DesignWizard

A Tool that Gives Support to Automatically Check Your Code Against Design Rules

João Brunet

Dalton Guerrero and Jorge Figueiredo

Federal University of Campina Grande (UFCG), Brazil.

24/06/2009

Agenda

1. Motivation
2. Design Test
3. DesignWizard
4. Experience
5. Benefits
6. Related Tools

Motivation

- Check code against design rules is an important activity to guarantee quality on source code
  - Design Review
    - Manual Process
    - Error-prone
    - Does not scale

Motivation

- Check code against design rules is an important activity to guarantee quality on source code
  - Design Review
    - Manual Process
    - Error-prone
    - Does not scale

The Problem

- Gap between state-of-the-art and the state-of-the-practice
  - Approaches from the state-of-the-art are difficult to use

The Problem

Lack of structural conformance checking

Design decisions are violated

Design Test

Design Test

A test that checks whether an implementation complies with a given design rule expressed as an algorithm.

What's the big deal?

- **It is written in the target programming language**
- Automated test

Example of Design Rule

Only *controller* package shall access the *dao* package of the *peer* component.

Example of Design Test

Only *controller* package shall access the *dao* package of the *peer* component.

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
```

Example of Design Test

Only *controller* package shall access the *dao* package of the *peer* component.

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
```

- Facts about the code (an API)

Example of Design Test

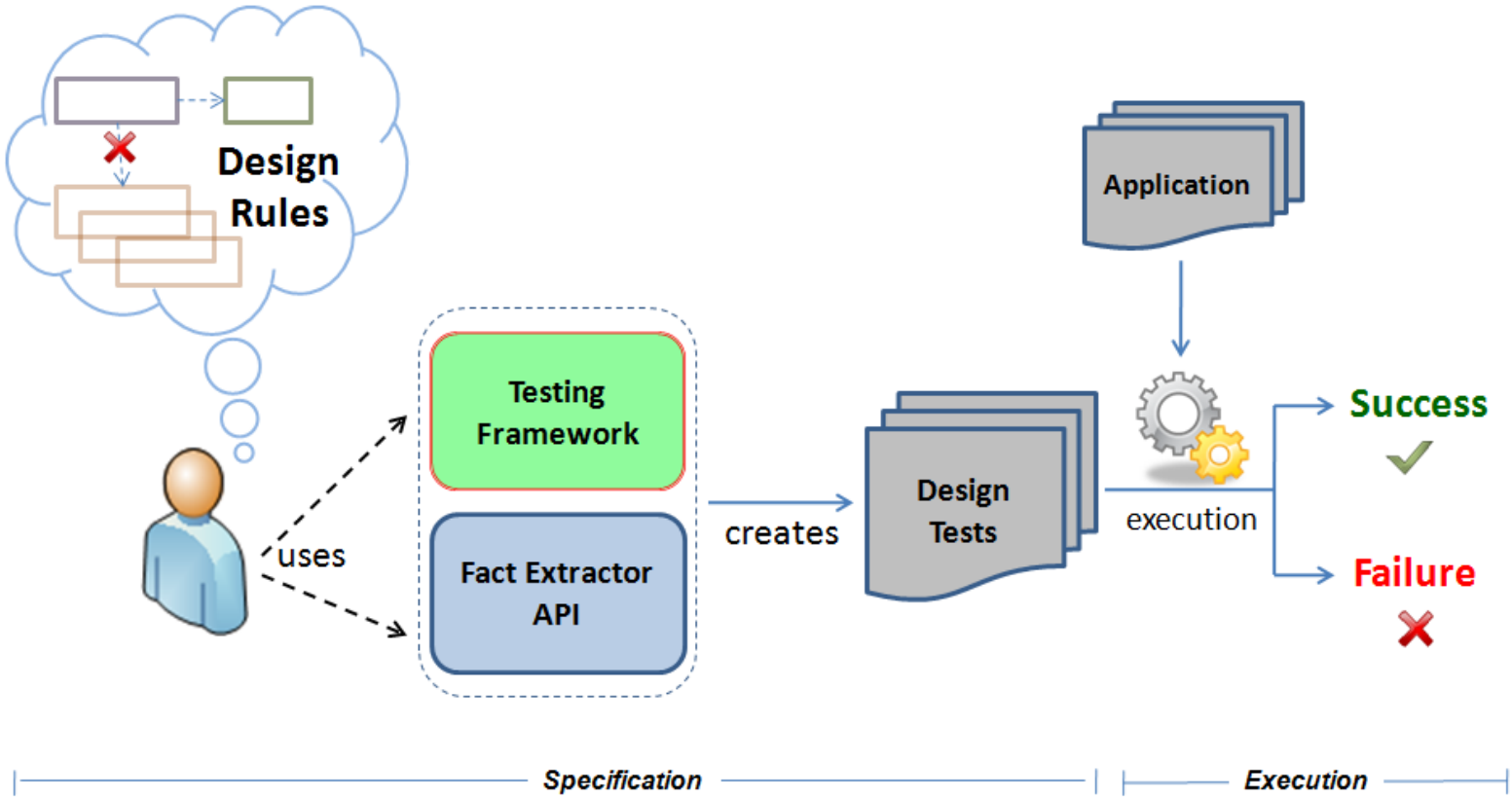
Only *controller* package shall access the *dao* package of the *peer* component.

