Topic merge scenarios for knowledge federation

Book Section

Topic Merge Scenarios for Knowledge Federation

Jack Park
Knowledge Media Institute, Open University, UK
jackpark@gmail.com

Abstract. Climate change is a growing concern to humankind, since the dominant view argues for rapid, significant changes in human behavior to avert catastrophic consequences. This is a complex problem, known as a wicked problem. A productive way forward is through creative, critical dialogue. Such dialogue requires new kinds of socio-technical infrastructure. We offer a socio-technical infrastructure, described as a boundary infrastructure, based on improvements to existing and emerging Issue-based Information Systems (IBIS) conversation platforms. IBIS is an emerging *lingua franca* of structured discourse. We report on a core function in that boundary infrastructure: the topic merge architectures of topic mapping frameworks. We use two scenarios known to exist in our research platform to develop a novel merge architecture that supports *virtual merges* of topics.

Keywords: topic maps, merge algorithms, virtual merge

1.0 Background

Climate change is a growing concern to humankind, since the dominant view argues for rapid, significant changes in human behavior to avert catastrophic consequences. Those changes are proposed against a backdrop of lifestyle and economic change due to proposed and emerging mitigation plans. This is a complex problem, known as a wicked problem (Conklin, 2005). A productive way forward is through creative, critical dialogue. Such dialogue requires new kinds of socio-technical infrastructure. As part of a research and development program aimed at a prototype of a new kind of socio-technical infrastructure, we are developing an open source collective sensemaking platform we call Bloomer[1]. Bloomer serves two purposes in our research. One purpose is to support a thesis project that explores the ability to *federate* structured conversations (Park, 2010); the other purpose is to provide a socio-technical infrastructure, a boundary infrastructure (Bowker & Star, 1999) for collective intelligence, described as a *knowledge garden* (Park, 2008). From (Bowker & Star, 1999, p.313):

> "Any working infrastructure serves multiple communities of practice simultaneously be these within a single organization or distributed across multiple organizations….Boundary infrastructures by and large do the work that is required to keep things moving along. Because they deal in regimes and networks of boundary objects (and not of unitary, well-defined objects), boundary infrastructures have sufficient play to allow for location variation together with sufficient consistent structure to allow for the full array of bureaucratic tools (forms, statistics, and so forth) to be applied."

Bloomer combines a variety of web-based portals, including MediaWiki[2], Drupal[3], Cohere[4], and others, each communicating with a topic map platform we call TopicSpaces. To MediaWiki, we added a new extension that enables IBIS conversations to be conducted in the wiki. From this combination of platforms and from our goal to federate information resources, we derive novel topic merging situations, two of which we describe in this report.

Our use of the term *federate* entails topic maps. We posit that subject-centric merging of topics captured in social sensemaking settings offers opportunities for collaboration based on participants' discovery of like-minded others. Federation, in our sense of the word, is both a noun and serves as an act. The federation act is that of topic maps merging processes; a federation, the noun, is a collection of people, information resources, and the boundary

[1] Bloomer: http://code.google.com/p/bloomer/
[2] MediaWiki: http://www.mediawiki.org/
[3] Drupal: http://drupal.org/
[4] Cohere: http://cohere.open.ac.uk/

infrastructure together with its many related boundary objects (Star, 1989). A topic in a topic map, like a concept drawn on a chalk board, is an instance of a boundary object; it is co-owned by all participants and serves as a place where resources are shared.

We describe our research in relation to two scenarios that arise in the Bloomer platform. When one creates a federation that facilitates social contributions from many participants on varieties of user interface platforms, one of our scenarios emerges: Federating representations of human participants—humans as topics. As we shall soon see, identifiers of the same individual can present to the federation as distinctly different depending on the platform used. Our second scenario is precisely that which is the subject of our thesis research: federation of structured discourse (Park, 2010). We describe these scenarios in more detail below after a brief review of merge technology issues.

