Topic Maps

(Overview and Basic Concepts)

(XSLT Example)

Comments to: patrick@durusau.net
or (more usefully)
sc34wg3@isotopicmaps.org (requires subscription)
or
topicmapmail@infoloom.com (requires subscription)

30. July 2009

Table of Contents

1.Introducing Topic Maps

This year (2009) is the tenth anniversary of the first version of topic maps becoming an ISO standard. Topic maps are in use by governments, stock exchanges, any number of educational institutions and others. An introduction may seem odd under those circumstances but there is no guide to topic maps and the standards that define them. This document is a thumbnail sketch of topic maps and the actual standards constitute the normative definitions and rules for topic maps.

1.1. Merging Indexes: The Original Use Case

Topic maps are the answer to what looks like a simple question: How to merge two or more indexes from books in the same area? Walking through an example of the original use case will illustrate some of the problems that topic maps solve and the benefits that they can provide.

1.2. An Example with Basic Terminology

A small portion of the index to Michael Kay's *XSLT 2.0 and XPath 2.0*[1] reads:

> **Sorting, 242. see also ....xsl:sort**
> collations and, 106, 459
> distinct value, Xquery and, 750

Indexes have entries which in topic maps are said to be subjects we want to talk about and so are represented by **topics.** Some of the those subjects (by no means all of them) that could be represented by topics include: sorting, xsl:sort, collations, distinct value and Xquery. Those should be familiar to anyone who has ever used the index in the back of a book.

Book indexes record where a reader can find more information about a subject in a book. In topic maps we call those **occurrences** of a topic. In this particular entry, we find that more information on sorting is found on page 242, collations, on pages 106 and 459, and so on. Part of the "map" in topic maps is that a subject has been found (past tense) and its location recorded, just as in an index.

Indexes can also indicate relationships between subjects in the index. In a topic map those relationships are called **associations**. In this index fragment there is a relationship between sorting and collations as well as between sorting, distinct value and XQuery.

Associations are used in topic maps to provide explicit information about relationships. Information that answers questions such as: What is the relationship between sorting and collations? Or for that matter, why see xsl:sort?

1.3. One Index Plus Another Index = Trouble

We could know more about the entries in the index to Kay's book but it doesn't look all that complicated. For comparison and as another index to merge with that one, let's take a look at a small portion of the index of Jeni Tennison's *XSLT 2.0*[2]:

> Sorting

1 Reproduced with permission of the author.
2 Reproduced with permission of the author.

flexible
flexible sort orders, 409
flexible sort values, 412
overview, 409[3]

So, what happens if we do a crude “merge” on the index fragments from Kay and Tennison:

Sorting, 242. see also ....xsl:sort
collations and, 106, 459
distinct value, Xquery and, 750
flexible
flexible sort orders, 409
flexible sort values, 412
overview, 409

Wow! That is both ugly and useless. Even if you own both books, which page number (**occurrence**) goes with each book has been lost. A relationship between the book and the index in each case exists but it is only implied and there is no representative for the book in the index.

One requirement for useful merging of indexes is to keep track of which book has the information a user may want. And that information should be associated with page numbers (**occurrences**). Which means we are going to need to represent the book (**topic**), the entries (**topics**) and decide how to associate the page numbers (**occurrences**) with it.

So that the page numbers are kept with the work where they occur, let's try using the respective author's last names:

Sorting
Kay - 242
Kay - collations and, 106, 459
Kay - distinct value, Xquery and, 750
Tennison - flexible
Tennison - flexible sort orders, 409
Tennison - flexible sort values, 412
Tennison - overview, 409
Kay – xsl:sort

That is a “solution” to the issue of what page number goes with which work. But it is a problematic solution as well as not solving more fundamental issues.

3 Indexes are generally prepared by publishers and not the author and in this case it shows. I was puzzled by “flexible” sorting in part because the alternative would have to be “inflexible” sorting, which given the nature of sorting doesn't make a lot of sense.

The most obvious problem is that no one who hasn't read this document will know that Kay = “XSLT 2.0 and XPath 2.0” and that Tennison = “Beginning XSLT 2.0.” Or for that matter, which “Kay” I may be talking about.[4]

Briefly the problem is what subject/topic is being identified by a bare string (the entry in an index)? Without more, given that users use the same string for different subjects and different strings for the same subject, something more is needed. Something that a traditional index lacks. Such that users working independently can ascertain what subject was meant. The TMDM (ISO 13250-2) defines a solution to that issue.

