

Extending the UML to Support Evolution Management

Robert France James M. Bieman
Computer Science Department
Colorado State University
Fort Collins, Colorado, USA
{france,bieman}@cs.colostate.edu

Ray Trask
IT Architecture
Qwest
Dublin, Ohio, USA
ray.trask@qwest.com

1. Overview

It is well-known that uncontrolled change to systems can lead to increased evolution costs as a result of deteriorating structure and compromised system qualities. In organizations with large integrated systems, *enterprise architects* are responsible for ensuring that (1) system evolution occurs in a manner consistent with changing business goals, and (2) changes in one or more systems do not impact negatively on the services provided by other systems within the organization. This often involves resolving conflicting requests for changes and making tradeoffs on the quality of services provided by the systems.

We are currently developing a model-driven, repository-based approach to system evolution management. Our primary domain of interest is the telecommunication IT system area, but the techniques are applicable to organizations with integrated systems of systems. The approach is model-driven in the sense that data and object-oriented (OO) models (expressed in the Unified Modeling Language (UML) [1]) are used to represent system implementations at various levels of abstractions (business requirements, system requirements, logical and physical design).

Our work with the UML has identified deficiencies in the UML, the most significant being the poorly defined relationships among concepts in different UML model types. To address this problem we have developed lightweight extensions to UML Use Cases and Class Diagrams to support traceability of concepts across models.

The central element in our model-based approach is a repository of interrelated system artifacts. System artifacts include existing implementation artifacts and their models, as well as models and implementations that are under development in current projects. Traceability links relate elements within and across system artifacts in the repository. Using the repository, enterprise architects and system developers can carry out the following analyses in support of system evolution activities:

- Perform impact analyses: The relationships among the artifacts in the repository (for example, data/object create, read, update, and delete relationships between applications and data/objects) can be used to determine the impact of planned changes and new features on existing systems and other ongoing projects.
- Perform gap analyses: The repository can be used to determine what parts of a system are under development or already exists, and what parts need to be obtained from outside vendors or be built in-house. From an evolution perspective, gap analysis can yield insights that can determine the strategy for incorporating a change.
- Perform redundancy analyses: The repository can be used to determine whether proposals for new system features can already be met by existing systems and to determine wasteful overlaps in system functionality.

References

- [1] T. O. M. G. (OMG). Unified Modeling Language. Version 1.3, OMG, <http://www.omg.org>, June 1999.