In the following, we will describe concepts of merging of topics. Our research supports two kinds of merge actions: those which are software agent-based, guided by rules that inspect and vote for or against a merge between two topics, and those which are the result of human direction. Topic merging entails a single consideration when comparing two representations—called *subject proxies*: *do these two subject proxies represent the same subject?* That single question raises opportunities for research and innovation. In the remaining parts of this report, we examine a popular implementation of topic map merging technology, compare that to perceptions of the federation process that require merging, and then describe a new approach to implementation of merge processes we call *virtual merging*. We then describe two merge scenarios appropriate to the Bloomer project, its mission, and our thesis research.

1.1 Topic Maps Technologies

For this research, we identify two distinct approaches (among possibly many) to the fabrication of topic map platforms. A popular platform is based on the XML topic maps standard, two implementations of which are known by the acronyms XTM[5] (Pepper & Moore, 2001) and TMDM[6] (Garshol & Moore, 2008). Another implementation approach is known as by its acronym TMRM[7] (Durusau et al., 2007). We explore differences in the two approaches to topic maps architectures distinguished by the XML and TMRM approaches. In the following, the term *locator* refers to an identifier (string) for an object that is unique to the database in which the object is stored.

In XML topic maps, three primary objects exist: *topics, associations,* and *occurrences*. A topic map is a collection of topic and association objects. Topics serve as containers for occurrence objects.

With TMRM, a topic map is a collection of subject proxies, where a subject proxy, like a topic object in XML topic maps, serves the purpose of containing representations of the subject for which it stands. Thus, the term *subject proxy* is an analog for a *topic*. The difference is this: a subject proxy is a container for property objects, subject properties. With the TMRM, a map's author may create property types—known as *keys*—to suit particular needs. In XML topic maps, a prescribed choice of property types is available, while others can be fabricated through the use of occurrence objects. In our view, both approaches serve the same purposes.

TMRM implementations do not distinguish between the concept types *topics*, *occurrences*, and *associations*; every concept type is always a subject proxy—a topic. Every topic is represented by collections of subject property objects—key-value pairs. In the TMRM, keys—property types—are defined as topics in the map; the specific locators of such topics serve as *keys* in property objects. While there are other means to accomplish the same ends, the overriding requirement of the TMRM is that the definition of those keys required for subject identification be recorded in a public document known as a *legend*.

TMRM specifies that, if there is any relationship (association) asserted between two topics, the association type is a defined topic, and the instance of that association linked between two *actor* topics is, itself, a subject proxy. Roles played by each actor are also topics in the map. Occurrences are topics that represent instances of *things* which *occur* 'out there'—recall, a map is a representation of some territory; occurrences are representations of topics in that territory. A particular web page, for instance, is represented by a subject proxy in a map.

---

[5] XTM: XML Topic Maps

[6] TMDM: Topic Maps Data Model

[7] TMRM: Topic Maps Reference Model

In a TMRM implementation, we have the opportunity to model an XML topic map by simply declaring property types that mimic those declared by either XTM or TMDM. Doing so allows us to inherit useful merging algorithms found in existing platforms. At the same time, the TMRM allows us to explore other merge architectures. The opportunity to explore merge architectures animates this research. We articulate this research through the following desiderata:

- Provenance—used here to mean identification of sources of resources accumulated in any representation of any subject—is of sufficient importance that merge processes that are loss-free in terms of provenance maintenance are important
- Contested merges are those combinations of topic representations that may, at some later time, become suspect; at issue are two aspects of the merge process:
    - Merge accountability—records, histories of merges
    - "Unwindability"—the ability to "un-merge", to unwind a merge and return the representations to their original forms.

That short list of issues leads to this proposition: it should be possible to perform a topic merge in such a way that:

- Provenance is fully maintained for every resource engaged in a topic merge
- All merges are accountable in terms of reasons offered for merge decisions
- All merges are contestable
- Any merge can be unwound, complete with reasons given for such decisions

Merge processes ask questions of subject identity. A trivial and rarely accurate form of subject identification lies in names for things. This author's name, when entered into a web query, is highly ambiguous—many *hits* occur, few of which are correct. But, names for things offer hints; combine hints with other properties such as roles played and a search returns greatly refined results. Combine this author's name with the term "topic map" and web query returns accurate results. Using that observation as a point of departure, we can compare the XML and TMRM architectures in terms of merging.

