Dakar Testing

Authors
Damien Cassou & Karsten Kuche

Company
Georg Heeg eK

Application
Dakar Testing

Availability
Cincom Public Store - package: Dakar Testing

Keywords
unit-testing, sunit, tdd, user experience, work flow, user interaction

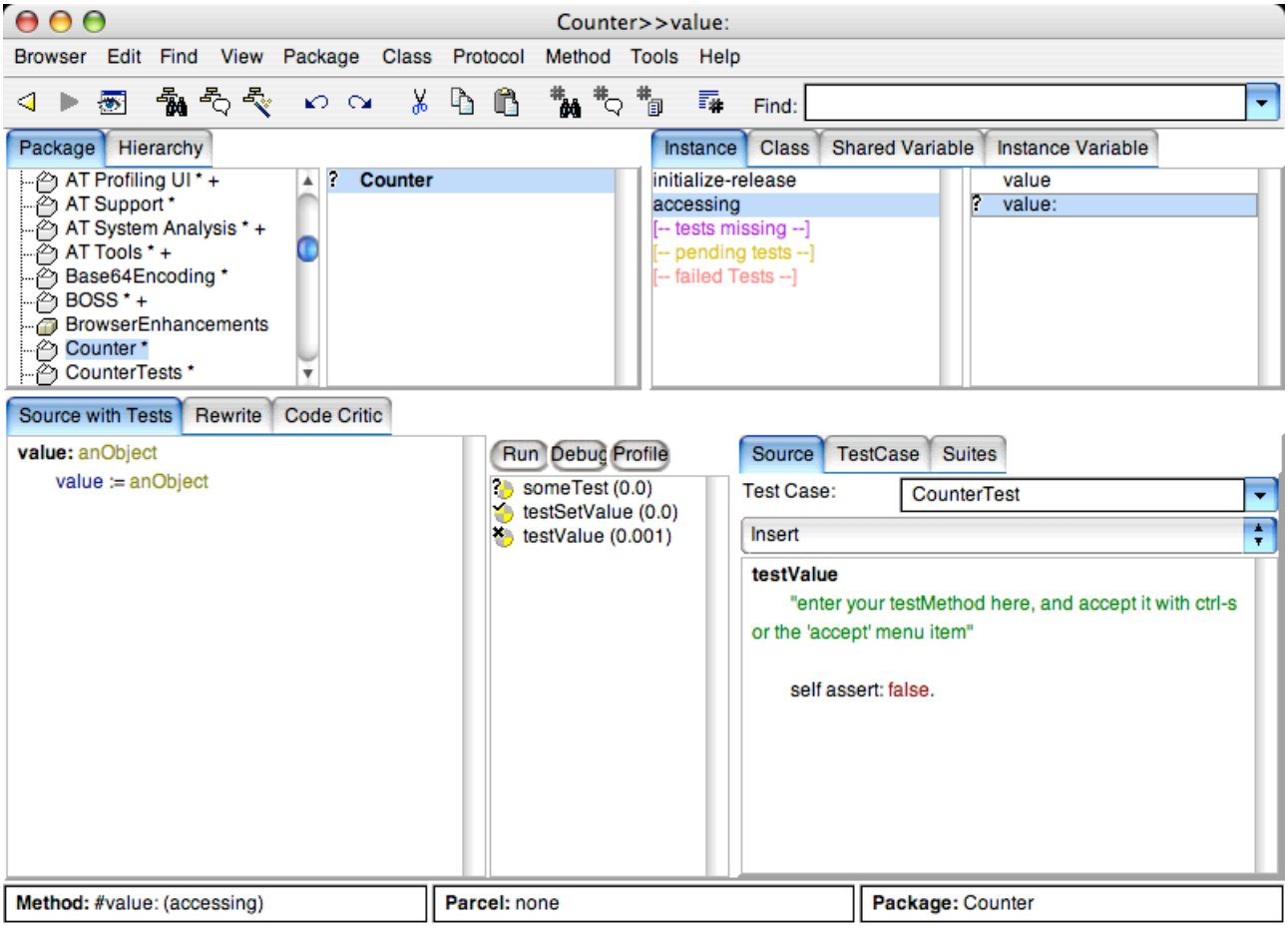
Description
Whereas unit tests are known to be part of the best practices in software development, they are seldom used because of the difficulties to create them and maintain them. In fact, when writing software, developers often have to switch between the development process and the testing process which is kind of disturbing. The aim of this tool is to improve the workflow and process of unit testing and to allow more efficiency and less disturbance to the developer.

