XML:
Knitting the Web Together

Henry S. Thompson
HCRC Language Technology Group
Division of Informatics
University of Edinburgh

The old Web

- Mark 1
  - Hand-authored HTML
  - Marking up human-generated prose
  - For humans to read
- (Mark 1a)
  - WYSIWYG-authored HTML
- Mark 2
  - Mechanically generated HTML
  - Marking up non-prose data
  - For machines to read

The new Web

- Part 1
  - Machine-created XML
  - Marking up application data
  - For other applications to process
- Part 2
  - Human-authored XML
  - Adding value to existing XML
  - For humans and machines to process
- Part 3
  - . . . distributed in space and time

XML is ASCII for the 21st century

- ASCII (ISO 646) solved a fundamental interchange problem for flat text documents
  - What bits encode what characters
    - (For a pretty parochial definition of 'character')
- UNICODE/ISO 10646 extends that solution to the whole world
- XML thought it was doing the same for simple tree-structured documents
  - The emphasis in the XML design was on simplifying SGML to move it to the

Web
XML didn't touch SGML's architectural vision
flexible linearisation/transfer syntax
for tree-structured documents with internal links

The essence of XML

It's a markup language used for annotating text
It *is* concerned with logical structure
to identify sections, titles, section headers, chapters, paragraphs,…
It is *not* concerned with appearance
you say 'this is a subtitle'
not 'this is in bold, 14pt, centered'
you say 'this is an example'
not 'this is in verbatim, indented by 5pts, ragged right'

Why is XML a big deal?

It is an official W3C Recommendation
It is vendor-independent, platform independent, application independent, …
unlike Word documents, RTF documents, PDF documents, Postscript documents, …
It is human readable
ditto (for most values of 'human')

Unformatted text

Internet-based Application Architectures for the 21st Century:
The Role of XML
Let's skip straight to an example of XML syntax for a simple bit of structure:
<tip><emph>Never</emph> stand up in a canoe!</tip>

XML marked up text

<article>
<title> Internet-based Application Architectures for the 21st Century:
</title>
<subtitle>The Role of XML</subtitle>
<section>
<para> Let's skip <emph>straight</emph> to an example of XML syntax for a simple bit of structure:</para>
<example> <tip><emph>Never</emph> stand up in a canoe!

```
&lt;/tip></example>
 </para>
 </section>
</article>
```

Who is in charge of XML?

XML is a W3C Recommendation

The W3C is *The World Wide Web Consortium*, a voluntary association of companies and non-profit organisations. Membership costs serious money, confers voting rights. Complex procedures, with the Chairman (Tim Berners-Lee) having ultimate authority, guided by a committee of the whole called the Advisory Council.

The XML recommendation was written by the W3C's XML Working Group.

The essence of XML, try again

It's a markup language used for transferring data

It *is* concerned with data models

- to convert between application-appropriate and transfer-appropriate forms

It is *not* concerned with human beings

- It's produced and consumed by programs

What just happened!?

The whole transfer syntax story just went meta, that's what happened!

XML has been a runaway success, on a *much* greater scale than its designers anticipated

- Not for the reason they had hoped
  - Because separation of form from content is *right*
- But for a reason they barely thought about
  - Data must travel the web

Tree structured documents are a useable transfer syntax for just about anything

- So data-oriented web users think of XML as a transfer mechanism for their data

---

Components of the XML family

XSLT

- Transforming XML

XLink/XPointer

- Connecting XML documents

- XML Schema
  - Defining XML document families
- XML Protocols
  - XML-based communication

XSLT: Structure into form

- There is a stylesheet language called XSLT
  - Rules for transforming from one vocabulary to another
    - Common case: output vocabulary is HTML
    - Coming soon: HQ print-orientated vocabulary
- For example

```
<template match='emph'>
<I><apply-templates/></I>
</template>
```

  - will do part of the Transformation job

XSLT Status:

- W3C approved REC since November 1999
- Three or four fully conformant implementations
  - All free
  - Including IE5
    - As of last week J
- Most are offline
  - Written in Java
- IE5 is online
  - Written in C++
  - file://Events/events/XML%20World/xmleur01/Speaker/work/xmlschema/structures/structures.xml

---

What is XLink

- Together with XPointer, a reconstruction and enrichment of the hyperlink concept at the heart of the web
- Browsing is not the only application
  - "Follow Me" is not the only link semantics
- Take HTML's <A HREF="…"/>, and do it right
  - Not tied to a particular element type
  - Not restricted to two endpoints
  - Not restricted to be inline
- A careful separation between
  - The ontology and its notation (XLink)
  - The syntax of resource identification (Xpointe/XPathr)

