Analysis of Multi-Dimensional Code Couplings

Fabian Beck


primary supervisor: Stephan Diehl, University of Trier, Germany
secondary supervisor: Martin Pinzger, University of Klagenfurt, Austria

Fabian Beck is now with
Definition: Code Coupling

“If changing one module in a program requires changing another module, then coupling exists.”

Martin Fowler, 2001
Code Coupling – an Example

```java
/**
 * @author fabian
 */
class A {
    String name;
    B b = new B();

    void doSomething() {
        b.doSomethingElse();
    }

    String toString() {
        return name;
    }
}

/**
 * @author fabian
 */
class B {
    String name;

    void doSomethingElse() {
        ...
    }

    String toString() {
        return name;
    }
}
```
Goal: understand multi-dimensional code couplings
Part I

VISUAL COMPARISON OF CODE COUPLINGS

SOFTVIS 2010, VISSOFT 2011, INFOVIS 2011 (TVCG), INFORMATION VISUALIZATION
Visualizing Code Couplings

- scalable visualization
- compare concepts of coupling

Graph visualization

two visualization approaches
Node-Link Visualization

Inheritance  Aggregation  Usage

interactive comparison
Part II

CONGRUENCE OF COUPLING AND MODULARITY

ESEC/FSE 2011
Low Coupling and High Cohesion

Stevens, Myers and Constantine, 1974

Different modules are coupled loosely while being cohesive themselves

Information Hiding

Parnas, 1971/72

Hiding design decisions within modules

Conway’s Law

Conway, 1968

Products mirror the structure of the organization by which they were developed

Which principles are applied in practice?

and others
conjectured relationships:

<table>
<thead>
<tr>
<th>concept of coupling</th>
<th>modularity principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>structural (direct)</td>
<td>Low Coupling and High Cohesion</td>
</tr>
<tr>
<td>structural (indirect)</td>
<td>Information Hiding</td>
</tr>
<tr>
<td>historical (co-change)</td>
<td>Information Hiding</td>
</tr>
<tr>
<td>historical (author)</td>
<td>Conway’s Law</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

application of modularity principle ⇔ congruence of coupling and modularity
study on 16 software projects and 17 concepts of coupling

<table>
<thead>
<tr>
<th>concept of coupling</th>
<th>moudarity principle</th>
</tr>
</thead>
<tbody>
<tr>
<td>structural (direct)</td>
<td>Low Coupling and High Cohesion</td>
</tr>
<tr>
<td>structural (indirect)</td>
<td>Information Hiding</td>
</tr>
<tr>
<td>historical (co-change)</td>
<td>Information Hiding</td>
</tr>
<tr>
<td>historical (author)</td>
<td>Conway’s Law</td>
</tr>
<tr>
<td>...</td>
<td>...</td>
</tr>
</tbody>
</table>

none of the principles is dominating
Part III

COMPONENT EXTRACTION

WCRE 2010, WSR 2012
Problem: Extract a Component

goal: independent development

1. identify key entities
2. shape the components
3. extract a contract

minimal cut

contract

extracted component

user

original component
Multi-Dimensional Coupling

Study
- different suitability of coupling concepts
- combination of concepts does not lead to improvements
CONCLUSION
**Lesson:** developers need to stay in control

**Challenge:** design a modularization tool leveraging multi-dimensional code couplings
Summary

Part I
Visual Comparison of Code Couplings

Part II
Congruence of Coupling and Modularity

Part III
Component Extraction

empirical results and tools for understanding multi-dimensional code couplings