Preemptive Reification

Steven R. Newcomb

Coolheads Consulting
1527 Northaven Drive
Allen, Texas 75002 USA
srn@coolheads.com
http://www.coolheads.com

Abstract. It is useful to express the location of a node in a semantic graph in terms of a sequence of knowledge-bearing arcs that lead to the node from a node that is used as a point of reference. If the arcs on which such "graph-based addressing expressions" depend disappear, the expressions become invalid.

When an arc is reified as a node, the reified arc may or may not be removed. If it is not removed, the semantic graph may exhibit unnecessary complexity, ambiguity, and lack of parsimony. Alternatively, if the original arc is removed, the value of dependent graph-based addressing expressions may be lost.

For systems intended to support collaborative knowledge aggregation, such as the Semantic Web, one way to resolve this dilemma is to disallow "lazy" reification, and, in effect, preemptively reify everything that may, at any future time, require reification. Preemptively reified nodes can be "virtual" until they are actually needed, thus effectively maintaining parsimony.

The Reference Model of the Topic Maps paradigm, now under development in the ISO, shows how the need for "lazy" reification can be avoided by means of a fixed set of arc types and rules used to represent fully elaborated assertions.

1 Introduction

A demonstration of a certain critical difference between Topic Maps and RDF was Michael Sperberg-McQueen's wrap-up keynote at the Extreme Markup Languages Conference (`www.extrememarkup.com`) in August, 2001.

Sperberg-McQueen used colored ribbons to represent arcs, and several volunteers to represent nodes. He began by using a blue ribbon to represent the statement that "Tom buttered the bread".

```
Tom -------> the bread
      (blue
      ribbon)
```

"Now," Sperberg-McQueen said, "what if I want to say that Tom buttered the bread with a knife, on Friday? In order to attach the knife to this statement, I need a node for the knife, a node for 'on Friday', and I also need a node to represent the buttering itself. (There must be some sort of a 'buttering event' going on here.)"

```
                the buttering event
                 /\
                /  \
               /    \
              /      \
             /        \
            /          \
          Tom -------> the bread
               (blue
               ribbon)
```

Now, with the buttering event in existence, it was possible to use a ribbon to connect it to the knife, and yet another ribbon to connect it to "on Friday".

```
knife --------- the buttering event
              /  /\
on Friday ---+  /  \
               /    \
              /      \
             /        \
            /          \
          Tom -------> the bread
               (blue
               ribbon)
```

Once "the buttering event" existed as a distinct node, it was no trouble at all to say anything about that event. However, *before* there was a buttering event, there was no way to say anything about it.

Once the "buttering event" existed, there was no further need for the original blue ribbon connecting Tom to the bread. In the interests of parsimony and the avoidance of ambiguity, the blue ribbon should disappear:

```
knife --------- the buttering event
              /  /\
on Friday ---+  /  \
               /    \
              /      \
             /        \
            /          \
          Tom          the bread
```

In Topic Maps, unlike RDF, there is no way to say "Tom buttered the bread" without creating an explicit "buttering event" – a "buttering association" between Tom and the bread. Instead of making a direct connection between Tom and the bread, the Topic Maps paradigm requires the creation of a "buttering event" node. In other words, what in RDF might have been an arc that might later be reified as a node only if and when necessary ("lazy reification"), in a Topic Map is "preemptively reified."

The advantage of preemptive reification is that we can always say something new about anything that already exists, without having to choose between the two evils of creating redundant connections, on the one hand, or invalidating existing lore about how things are connected together, on the other. If someone wants to say something about Tom's buttering of the bread, there is guaranteed to be something to which those remarks can be attached.

2 The Depth of Preemptive Reification

If we accept the idea of forbidding lazy reification, and demanding some degree of preemptive reification, we must also accept that there is some level of recursion below which reification – or rather "reification *in situ*[1]" – is forbidden.

Is it advisable, or even practical, to forbid *in situ* reification at a specific level of recursion? The Topic Maps Reference Model answers this question affirmatively. It uses the yardstick of "substantiveness with respect to the semantics of the assertion" to justify its decisions as to the level at which further *in situ* reification is forbidden. This doctrine of substantiveness is intuitively satisfactory when it is applied in such a way that, for each remaining "unreifiable" arc, there is no reason to talk about it other than to discuss the Reference Model itself. In other words, in the Topic Maps Reference Model, *in situ* reification does not occur when its only conceivable purpose would be to allow assertions to be made about the subterranean mechanics of the graphic representation of an assertion.

Before the various components of an assertion can be discussed, it's necessary to introduce a few of the basic concepts of Topic Maps.

