XML Metadata Standards and Topic Maps

Erik Wilde

Outline

- what is XML?
  - a syntax (not a data model!)
- what is the data model behind XML?
  - XML Information Set (basically, trees...)
- what can be described with XML?
  - describing the content syntactically (schemas)
  - describing the content abstractly (metadata)
- XML metadata is outside of XML documents
- ISO Topic Maps
  - a "schema language" for meta data

Extensible Markup Language

- standardized by the W3C in February 1998
- a subset (aka *profile*) of SGML (ISO 8879)
- coming from a document world
  - data are documents
- defined in syntax
  - no abstract data model
- problems in many real-world scenarios
  - how to compare XML documents
    - attribute order, white space, namespace prefixes, ...
  - how to search for data within documents
    - query languages operate on abstract data models
  - often data are not documents

Why XML at all?

- because it's simple
  - easily understandable, human-readable
- because of the available tools
  - it's easy to find (free) XML software
- because of improved interoperability
  - all others do it!
  - easy to interface with other XML applications
- because it's versatile
  - the data model behind XML is very versatile

XML Information Set

- several XML applications need a data model
  - style sheets for XML (CSS, XSL)
  - interfaces to programming languages (DOM)
  - XML transformation languages (XSLT)
  - XML fragment identifiers (XPointer)
  - XML query languages (XQuery)
- XML does not have a real data model
  - implicitly defined, but not authoritatively
- XML Information Set (XML Infoset)
  - describes a set of *information items*
  - each XML document is a set of such items

XML Infoset Essentials

- only Namespace-compliant XML allowed!
- so what's in the Infoset?
  - elements
  - attributes
  - Namespace declarations and prefixes
  - comments
  - processing instructions
- and what's not in the Infoset?
  - whitespace within element tags
  - the order of attributes within element tags
  - any information about the DTD

XML Schema Languages

- XML represents structured Information
  - XML Infoset defines the data model (trees)
  - XML 1.0 defines a character-based syntax
- XML 1.0 also defines DTDs
  - element types and their content models
  - attributes and their data types
- every XML application has to support DTDs
  - the only globally accepted schema language
  - almost 20 years old
  - many drawbacks for non-document scenarios

XML Schema

- developed because of user demand
  - B2B scenarios need better data types
  - data modeling needs better structuring
- XML Schema W3C standard since 5/2001
  - implementations available
  - rapid adoption is very likely
- Part I defines structuring mechanisms
  - element types may be derived from each other
- Part II defines a data type vocabulary
  - a set of application-oriented *simple types*

Schemas and Metadata

- XML resources may contain any type of data
  - documents (as originally intended by SGML)
  - order forms (as is common in B2B scenarios)
  - generic things such as RPC requests and responses
    - SOAP and XML RPC are two popular variants
  - or even information about other resources
- XML metadata describes data resources
  - not necessarily XML data (eg, image descriptions)
  - not necessarily attached to the resources
    - making comments on other people's resources
  - metadata is also data (ie, structured information)
  - XML metadata needs schema definitions

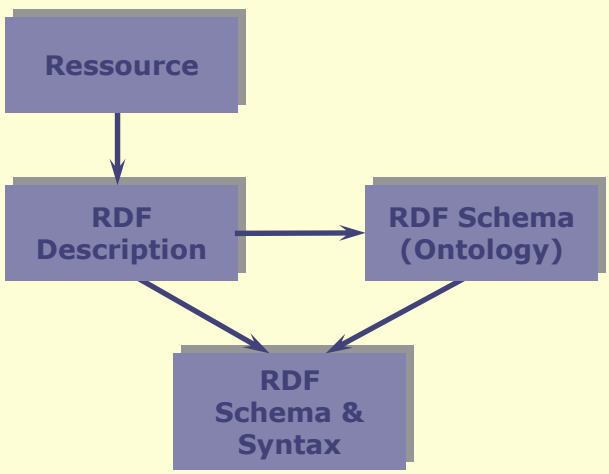
XMLizing the World...

