

Automatic Generation of Rule-based Software Configuration Management Systems

Guozheng Ge, E. James Whitehead, Jr. (Advisor)

Dept. of Computer Science

University of California, Santa Cruz

{guozheng, ejw}@cs.ucsc.edu

ABSTRACT

We propose a model-driven methodology and toolset for automatic SCM system repository creation and feature composition using code generation and rule engine technologies.

Categories and Subject Descriptors

D.2.1 [Requirements/Specifications]: Elicitation Methods – *rapid prototyping*; I.2.2 [Automatic Programming]: Program Synthesis.

General Terms

Design, Languages, Theory

1. RESEARCH OPPORTUNITY

Software configuration Management (SCM) systems [1,2] provide control over the evolution of complex software systems. Most existing SCM systems are built as monolithic systems using proprietary data models, unique system architectures and non-reusable components. SCM customization and evolution are both difficult and expensive. Many users lack standard SCM domain knowledge and modeling tools to compare features and capabilities of various SCM systems on a common ground. Few rapid development solutions are available for SCM feature automation and customization. Once an SCM system gets deployed, it is difficult and expensive to change its feature configuration and redefine operational semantics when user requirements change over time. It is an important and practical research problem to develop solutions for automated SCM system generation in which users design data models and semantic models to drive a code generator and automatically produce an SCM system that implements these model specifications. When customers change model specifications to reflect new requirements, an updated SCM system can be quickly generated again.

2. APPROACH OVERVIEW

Containment Modeling Framework (CMF) [3] has been successfully applied to define and compare repository models for more than twenty Content Management and SCM systems with a standard set of modeling primitives. This opens the possibility to automatically generate SCM repositories based on their CMF descriptions. We have also designed a Semantic Modeling Framework (SMF) to specify SCM features and operations on top of the generated repository. With the semantic model specifications, we can also automatically generate these feature implementations. Knitting CMF and SMF together, a prototype SCM system generator called Bamboo has been

developed and it can generate a replica of Revision Control System (RCS) now. Other advanced features, such as advanced version control, workspace management, distributed collaboration, process support, accounting and auditing etc., will be identified and integrated into SMF after a detailed SCM domain analysis process using FODA [4].

Various SCM features and operations are encoded as rules using special rule languages [5,6]. The rules are centrally stored and managed in a rule repository. A standalone rule engine queries the rule repository and executes matching rules to provide SCM services. Centralized management of SCM logic makes dynamic change and customization of SCM features possible and it simplifies the code generator design because rules can have recurring structures and patterns.

Combining the CMF, SMF and rule engine technologies together, we propose an automatic generator for rule-based SCM systems, in which we generate the repository layer based on containment models and the SCM application logic layer based on semantic models. The generator translates the models into executable Java code as well as rule files. Java code implements repository initialization, repository API, SCM utilities services, glue code for business rules, and a demo command line client. The business rules are executed in the rule engine to provide SCM services. When users change SCM requirements, they redesign the repository and semantic models (with assistance from a visual modeling tool that we will create) and an updated SCM system can be quickly generated according to the new model specifications.

3. REFERENCES

- [1] J. Estublier, D. Leblang, G. Clemm, R. Conradi, A. v. d. Hoek, W. Tichy, and D. Wilborg-Weber, "Impact of the Research Community on the Field of Software Configuration Management." *Software Engineering Notes*, vol. 27, pp. 31-39, September 2002.
- [2] S. Dart, "Concepts in Configuration Management Systems." *Proc SCM-3*, Trondheim, Norway, 1991.
- [3] E. J. Whitehead, Jr. and D. Gordon, "Uniform Comparison of Configuration Management Data Models." *Proc SCM-11*, Portland, Oregon, 2003.
- [4] K. C. Kang, S. G. Cohen, J. A. Hess, N. W. E., and A. S. Peterson, "Feature-Oriented Domain Analysis (FODA) Feasibility Study (CMU/SEI-90-TR-21, ADA 235785)." the Software Engineering Institute, Carnegie Mellon University, Pittsburgh, PA, 1990.
- [5] J. Widom and S. Ceri, "Active Database Systems", Morgan Kaufmann, 1996.
- [6] "Defining Business Rules ~ What Are They Really?" the Business Rules Group, July, 2000, http://www.businessrulesgroup.org/first_paper/BRG-whatBR_3ed.pdf.