There is a deeper problem with this “merged” index. If you are familiar with XSLT 2.0, you remember that the word “flexible” occurs only twice, once in a note and the other time in an example.[5] As far as you can remember, you haven't seen any concept of “flexible sorting” in XSLT 2.0. And Kay doesn't mention “flexible sorting” in the index to his book. Is this something new? Did Kay leave something out? Actually not. Both books cover the same material but use different terminology when they do.

When Kay says: “see xsl:sort,” Tennison says, “flexible sort orders.” But there isn't anything in the index to indicate that equivalence. What would be very useful would be to have information listed any way a user may be looking for it.

Here is one attempt to solve the subject equivalence issue, at least as far as xsl:sort vs. flexible sort orders:

Sorting

Kay - 242
Kay - collations and, 106, 459
Kay - distinct value, Xquery and, 750
Tennison - flexible
Tennison - flexible sort orders, 409
Tennison - flexible sort values, 412
Tennison - overview, 409
xsl:sort
Kay – see xsl:sort
Tennison - flexible sort orders, 409
Tennison - flexible sort values, 412
Tennison - overview, 409

One obvious problem is that the knowledge that xsl:sort and “flexible sorting” are the same subject isn't represented. At least not explicitly. Which means that anyone who wants to merge this example with another index will have to guess as to why Tennison's “flexible sorting” is listed under xsl:sort.

Being able to merge indexes, particularly if those indexes could extend beyond books would be quite valuable. Once information is found by one user, it could be more easily found by other users. It is the difference between having a common city map and every citizen making their own map. Both are

4 The “Whitepages” report 24 Michael Kay's in Texas alone.

5 XSL Transformations (XSLT) Version 2.0, http://www.w3.org/TR/xslt20/, in the first note in 11.9.1 Shallow Copy, and, 14.4 Examples of Grouping, in the example titled: “Example: A Composite Grouping Key.”

possible but the first option, having a common map, is the least expensive and most useful for all.

But as the experiment with indexes shows, we have to solve several problems at the same time. We have to identify subjects (represented by **topics**), keep places where we have seen those subjects (**occurrences**) linked up to those subjects. And, be able to explicitly represent relationships between subjects (**associations**). That is doable, but only with some rules to govern that process. Next stop: ISO 13250-2, Topic Maps Data Model (or Defining the Rules of the Road).

2. Defining Rules of the Road (Topic Maps Data Model, ISO 13250-2)

2.1. General Comments

The Topic Maps Data Model (TMDM), along with the syntaxes for topic maps (ISO 13250-3, ISO 13250-6), enables independently constructed topic maps to meaningfully merge with each other, unlike our index example.

In order to have interchange of subject based information between two parties there has to be a basis for that interchange. The TMDM, defined a metamodel using the XML Information Infoset (XML Infoset) that exchange. It also serves as the basis for standards that define canonical syntaxes, querying, constraints and other matter relative to topic maps.

The TMDM is what the Topic Maps Reference Model (ISO 13250-5) calls a **legend**. Just like the legend on a map, a topic maps legend defines all the rules for a map. Such as how to represent subjects, how subjects are identified, how to compare those identifications, what to do when two or more subject representatives represent the same subject and other rules.

The rules of the TMDM are best learned by reading the text of the TMDM itself but there are several themes that should be called out for special attention.

2.2. Identifying Subjects

One of the problems we faced in trying to merge the indexes was how to know what terms went with what subjects? Authors use different names for the same subjects and the same names for different subjects. As a legend, the TMDM specifies ways to identify any subject we want to talk about.

The TMDM divides the subjects we want to talk about into two very large categories. First, there are subjects we want to talk about that have network addresses. The XSLT 2.0 specification, the Wall Street Journal website, the Google search homepage, all of those have network addresses.

The important thing about having a network address is that any user who wants to know what subject is being identified, in theory at any rate, can visit that address to see the “subject.” (In topic maps language a subject with a ne twork address is identified by a **subject locator**.)

Second, there are subjects we want to talk about but they have no network addresses. Michael Kay, as a person, for example, cannot be viewed over the network. And as a matter of fact, there are 28 Michael Kays in Texas (United States) alone. Given that we can't view the Michael Kay who edited XSLT 2.0 over the network, how will we know when we are talking about the same Michael Kay?