- should everything be XML?
  - structured data would be an appropriate target
  - but what about GIF, JPEG, MPEG, ... ?
- everything should be described using XML
  - descriptions of resources are metadata
  - metadata is structured data
  - metadata should be in XML
- so there must be an XML metadata standard
  - TimBL's favorite: *Resource Description Framework*
  - coming from ISO standardization: *Topic Maps*

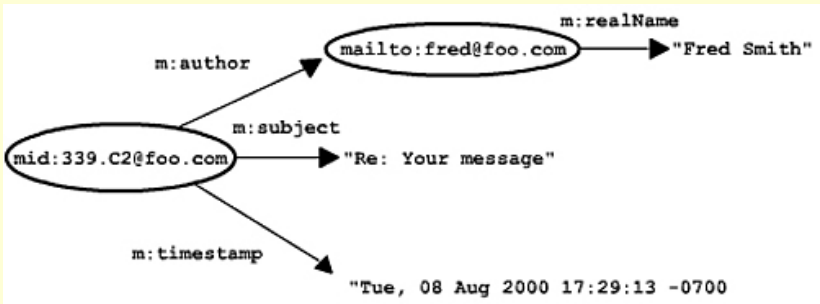
Resource Description Framework

- RDF starts with a data model
  - and defines an XML syntax for representation
- everything in RDF can be represented by a graph with nodes and arcs
  - each node is a resource
  - each arc represents a property
  - properties and resources are named with URIs
- describes the whole Web and beyond
  - anything which can be named with a URI
  - which is almost anything (phone, tv-channels, ...)
- RDF graphs describe logical assertions

RDF Metadata



RDF-based Email Description



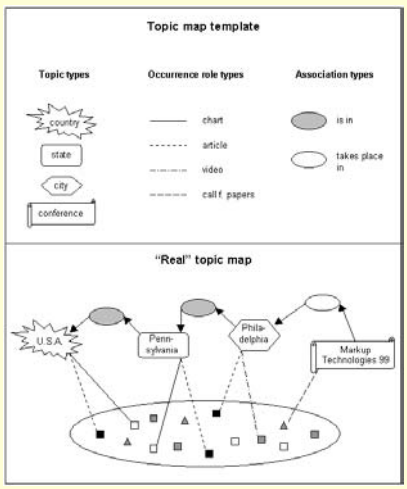
But ... what is it good for?

- ask questions about the email
  - who sent me mail on a particular topic?
  - get me all the mail from Fred Smith
  - who where the people who I mailed with on Friday?
- join the email graphs with other ones
  - address books
  - home pages
  - browser history
  - organizational affiliations

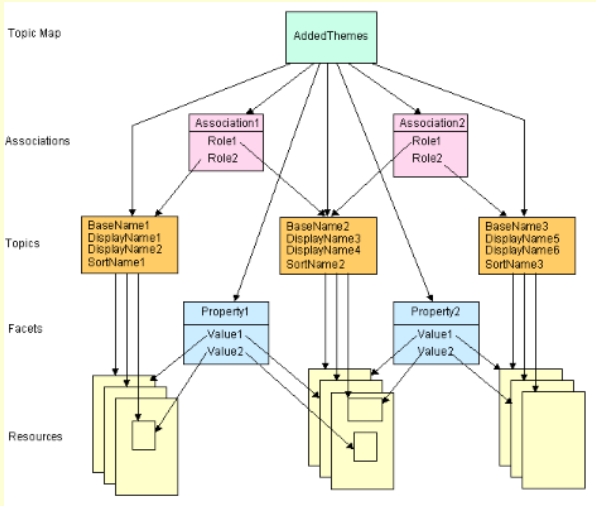
Topic Maps

- Topics are "things of interest"
  - loosely defined, widely usable
  - each Topic has name(s) and/or occurrence(s)
  - Topics have "topic types" (which are Topics...)
- Associations are used to connect Topics
  - the have an "association type" (which is a Topic...)
  - Topics references in Associations have an "association role type" (which are Topics...)
- Topic Occurrences point to resources
  - anything addressable by a name (URI)
  - described by an "occurrence role type" (a Topic...)