1.1 Scenarios

We identify two scenarios that we are experiencing in our work with the Bloomer project. One is an artifact of federating different platforms—specifically author identity; the other relates precisely to our thesis research: federating structured conversations. Specifically, the first scenario involves duplication of topics that represent actual human participants in sensemaking activities at different portals when introduced to the federation server, TopicSpaces. The second scenario involves federation of structured conversations accumulated at various portals. We introduce each next.

1.1.1 Federating Participants

This federation issue is best described by our first encounter with the event. Consider that each participant needs an identity at each physical location where the participant either participates, or is modeled—represented—in a topic map. Consider the case of this author: logged in at a TopicSpaces federation server, identified there by the unique login name used. Same author, logged in at a MediaWiki instance, participating in a structured conversation which is being sent to the federation server. In the wiki, the very same individual logs in with the very same login name. The wiki, however, exhibits a quirky behavior: it capitalizes the first letter of the login name and transmits that to the federation server, where that author identity is not recognized. Thus, the scenario…

We have the same individual using the same identifier, but that identifier is mutated through a software artifact based on a particular wiki behavior. The federation server is programmed to deal with unknown authors by creating a new subject proxy for those that are not known to it. Since that method is algorithmic, when that same unknown individual enters the server in a different structure, the algorithm now knows that subject. Still, we experience the

fact that we now have two different representations—subject proxies—for the very same subject. This situation would be hard to detect with algorithmic merge detection processes; it does, however, submit to relatively easy detection when suspected, availing the opportunity for manual merge commands from appropriately credentialed administrators.

1.1.2 Federating Structured Conversations

Our use of the term *structured conversation* refers specifically to the Issue-based Information Systems (IBIS) approach (Rittel & Webber, 1973; Conklin et al., 2003; Conklin, 2005). IBIS conversations are found under the names *dialogue map, issue map,* and *argument map*. In this approach, a *graph* represents a conversation in the form of nodes, which represent questions, answers, or arguments, are coupled together with labeled arcs. Figure 1 is an example of an IBIS *issue map* created using Compendium[8] (arc labels not shown).

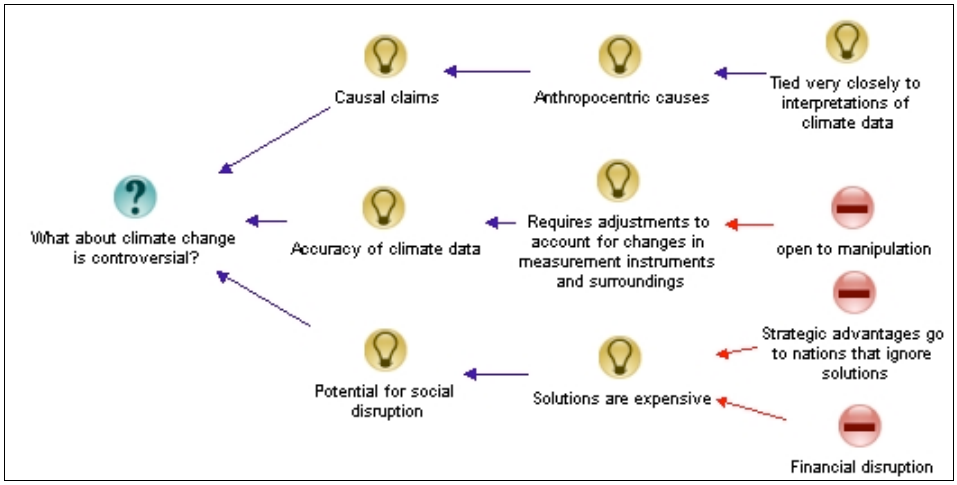

Figure 1. Structured Conversation

When we federate conversations, we seek to provide views into conversations that avoid duplicate nodes. Thus, we merge conversations, node-by-node as described below. To introduce our problem space, consider two trivial IBIS conversations that, to native English speakers, appear to begin with the same question. Figure 2 presents those two conversations together.

