

Aspect Tools for orchideo

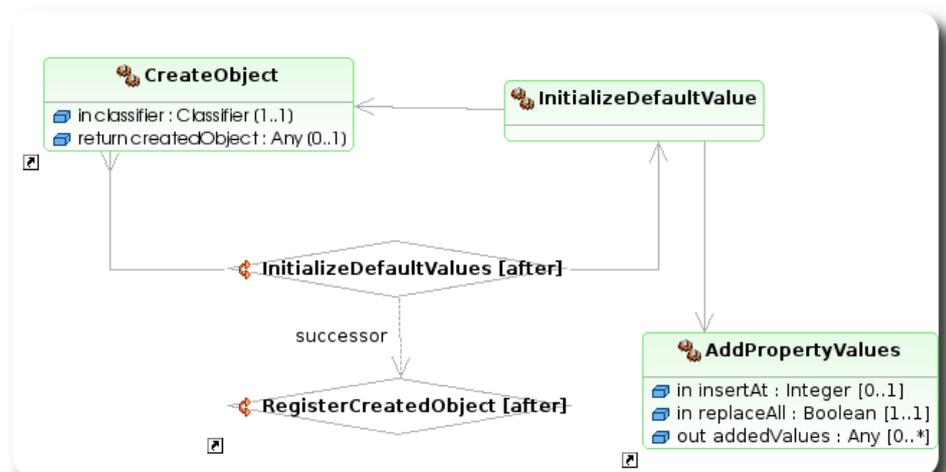
Bachelor Project Proposal, WS 2009/2010 – SS 2010
Software Architecture Group, Prof. Dr. Robert Hirschfeld

Project Description

The orchideo|engine is a model-driven and aspect-oriented runtime environment that allows for a modular construction of arbitrary runtime systems. Each individual feature of a particular runtime system is formally specified as an aspect that declares its capabilities and cross-cutting traits. The orchideo|engine is written in Java, as are the aspects' implementations. For business applications, typical aspects would, for example, cover persistence, relation management, and business constraint validation. These and other aspects are implemented in the orchideo|objects framework, which has been successfully used to develop complex business applications.

Aspect-orientation is still a relatively new programming paradigm. Developers used to working with object-oriented systems often have problems grasping the concepts of aspects. For one, it is

hard for them to identify cross-cutting concerns and formalize them as aspects. Moreover, even if a set of aspects is already given, it is usually not obvious which aspects are actually woven into a particular code fragment. Questions like “How do I know what exactly happens if I call this or that method?” are very common. Another typical issue is error handling. A seemingly simple statement in a developer's code may lead to a cascade of aspects being woven into the flow of execution. If an error occurs in any of these aspects, it is not always easy for a developer to infer the actual cause of the error by reading the message or stack trace. This problem is even more severe in the orchideo|engine, since aspects are not woven directly into the Java bytecode, but instead handled at runtime by an interceptor pattern. A Java stack trace thus does not show the full extent of a program's execution flow. Closely related to this problem is the awkwardness of debugging aspect-oriented software systems.

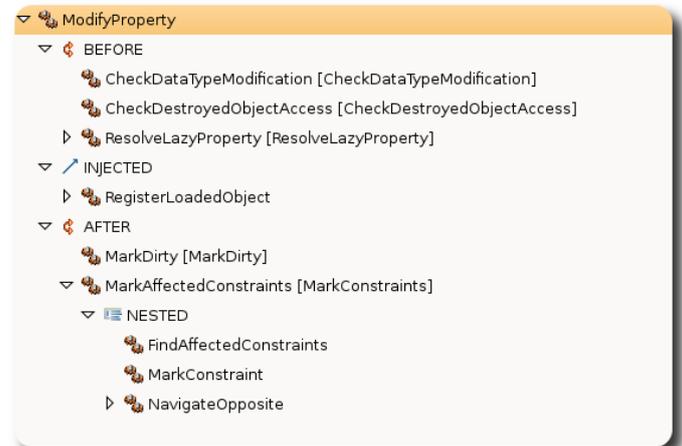


The Project

From the problems described above, it is obvious that a lack of appropriate and easy-to-use tools for developers currently hinders the adoption of aspect-orientation. The goal of the project is to explore directions to improve the aspect-tooling for the orchideo|engine. One major area of work will be impact analysis. We plan to provide an easy and intuitive way for a developer to determine the impact of a certain action within a given configuration. It should be easy to find out which actions are advised under certain circumstances. This information is to be provided in an intuitive and visually appealing user interface.

The second area to focus on will be debugging and error handling. Tools should be provided to simplify the analysis of long and complicated error dumps. At runtime, heuristics may be capable of detecting repetitive error dumps and provide only compacted information for subsequent dumps. In the debugger, we may aid developers by showing the execution flow through the orchideo|engine in addition to the regular Java stack.

The implementation language in this project is Java. Tools for the orchideo|engine are realized as Eclipse plug-ins. All extensions and new tools delivered in this project shall fit in this architecture.



Organization

A group of six to eight (6-8) students may participate in the project. Organization and tasks will be determined by the project participants, based on the Extreme Programming development process. The project will be carried out at the Hasso Plattner Institute in Potsdam. Project participants are expected to communicate with our partner via email, chat, or voice on a regular basis. In WS 2009/2010, participants will work on initial design sketches and prototypes. Main steps in design and implementation are to be executed in SS 2010. Expected results include a working software system accompanied by appropriate documentation.

Partner & Kontakt

Achim Demelt, eXXcellent solutions GmbH, Ulm (www.excellent.de, a.demelt@excellent.de)

Prof. Dr. Robert Hirschfeld, Dr. Michael Haupt, Software Architecture Group, Hasso Plattner Institute, Potsdam (www.hpi.uni-potsdam.de/swa, hirschfeld@hpi.uni-potsdam.de, michael.haupt@hpi.uni-potsdam.de)