At the Reference Model level, a topic map consists of nodes that reify **subjects** ("subjects of conversation" – specific semantics, ideas, concepts, etc.), and arcs. There are exactly four arc types.[2] Some subjects are **assertions**. An assertion is reified by a node that serves as certain specific ends of a specific set of arc types. Assertions are strongly-typed relationships between subjects; the

[1] In Topic Maps, literally anything – any subject of any kind, without exception – can be reified as a node. Even an arc that is non-reifiable according to the Topic Maps Reference Model is reifiable. The kind of reification that is forbidden by the Topic Maps Reference Model, below a certain "floor" level, is a specialized kind of reification, *in situ* reification, in which the reifying node literally replaces what would otherwise be an arc in exactly the same place (*"in situ"*) in the graph. In other words, an *in situ* reifying node will always necessarily be encountered in the course of traversing the path described by the arc that has been replaced by a node and two new arcs.

[2] Since the "arcs" used to describe the Reference Model are all bidirectional, eight distinct arc types are required in order to describe the Reference Model in RDF terms.

subjects related by an assertion are called "**role players**" in the assertion. Each played **role** is also a subject. An assertion can optionally be an instance of an **assertion pattern**, which is, like everything else in Topic Maps, a subject. For each role defined by an assertion pattern, there can optionally be a set of **role player constraints**. In addition, an assertion has one or more **scope**s, each of which is a set of subjects; each scope is a subject. (Scopes are used as a kind of shorthand, to qualify the semantics of an assertion with a level of specificity that would be burdensome to handle by means of traditional subtyping.)

In a topic map, there is a one-to-one correspondence between subjects and nodes; every node is the surrogate for exactly one unique subject. At the usual, application-defined level at which "statements" are made in RDF, "assertions" are made in Topic Maps. However, because of preemptive reification, scoping, the more detailed nature of assertions, and the fact that assertions are n-ary and n-directional, Topic Maps "assertions" are not exactly the same as RDF "statements". Several RDF statements are required in order to represent a single Topic Maps assertion. The translation of an RDF statement into a Topic Maps assertion is an up-translation.

2.1 Assertions and Role Players

It is beyond the scope of this position paper to survey the Topic Maps Reference Model,[3] but a description of one aspect of it should be sufficient to illustrate how the yardstick of "substantiveness with respect to the semantics of the assertion" can be used to determine where the "floor" of reification is.

The following elaboration of Sperberg-McQueen's example shows the two roles involved in a "buttering event": the "butterer" role and the "buttered" role.

```
          the buttering event
           /\
          /  \
         /    \
(role:  /      \ (role:
butterer)     buttered thing)
      /          \
    Tom         the bread
```

We may wish to make assertions about all of the following substantive aspects of the buttering event:

- The buttering event assertion (node A in the diagram below),
- the role of "butterer" in any buttering event assertion (node R1), and the role of "buttered thing" (R2),
- Tom himself (X1) and the bread itself (X2), and

[3] ...and, in any case, the Topic Maps Reference Model is a work in progress.

– the fact that Tom plays the butterer role in this assertion (C1),[4] and the fact that the bread plays the buttered role (C2).

```
                        Tom  (X1)
                             /
 the fact that Tom          /
 plays the butterer role (C1)
                          /\
                         /  \
                        /   (R1) role: butterer
                       /
 the buttering event (A)
                       \
                        \   (R2) role: buttered
                         \   /         thing
the fact that the bread   \ /
plays the buttered role   (C2)
                            \
                             \
                  the bread  (X2)
```

None of the remaining arcs can conceivably be interesting *from the perspective of the substance of the buttering event assertion.* It is hard to imagine why anyone who is more interested in the buttering event than in the underlying mechanics of assertions in topic maps would ever want to make assertions about:

arc A-C1 The aspect of this assertion that the aspect of this assertion that Tom plays the butterer role is one of the casting aspects of this assertion.
arc C1-X1 The aspect of this assertion that it is Tom who plays the butterer role in the aspect of this assertion that Tom plays the butterer role.
arc C1-R1 The aspect of this assertion that it is the "butterer" role that is the role aspect of the aspect of this assertion that Tom plays the butterer role.

3 Parsimony

The Semantic Web's ability to protect the value of investments made in graph-based addressing expressions may be critically important, but so is parsimony. Fortunately, if there are no explicit assertions about them, it won't be necessary for Topic Maps systems to physically instantiate every preemptively reified node, as long as such systems behave in the same way that they would have behaved if such nodes actually existed. The important thing is to agree that these preemptively reified nodes *virtually* exist, even when nobody has yet made assertions about them.

[4] The letter *C* here is a mnemonic for the idea of *casting*, a metaphor drawn from the jargon of theatrical productions. A *C* node always represents the fact that a specific role player has been cast in a specific role in a specific assertion.