Managing semantics with content using DITA XML

presented by Erik Hennum
IBM STG User Technology

The talk at a glance

- The need for content semantics
- Background about SKOS and DITA
- The XML implementation
- Lessons learned, limitations, and future directions
Content – the other semantics

- **The human-readable documents (discourse)**
  Not the values that are processed only by software (data)
  Classification identifies what the content is about

- **Important for the vision of the Semantic Web**
  The worldwide distributed database should support text blobs
  The issues of semantic interoperability and integration also apply

- **Why the semantics of content are useful**
  Discover the relevant content
  Filter the irrelevant content
  Compose views of content based on relevance

- **Content provides a human interface for data semantics**
  Where people have to understand a thing or an activity
  Isolated fragments of black box text can’t handle all cases

Problems with divorcing the semantics from the content

- **Typically semantics and content are maintained separately**
  By different people and with different tools

- **Classifiers have to read and understand the content**
  Expensive if they do and inaccurate if they don’t

- **Classification and content have to be maintained in parallel**
  Hard to coordinate and inaccurate when it isn’t coordinated

- **Focus on the tool instead of the content**
Benefits of keeping the semantics close to the content

- Use document tools to define the semantics
- Writers maintain the content and its classification
  Two different ways of expressing the same subject matter
  Use the classification to improve the content
- Low barrier to entry

Theoretical background

- **Hypertext theory**
  “The relationship between hypertext and semantic networks has long
  been realized.” – Horrocks, McGuinness, and Welty
  “Hypertext can be seen as a logic representation, where semantics
  are encoded in both the textual nodes and the graph of links.”
  – Millard, Gibbins, Michaelides, and Weal
  Coarse-grained, evolving, tacit, or contextual knowledge benefit from
  a less formal representation
  – Shipman and Marshall

- **TopicMaps standard**
  Formalized content relationships – table-of-contents, index, ...
Ingredients of the solution

- **SKOS**
  W3C RDF vocabulary (Simple Knowledge Organization System)
  Formal concepts and their relationships

- **DITA**
  OASIS XML standard (Darwin Information Typing Architecture)
  Human-readable, semantic content objects and their relationships

- **DITA taxonomy specialization**
  Extends DITA to provide an authorable XML format for the SKOS model
  Uses hypertext relationships to specify semantic relationships
  Defines a taxonomy
  Classifies the content

**SKOS (Simple Knowledge Organization System)**

- **W3C Public Working Draft**
  “… expresses the basic structure and content of concept schemes such as
  thesauri, classification schemes, subject heading lists, taxonomies, ‘folksonomies’, other types of controlled vocabulary, and
  also concept schemes embedded in glossaries and terminologies.”

- **Fills a hole in the RDF stack**
  Between ad hoc RDF properties and fullblown OWL ontologies

- **Enriched by a cross-section of perspectives**
  Library Science experts
  Terminology experts
  RDF and TopicMaps standards leaders
  Open Source project leads (content management)
DITA (Darwin Information Typing Architecture)

**An OASIS XML standard**
Goal of usability for writers and vocabulary designers (not just processing)

**High-level features:**

1. **Topics (documents)**
   Human-readable content objects
   Emphasis on semantic focus

2. **Maps**
   Hierarchical or associative relationships between topics

3. **Specialization**
   Extensibility to add modular XML vocabularies
   Increase semantic precision and constrain content structures

---

### DITA 1: A topic is a content object

```xml
<topic id="xmlmarkup" xml:lang="en-us">
  <title>About XML markup</title>
  <shortdesc>Fundamentally, markup applies an annotation to a delimited span of content.</shortdesc>
  <body>
    <p>The annotation provides</p>
    <ul>
      <li>Indicates the semantic</li>
      <li>Specifies the properties</li>
      <li>Constrains the structure</li>
    </ul>
    <example>...</example>
  </body>
</topic>
```

- **Semantic focus, granularity, and independence**
  Rich text elements from HTML such as `<p>`, `<ul>`, `<dl>`, ...
  Emphasis on structure and semantics of content instead of presentation
DITA 2: A map defines relationships between topics

```xml
<map>
  ...
  <topicref navtitle="Creating Web services"
    ...
  <topicref navtitle="Creating Web service application"
    ...
  ...
</map>
```

- Relationships and properties defined outside of the topics
  - Hierarchical relationships – for instance, a navigation such as a sitemap
  - Matrix or group associative relationships – for instance, related links

- A topic can have different relationships in each context
  - Organize subsets of the same content in many different ways

