Abstract

Metadata usage often depends on schemas for metadata, which are important to convey the meaning of the metadata. We propose an architecture where users can extend the schema used by a system for managing referential metadata. Users can plug new schemas and install custom filters for exporting metadata, so that users are not forced to limit their metadata to a fixed schema. The goal of this architecture is to provide users with a system that helps them managing their referatory, enables them with powerful tools to adapt the tool to their metadata, and still makes it possible to collect the metadata of several users in a central storage and exploit the common facets of the metadata. Our system is based on a specialized schema language, which has been built on top of the XML schema languages XML Schema and Schematron.
Outline

- Referatories and our Goals
- BibTeXML Metadata Management Tool
- BibSchema Language
- Implementation
- Q&A

What is Metadata?

- data about existing resources
  - describing the resources
  - using pre-defined schemas for description
    - metadata without a schema is not very useful
  1. metadata using standardized schemas
    - semantics are (often...) strictly defined
    - limited vocabulary, trade-off usability/complexity
  2. metadata using proprietary schemas
    - defined by users or user groups
    - optimized for the requirements of a closed group
      - easy to use for the schema's creators
      - hard to use and understand for other people
    - possible mappings to metadata standards
Where is Metadata?

- inside people's heads
  - some of it would be rather easy to formalize
  - other things are very vague and hard to formalize
- in people's notes
  - people forget, so they try to save their metadata
  - note "schemas" are informal and dynamic
  - notes are hard to formalize and classify
- on people's computers
  - many users keep text files for notes
  - others use existing applications
  - bibliographies are collections of metadata

Our Goals

- people know a lot about resources
  - much of this information is inaccessible
  - some is machine-readable, but off-line
- help people manage their metadata
  - giving them powerful tools
- help people share their metadata
  - show them that sharing helps them and others
- don't be dogmatic about metadata schemas
  - people use different schemas
  - don't hardcore metadata schemas
- reduce coding as much as possible
  - less coding makes software more flexible
Referatories

- a collection of references to resources
  - ISBN numbers
  - ISSN numbers, volume(number) and pages
  - URIs (Web browser bookmarks)
    - there are dozens of existing URI schemes
    - basically, anything that can be referred to
- many referatories contain metadata
  - authors of books
  - mirrors of Web pages
  - language information about resources
  - abstracts of papers
  - keywords describing resources

BibTeX as Metadata

- many people keep BibTeX files
  - for their bibliographic references
  - for references to other resources
- BibTeX is extensible (+)
  - unknown fields are ignored by BibTeX
  - fields may have any content (HTML, XML, ...)
- BibTeX is strange and fuzzy (-)
  - BibTeX schema is not nicely designed
  - some handling is buried in BibTeX style files
  - people use BibTeX differently
    - author and editor fields
    - inbook and incollection entry types
How to collect Metadata?

- convince rather than force people
  - give them tools that are attractive
  - design the tools so that people like them
  - ... and that the tools create re-usable metadata
- many people don't like centralized systems
  - many years of work go into the bibliography
  - they want to "keep it"
- design centralized schemas
  - people use their local data (but centralized schemas)
  - people can register and use schema extensions
    - designed by them (unknown to anybody else)
    - designed by a group (used in a closed community)

BibTeX Example

@misc{xmlns10,
  author = "Tim Bray and Dave Hollander and Andrew Layman",
  title = "Namespaces in XML",
  howpublished = "W3C, REC-xml-names-19990114",
  month = "January",
  year = 1999,
  url = "http://www.w3.org/TR/1999/REC-xml-names-19990114",
  topic = "xml[0.8] xmlns[1]"}
Why not use BibTeX?

- BibTeX is weird
  - many things are "defined by the program"
  - there is no authoritative BibTeX grammar
  - name handling uses clever but complex rules
  - character handling is very TeX-specific
  - structured fields are hard to implement
    - requiring some non-BibTeX syntax
  - defining new schemas is very hard
    - must be programmed in a BibTeX style file
- not a lot of BibTeX software is available
  - BibTeX (the program) and bibclean
- XML is the standard for structured data
  - many tools available (schema languages, XSLT, ...)

BibTeXXML

- XML-based language for referatories
  - many tools for working with XML
- based on a specific schema language
  - completely configurable
  - mapping the schema language to others
    - reduce the amount of programming
    - much easier to port to other platforms
BibTeXML Entry Example

```xml
<s:misc key="xmlns10">
  <s:author>
    <b:bibperson>
      <b:firstname>Tim</b:firstname><b:lastname>Bray</b:lastname>
    </b:bibperson>
  </s:author>
...
  <s:title>Namespaces in XML</s:title>
  <s:howpublished>W3C, REC-xml-names-19990114</s:howpublished>
  <s:month>January</s:month>
  <s:year>1999</s:year>
  <u:uri>http://www.w3.org/TR/1999/REC-xml-names-19990114</u:uri>
  <t:topics>
    <t:topic name="xml" weight="0.8"/>
    <t:topic name="xmlns" weight="1"/>
  </t:topics>
</s:misc>
```