Details

Features
- linking methods and their tests (manually with the mouse)
- creation of suite, which is a collection of tests and/or other suites
- everything is done in one browser, within the model
    - running tests
    - seeing the states
    - creating and managing suites
    - writing helper methods
- duration of the test is stored and displayed in front of each test
- a pending state informing about outdated tests

- methods under test display the state of the associated tests
- you see what is tested and what is untested
- change a method and its tests are automatically invalidated (set to pending)
- automatic storage of links and suites using source code that can be published and merged
- dynamic protocols for tested and untested methods with their tests

Screenshots



Here, one can see the refactoring browser enhanced with our tool.

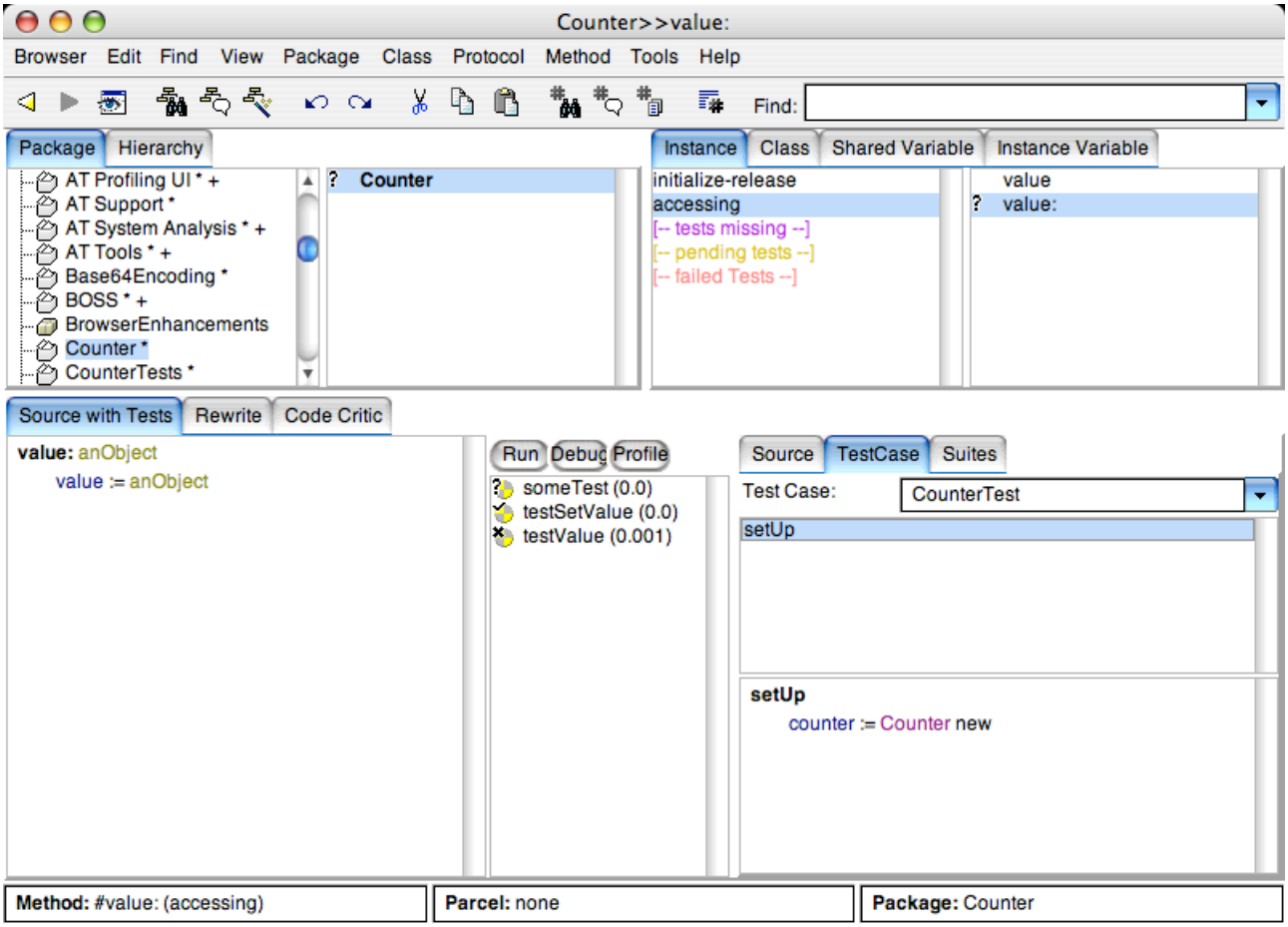
Multiple things can be noticed:

