A Unified Approach to Architecture Conformance Checking

Andrea Caracciolo, Mircea Filip Lungu, Oscar Nierstrasz

http://scg.unibe.ch
Architecture Erosion

Design ≠ Code
Architecture Erosion

Compliance Monitoring

RULES
1. you can....
2. you can't...
3. you can....
4. you can't
Compliance Monitoring

60% non-automated techniques *

* How Do Software Architects Specify and Validate Quality Requirements? (ECSA 2014)
Compliance Monitoring
Compliance Monitoring
Compliance Monitoring

Specification Formalisms

A
B

c

JAVA
Compliance Monitoring

Reports
Compliance Monitoring
Compliance Monitoring
A Unified Approach

Persistence cannot depend on Service Impl must have annotation "@Service"
System cannot contain cycles
Rules

Persistence cannot depend on Service
Service must have annotation "@Service"
System cannot contain cycles
Rules

Persistence cannot depend on Service
Service must have annotation "@Service"
System cannot contain cycles

Entities

Persistence = Package with name: "app.*persistence.*"
Service = Package with name: "app.*service.*"
Rules

Persistence cannot depend on Service
only Service can have annotation “@Service”
System cannot contain cycles

Entities

Persistence = Package with name:"app.*persistence.*"
Service = Package with name:"app.*service.*"
Rules

Persistence cannot depend on Service
only Service can have annotation "@Service"
System cannot contain cycles

Entities

Persistence = Package with name: "app.*persistence.*"
Service = Package with name: "app.*service.*",
        name! : "*persistence*",

Rules

Maintainability

*Method* can only be named "String"
only *Package* can contain dead methods

Compatibility

*XMLTag* must have attribute “String"
*WebResource* must have content "String"

Reliability

Method must catch *Class*

Performance

*WebResource* must have latency < *int ms*
*WebResource* must handle load from *int users*
Analysis

Persistence cannot depend on Service

depend-on(app.xx.persistence.yy, app.xx.service)
depend-on(app.xx.persistence.yy2, app.xx.service)
depend-on(app.xx.persistence.yy3, app.xx.service)
Persistence cannot depend on Service

\[
\text{depend-on(app.xx.persistence.yy, app.xx.service)} = \text{True} \\
\text{depend-on(app.xx.persistence.yy2, app.xx.service)} = \text{False} \\
\text{depend-on(app.xx.persistence.yy3, app.xx.service)} = \text{False}
\]
Dictō

Persistence cannot depend on Service
Service must have annotation "@Service"
System cannot contain cycles

Statistics
Dicto
Automated Architectural Tests. With emphasis on the Diff. What's this?

- Repository: https://github.com/studer-raimann/ILIAS
- Commit: ec5af77e96c6ab3e6b6ca0e5461d7d7d8015b833a
- Compared to Commit: 1d8c0bc57f27a02a0df2a5ab82a541dec10ee46

742 468 0
Total Violations Added Violations Resolved Violations

- only GUIClasses can depend on iILanguage

- WholelliasCodebase cannot depend on SuppressErrors

- iIExceptionsWithoutTopLevelException can only depend on iIExceptions

- WholelliasCodebase cannot depend on eval

- WholelliasCodebase cannot depend on exitOrDie

Exit and die are a bad idea in both development and production: In development you have no idea what went wrong and in production the user receives a white page and has no idea what's going on. The implemented exception handling does not work if you use exit or die.

If you want to send a file consider using: Services/FileDelivery.

Exception: Currently if you want to output json you most likely have to use exit() at the moment.

| Added Violations | 1 |
| Resolved Violations | 0 |
| All Violations | 228 |

Newly Introduced Violations
iIFileDelivery depends on exit/die
Dictō

Persistence cannot depend on Service
Service must have annotation "@Service"
System cannot contain cycles
TestMethods cannot be invoked

Move this trailing comment on the previous empty line.

Remove this exit call or ensure it is really required.

Move this file to a named package.

Replace this usage of System.out or System.err by a log...
Evaluation

Open source project
LMS - PHP (1.8M LOC)
12 service providers, 900’000+ users

Large size company
B2B - Java EE (50K LOC)
1’000 employees

Medium size company
various - Java EE / .NET
100 employees
Evaluation

**Impact**
Process? Culture?

**Value**
Cost–effectiveness? Quality?

**Applicability**
Expressivity? Usability?
**Summary**

**Dictō**

A uniform, readable, executable DSL for specifying architectural constraints.

Andrea Caracciolo  
@ scg.unibe.ch/dicto
Discussion Topics

- how to streamline/incentivize compliance monitoring?
- which are the common obstacles?

Persistence cannot depend on Service
Service must have annotation "@Service"
System cannot contain cycles

Andrea Caracciolo
@ scg.unibe.ch/dicto