Design Test Pseudocode

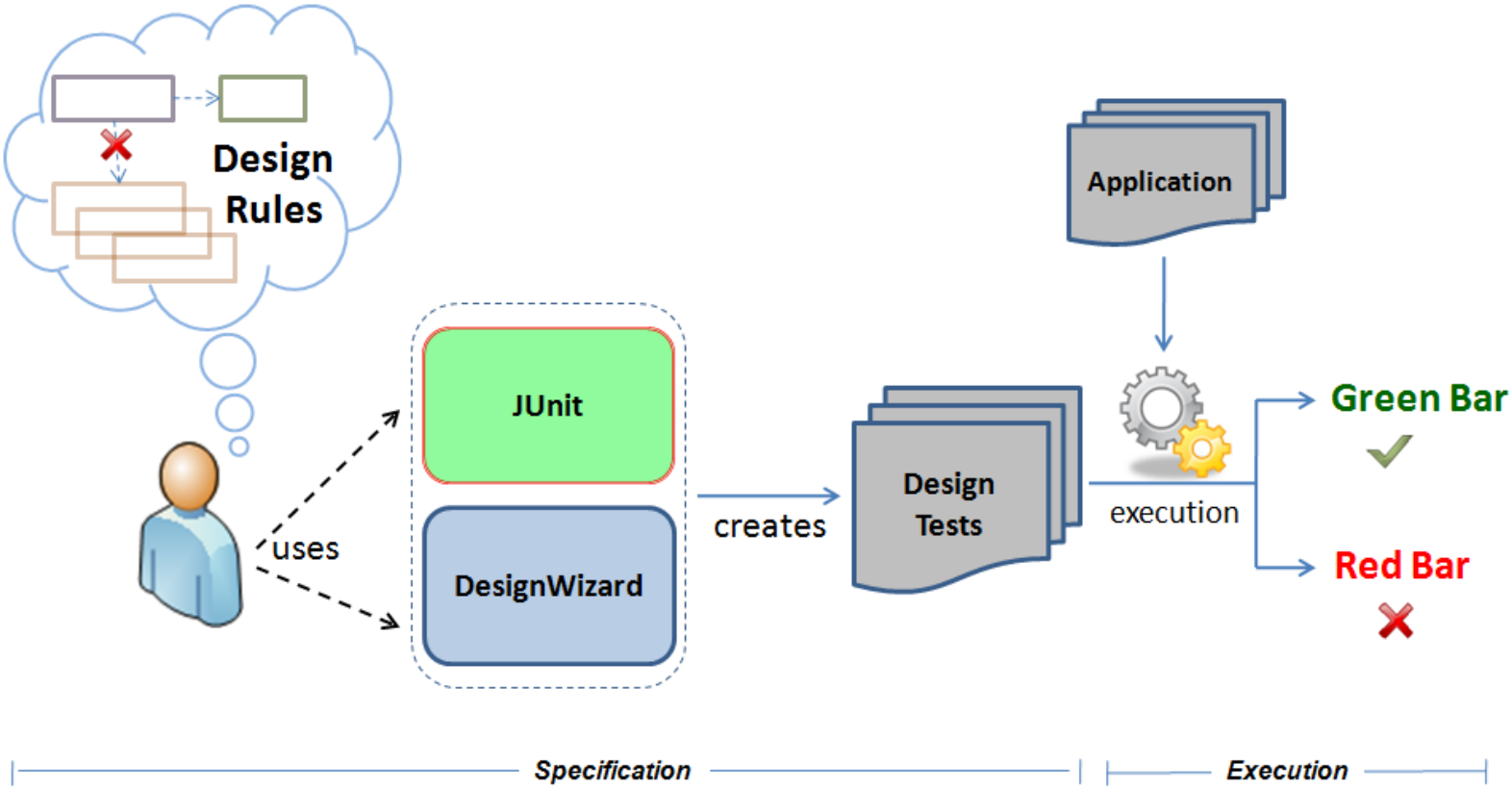
```
1  daoPackage = org.ourgrid.peer.dao
2  controllerPackage = org.ourgrid.peer.controller
3  callers = daoPackage.getCallers()
4  FOR each caller IN callers DO
5    assert ( caller == daoPackage ) || ( caller == controllerPackage )
6  ENDFOR
```

