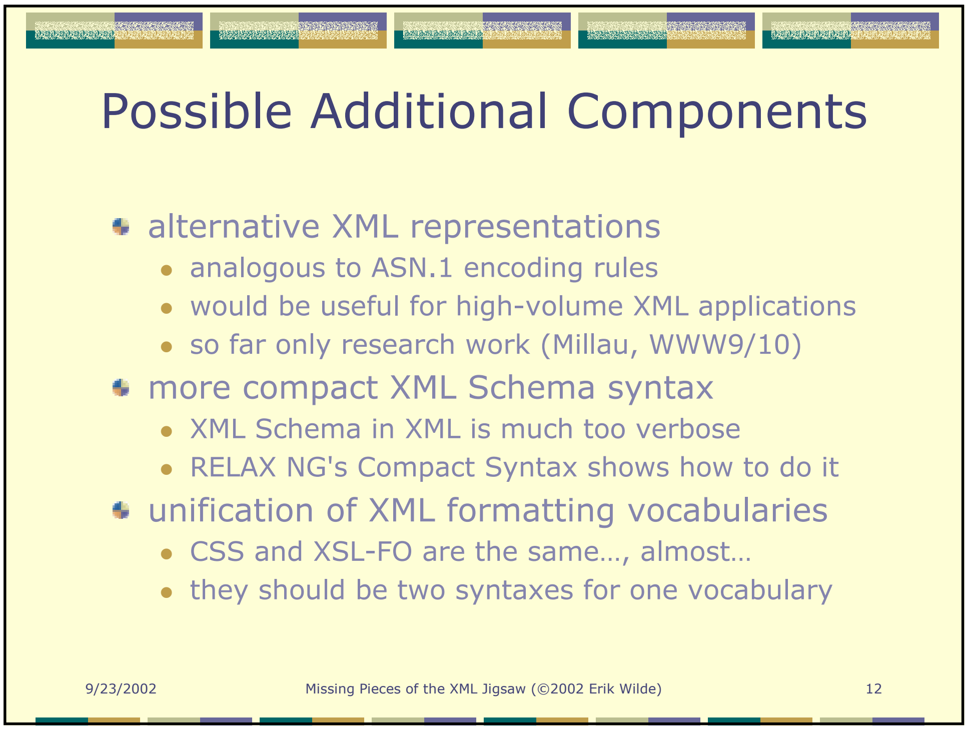
Infoset vs. Document

- the Infoset is a rather abstract construct
  - it does not have a particular representation
  - it does not have a particular interface
  - it often is produced by a parser
    - but it may also be synthesized
- the Infoset is more than a parse tree
  - it omits some information
    - syntax details deemed irrelevant
  - it adds some information (schema-derived)
    - type information about attributes
    - default attributes from a schema
    - references for `ID`/`IDREF(S)` attributes
- Infoset = Abstraction + Semantics



Some XML Standards

	Data model	Interfaces	Representations
XML	XML Infoset	DOM, SAX, JDOM	XML, Canonical XML
XML Schema	Schema Components	(Apache Schema Component API)	XML Schema XML Representation
Formatting	(CSS)	SAC, CSS-OM (DOM)	CSS
	(XSL-FO)	n/a	XSL-FO
Hyperlinking	(XLink)	n/a	XLink, HLink
HTML	(HTML 4.01)	DOM HTML	HTML 4.01, XHTML 1.0, XHTML 1.1
Selection	XPath Node Tree	DOM XPath	XPath



Possible Additional Components

- alternative XML representations
  - analogous to ASN.1 encoding rules
  - would be useful for high-volume XML applications
  - so far only research work (Millau, WWW9/10)
- more compact XML Schema syntax
  - XML Schema in XML is much too verbose
  - RELAX NG's Compact Syntax shows how to do it
- unification of XML formatting vocabularies
  - CSS and XSL-FO are the same…, almost…
  - they should be two syntaxes for one vocabulary

Case Study: Hyperlinks in XML

- XLink defines hyperlinks for XML
  - defines a number of global attributes
  - link recognition is based on the XLink Namespace
    - `<mylink xlink:type="simple" xlink:href="…" …`
- XLink also implicitly defines a linking model
  - link topologies (n:m, directionality)
  - link locations (inline, out-of-line, third-party)
  - link semantics (link behavior)
- XLink syntax does not make everybody happy
  - significantly hurts the adoption of XLink
  - e.g., XHTML cannot use XLink syntax
    - introduced their own vocabulary (HLink)
  - XLink syntax should be separate from the model

An Open XML Data Model

- XLinks extends XML's data model
  - additional information about resource fragments
- link information may come from any source
  - from XLinks within the document
  - from XLinks retrieved from an external linkbase
  - from non-XLink information from any data source
- currently links are not reflected in the Infoset
  - programming with links is cumbersome
    1. find all the relevant information (via the Namespace)
    2. check for integrity as defined by XLink's constraints
    3. process the link information
  - could be made easier if integrated into the pipeline
    - currently no spec for a processing pipeline

Infoset Extension: An Example

- XML Schema extends the Infoset
  - XML Schema validation is defined as Infoset augmentation (adding items and properties)
  - XML Schema defines a number of *Post Schema Validation Infoset (PSVI)* contributions
- PSVI is hardwired into other W3C specs
  - the XPath/XQuery data model uses the PSVI
  - currently there is no API for PSVI
    - no active work on PSVI DOM module
    - validation is easy, using the results is not!
- XLink is similar to XML Schema
  - it extends the Infoset of an XML document

Generic Infoset Extensions

- defining Infoset extensions can be easy
  - define a Namespace URI for the extension
  - define possible dependencies with other extensions
  - define the Infoset extensions
    - additional *properties* of existing *items*
    - additional *items* defines by the extension
- extensibility must be supported everywhere
  - define a DOM module for generic Infoset extensions
    - if better support is required, define a specific module
  - define XPath mechanisms for using extensions
    - for example, define "extension axes" for XPaths
  - define CSS selectors for extensions
    - CSS currently is based on XML rather than the Infoset

XLink NG

- XLink NG only defines the data model
- the XLink data model is an Infoset extension
  - based on the XML Infoset core
    - defining a *link information item*
    - as well as *link properties* pointing to *link items*
  - XLink 1.0 is one possible syntax for it
  - XHTML's HLink is another possible syntax
- future work is based on the data model
  - a protocol for accessing linkbases
  - APIs for programming with XML hypermedia

The 80/20 Split

- many people don't like the idea of extensibility
  - things get more complicated
  - software must be more robust
  - and: many people don't like Namespaces
- is it worth the effort?
  - depends on the point of view…
    - if you are happy with plain XML, it is not
    - if you see XML as a foundation for more advanced data models supporting additional semantics, it is
- political questions are important
  - W3C has very many people from a lot of companies with very different goals and backgrounds

Future Work & Conclusions

- the basic idea seems to be feasible
- closer look at some questions of support
  - is a generic API (DOM/SAX/JDOM) doable/reasonable
  - is there a clean way of integration into XPath?
- XML needs more work
  - XML 1.0 is fine, but rather simple
    - a way to exchange labeled trees
  - additional semantics should be pluggable into XML
    - in terms of the data model, not only the syntax
- XML standardization is complex
  - hard to talk to the right people
  - hard to convince all the necessary people

Thank You! — Q&A