The TMDM solves that problem by allowing for the creation of an information resource with a network address that **indicates** the subject that is being discussed. That is a resource that a user can visit to decide if the subject which has no network address is the same subject they are talking about. (The

address of a network resource that indicates a subject is called a **subject identifier.**)

To peek ahead at the XML syntax, here is the identification of a subject that has no network address and one that has a network address:

Subject without network address:

```
<topic id="mike_kay">
    <name>
        <value>Michael Kay</value>
    </name>
    <subjectIdentifier href="http://www.durusau.net/psi/Michael_Kay/"/>
</topic>
```

Subject with a network address:

```
<topic id="xslt.2.0">
    <name>
        <value>XSLT 2.0</value
    </name>
    <subjectLocator href="http://www.w3.org/TR/xslt20/"/>
</topic>
```

2.3. Merging Rules

The TMDM defines merging rules for topic and other topic map constructs found in a topic map. As a consequence of those definitions, any topic map engine that follows them will produce the same results as any other engine. The full story of those rules is told in the TMDM (ISO 13250-2) but suffice it here to say that rules for comparison of subject identifiers or subject locators, rules for what happens with either identifiers or locators match, and all of the constructs necessary to create a topic map are defined by the TMDM.

2.4. Back to the Index Example

To illustrate a part of what the TMDM defines, recall that there was a relationship in the Kay index that we wanted to represent. The TMDM defines the constructs (but not the syntax) for representing such relationships.

Recall from part of our example:

> **Sorting, 242. see also ....xsl:sort**
> collations and, 106, 459

Let's sketch out how to represent the relationship between xsl:sort and collations.

First, we want to treat both of those as subjects so they will be represented by topics. Neither one has a network address so we will have to create a resource to identify them.

What we haven't discussed is what to call this relationship and any parts of it. Relationships themselves are known as **associations**. And in an association there are **roles**, that is what part is being played in the relationship. We would represent a marriage relationship, for example, with an association and we would say that husband and wife are the roles in that relationship. That allows us to discuss the

relationship without talking about talking about any marriage in particular.

If we were talking about some marriage in particular, the people who are in the husband and wife roles would be called **role players**.

Let's see, we need representatives for:

xsl:sort (subject, role player)

collation (subject, role player)

function (subject, role in the association)

input2function (subject, role in the association)

type (subject, for the association, alters_sort)

As XSLT type know, “collation” is an attribute on the <xsl:sort> element. An association would show that “collation” plays the role of “input2function” in this association. That may sound like a long way to go for little gain but consider this: Once encoded, it will be possible to determine every place where collation appears as an attribute and what is more, what other inputs will have an affect on a sort. Suddenly that doesn't look quite so academic.

The TMDM defines one part of what is needed for interchange, a common model. But a syntax is necessary if information is to move from one computer to another. The basis for that communication is defined by ISO 13250-3 Topic Maps XML Syntax (or A Topic Maps Language).

3. A Topic Map Language (Topic Maps XML Syntax, ISO 13250-3)

Having a meta model that defines interchange (TMDM) is a necessary but not sufficient condition to enable interchange. XTM, the XML syntax for topic maps, is based on a mapping to the TMDM and serves as the interchange format for topic maps.

Using parts of our prior examples, what follows are samples of XTM 2.0 syntax:

To represent Michael Kay as an author, we need a topic that has a pointer to a resource that identifies Michael Kay. Here is one possibility:

```
<topic id="mike_kay">
    <name>
        <value>Michael Kay</value>
    </name>
    <subjectIdentifier href="http://www.durusau.net/psi/Michael_Kay/"/>
</topic>
```

On the other hand, to represent a network resource we would write:

```
<topic id="xslt.2.0">
    <name>
        <value>XSLT 2.0</value
    </name>
    <subjectLocator href="http://www.w3.org/TR/xslt20/"/>
</topic>
```

What is important in both cases is that a user can follow those links to ascertain if they are talking

about the same subject.[6] Software, of course, cannot understand the content returned for either one and so make merging decisions on the basis of string comparisons.

Assuming topics that represent “editor,” “standard,” and “editor_of” we can represent Michael Kay's relationship to the XSLT 2.0 standard with:

```
<association>
        <type>
                <topicRef href=”#editor_of”/>
        </type>
<role>
        <type>
                <topicRef href=”#editor”/>
        </type>
                <topicRef href=”#mike_kay”/>
</role>
<role>
        <type>
                <topicRef href=”#standard”/>
        </type>
                <topicRef href=”#xslt.2.0”/>
</role>
</association>
```

The href values given in this example presume topics in the current topic map but they could just as easily be references to locations outside of the topic map.