- Facts about the code (an API)
- Assertion routines

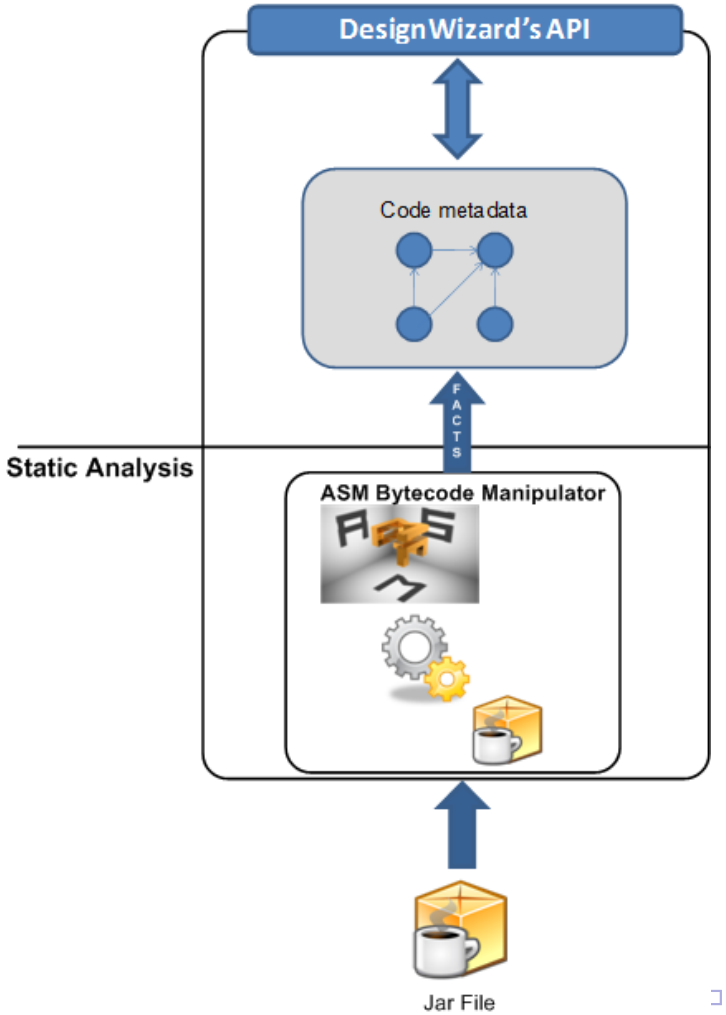
Checking Code against Design Rules with Design Tests