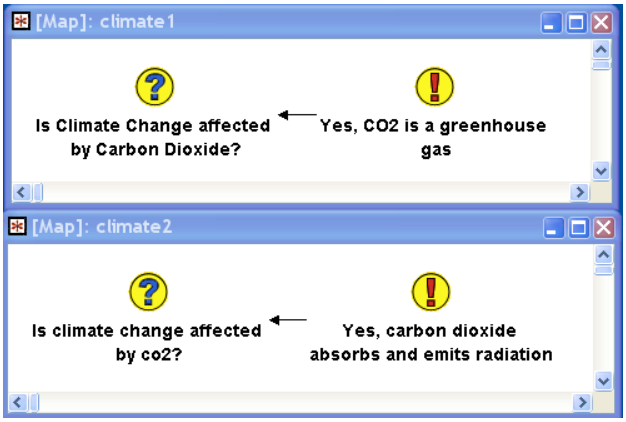

Figure 2: Two IBIS Conversations

[8] Compendium: http://compendium.open.ac.uk/

Readers will recognize "carbon dioxide" and "co2" as names for the same gas. When those two conversations are imported into a topic map, the merge agent will ask if the two conversations are *about* the same subject. In this example, the subject is a question related to a causal factor in climate change. We see two subjects entailed in that question: *climate change*, and *carbon dioxide/co2*. Determination of subject sameness in this example relies on a synonym test.

Our definition of conversation sameness centers on the context of the conversation. In the example, the context resides in a question, and that question, in both cases, is the same. Figure 3 illustrates a merged conversation.

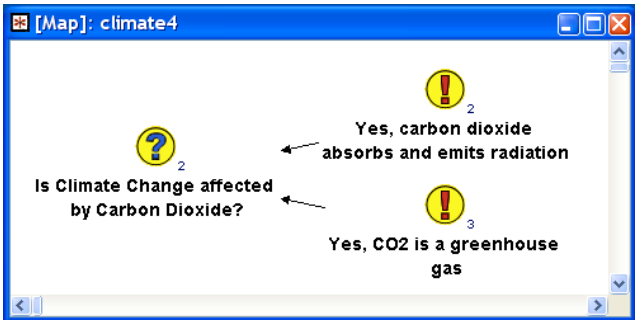

Figure 3. A Merged IBIS Conversation

For conversations of greater depth than illustrated, each node within the merged result is then treated as a merge subject and compared to its *siblings*. That is, each answer to each question is compared against other answers to the same question.

While synonym detection is a trivial lookup in a topic map, conversation merging remains a complex process. As an example, sentence structure is illustrated, again, by two sentences that ask the same question:

- `How does carbon dioxide affect climate?`
- `How is climate affected by carbon dioxide?`

Greater complexity entails technologies such as natural language processing (NLP) and emerging forms of *machine reading* (Etzioni, et al, 2007). Our research seeks to fabricate a platform with which to explore the territory of machine reading to support merge decisions in conversation federation.

Merges, once performed, can be contested. Merge decisions must be transparent—merge rules that suggest a merge must inject their reasons into the merge process for later inspection. At the same time, merges should be capable of being reversed. We turn now to a look at merge technologies, one representative of traditional topic maps processes, and a new platform that answers questions similar to ours.

2.0 Related Work

In this section, we examine an instance of *algorithmic* merge detection and execution. We then examine a project that recognizes an important aspect of our desiderata, and implements a solution that is remarkably similar to the work we describe here.

