Design Tests

Federal University of Campina Grande
Department of Computer Science
João Brunet, Dalton Guerrero and Jorge Figueiredo
http://www.designwizard.org

The Problem

- Gap between the state-of-the art and the state-of-the-practice in conformance checking area
- Approaches from the state-of-the-art are difficult to use
- Design decisions are violated

Objectives

- Bridge the gap between state-of-the art and the state-of-the-practice in conformance checking
- Build an automated an simple to use approach to check conformance

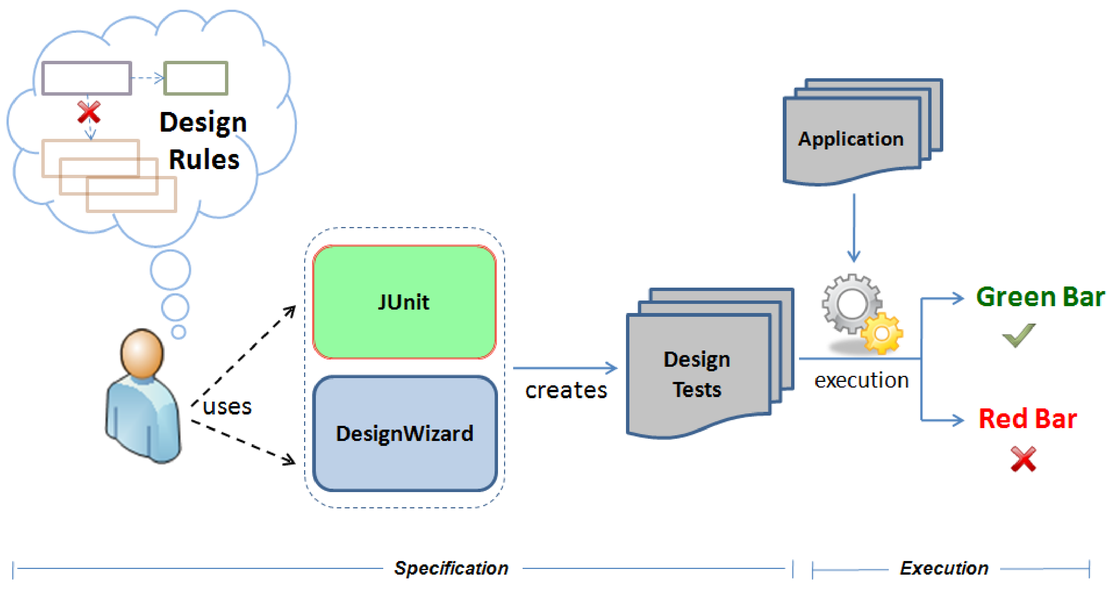
Design Test

- A test that checks whether an implementation complies with a given design rule expressed as an algorithm
- Automated test
- Written in the target programming language

Design Test Pseudocode

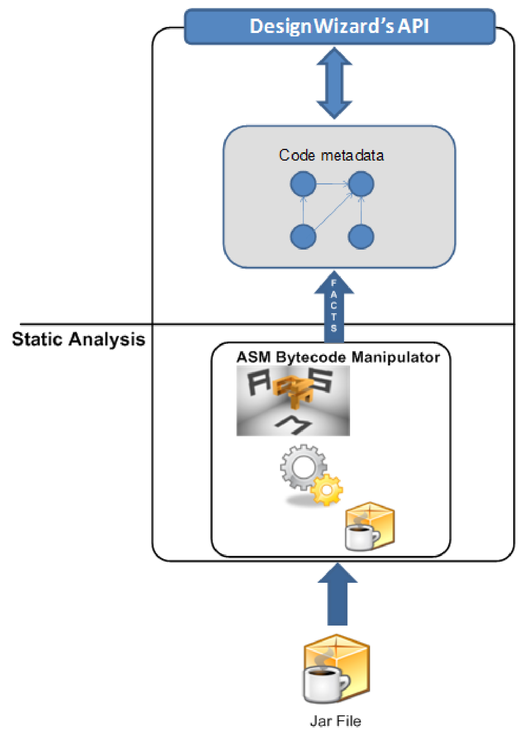
```
1  daoPackage = org.ourgrid.peer.dao
2  controllerPackage = org.ourgrid.peer.controller
3  callers = daoPackage.getCallers()
4  FOR each caller IN callers DO
5    assert ( caller == daoPackage ) || ( caller == controllerPackage )
6  ENDFOR
```

Checking Conformance with DT



DesignWizard

- Extracts facts from java *bytecode*
- Models the facts in a graph, in which nodes are entities and edges are the relationships among them
- Exposes an API to provide access to the information extracted



Early Evaluation

- OurGrid (111,790 LOC)
  - 10 violations (7 severe and 3 acceptable)
  - All developers understood what the test checks
  - developers also appreciated how conformance checking accommodates into the testing process
  - Violations caused design re-discussion

Acknowledgements

This work has been partially supported and developed in collaboration with CPM Braxis and FINEP/MCT/Brazil through grant number 3156/06.

Related Work

[1] J. Aldrich, C. Chambers, and D. Notkin. ArchJava: connecting software architecture to implementation. Proceedings of the 24th International Conference on Software Engineering, pages 187–197, 2002.

[2] G. Fairbanks, D. Garlan, and W. Scherlis. Design fragments make using frameworks easier. In Proceedings of the 2006 OOPSLA Conference, volume 41, pages 75–88. ACM New York, NY, USA, 2006.

[3] G. Murphy, D. Notkin, and K. Sullivan. Software reflexion models: bridging the gap between design and implementation. Software Engineering, IEEE Transactions on, 27(4):364–380, Apr 2001.