---

DITA 3: New markup can be specialized

```xml
General topic
<topic id="installstorage">
  <title>Installing a hard drive</title>
  <body>
    <ol>
      <li><ph>Unscrew the cover.</ph></li>
      <itemgroup>The drive...</itemgroup>
      <li><ph>Insert the drive...</ph></li>
      <itemgroup>If you feel...</itemgroup>
    </ol>
  </body>
</topic>

Specialized task
<task id="installstorage">
  <title>Installing a hard drive</title>
  <taskbody>
    <steps>
      <step><cmd>Unscrew the cover.</cmd>
      <stepresult>The drive...</stepresult>
    </step>
    <step><cmd>Insert the drive...</cmd>
      <info>If you feel resistance...</info>
    </step>
    <step><cmd>Insert the drive...</cmd>
      <info>If you feel resistance...</info>
    </step>
  </steps>
</taskbody>
</task>
```

- Derive new XML markup from existing markup
  - Extension by substitution to increase semantic precision
  - Modules pluggable into the base DITA vocabulary

- Adapts to support new requirements
Using the DITA taxonomy specialization

Steps:
1. Write the DITA subject documents
   Equivalent to a formal concept in SKOS
   Properties include preferred label, description, alternate label, scope note, ...
2. Write the DITA taxonomy map
   Equivalent to a SKOS scheme
   Defines hierarchical and associative relationships between subjects
3. Add the classification to the DITA navigation map
   Specify the classification while organizing the content

Step 1: Define a subject with a topic document

- An extension on the familiar markup for conceptual documents
  Specialized markup prompts the writer for the SKOS properties
- A readable explanation for users or other writers
  Some properties for internal consumption (scopeNote, changeNote)
Step 2: Define a taxonomy with a hypertext map

- An extension of the familiar markup for hypertext relationships
  - Applied to the documents that define formal subjects
- Hierarchy of subject documents
  - Like a sitemap but for hasKind, hasPart, or hasInstance relationships
- Associations between subject documents
  - Like related links

```
<subjectScheme ...
  <hasKind>
    <subjectdef navtitle="Application server technology" ...
      <subjectdef navtitle="Web Services" href="WebServices.dita">
        ...
    </subjectdef>
    ...
  </hasKind>
  <relatedSubjects>
    <subjectdef navtitle="Service Oriented Architecture" ...
      <subjectdef navtitle="Web Services" href="WebServices.dita"/>
        ...
    </subjectdef>
  </relatedSubjects>
</subjectScheme>
```

Step 3: Classify the content with a hypertext map

- Insert the classification into the standard navigation
  - Use the <subjectref> element to refer to subject documents
    - Classify individual content documents or entire collections of documents
- Most writers work with this view
  - Information architect maintains the subject documents and taxonomy map
- Change the taxonomy without changing the classification

```
<map>
  ...
  <topicref navtitle="Creating Web service applications"
            href="creatingwsapp.dita">
    <topicsubject>
      <subjectref navtitle="Web Services" href="WebServices.dita"/>
      <subjectref navtitle="Application Development" ...
        ...
    </subjectref>
  </topicsubject>
</map>
```
Manage the runtime representation in SKOS RDF

- XSLT transforms convert DITA source files to runtime SKOS RDF
- Also transform content to HTML pages (where appropriate)
- RDF APIs can query or traverse the SKOS model

Runtime 1: the Open Source SWED facet browser

Subjects,

http://www.swed.org.uk/swed/servlet/Entry?action=v

2006 Semantic Technology Conference
San Jose, California ● March 6-9, 2006
Runtime 2: IBM STG product information browser

Snapshot of work in progress

Demonstration
Variation 1: A document as a subject and content

```
<Map>
  ...
  <topicref navtitle="Creating Web service applications" href="creatingwsapp.dita">
    <topicsubject>
      <subjectref navtitle="Web Services" href="WebServices.dita"/>
    </topicsubject>
  ...
  <topicref navtitle="Glossary"/>
  ...
  <topicref navtitle="Web Services" href="WebServices.dita"/>
  ...
</Map>
```

- **Useful when readers might need the subject definition**
  Typically unfamiliar glossary terms or conceptual background

- **Refer to the same topic with different elements**
  Use `<subjectref>` to classify content with the subject
  Use `<topicref>` to include the subject document in the navigation

- **Standard formatting can process the specialized subject topic**

Variation 2: Classify HTML or PDF documents

```
<Map>
  ...
  <topicref navtitle="Web Services Activity" format="html" href="http://www.w3.org/2002/ws/">
    <topicsubject>
      <subjectref navtitle="Web Services" href="WebServices.dita"/>
    </topicsubject>
  ...  
  <topicref navtitle="Business Process Execution Language 1.1" format="pdf" href="ws-bpel.pdf">
    <topicsubject>
      <subjectref navtitle="Web Services" href="WebServices.dita"/>
    </topicsubject>
  ...
</Map>
```

