UCI
University of California, Irvine

DAS-BOOT
Design- And Specification-Based Object-Oriented Testing

DAS-BOOT is a design- and specification-based tool for testing object-oriented systems. The current prototype:

- implements improved specification-based test coverage criteria suitable for object-oriented software systems whose behavioral specification is modeled as finite state machines (FSM);
- generates test cases, test drivers and embedded test oracles with little interaction required by the human tester.

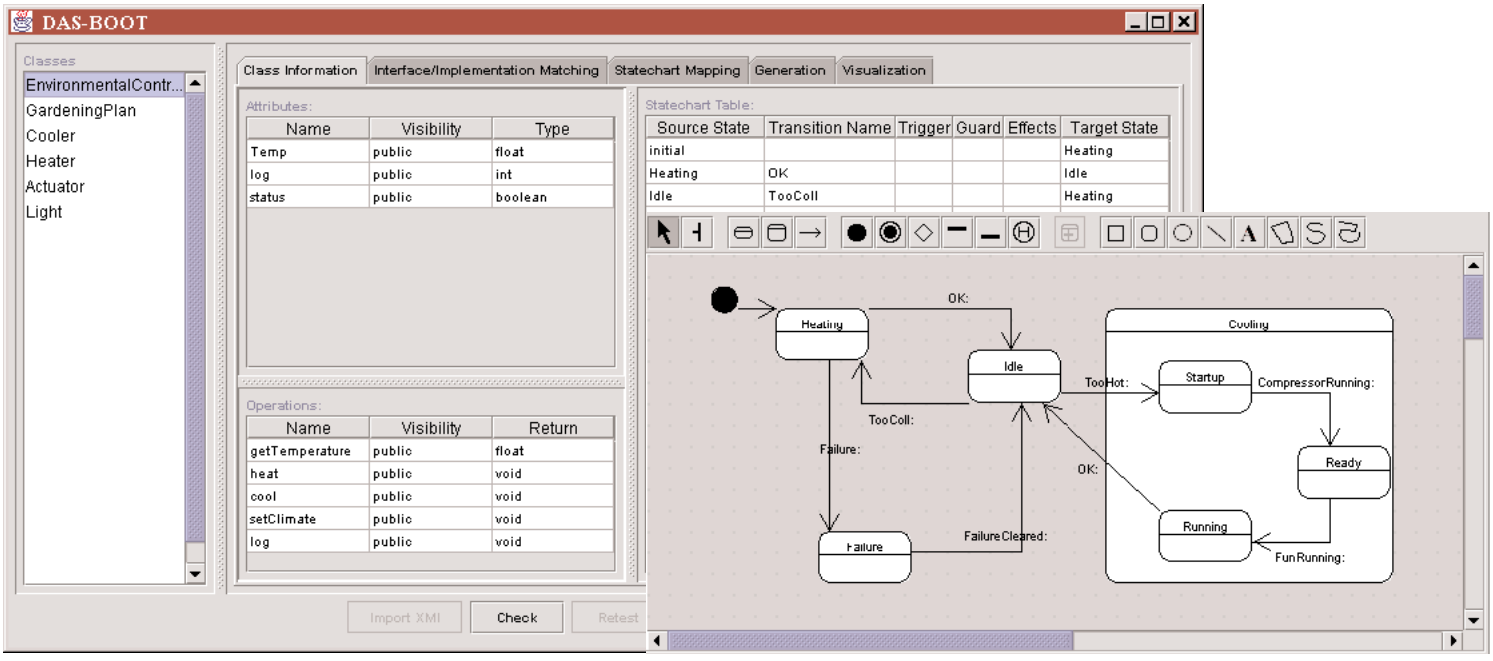
DAS-BOOT demonstrates these capabilities by testing Java classes based upon UML Statechart diagrams (a widely accepted notation for requirements & design specification). DAS-BOOT takes as input (1) a Java class to be tested, (2) a Statechart specification of the class behavior, and (3) a FSM-based test coverage criterion. From this information, DAS-BOOT produces as output (1) test drivers, (2) test oracles, and (3) test cases. The test drivers automatically execute the Java class over test cases satisfying the test coverage criterion and embody test oracles that automatically check the Java class behavior against the Statechart specification.