Or we can model the occurrences for the subject “collation” as follows:

```
<topic id="collation">
        <name>
                <value>collation</value>
        </name>
        <subjectIdentifier href=”http://www.durusau.net/psi/collation/”/>
<occurrence>
<scope>
        <topicRef href=”#xslt_2.0_xpath_2.0”/>
</scope>
<resourceData>106</resourceData>
</occurrence>
<occurrence>
<scope>
        <topicRef href=”#xslt_2.0_xpath_2.0”/>
</scope>
<resourceData>459</resourceData>
</occurrence>
</topic>
```

6 The distinction between addressable and non-addressable subjects allows topic maps to avoid dubious re-direction mechanisms.

While all that may seem straightforward to the average reader, the question for interchanging topic maps (and merging after interchange) is whether different topic map processors are getting the same results from the same topic maps. If they are not, then all of the careful identification of subjects and the production of the topic map in syntax has been in vain. A method for testing topic map processors is defined by Topic Maps Canonical Syntax ISO 13250-4 (or, Following All the Rules).

4. Following All the Rules (Topic Maps Canonical Syntax, ISO 13250-4)

If there are any parts to ISO 13250 that can be skipped entirely without diminishing your appreciation of topic maps, then Topic Maps Canonical Syntax, ISO 13250-4 is it.

CXTM, as it is known to the topic maps community, is used to create serializations that insure that two topic map engines, when given equivalent input, are capable of creating equivalent output. It is of interest to developers and then only for use in test suites for topic map software.

5. Subjects, Subjects Everywhere (Topic Maps Reference Model, ISO 13250-5)

It would be deeply ironic for a standard premised on users having different identifications of subjects which prevents semantic integration, to then say that there is only one way to talk about subjects that solves that issue. One of the impediments to semantic integration today is that every solution sees itself as the only solution and if users would just all use “insert brand X” of integration technology, all would be well.

Except that no software system to date, including topic maps has gained universal acclaim for semantic integration. Rather than continue in that vein, the topic maps reference model (TMRM) posits a fundamental view of the representatives of subjects that has a minimal set of ontological presumptions.

Briefly stated, the TMRM assumes that all subjects are represented by proxies which are composed of key/values pairs. Every key is a label for a proxy that represents the subject of that key. Values may or may not be labels of proxies.

It doesn't take long thinking about such a model to realize two things about it:

1) The structure of information systems (the “keys”) represent subjects just as much as the structure itself may represent other subjects, and,

2) That subject identification always depends upon other subjects, which may or may not be explicitly identified, that is to say that subject identification is always recursive.

The recursive nature of subject identification isn't troubling because all systems will pick a place that seems suitable to it to end the recursion. For some purposes, it may be sufficient to treat all column headers in a relational database as primitives. For others, such as integration with another relational database, that might be a poor choice.

But knowing the properties that are attributed to representatives of subjects is insufficient to enable semantic integration. The TMRM defines the concept of a legend, which like the legend on any map defines the rules for that map. Among the things that a legend could define would be ontological assumptions such as data primitives, the basis on which subject representatives are compared, what happens in the case of equivalence and for all other matters that seems needful to the author of the legend.

It isn't possible to “implement” the TMRM as it is more of a set of premises as to what is required in order to define a legend. The TMRM does not define any equivalence operations for example. Just as there are an unbounded number of ways in which subjects can be identified, there are an equally unlimited number of ways to determine when subject representatives represent the same subject. What the TMRM provides is a formal model upon which to base the disclosure of legends that make semantic integration possible.[7]

From the highly abstract Topic Maps Reference Model, which is useful primarily to information theorists and those who want to have repeatable and disclosed semantic integration of heterogeneous information resources we move to address the needs of a “texting” generation. That would be Topic Maps Texting (Topic Maps Compact Syntax, ISO 13250-6).

6. Topic Map Texting (Topic Maps Compact Syntax (CTM), ISO 13250-6)

As we saw in the examples of the XML syntax defined by ISO 13250-3, even simple topic map fragments can become quite verbose. That poses problems both for use in examples (to get more material on a screen) as well as when a person is hand authoring a topic map. The other factor that influenced the development of a compact syntax was the need for a common syntactic basis for both a constraint language (ISO 19756) and a query language (ISO 18048).

