An XML Based Approach to Support the Evolution of Model-to-Model Traceability Links

Jonathan I. Maletic, Michael L. Collard, Bonita Simoes
Software Development Laboratory
Department of Computer Science
Kent State University
Traceability

\[ 2^M \times 2^K \] – number of possible links
Concrete Example
Why is Traceability Difficult?

• Traceability is lost when changes occur
  ▪ *A design document is outdated the second a line of source code is typed in and compiled*

• A single change to a model may cause a large number of changes to many other models

• Little or no support for evolution of links
Research Question

• How do you detect that the validity of a link suspect - due to evolution of the system model?

• Can we provide a mechanism to support the evolution of model-to-model traceability links?
Evolution of Links

UML Class Model
Ver 1.0

Source Code Model
Ver 1.0

<SDML> @ KSU
Our Approach

• Use XML to represent models – making them interoperable

• Use XML to represent links - XPath, XLink

• Use fine grained syntactic differencing supports link evolution
Model Representation

• Models are represented in XML without any schema restrictions

• Source code is represented as srcML, an efficient lightweight markup of the source code (other representations could be used)

• Class models can be represented in similar manner (e.g., classML, XMI)
Link Representation

- XPath forms address paths between elements of models – can easily address any location in the model

- Link meta-data can be stored as part of the link

- Links stored externally with decoupled from on model semantics
Changes to Models

• Changes can be viewed at different levels of granularity
  ▪ Physical: file, line (current technology)
  ▪ Syntactic: class, method, statement sub-statement

• Need syntactic differencing mechanisms to support evolution of links – if linking to non-physical nodes
Link Evolution

• Depends on the granularity of the link and change detection mechanism

• Link granularity refers to the XML subtree(s) affected

• Change granularity depends on the what actually changed in the model
Meta-differencing

• Further leverage our XML representation – extending it with difference information (srcDiff)
• Currently we convert diff information in conjunction with srcML resulting in srcDiff
• Queries on the differences can be made to determine exact types of syntactic structures that were changed
Scalability

• Number of possible links is quite large

• Need to translate models into XML – with both space/time issues

• Storage of links (external to the models)

• Traversal of the links – evaluation of XPath
Conclusions

• XML can be used to support efficient/easy interoperability

• Traceability is needed at different levels of granularity

• Change detection is required at the same level and type as the link to support evolution
Resources

• src2srcml & srcML DTD
  ▪ GPL
  ▪ Commercial license also - ask

• srcDiff – close to a release for a tool
  ▪ Will be GPL

• www.sdml.info