Specification-based test coverage criteria (in this case FSM-based criteria) are meant to define what to test based upon covering a specification (model) of the component under test. These criteria can be used in two very distinct ways. First, they can be used to test or simulate the model during the specification phase and thereby detect defects early in the software lifecycle. Second, they can be used to indicate how to test the implementation to ensure that the behavioral model is adequately covered, thereby testing based upon what the component is required to do rather than merely what it actually does (as is indicated by code-based coverage criteria). DAS-BOOT utilizes FSM-based coverage criteria in both ways: to ensure that the finite state model of intended behavior has been covered either to test the model itself or to test an implementation meant to reflect the model.

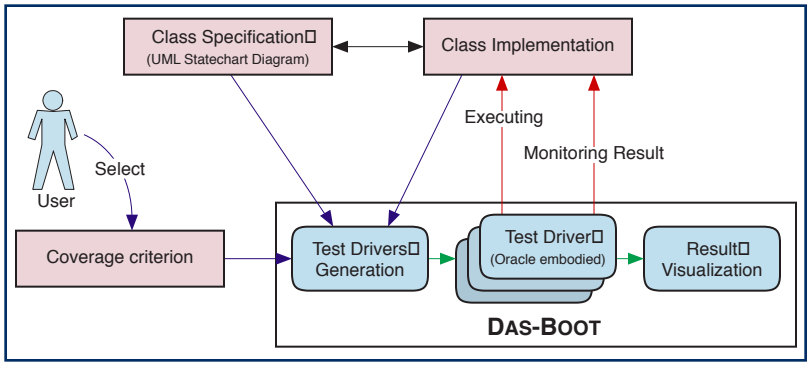
DAS-BOOT allows the user to choose from a limited set of FSM-based test criteria based upon Chow's W method. The W method requires covering the tree of possible sequences of transitions from the initial state without revisiting states. It may be important, however, to revisit states after a transition sequence is exercised, and some failures will appear only when states are multiply-visited. Thus, DAS-BOOT also permits the tester to choose how many times a state is visited (currently the choice is for one, two or three maximum visits). We are considering alternative approaches such as determining when and how often a particular state is revisited based on the granularity (range of values) associated with the state or based upon the dependencies between states and/or transitions in the sequence.

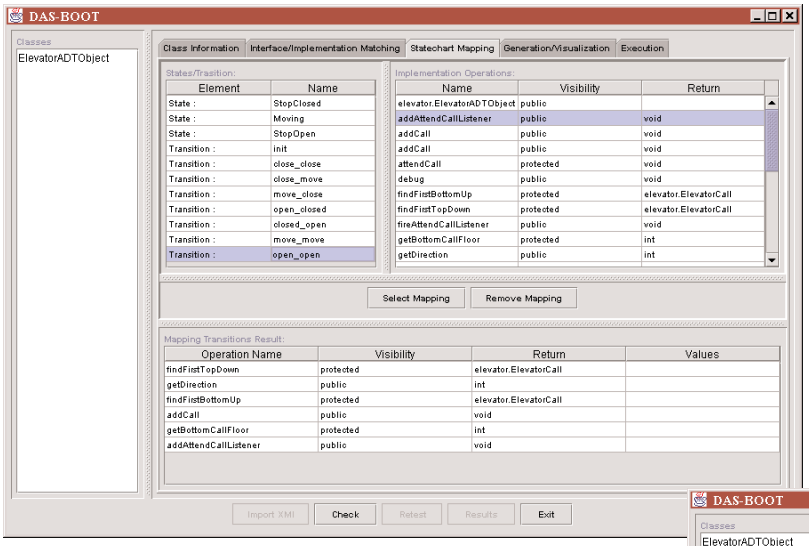
Using specification-based test criteria to test (or simulate) the specification model, is straight-forward, because the artifact upon which test cases and/or coverage measures are based are the same as the artifact being tested. On the other hand, using criteria based on the specification (or FSM model) to test the implementation requires associating or mapping the test requirements to the implemented objects under test – this is called a representation mapping. DAS-BOOT assists the developer/tester in defining the representation mapping.