Checking Code against Design Rules with Design Tests



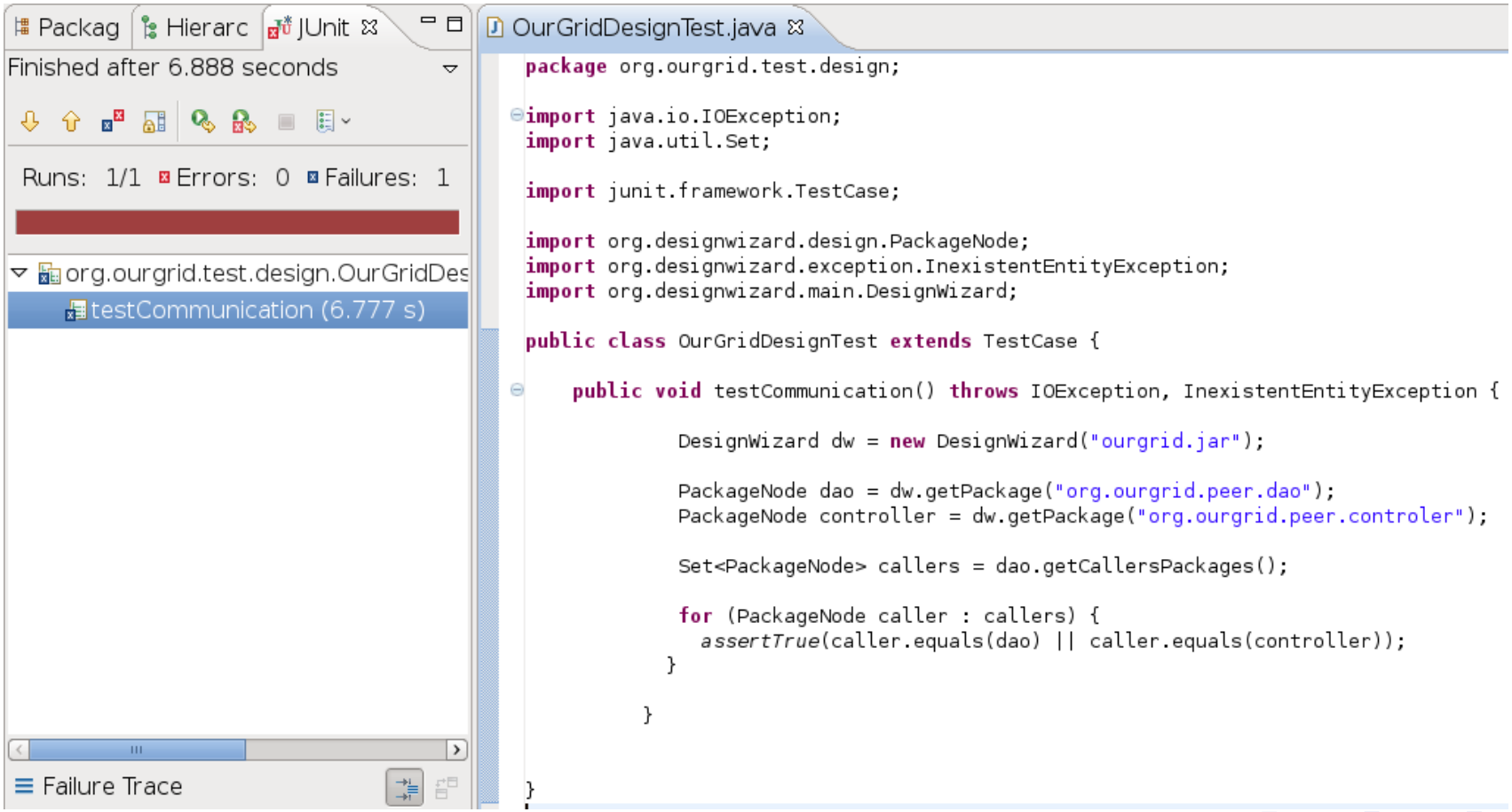
DesignWizard



Some Of DesignWizard's API

- getAllPackages()
- getAllSubClasses()
- getStaticMethods()
- getImplementedInterfaces()
- ...

