

Unicase – a unified CASE Tool

Jonas Helming, Maximilian Koegel

Technische Universitaet Muenchen
Department for Computer Science, Chair for Applied Software Engineering
Bolzmanstraße 3, D-85748 Garching bei München, Germany
{helming, koegel}@in.tum.de

***Abstract:** Many tools in software engineering projects support the visualization and collaborative modification of a custom set of artifacts. These include tools for requirements engineering, UML tools for the design, project management tools, developer tools and many more. In typical software engineering projects the artifacts from the respective tools are stored separately in various repositories including databases, file systems or software configuration management systems. To achieve traceability between these artifacts, different tools and especially their underlying models have to be integrated. This is a non-trivial task and is especially difficult due to limited extensibility of commercial tools in particular. We propose a unified CASE tool, which offers a unified repository for software engineering projects and specific views on this model for project participants like architects, project manager or developers. Unicase is an open-source project based on the Eclipse platform.*

1 Motivation

The integration of a heterogeneous tool-landscape is a difficult endeavor for software engineering projects. The need for integration is triggered by the requirement for traceability between various artifacts. On one hand, parts of the specification shall be traceable inside of the system model to the origin requirements. On the other hand parts of the project model like persons, iterations or bug reports should be traceable to the affected part of the system model. Both kinds of traceability are essential preconditions for an effective management of change.

The difficulty of tool integration in software engineering projects is dimorphic: (1) First from a design point of view the integration of different tools is constrained by missing or not well-documented APIs and poor use of open exchange standards. (2) Second from an analytical point of view the underlying models of different tools have to be integrated. Many tools lack in extensibility and adaptability to integrate their model with other repositories.

2 Proposed Solution

This presentation introduces a unified CASE tool: “unicase”. The tool essentially consists of three parts: (1) A uniform and adaptable model for software engineering

projects, (2) a unified repository for collaborative modification of the model, (3) unified views for textual, tabular and graphical browsing and modification of the model, tailored to specific activities.

2.1 Unified Model

We propose a unified model for software engineering projects. This model explicitly connects artifacts from the system model, i.e. the model of the system under construction, with the project model, i.e. the model of the project itself. The system model contains model elements like Requirements or Use Cases but also includes the design using artifacts such as UML ([8]) classes or flow charts. The project model contains management artifacts such as tasks or iterations as well as the organizational structure. Both models define which artifact types can be linked and the unified model defines which artifact types from the different models can be linked. The former are also referred to as *intra-model* links while the latter are *inter-model* links. The model implementation is based on the Eclipse Modeling Framework and therefore easily extensible. A modification of the unified model is even possible during project run-time with existing model instances.

2.2 Unified Repository

The unified repository is a server node which is able to store and version any instance of the unified model. It supports collaboration by providing the changes between two versions of a model instance and thereby providing an update and commit operation as known from other software configuration management systems such as Subversion. The server implementation is independent of the specific model in use.

2.3 Unified Views

Unicase provides basic views to browse and modify instances of the unified model, including graphical draw panes. These views are independent from a specific model. They work generically with any instance of a unified model. As specific activities in a software engineering project require customized views, unicase provides various specific views such as a task view for the developer or status views for the project manager. Custom views can be added easily by a plug-in mechanism.

3 Conclusion

We are currently evaluating unicase in a student project with 25 participants with an industrial client – the Airport Munich. So far we identified two essential shortcomings of the approach, we are currently working on: (1) Effective use of the system requires a better integration with source code in terms of possibilities for code generation and traceability to parts of the source code. (2) Large projects generate huge models and require innovative approaches to consolidate them over time in terms of managing obsolete and duplicate artifacts.