Consider the merge determination algorithm from Ontopia's open source topic map platform[9], which we have chosen to examine since the platform is at once a popular and reliable instance of a commercial topic map product. In the following analysis, we examine two objects in Ontopia's XML topic map platform; the two objects examined are considered the equivalent of property objects defined by the TMRM document (Durusau et al., 2007). The Ontopia merge algorithm tests for overlaps between *subject identifiers* of two topics being compared, or overlaps between *item identifiers* of those two topics, or overlaps between *subject identifiers* and *item identifiers* of the two topics. Failing those tests, a further test asks if both topics *reify* the same object. Specifically, these tests are found in the Ontopia source code in this class:

`net.ontopia.topicmaps.utils.MergeUtils`

[9] Ontopia: http://code.google.com/p/ontopia/

and implemented in this method in that class:

```
public static boolean shouldMerge(TopicIF t1, TopicIF t2)
```

The method returns the value true if topic t1 is determined to be *about* the same subject as topic t2. In more detail, the algorithm relies on two particular property types: *item identifiers*, and *subject identifiers*. We shall skip the reification process since it is not represented in TMRM implementations with which we are familiar. From (Garshol & Moore, 2008), a subject identifier is indirectly defined:

> "information resource that is referred to from a topic map in an attempt to unambiguously identify the subject represented by a topic to a human being"

A subject identifier bears strong resemblance to the Internet's Uniform Resource Identifier (URI) (Berners-Lee et al., 1998). An example of a subject identifier for a particular term defined in the TMDM is this:

```
http://psi.topicmaps.org/iso13250/glossary/association-role
```

The TMDM similarly defines an item identifier:

> "locator assigned to an information item in order to allow it to be referred to"

For completeness, we borrow two more definitions from the TMDM:

- information resource: "a representation of a resource as a sequence of bytes; it could thus potentially be retrieved over a network"
- locator: "string conforming to some locator notation that references one or more information resources"

An information resource can be a URI as illustrated, or a URL—a direct address on the Internet of some web page of interest. Defined as string objects, an information resource is open to simple comparisons by the algorithm. Simple string comparisons are primary engines of subject identity comparison in the Ontopia algorithm.

When the Ontopia platform decides to merge two topics, one is chosen as a recipient, the other as a donor. There is then the equivalent of a *set union* of resources performed, where resources taken from the donor are included, without duplication, into the recipient. The donor is then removed from the map, leaving one topic that represents the sum of all resources known to that map as representations of that subject.

Aki Kivelä (2010) mirrors an aspect of topic merging reflected in our desiderata when he says:

> "After merge it is however impossible to solve where some piece of information originated from."

The Kivelä comment becomes important when precise provenance is required to create a view that separates, say, sources of information. Further, it is sometimes the case that a topic merge is later contested; the merge might have been triggered by a misinterpretation of or by false information. When such situations exist, a case is made to find an alternate solution to merge platform architectures. The Wandora platform[10] posits a layered topic map architecture (Kivelä, 2010) that is remarkably similar to the *virtual proxy* solution we describe below. Figure 4 illustrated an added proxy that contains the *subject identifiers* from two proxies that have been merged.

[10] Wandora: http://www.wandora.org/wandora/wiki/index.php?title=Main_Page

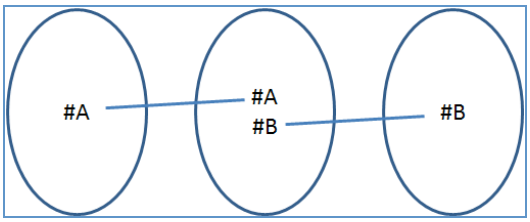

Figure 4: Layered Merged Topics—After (Kivelä, 2010)

Figure 4, as we shall see, is remarkably similar to the virtual merge approach we take, as described next.

3.0 A Research Solution

We begin this inquiry by asking this question: *Why perform set-union merges?* We acknowledge one well-known reason: the result is a single object which consumes less space in the map and which provides the equivalent of a *joined query result* when fetched from a database. That is, the merge operation co-located all resources necessary to re-construct the subject when requested. To visualize the alternative—that situation where no set-union was performed, meaning there are multiple objects representing the same subject in the database—a *join* operation must be performed either inside the database or following multiple queries to gather all related resources. That is, indeed, an important consideration.