BibSchema

- do not hardcode any metadata schema
- schema language for metadata
  - entry types for metadata "records"
  - entry fields for record "contents"
  - field formats for content "structure"
- entry types defined by field occurrences
  - small grammar-like vocabulary
  - referring to field definitions
  - additional fields are always legal
- entry fields defined by allowed content
  - some special cases (for example, persons)
  - in general, a XML Schema type definition
BibSchema Example

```xml
<bs:schema xmlns="http://www.w3.org/2001/XMLSchema"
    xmlns:bs="http://dret.net/xmlns/bibtexml/bibschema"
    defaultRefAndTargetNS="http://dret.net/xmlns/bibtexml/standard">
    <bs:entry name="misc">
        <bs:minOne>
            <bs:field ref="author"/>
            <bs:field ref="title"/>
            <bs:field ref="howpublished"/>
            <bs:field ref="month"/>
            <bs:field ref="year"/>
            <bs:field ref="note"/>
        </bs:minOne>
    </bs:entry>
    <bs:field name="author" isPerson="true" repeatable="true"/>
    <bs:field name="title" isSpecialText="true"/>
    <bs:field name="month">
        <restriction base="string">
            <enumeration value="January"/>
        </restriction>
    </bs:field>
</bs:schema>
```

BibSchema Basic Design
BibSchema Usage

- server is based on one schema
  - the main schema of the referatory
- additional schemas may be installed
  - supporting additional metadata in entries
- metadata may be exported
  - in native XML format
  - in any other format generated by XSLT

BibSchema Extension Example

```xml
<bs:schema xmlns="http://www.w3.org/2001/XMLSchema"
           xmlns:bs="http://dret.net/xmlns/bibtexml/bibschema"
           defaultRefAndTargetNS="http://dret.net/xmlns/bibtexml/topic">
  <bs:field name="topics">
    <sequence>
      <element name="topic" maxOccurs="unbounded">
        <complexType>
          <attribute name="name" type="NCName"/>
          <attribute name="weight">
            <simpleType>
              <restriction base="decimal">
                <minInclusive value="0"/>
                <maxInclusive value="1"/>
                <fractionDigits value="1"/>
              </restriction>
            </simpleType>
          </attribute>
        </complexType>
      </element>
    </sequence>
  </bs:field>
</bs:schema>
```
**System Implementation**

- based on a standard LAMP implementation
  - Linux/Apache/MySQL/PHP
  - very simple Web-based interfaces
- user interface
  - data import
  - data export using transformations
  - searching in the database
- management interface
  - all normal user functions
  - managing supported schemas
  - managing export formats
  - managing users and user groups

---

**BibTeXML Entry Example**

```xml
<b:bibliography xmlns:b="http://dret.net/xmlns/bibtexml/base"
    xmlns:s="http://dret.net/xmlns/bibtexml/standard"
    xmlns:t="http://dret.net/xmlns/bibtexml/topic"
    xmlns:u="http://dret.net/xmlns/bibtexml/uri">
  <b:entries>
    <s:misc key="xmlns10">
      <s:author>
        <b:bibperson>
          <b:firstname>Tim</b:firstname><b:lastname>Bray</b:lastname>
        </b:bibperson>
      </s:author>
      ... 
      <s:title>Namespaces in XML</s:title>
      <s:howpublished>W3C, REC-xml-names-19990114</s:howpublished>
      <s:month>January</s:month>
      <s:year>1999</s:year>
      <u:uri>http://www.w3.org/TR/1999/REC-xml-names-19990114</u:uri>
      <t:topics>
        <t:topic name="xml" weight="0.8"/>
        <t:topic name="xmlns" weight="1"/>
      </t:topics>
    </s:misc>
  </b:entries>
</b:bibliography>
```
Search Interface

Advanced Search

Search Result Interface

Advanced Search
Management Interface

WELCOME TO THE
BIBLIOGRAPHY MANAGEMENT SYSTEM

Search
Selection
Data Import
Schema Management
Export Format Management
Macro Management
User Management
MySQL Admin

Metadata Schema Management

SCHEMA MANAGEMENT

Schema Overview

Namespace | Details
-----------|---------
http://bibliowrl.org/common-extension | BibSchema XML Schema Schematron Delete
http://bibliowrl.org/standard | BibSchema XML Schema Schematron Delete

Schema Import
Select the Bib Schema you want to import or just validate:
- Import schema / Only validate schema
- Path: [ ]

September 2003 (© Erik Wilde)
Metadata Schema Import

1. the BibSchema is uploaded
   - only allowed for administrators
2. converted using XSLT into two schemas
   - an XML Schema describing the fields
   - a Schematron schema describing constraints
3. all three schemas are installed
   - can be reviewed by administrators
4. uploading entries uses the schemas
   - conceptually: BibSchema validation
   - implementation: XML Schema & Schematron
     - currently very slow (XSLT is very slow)

Export Format Management

- Export Format Management Table
  - Mode: BibTeX, Example
  - Info: base, author, title, year, month, number, topics, default
  - XSLT Precedence: base, author, title, year, month, number, topics, default
  - Translation: up, down, remove, details

* Higher position means higher import precedence.

- Mode Name: 
  - Info: up, down, remove, details

* The first 3 characters of the mode name will be the extension of the export file for this mode.
Export Formats

- collections of XSLT programs
  - and translation tables
- export format is a container XSLT
  - generated on the fly for export requests
  - importing a number of installed XSLTs
  - import precedence can be configured
- encouraging re-use of XSLT modules
- translation tables are simple XML documents
  - XML is based on Unicode (90'000 characters)
  - export formats often are more limited
  - a set of mappings (e.g., ã → \"{a}\")

Conclusions

- supporting metadata handling
  - better management of their own metadata
  - access to metadata of others
  - easy conversion to different formats
    - BibTeX for writing papers
    - HTML for having it available online
    - HTML for using it as bookmarks
    - RDF for shipping it to partners
- prototype implementation
  - performance problems for large data volumes
  - management functions need to be improved
  - central service provision would be ideal
Thank You! — Q&A

Project Page: http://dret.net/projects/bibtexml/

Paper: http://dret.net/netdret/publications#wil03j