Abstract

Nowadays, software modeling is an essential part of enterprise software projects. The Unified Modeling Language (UML) is the most widely employed software modeling language. Through UML's various diagrams, designers can effectively get a visual way to specify requirements and design architecture.

The test-driven development method is a lightweight software implementation process, which originates from the Extreme Programming (XP) paradigm. As one of the essential features of XP, the test-driven development method intends to let developers produce simple designs, which can be immediately converted into run-able unit-tests, and these unit-tests will guide code-writing. Many successful XP practices present the test-driven development as an efficient way to prevent creation of bugs in the implementation stage. For this reason, the test-driven development will improve the quality of software systems. However, while developing software that includes complicated logical or mathematic contents, it is necessary to correctly and completely model the system before starting the implementation. Because the test-driven development method includes simple design only, it cannot be solely employed in such contexts.

The goals of this project are to use UML and test-driven development methods to develop an elevator simulator system and to design a software process which combines UML and test-driven development. This thesis depicts essential concepts of UML, test-driven development, and elevator control technologies. Through presenting development process of this elevator simulator system, details of design with UML and implementation with test-driven development are depicted. Several experiences to effective combination of UML and test-driven development are also considered, as well as efficient design of unit-tests.

Keywords: UML, test-driven development, elevator simulator system, group-control, unit-test.