Chapter XV

DPSSEE: A Distributed Proactive Semantic Software Engineering Environment

Donghua Deng, University of California, Irvine, USA
Phillip C.-Y. Sheu, University of California, Irvine, USA

Abstract

This chapter presents a distributed proactive semantic software engineering environment (DPSSEE) that incorporates logic rules into a software development process to capture the semantics from all levels of the software life cycle. It introduces the syntax and semantics of the Semantic Description Language (SDL). It also introduces model-based testing and maintenance. Several working scenarios are discussed to illustrate the use of semantic rules for workflow control, design consistency checking, testing, and maintenance.

Introduction

Huge amounts of information can be derived in a software development process. Such information includes programming objects (e.g., projects, modules, functions, classes, variables, statements, etc.), their relationships and the experience knowledge that can be derived. Appropriate management and utilization of this information can guide the developers, speed up a software development process, and allow the developers to incorporate personalized logic rules to control the environment.

Copyright © 2007, Idea Group Inc. Copying or distributing in print or electronic forms without written permission of Idea Group Inc. is prohibited.
In this chapter, we describe a distributed proactive semantic software engineering environment (DPSSEE) that incorporates a set of advanced knowledge representation schemes and knowledge-based function modules that can capture and utilize the semantics of a software development process.

Compared with other disciplines, the nature of information for software development is much less observational. In DPSSEE, we aim to investigate such nature in a rational and empirical way. We formulate an appropriate model for a software development environment by extracting objects from a set of function modules (e.g., design tools, version control tools, project management tools, etc.), so that the model can then be used to support more systematic, complete, and efficient project management and software analysis, implementation, testing, and maintenance.

In addition, a user can assert more semantics based on experiences, observations, and so forth. Such additional semantics is expressed in terms of constraint rules and proactive rules. They can be used to strengthen the functionality of each function module, and make a software development environment more efficient and intelligent.

The organization of this chapter is as follows. The Background section reviews the work related to knowledge-based software engineering at different levels. The System Architecture section illustrates the overall architecture of DPSSEE. The Semantic of Software Engineering section introduces the syntax and semantics of the Semantics Description Language (SDL). The Semantic Software Development Model (SSDM) section illustrates the structure, annotation, and construction of the SSDM. The Semantic Project Management Model (SPMM) section describes the construction of the SPMM. The Model-Based Testing and Maintenance section presents an approach to model-based testing and maintenance based on the semantics provided by SSDM. The Rule Generation and Applications section discusses the syntax and applications of proactive and constraint rules. Finally, the chapter ends with the Conclusions section.
ISW: Interest-Based Small World Network of P2P

[www.igi-global.com/article/isw/88925?camid=4v1a](www.igi-global.com/article/isw/88925?camid=4v1a)