XLink/XPointer status

- In Candidate Recommendation phase
- Several near-complete implementations recently announced
- Retrospective integration with e.g. XHTML and SVG underway

XML Schema: some details

- XML Schema is a language for defining the structure of XML documents
  - Notated in XML itself
- So there are elements defined for use in schemas to define. . .
  - Elements :-)
  - Attributes
  - Types
- A type is a collection of constraints on element content and attribute values
- A type may be either
  - Simple, for constraining string values
  - Complex, for constraining elements which contain other elements

A simple example

```
<!ELEMENT text (#PCDATA|emph|name)*>
<!ATTLIST text
      timestamp NMTOKEN #REQUIRED>
<xs:element name="text">
 <xs:complexType content="mixed">
 <xs:choice minOccurs='0'
       maxOccurs='unbounded'
  <xs:element ref="emph"/>
  <xs:element ref="name"/>
 </xs:choice>
 <xs:attribute name="timestamp"
       type="date" minOccurs="1"/>
</xs:complexType></xs:element>
```

XML Schema Status

- Last Call finished in May
- Entering Candidate Recommendation very soon
  - Small number of weeks
- At least five (partial) implementations

- Three free
- Big players strongly committed
  - IBM/Lotus, Oracle, Microsoft
- W3C eating its own cooking
  - Subsequent RECs based on XML Schema

XML Protocols

- Replace application-specific wire protocols with XML
- Define an XML messaging story just above the transport layer
- Use the modularity of XML Schema to allow application-specific specialisation of payload
- Lack of consensus about exactly what the right level is

XML Protocol Status

- W3C Working Group formation just announced
- First meeting next month
- Starting points
  - XML RPC
  - SOAP
    - Microsoft just announced a major development effort

Linking vs. Messaging

- People tend to think about distributed applications at too low a level
  - RPC
  - Messages
- E-business and E-commerce are struggling to use XML versions of these technologies
  - With less success than originally expected
- I think distributed, dynamic documents are a better fit

Conclusions

- XML has a lot to offer e-Business and e-Commerce
  - Separating hype from reality is not easy
  - Careful requirements analysis is still the only sensible starting point
- Old paradigms are not always the right model
  - Creative exploration/exploitation of new architectures is needed
- Pilot first, before you bet the company
  - Look for help from established practitioners
- Start now, if you haven't already!

XML and e-Business

- Ed Feigenbaum once described Terry Winograd's work as "a breakthrough in enthusiasm"
  - I worry sometimes if XML and e-business is vulnerable to the same criticism
- Negotiation between producers and consumers is the key
  - If you can't describe what you want, you can't have it
  - If you can't describe what you've got, no-one will use it
  - If you can't dicker, you'll always lose
- So as far as I can see, for e-Business to be successful the Web badly needs a solution to the metadata problem

---

What is the metadata problem

- There's been a lot of talk about metadata.
- What is metadata?
  - It's just data.
  - But it's data *about* other data
  - Data intended for machine consumption
- What could metadata do for us?
  - Give search engines something to work with that is designed for their needs.
  - Give us all a place to record what a document, or any other resource or service, is *for* or *about*.

Requirements for metadata

- What would we need to make this work?
  - A standard syntax, so metadata can be recognised as such;
  - One or more standard vocabularies, so search engines, producers and consumers all speak the same language;
  - Lots of resources with metadata attached;
  - Attribution and trust
    - Is this resource *really* about Pamela Anderson?

Some choices for the GRID

- Design our own languages/data structures for describing problem and resource components
- Just define the ontologies, and use an existing data modelling meta-language
  - Entity-Relation
  - UML
  - RDF
  - Topic Maps

XML Schema and RDF are the W3C-designed vehicles of choice

What is RDF?

RDF is actually two standardisation efforts, under the aegis of the W3C. It stands for Resource Description Framework (in other words, metadata).

The two efforts are:

Standardising the syntax and abstract semantics;

RDF Model and Syntax

Providing a standard way of *defining* standard vocabularies (but *not* actually defining any).

RDF Schema

Distributed Dynamic Documents

Ted Nelson identified a powerful link semantics over twenty years ago

He called it *transclusion*

We're only just able to implement it

A document with transclusions in it is synthesised from the parts it points to

The separation of form from content is crucial here

First you pull it together

Then you render it

In the dynamic case, if what you're point at changes

You re-knit, and re-style

I've used document language, but the layered story works here too