- **Local or remote resources in other formats**
  Use the format attribute to distinguish from DITA content documents
  Supply a title for readable source and simple processing

- **Classify the resource with DITA subject documents as usual**
Variation 3: Define subjects in SKOS or OWL

```
<subjectScheme ...
  <hasKind>
    <subjectdef navtitle="Application server technology"
      format="skos" href="http://some.org/subjects.rdf#appserver">...
    <subjectdef navtitle="Web Services" href="WebServices.dita">
      ...
  </subjectScheme>
```

- Integrating with public semantics or local ontologies
  - Use the format attribute to distinguish from DITA subject documents
  - Best to supply a title for readable source and simple processing
- Use external subjects in taxonomy or classification
  - RDF-based or TopicMaps formats
- Extend shared general semantics for local specific semantics
  - External subjects provide the trunk and branches of the taxonomy
  - DITA subjects provide the twigs and leaves for content semantics

Variation 4: Tagging without a taxonomy

**Steps:**

1. Define the subject documents as usual
   - Don’t organize the subjects in a taxonomy
2. Classify the content with the subjects as usual

**Benefits of defining tags as subjects:**

- Minimize single tags with many meanings
- Minimize multiple tags with the same meaning
- Increase the semantic precision of tagging
Challenges for writers

- **Focusing on the subject meaning instead of the label**
  The titles are only reminders of the subject definition and content meaning

- **Defining the taxonomy hierarchy**
  Creating hasKind relationships instead of hasPart relationships
  Applying the intersection of existing subjects instead of defining a new compound subject
  Balance – consistent depth of coverage
  Pragmatic discipline – avoid obsession with subdivision of meaning

- **Modifying the content during classifying**
  Avoid treating the current content or navigation as a carved in stone

Benefits of the approach

- **Leverage document familiarity and infrastructure**
  Demystification – you already create semantics when you create content
  Apply hypertext understanding to taxonomy definition and classification
  Use content tools to edit, format, and archive the semantic definitions
  A low cost and open solution for basic semantics of content

- **Improve the content**
  Identify holes in your coverage of the subject matter
  Identify content with a blurred focus
  Identify duplicate content and avoid contradiction nightmares
Limitations of the approach

- **Not a format for rigorous ontologies**
  Useful where an ontology would be hard to create or maintain
  Integrate content semantics with ontologies as part of a continuum

- **Not a format for common linguistics**
  Best in a domain with precise concepts

- **Not a method for classification of static inventories**
  Best where content is maintained with semantic focus
  Integrate with text mining (such as UIMA) for large content archives

- **Not a power tool for semantic management**
  Can provide a document interface for semi-formal knowledge

Future directions

- **Add a display-oriented map for subjects**
  Separate presentation from taxonomy definition concerns (SKOS idea)

- **Represent deeper knowledge about the subjects**
  Associative relationships and properties for subjects
  Specialize the subject definition elements – aligns with SKOS
  speculations about subclassing in parallel with conceptual hierarchy

- **Leverage more of the content semantics**
  Specialized DITA markup provides semantic annotation for content

- **Public definitions of technical subjects**
  ACM taxonomy not detailed enough
  CIM model isn’t user oriented
Future: specialized subject relationships

- Source for a specialized associative relationship
  
  ```
  <containerFor>
    <subjectdef href="toolbox.xml"/>
    <subjectdef href="tools.xml"/>
  </containerFor>
  ```

- Runtime RDF
  
  ```
  <rdf:Property rdf:about="&garage;containerFor">
    <rdfs:subPropertyOf rdf:resource="&skos;related"/>
    ...
    <skos:Concept rdf:about="&garage;toolbox">
      <garage:containerFor
        rdf:resource="&contentSubject#tools"/>
  ```

Summary

- Importance of semantics for content
  Not data vs documents but both
  Not formal vs informal semantics but a continuum

- Maintain semantic declarations with the content
  Improve the content instead of trying to bolt on semantic precision

- Leverage SKOS and DITA standards and tooling
  Use the familiarity of writers with hypertext and documents:
  1. Define subject documents
  2. Organize subjects in a taxonomy map
  3. Classify content in a navigation map
## Resources

- **SKOS**
  - W3C - http://www.w3.org/2004/02/skos/

- **DITA**
  - OASIS – http://www.oasis-open.org/committees/dita
  - Cover page – http://xml.coverpages.org/dita.html
  - Forum – http://groups.yahoo.com/group/dita-users/
  - DITA Open Toolkit – http://dita-ot.sourceforge.net/

- **DITA Taxonomy specialization**
  - Plugin available at the SourceForge site for the DITA Open Toolkit

Feedback welcome – **Erik Hennum – ehennum@us.ibm.com**