

Enhanced Traceability for the Unified Process

Patrick Maeder, Ilka Philippow and Matthias Riebisch
Software Systems/Process Informatics Group
Technical University of Ilmenau
D-98693 Ilmenau, Germany

patrick.maeder|ilka.philippow|matthias.riebisch@tu-ilmenau.de

Complex, business critical software systems have to adapt to frequently changing needs. Evolutionary development processes have been developed to enable short responses to changes. In complex settings the changes bear high risks, such as incomplete implementation, misunderstood dependencies, missing comprehension and lacking coverage. To manage these risks, the concept of traceability has been developed and introduced to most development process standards. However, we have to state that traceability is poorly used in practice, mostly limited to tracing requirements. Nevertheless, traceability links are needed to connect design and implementation elements as well. To be useful, traceability links have to be defined at a fine-grained level and maintained in a correct and complete state. Since a high number of links has to be managed, and the maintenance tasks have to be carried out manually, an extremely high effort is required. Tool support would be very helpful, but it is only possible if the development methods support the traceability links. Most design methods claim to support the concept of traceability, although their definitions of artefacts, relations and activities are too imprecise to define rules for traceability links based on them.

It is necessary to refine the definition of development methods to a level that enables the establishment of traceability links during the developer's activities. A development activity conserved in this way, has to store additionally the reasons and decisions for that activity. Furthermore, it is necessary to perform this definition for concrete development methods, which are widely used in the industrial practice. For our work we have chosen the Unified Process UP to define a process-specific model of traceability links. Although, Letelier showed in [4] the application of his meta-model for the UP, his definitions are not detailed enough to derive rules for traceability links. The UP description by its authors offers traceability as one of its features, but it is lacking a detailed description of how and between which artefacts the traceability links should be established. Additionally, works are necessary to give a unique syntactical and semantical definition of traceability links.

The contribution of our work consists in analysis and classification of all UP artefacts regarding traceability aspects.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, to republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Copyright 200X ACM X-XXXXX-XX-X/XX/XX ...\$5.00.

Based on that, all required links between the artefacts of the UP activities of requirements engineering and design are chosen out of the amount of all possible valid links. Additionally, a syntactical and semantical definition of traceability links is established, customized to the UP's methods. These steps lead us to a milestone and provide a basis for further works e.g., empirical investigations for rules concerning the suitable level of detail for traceability links, or for rules how far to follow traceability links during the impact analysis of a change.

As results of these works, guidelines for the level of detail and rules for the verification of traceability links have been established. The analysis of the UP and the customisation of the traceability link model are performed during practical development projects.

Currently, the developed traceability link model is refined towards a detailed level of coverage of the methodical activities, to facilitate appropriate tool support for the creation, update and verification of the traceability links with a minimum interaction with the developer. To extend the UP's development methods, the architectural design method Qasar [1] is integrated into them.

Other development methods and processes like Fusion [2] and Refactoring [3] are currently investigated aiming towards a generally usable traceability model. For the realisation of tool support we have started the implementation of extensions for existing UML tools. The extensions will support the developer by the establishment of traceability links in the background while modelling and by maintaining the consistency of existing links during changes of artefacts. To extend the practical application of traceability links, traceability link model is currently implemented by two major CASE-Tool vendors. Furthermore, it is integrated to project management methods and software process models.

1. REFERENCES

- [1] J. Bosch. *Design and Use of Software Architectures : Adopting and evolving a product-line approach*. Addison-Wesley, 2000.
- [2] D. Coleman. *Object-Oriented Development: The Fusion Method*. Prentice-Hall, 1994.
- [3] M. Fowler. *Refactoring: Improving the Design of Existing Code*. Addison Wesley, 1999.
- [4] P. Letelier. A framework for requirements traceability in UML-based projects. In *Proceedings of 1st International Workshop on Traceability in Emerging Forms of Software Engineering*, Edinburgh, UK, Sept. 2002.