We borrow and adapt a concept from the Ted Nelson *Xanadu* play book, his *virtual file* architecture (Nelson, 1999). The virtual file concept entails a large body of text created in a persistent way, and a *file* that is created as a document that contains a list of *links* into the large body of text (Figure 5). If some text is to be modified, new text is created at the end of the large body, and appropriate pointers in the virtual file are adjusted.

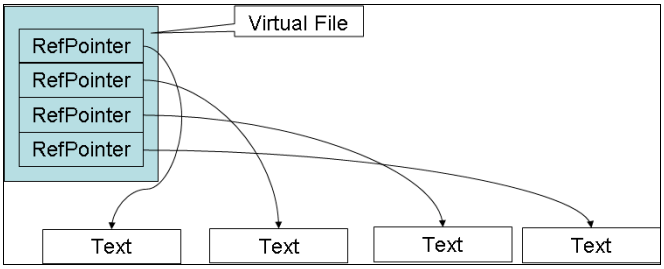

Figure 5: Virtual File System after (Nelson, 1999)

In the simplest expression of Nelson's virtual file concept, we implement a *virtual proxy* that serves as a *binding point* for all merged proxies (Figure 6) for a given subject. In this implementation, one creates *merge assertions* (associations) that specify the nature of the merge. For instance, one proxy is designated the *original proxy* assertion and another is designated a *merged proxy* assertion; in each case, the *justifications* for the merge are presented in the merge assertions—rounded rectangles that connect each subject proxy to the virtual proxy. Since each merge assertion is, itself, a subject, each merge is thus a candidate for social intervention in a contested domain; each merge assertion is *contestable*. The virtual proxy itself gains a set union of *subject identity* properties from each merged proxy. The virtual proxy becomes the core target for queries that seek subjects. When timestamps are included in the merge assertions, one gains a *temporal* view of the history of a map-mediated subject.

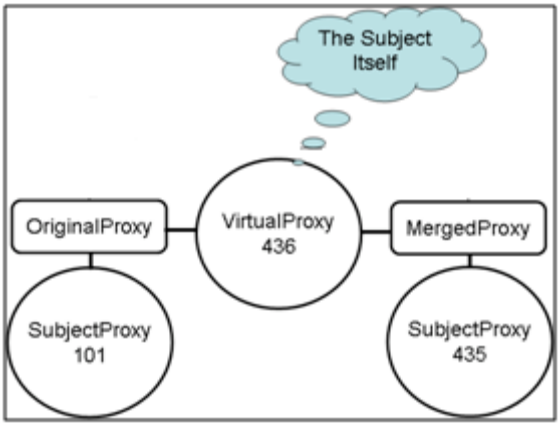

Figure 6: Virtual Merge Graph

A virtual proxy allows us to maintain the separate identities of different representations of the same subject. In the case of merging representations of users that come into a federation with different object identities, we allow the separate identities to remain to facilitate continued actions of algorithmic processes. Views requested on those participants are joined together as needed.

In the scenario where contested merges call for reversing a particular merge, the merge assertion can be unlinked to release the contested proxy from its merge. Depending upon federation requirements, it may be reasonable to re-link that merge assertion into a subject that federates contested merges for purposes of historical analysis.

Performance aspects of information retrieval are in play when one plans for a federation infrastructure that entails merging resources. Specific to our interest is the cost of *joins* associated with production of views. In the set-union merge as described for the Ontopia algorithm, the join occurs during the merge process—the join is performed one time only. In the case of a virtual merge algorithm, joins are expected to occur during view production, on demand, with potential impact on system performance. At the implementation level, it is reasonable to consider performing a join just once and sequestering it in the virtual proxy. If a proxy merge is reversed, the join will need to be re-performed at that time.

