

Better Living with xlinkit

Christian Nentwich, Wolfgang Emmerich and Anthony Finkelstein

Department of Computer Science

University College London, London WC1E 6BT

{c.nentwich,w.emmerich,a.finkelstein}@cs.ucl.ac.uk

1 Background

XML has become popular in software engineering as an exchange format that can bridge heterogeneity and distribution problems. Many specification languages are now available in XML, as exemplified by the XMI standard [5], which can be used to store UML models.

We have worked extensively in the area of consistency checking of distributed XML documents [3, 4, 2, 1] and have produced a formal language for specifying consistency rules between distributed documents. We have also devised a method for evaluating this language in order to obtain a set of hyperlinks. These *consistency links* show which elements in documents are consistent or inconsistent with respect to a rule.

```
<globalset id="$classes"
  xpath="//Foundation.Core.Class[@xmi.id]"/>
<globalset id="$stateschemas"
  xpath="/z/schemadef[@purpose='state']"/>

<consistencyrule id="r1">
  <description>
    Every class in the UML model must have a
    state schema in a Z specification
  </description>

  <forall var="c" in="$classes">
    <exists var="s" in="$stateschemas">
      <equal op1="$c/ModelElement.name/text()"
        op2="normalize-space($s/text()[1])"/>
    </exists>
  </forall>
</consistencyrule>
```

Figure 1: Example rule in XML

Figure 1 shows an example rule, written in the XML encoding of our language, that specifies a constraint between UML documents and a Z schema. The language is basically first order logic, with the restriction that the sets it works on are finite - because they are taken from documents - and no functions are allowed. Our checker can now be used to apply this rule to a UML model and a Z specification. For each schema that matches a UML class, a hyperlink is generated between the pair. If some

UML class has no schema, an inconsistent link is generated. We have demonstrated elsewhere [3] that our language is powerful enough to express the constraints of the UML Foundation/Core package.

We have implemented our technology as a light-weight, web based consistency checking service on which we intend to build our future work. This service, together with interactive examples, can be used freely at <http://www.xlinkit.com>.

2 Research outline

The biggest problem to be tackled in the near future is that we have to check the global consistency of the whole set of documents every time, i.e. check all rules against all documents. An *incremental* consistency checking scheme would be preferable and we have already made some inroads in this area.

We have implemented an algorithm which computes the difference between two XML documents. Given an old and a new version of a document it is thus possible to identify the changes in their tree structure. Since our consistency rules make reference to the elements they are checking using XPath, we can exploit this information to compute an “intersection” between the changes and the element selection. This intersection will tell us which rules have to be rechecked, thus minimising the number of rules that have to be included in a check. Early results are encouraging, suggesting a speedup of a factor of at least 5 over checking the whole set of rules.

It is worthwhile investigating if some benefit can be derived from analysing consistency rules for overlap. For example, one rule may be testing a subset of the properties of another rule or properties may be inferred from rules. In that case, it may be possible to speed up processing by reusing the output of some rules to speed up others.

At the moment, our tool generates a *consistent link* be-

tween a set of elements if these elements conform to a rule and an *inconsistent link* if they violate a rule. It may be possible to find the *degree* of inconsistency of a set of elements by picking out the subset that causes a formula to fail.

Consider the formula $a \wedge b \wedge c$. Suppose a and b are true and c is false. If this formula was consistency rule, the elements assigned to a , b and c would be linked together as inconsistent since this combination makes the formula false. The user of the consistency checker would then examine the elements and most likely spot that c causes the problem. We are investigating whether it is possible to specify the consistency status of individual elements rather than the combination of all three. In the example, we would like to see the following information: a is consistent, b is consistent and c is inconsistent. We will have to solve the problem of how we can generalise such a system to work for arbitrary first order logic expressions.

Another area that we are interested is *resolution* of inconsistency given hyperlinks as diagnostic information. While it is not desirable and sometimes impossible to enforce global consistency, it may be possible to classify inconsistencies in order to determine a mode of action: some inconsistencies are trivial and can be removed immediately and possible algorithmically. Other inconsistencies may be related to fundamental disagreements or differences in understanding and may require human intervention. The author of the consistency rules may be aware which rules check simple properties and which implement vital business decisions, and thus may be able to help by providing annotations to support resolution.

The consistency links we generate are not semantically meaningful when considered on their own. They do contain references to the rule that generated them, but no reference to who executed the consistency check and no description of their overall characteristics. As a consequence, a set of links that is meaningful to one developer may not be meaningful to others at best and confusing at worst. We will try to identify suitable metadata to attach to linkbases as a descriptive mechanism to alleviate these problems.

We are particularly interested in integrating our approach with a workflow model. Such a model would decide the points in time at which consistency rules have to hold and allocate responsibility for resolution, effectively using the rules as postconditions for actions and using the workflow to drive consistency checking. Looking at the problem from a different point of view, perhaps we can use the consistency status of the system to drive the workflow and make it more adaptive.

For example, being aware of a large number of inconsistencies may influence the decision of managers to move between phases in an iterative lifecycle. In other words, it may be possible to use the level of global consistency as a *metric*.

3 Summary

We have a working system with a formal basis, xlinkit, for checking the consistency of distributed documents. xlinkit has evolved in several stages and been continuously improved with respect to expressiveness, scalability and performance. We intend to use it experimentally to explore several routes in consistency checking, including incremental checks, refinement of diagnostic information, workflow integration and resolution of inconsistency.

We are particularly keen to see wide scale use of our technology and to obtain feedback so as to improve it. xlinkit is available for evaluation and research and can be used as a plugin consistency checker.

References

- [1] E. Ellmer, W. Emmerich, A. Finkelstein, D. Smolko, and A. Zisman. Consistency Management of Distributed Documents using XML and Related Technologies. Research Note 99-94, University College London, Dept. of Computer Science, 1999. Submitted for Publication.
- [2] C. Nentwich, L. Capra, W. Emmerich, and A. Finkelstein. xlinkit: a Consistency Checking and Smart Link Generation Service. Research Note RN/00/66, University College London, Dept. of Computer Science, 2000. Submitted for Publication.
- [3] C. Nentwich, W. Emmerich, and A. Finkelstein. Checking Distributed Software Engineering Content. Research Note RN/01/10, University College London, Dept. of Computer Science, 2001. Submitted for Publication.
- [4] C. Nentwich, W. Emmerich, and A. Finkelstein. xlinkit: links that make sense. Research Note RN/01/06, University College London, Dept. of Computer Science, 2001. Submitted for Publication.
- [5] Object Management Group, 492 Old Connecticut Path, Framingham, MA 01701, USA. *XML Metadata Interchange (XMI) Specification 1.1*, November 2000.