The semantics of CTM, as it is known in the topic maps community are defined by the TMDM (ISO 13250-2).

To briefly illustrate the compactness of the notation, here are the examples that were written in XML as illustrations of ISO-13250-3:

```
<topic id="mike_kay">
        <name>
                <value>Michael Kay</value>
        </name>
        <subjectIdentifier href="http://www.durusau.net/psi/Michael_Kay/"/>
</topic>
```

Compare CTM:

```
mike_kay http://www.durusau.net/psi/Michael_Kay;
- “Michael Kay”.
```

Or, the markup for a simple association:

```
<association>
        <type>
                <topicRef href="#editor_of"/>
        </type>
<role>
        <type>
                <topicRef href="#editor"/>
        </type>
                <topicRef href="#mike_kay"/>
```

7 The Topic Maps Data Model (TMDM), ISO 13250-2, is an example of a legend.

```
</role>
<role>
        <type>
                <topicRef href=”#standard”/>
        </type>
                <topicRef href=”#xslt.2.0”/>
</role>
</association>
```

Compare CTM:

```
editor_of(standard: xslt.2.0, editor:mike_kay)
```

Or, the markup for a topic with occurrences:

```
<topic id="collation">
        <name>
                <value>collation</value>
        </name>
        <subjectIdentifier href=”http://www.durusau.net/psi/collation/”/>
<occurrence>
<scope>
        <topicRef href=”#xslt-2.0-xpath”/>
</scope>
<resourceData>106</resourceData>
</occurrence>
<occurrence>
<scope>
        <topicRef href=”#xslt-2.0-xpath”/>
</scope>
<resourceData>459</resourceData>
</occurrence>
</topic>
```

Compare CTM:

```
 - "collation";
    page: 106 @xslt-2.0-xpath;
    page: 459 @xslt-2.0-xpath;
    page: ??? @beginning-xpath.
```

CTM also defines mechanisms that assist in hand authoring tasks, such as templates that auto-generate portions of a topic map when supplied with the variables defined by the template.

7.I have a question... (Topic Maps Query Language, ISO 18048)

TMQL (Topic Maps Query Language) defines an expression language for the extraction of content from topic maps or any storage mechanism that is viewed as a topic map. In the current edition, only the extraction of content is addressed and updating of topic maps was deferred to a later edition.

The details of the query language are many and readers should consult the full standard for its actual details.

A query that would return all the books that were edited by Michael Kay could read:

```
select $book
where
is-edited-by (editing: $book, editor: Michael_Kay)
```

A query that would return all the books in a topic map that were written in English could read:

```
// book / name [ @ en ]
```

Those are not the only forms of the queries as TMQL enables equivalent queries to be expressed in different forms.

8. Restraints for Topic Map Authors (Topic Maps Constraint Language, ISO 19756)

TMCL (Topic Maps Constraint Language) enables topic map authors to impose constraints on various topic map constructs in a particular map. For example, a topic that represents a book may be required to participate in either an authored_by or edited_by association. The constraint language is in effect a declaration of an ontology for a particular topic map.

(Should I have examples here?)

9. Additional resources and reading

ISO 13250-2: Topic Maps — Data Model (TMDM)

ISO 13250-3: Topic Maps — XML Syntax (XTM 2.0)

ISO 13250-4: CXTM (Canonicalization)

ISO 13250-5: Topic Maps — Reference Model (TMRM)

ISO 13250-6: CTM (Compact Notation)

ISO 13250-7: GTM (Graphical Notation)

ISO 18048: TMQL

ISO 19756: TMCL

Questions:

1) Should I include 13250-7 GTM? Do we have enough there to discuss it?

2) The two main mailing lists should be listed, along with representative websites. What else?

3) More specifically, shouldn't there be links to sites that use topic maps?

4) I can probably get this down to 8-9 pages of substantive text by playing with the font sizes for the examples, etc. Suggestions on placement?

5) How can I better make this a framework into which other index type examples could be inserted? Reasoning that users have an easier time seeing the potential of topic maps in their own domains.

6) Yes, it needs a conclusion before Additional Reading. Now it just trickles off. Needs a strong finish.

7) Any and all comments/suggestions are welcome.