Design Test Execution



DW + JUnit

```
public class OurGridDesignTest extends TestCase {

    public void testCommunication() throws IOException, InexistentEntityException {

            DesignWizard dw = new DesignWizard("ourgrid.jar");

            PackageNode dao = dw.getPackage("org.ourgrid.peer.dao");
            PackageNode controller = dw.getPackage("org.ourgrid.peer.controler");

            Set<PackageNode> callers = dao.getCallersPackages();

            for (PackageNode caller : callers) {
              assertTrue(caller.equals(dao) || caller.equals(controller));
            }

    }


}
```

Figure: Design Test Code

DW + JUnit

```
public class OurGridDesignTest extends TestCase {

    public void testCommunication() throws IOException, InexistentEntityException {

            DesignWizard dw = new DesignWizard("ourgrid.jar");

            PackageNode dao = dw.getPackage("org.ourgrid.peer.dao");
            PackageNode controller = dw.getPackage("org.ourgrid.peer.controler");

            Set<PackageNode> callers = dao.getCallersPackages();

            for (PackageNode caller : callers) {
              assertTrue(caller.equals(dao) || caller.equals(controller));
            }

    }


}
```

Figure: Design Test Code

DW + JUnit

```
public class OurGridDesignTest extends TestCase {

    public void testCommunication() throws IOException, InexistentEntityException {

            DesignWizard dw = new DesignWizard("ourgrid.jar");

            PackageNode dao = dw.getPackage("org.ourgrid.peer.dao");
            PackageNode controller = dw.getPackage("org.ourgrid.peer.controler");

            Set<PackageNode> callers = dao.getCallersPackages();

            for (PackageNode caller : callers) {
              assertTrue(caller.equals(dao) || caller.equals(controller));
            }

    }


}
```

Figure: Design Test Code

DW + JUnit

```
public class OurGridDesignTest extends TestCase {

    public void testCommunication() throws IOException, InexistentEntityException {

            DesignWizard dw = new DesignWizard("ourgrid.jar");

            PackageNode dao = dw.getPackage("org.ourgrid.peer.dao");
            PackageNode controller = dw.getPackage("org.ourgrid.peer.controler");

            Set<PackageNode> callers = dao.getCallersPackages();

            for (PackageNode caller : callers) {
              assertTrue(caller.equals(dao) || caller.equals(controller));
            }

    }


}
```

Figure: Design Test Code

DW + JUnit

```
public class OurGridDesignTest extends TestCase {

    public void testCommunication() throws IOException, InexistentEntityException {

            DesignWizard dw = new DesignWizard("ourgrid.jar");

            PackageNode dao = dw.getPackage("org.ourgrid.peer.dao");
            PackageNode controller = dw.getPackage("org.ourgrid.peer.controler");

            Set<PackageNode> callers = dao.getCallersPackages();

            for (PackageNode caller : callers) {
              assertTrue(caller.equals(dao) || caller.equals(controller));
            }

    }


}
```

Figure: Design Test Code

DW + JUnit

```
public class OurGridDesignTest extends TestCase {

    public void testCommunication() throws IOException, InexistentEntityException {

            DesignWizard dw = new DesignWizard("ourgrid.jar");

            PackageNode dao = dw.getPackage("org.ourgrid.peer.dao");
            PackageNode controller = dw.getPackage("org.ourgrid.peer.controler");

            Set<PackageNode> callers = dao.getCallersPackages();

            for (PackageNode caller : callers) {
              assertTrue(caller.equals(dao) || caller.equals(controller));
            }

    }


}
```

Figure: Design Test Code

Early Evaluation

- OurGrid
  - 100KLOC
  - Execution time: 6.7 seconds
  - 10 violations

Early Evaluation

- All developers understood what the test was supposed to do
- They appreciated the way to check whether they are following the rules
- The test was added in the test suite of OurGrid in order to be performed daily

What else?

- Rules related to framework usage
- Bug patterns (HashCode and Equals implementation)
- Rules related to quality aspects

Benefits

- Easy to compose/maintain
- Easy to understand
- Easy to execute
- Executable documentation of the design rules
- Easy to be added in the process development

Related Work

- FindBugs
- PMD
- CheckStyle

More About DesignWizard

Site: http://www.designwizard.org

Main reference: BRUNET, João; GUERRERO, Dalton and FIGUEIREDO, Jorge. **Design Tests: An Approach to Programmatically Check your Code Against Design Rules.** To appear in Proceedings of the 31st International Conference on Software Engineering (ICSE 2009), New Ideas and Emerging Results, Vancouver, Canada, May 2009.

Thank you

Questions?