March 2, 2004

Implementations of GoF Design Patterns in Java and AspectJ

Code base documentation

Overview

The goal was to provide parallel implementations in AspectJ and Java that allow for direct comparisons. It has to be noted that most patterns offer variability in the implementation approach. We choose the one that appeared to be the most general. Sometimes, the AspectJ version implements a slightly different (or additional) case(s). This is due to the fact that AJ sometimes allows for additional functionality. Please refer to the web page and our OOPSLA ’02 paper for a detailed description of this work.

Example Setup

All pattern examples have a class called `Main`. This class is the driver for the particular example. The `Main` classes are extensively documented using `ajdoc`, describing the experimental setup and the assignment of roles to participants. In most cases, the differences between Java and AspectJ implementations are also mentioned.

Documentation (`ajdoc`)

While all files are extensively documented using `ajdoc` (the AspectJ version of `javadoc`), `ajdoc` is not yet compatible with the later AspectJ releases, so it is currently not possible to generate HTML documents from it. This will be added when `ajdoc` is updated.

Within the `ajdoc` documentation, we tried to separate type names used in our examples from role names (as presented in GoF). We show roles names in *italics* and actual type names in `code font`.

Questions, feedback, suggestions, etc.

The AODP web page is http://www.cs.ubc.ca/labs/spl/projects/aodps.html

Please send all questions, feedback, and suggestions to Jan Hannemann (jan@cs.ubc.ca). We are very much interested in improving our code. Please do not hesitate to drop us a line.

Appendix

This appendix outlines how to compile and run the examples provided. DOS batch files exist that automate these tasks somewhat. Note that the batch files only work in Windows environments. The following is a list of tasks and a description of what commands accomplish them. For compiling, running and generating documentation, two options are given. The first one is using a provided script; the second is the standard command-line option (longer, but will work on all operation systems).

A1: Using the Eclipse IDE

Setting up your system and running the examples

1. Install Eclipse (www.eclipse.org) and AJDT (www.eclipse.org/ajdt). Currently AJDT only works with release 2.1.X and not with version 3.0+ of Eclipse. Check the AJDT web page for more information and updates.
2. Create a new AspectJ project
3. Import the ZIP file with the AOP pattern examples (from the web page) into your project
4. Set the source directory to "Design Pattern Examples/src" (under: Project / Properties / Java Build Path / Source). Set/adjust the output directory is desired.
5. Compile
6. Run individual examples by selecting the appropriate source folder and selecting Run As / Java Application.

A2: Using other AspectJ-compatible IDEs

Note: the code base has not been tested with other IDEs. Chances are that this will work similar to the above, though.

A3: Using command-line compilation

Setting up your system

1. Install Java (version 1.4+) and AspectJ (version 1.1+)
2. Extract the ZIP file into a directory of your choice
3. Make sure your `CLASSPATH` contains the example's `src` directory
4. Change to that directory

Compile Java and AspectJ versions the design pattern examples. Choose one:

- Use the **`buildAllPatterns`** batch file (just call it from the examples root directory. Needs no arguments).
- `ajc –d bin @src/allPatterns.lst`

Run a compiled example (e.g. observer). Choose one:

- `testPattern observer` (this runs both Java and AspectJ versions)
- `java ca.ubc.cs.spl.aspectPatterns.examples.observer.java.Main` (for the Java version),
  `java ca.ubc.cs.spl.aspectPatterns.examples.observer.java.Main` (for the AspectJ version)