4.0 Concluding Remarks

We believe that our virtual merge architecture enables the federation of complex topics, those associated with human conversation. The enabling factors include maintenance of provenance and the ability to reverse a merge already performed. By coupling subject proxies with merge assertions that are, themselves, subjects, we promote full transparency of merge operations and facilitate debates related to each merge.

The ability to request views *by author*, that is, to view a conversation while excluding certain participants, provides greater flexibility in user control of views. A reputation and trust system added to a federation can provide metrics by which individuals can be excluded from views by setting thresholds in a view request.

The Bloomer platform is now at the full prototype stage, available for download as an open source project. We are beginning installations in several international settings that will provide us with early experience and feedback with which to improve the system and validate our virtual merge architecture.

5.0 References

Berners-Lee, T., R. Fielding, and L. Masinter (1998). “Uniform Resource Identifiers (URI): Generic Syntax”. Network Working Group: Request for Comments 2396. Online at http://www.ietf.org/rfc/rfc2396.txt

Bowker, Geoffrey, and Susan Leigh Star. (1999). *Sorting things out: classification and its consequences*. Cambridge, MA: MIT Press.

Conklin, Jeff, Albert Selvin, Simon Buckingham Shum, and Maarten Sierhuis (2003). "Facilitated Hypertext for Collective Sensemaking:15 Years on from gIBIS". Keynote Address*: Proceedings LAP'03: 8th International Working Conference on the*

Language-Action Perspective on Communication Modelling, H. Weigand, G. Goldkuhl and A. de Moor (Eds.) Tilburg, The Netherlands 1-2 July 2003

Conklin, Jeff (2005). *Dialogue Mapping: Building Shared Understanding of Wicked Problems*. Wiley.

Durusau, Patrick, Steve Newcomb, and Robert Barta (Editors) (2007). "Topic Maps Reference Model, 13250-5." Online at http://www.isotopicmaps.org/TMRM/TMRM-7.0/tmrm7.pdf

Etzioni, Oren, Michele Banko, and Michael J. Cafarella (2007). "Machine Reading". Proceedings of the 2007 AAAI Spring Symposium on Machine Reading

Garshol, Lars Marius, and Graham Moore (Editors) (2008). "Topic Maps—Data Model." Online at http://www.isotopicmaps.org/sam/sam-model/

Kivelä, Aki (2010). "Introduction to Layered Topic Maps". Online documentation for the open source Wandora topic map platform. Online at http://www.wandora.org/wandora/wiki/index.php?title=Introduction_to_Layered_Topic_Maps

Nelson, Theodor Holm, (1999). "Xanalogocal structure, needed now more than ever: parallel documents, deep links to content, deep versioning, and deep re-use". *ACM Computing Surveys*, Volume 31, Issue 4es, December, 1999.

Park, Jack (2008). "Knowledge Gardening as Knowledge Federation". In *Proceedings Knowledge Federation 2008: First International Workshop on Knowledge Federation*, Dubrovnik, Croatia. October 20-22, 2008.

Park, Jack (2010). "Boundary Infrastructures for IBIS Federation: Design Rationale, Implementation, and Evaluation". PhD Thesis Proposal kmi-10-01. Knowledge Media Institute, The Open University, Milton Keynes, UK. Online at http://kmi.open.ac.uk/publications/techreport/kmi-10-01

Pepper, Steve, and Graham Moore (Editors) (2001). "XML Topic Maps (XTM) 1.0." Online at http://www.topicmaps.org/xtm/

Rittel, H., and M. Webber (1973) "Dilemmas in a General Theory of Planning". *Policy Sciences*, Vol. 4, 155-169, Elsevier Scientific Publishing Company, Inc., Amsterdam.

Star, Susan Leigh. (1989). "The Structure of Ill-Structured Solutions: Heterogeneous Problem-Solving, Boundary Objects and Distributed Artificial Intelligence". In M. Huhns and L. Gasser, eds. *Distributed Artificial Intelligence 2*. San Mateo, CA: Morgan Kaufmann Publishers. 37-54