DAS-BOOT automates the testing process as much as possible by generating automated test drivers for each test case that not only



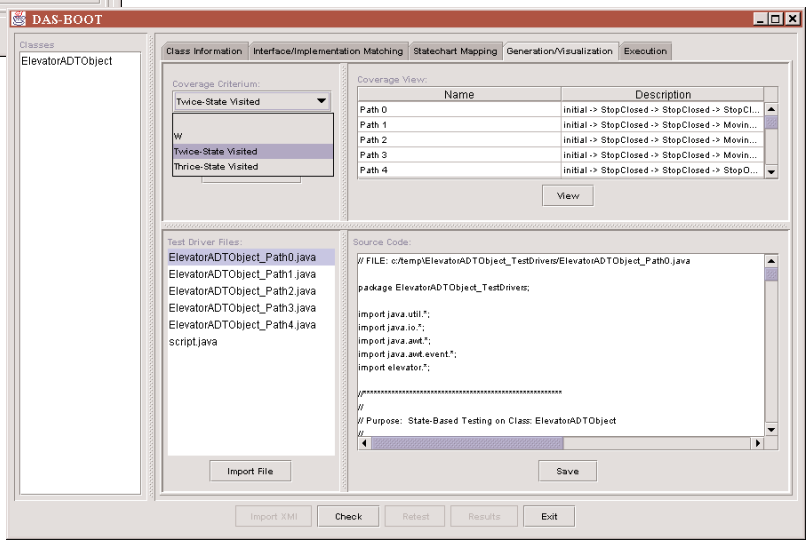
force required coverage but also ensure that the implementation behaves according to the specification or model. For FSMs, this is accomplished by not only force the coverage of the associated transition but also checking to ensure that the implementation behaves according to the model. Thus, for each transition (consisting of a source state, triggering event, action to be taken, and destination state) to be covered, a test driver is developed that puts the object into the source state, creates the circumstances leading to the triggering event, observes any actions taken and the destination state, and compares those actions and destination to those specified in the model.



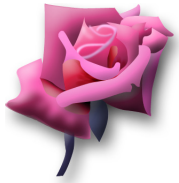


DAS-BOOT serves as a testbed for experimenting with approaches to specification-based testing. We are currently focusing on alternative FSM-based criteria since the specifications within DAS-BOOT are limited to UML Statecharts. Although Statecharts is a very rich notation, we intend to extend our specification environment to incorporate other notations for specifying component behavior. As we do this, we will prototype more appropriate specification-based test criteria. DAS-BOOT also provides a testbed for experimenting with approaches to defining representation mappings.

DAS-BOOT is integrated into the ARGUS-I architecture analysis toolset as the dynamic component analysis tool, thereby providing confidence in component quality before composing components into an architectural configuration.



Specification-based testing requires adding information to the specification/models, such as the data domain (set of allowable values) of each attribute, preconditions for functions. Adding test information (also called making test-ready) requires additional work, but is by far easier than creating new specifications or models just for the purpose of testing. This poses a basic drawback but also a major advantage of our approach: the models are not specifically developed for testing. On the other hand, with the current DAS-BOOT, they can be imported from a XMI\XML format and then augmented with testing information that enables the state based testing. This significantly reduces the overhead involved in using specification-based testing – thus, the advantage.



The **ROSATEA** tools are research prototypes, some of which have been successfully transitioned into use on real projects, but not yet as commercial systems. These tools were developed by the Research Organization for Specification- and Architectural-based Testing E (&) Analysis at the University of California at Irvine. The work was done in conjunction with the Perpetual Testing Projects sponsored by the DARPA's EDCS program.

The **DAS-BOOT** search and development effort is sponsored in part by: the Air Force Materiel Command, Air Force Research Laboratory, and the Defense Advanced Research Projects Agency under agreement number #F30602-97-2-0033. The views and conclusions contained herein are those of the researchers and should not be interpreted as representing the official position or policy, either expressed or implied, of the U.S. Government, AFMC, Rome Laboratory, DARPA, or the University of California, and no official endorsement should be inferred.

Freely Available Software

Information about UC Irvine's *Research Organization for Specification- and Architecture-based Testing E (&) Analysis* (ROSATEA), as well as its software, is available at:
http://www.ics.uci.edu/pub/rosatea/

June 1999

Contact Information
Professor Debra J. Richardson
Information and Computer Science
University of California
Irvine, California 92697-3425

url: http://www.ics.uci.edu/~djr
email: djr@ics.uci.edu
voice: 949-824-7353
fax: 949-824-1715