- the coding panel has been divided into two pieces. On the left you get the default text pane on which you write methods. On the right this is our tool; a description follows

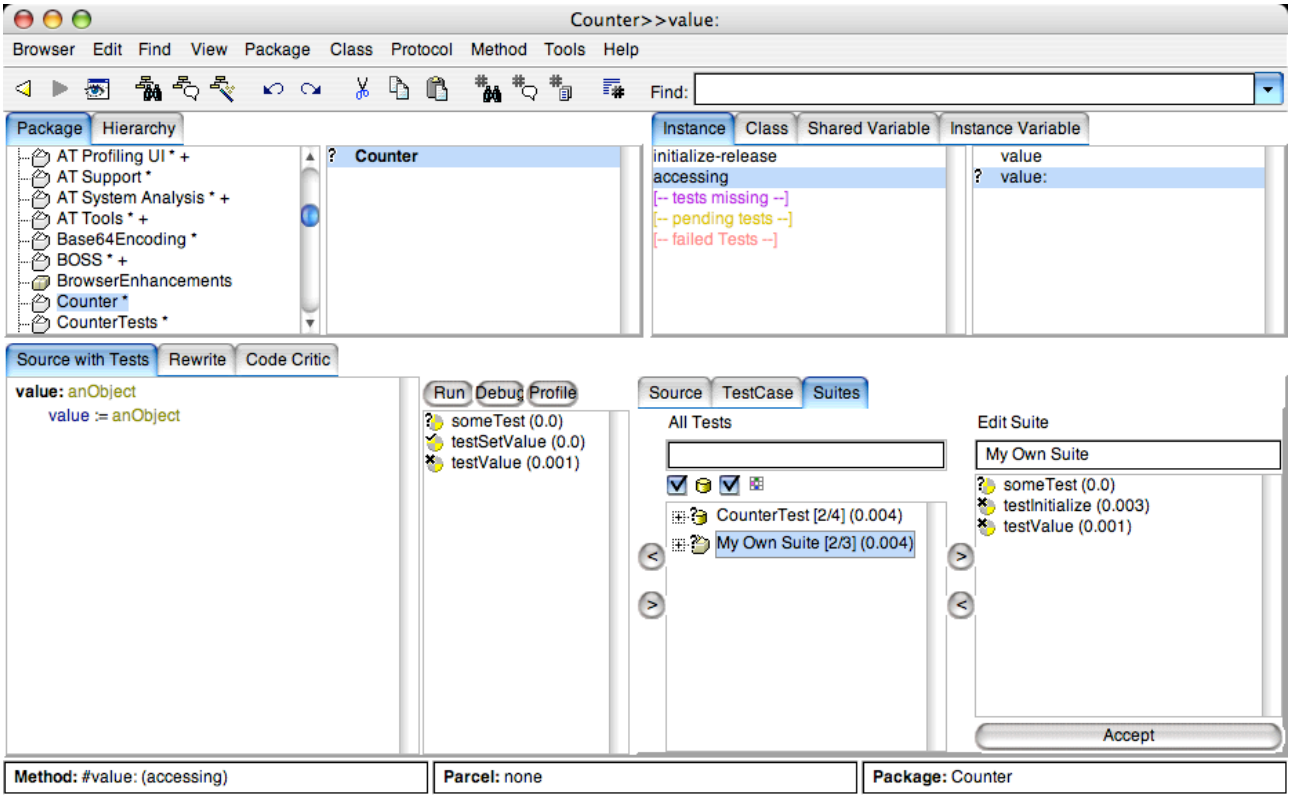
- on the top, in the protocol list, there are three dynamic protocols categorizing the methods. The content of this protocols is computed dynamically and if they are empty then the system makes them disappear.

The Source tab of the testing pane is used to create new test methods for the selected method. You can change the class into which this method will be compiled, and you can also insert templates for methods like #assert: and #deny:.

Tests are compiled in the selected testcase. If this class does not exist yet, then it is automatically created in a package that is also automatically created when not present. The names that are used for this creation is based on the selected class and protocol and just appends ‚Test' to it.



The TestCase tab is used to implement methods, that the selected TestClass should understand, like setUp and tearDown or whatever method you need for your tests.
It also displays a list of all implemented helper methods. These methods are compiled in the TestCase's ‚helper' protocol.



The Suites tab is used to provide an easy way to organize ones tests. Tests can be bundled into suites, which again behave like tests.

To assign a test or a suite to the selected method you just select them and drag them over the list of assigned methods. If you don‘t like drag and drop you can use the buttons as well.

In order to create new Suites you simply fill the right list and give the suite a name, then the suite will be created as soon as you hit the accept button.