A Simple Topic Map



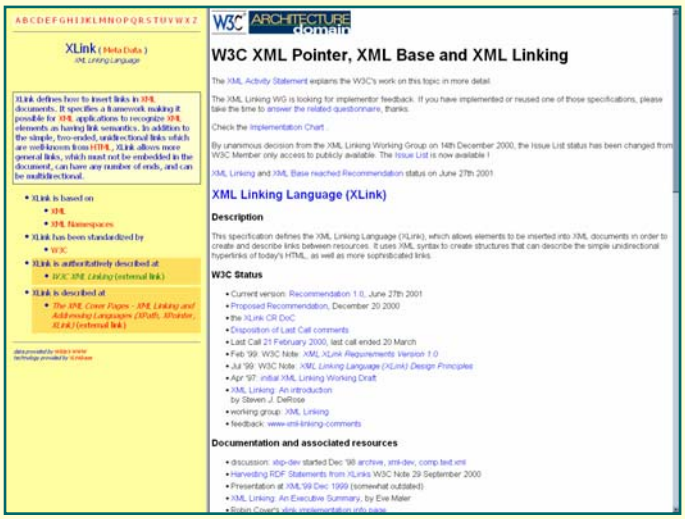
The Whole Picture



Topic Map View (I)

Topic Map View (II)

Topic Map View (III)



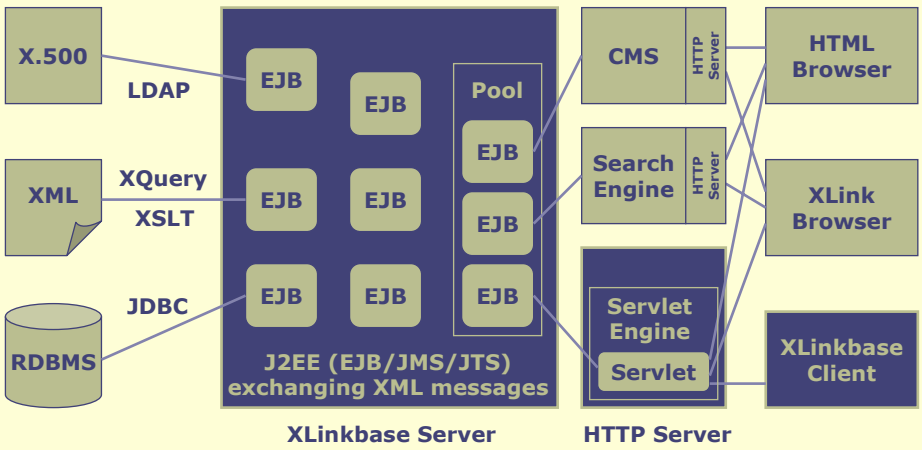
Comparison

RDF	Topic Maps
standardized by W3C	standardized by ISO
explicit data model	defined by syntax
properties have data types	associations aren't constrained
inherently distributed	centralized
separates schema from instance (resource description)	eveything (almost...) is a topic, there are no types

What is missing?

- for Topic Maps only
  - a clean way to separate schemas and instances
  - a constraint language for topic associations
  - a way to distribute Topic Maps
- for RDF only
  - a unified data model with XML Schema
- for both approaches
  - tools for creating and managing metadata
  - a query language for actually using metadata
  - support from a wide range of vendors & users
  - an approach for achieving vocabulary consensus
  - smart ways to handle distribution

XLinkbase System Architecture



XLinkbase Status

- where the implementation is going
  - currently concentrating on EJB environment
  - hard to keep up with commercial engines
  - case study with simpler model & implementation
  - case study for generating DHTML links
- where the concept is going
  - proof of concept with the case study
  - 1 or 2 DAs dealing with Topic Map distribution
  - looking into data model improvements
    - constraint language for associations
    - schema/instance separation or separability