Flexible Architecture Conformance Assessment with ConQAT

Florian Deissenboeck, Lars Heinemann, Benjamin Hummel, Elmar Juergens
Technische Universität München

ICSE 2010 Cape Town
Software Architecture

- Software architecture describes
  - Decomposition of a system into parts
  - Relations between the parts

- Architectural knowledge supports making changes to the system
- Problem: Decay of architecture decreases maintainability
Architectural Knowledge in Software Projects

Feilkas et al. „Loss of Architectural Knowledge during System Evolution“, ICPC 2009
Loss of Architectural Knowledge

Industrial Case study: Investigating extent of architectural decay and its reasons

<table>
<thead>
<tr>
<th>Project</th>
<th>kLOC</th>
<th>Problems in documentation</th>
<th>Architecture violations</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td>454</td>
<td>895</td>
<td>99</td>
</tr>
<tr>
<td>B</td>
<td>495</td>
<td>914</td>
<td>125</td>
</tr>
<tr>
<td>C</td>
<td>317</td>
<td>271</td>
<td>105</td>
</tr>
</tbody>
</table>

- Documentation typically very outdated
- Numerous relevant violations discovered
Goal:
- Detect architectural violations

Ingredients:
- Machine readable specification of the intended architecture
- Knowledge of the dependencies between systems’ artifacts
- Mapping between the architecture’s components and the system’s artifacts
Related Work

- Source code query languages
  - .QL
- Dependency matrix tools
  - Xdepend, Lattix LDM
- Reflexion model based tools
  - Bauhaus, Dependometer, SAVE, SonarJ, Sotograph, Structure101

Architecture specification depends on system artifacts
Not suited for communication with stakeholders
Fixed set of artifact types
Fixed notion of dependencies between elements of system
Challenge

- Artifacts of software systems of diverse nature
  - Source code, DSL artifacts, libraries, config files, DB artifacts, ...
- Diverse notions of dependencies
  - Call/use, Create (object creation), Clone relations, ...
- Change in artifact of one type may imply changes in artifacts of other types

→ Architecture Conformance Assessment Tool must be flexible
ConQAT (Continuous Quality Assessment Toolkit)

- **Graphical DSL** to specify analysis configuration
- **Processors** perform analysis operations
- **Edges** describe data flow between processors
- **Blocks** make configuration fragments reusable
Intended Architecture (Demo)

- Simple hierarchical component model
- Dependency Policies: allow, deny
Artifact Mapping (Demo)

- Mapping system artifacts to components via regular expressions
- For Java: Fully qualified class name
- Mapping types: include, exclude
Dependency extraction

- Extract dependency graph from system

- Example: Dependency extractor for Java
  - Uses byte code analysis to extract use dependencies
  - Advantage: No source code required
Assessment (Demo)

- Compares actual system architecture to specified intended architecture
- Dependencies are rated: valid, invalid, unnecessary

<table>
<thead>
<tr>
<th>Policy type</th>
<th>#deps &gt; 0</th>
<th>#deps = 0</th>
</tr>
</thead>
<tbody>
<tr>
<td>allow</td>
<td>valid</td>
<td>unnecessary</td>
</tr>
<tr>
<td>deny</td>
<td>invalid</td>
<td>valid</td>
</tr>
</tbody>
</table>

- Unmappable artifacts: orphans
Industrial Applications

- Assessment with initial architecture from documentation and interviews
- Experience: First assessment reveals many deviations
- Iterate: Refine architecture specification; rerun assessment
- Tolerations for violations that cannot be fixed immediately
- Run assessment continuously, e.g. in nightly build

Architecture description is kept up-to-date and turns into „living“ artifact
Flexible Configuration (Demo)

- Configuration via Pipes and Filter DSL
- Graphical setup of architecture assessment pipeline
- Enables support for arbitrary
  - Artifact types
  - Notions of dependency
Summary

- Continuous architecture conformance assessment
  - Keeps architecture description up-to-date
  - Detects architectural violations early
- Light-weight specification of intended architecture lowers adoption barrier
- Different types of artifacts and dependencies with low configuration effort for real world software

All tool support presented available as open source at http://www.conqat.org
Thank you.

Questions?