An Empirical Study of Software Architecture Change in Open-Source Software Systems

Duc Le*, Pooyan Behnamghader*, Joshua Garcia, Daniel Link, Arman Shahbazian, Nenad Medvidovic
Architecture of Two Software Versions

Apache Chukwa 0.3.0
How Systems End Up

Architecture Change

• Architecture change is caused by **introduction of design decisions** into the descriptive architecture that are either
  – **Unforeseen** by the prescriptive architecture (**drift**)  
  – **Violating** the prescriptive architecture (**erosion**)  

• Benefits of knowing your architecture and how it changes  
  – Less unexpected complexity  
  – Reduce maintenance effort  
  – Regain full control of development
The Big Picture of ARCADE

- **Architecture Recovery, Change, and Decay Evaluator (ARCADE)** is a workbench for software architecture comprehension

- ARCADE’s features
  - **Software architecture recovery**
    - Integrates 10 recovery methods
    - Supports Java and C
  - **Architecture change metrics**
    - System level
    - Component level
  - Architectural smell detection
  - Decay metrics
  - Mining implementation issues
Architectural Change Metrics

- **a2a**: architecture-to-architecture metric
  - System-level
  - A **distance measure** between two architectures

- **cvg**: cluster coverage metric
  - Component-level
  - The extent to which certain components **existed in an earlier version** of a system or were **added in a later version**
Empirical Study Setup

• Research questions
  – In what ways do architectures change at the **system level** and at the **component level**?
  – Do architecture changes at the **system** and **component** levels **occur concurrently**?
  – When does significant **architecture change** occur?

• Study architecture change in
  – 14 Apache open-source software systems (FOSS)
  – 572 versions (analysis based on **Major.Minor.Patch** versioning scheme)
  – 3 architecture recovery techniques
### Subject Systems

<table>
<thead>
<tr>
<th>System</th>
<th>Application Domain</th>
<th>Versions</th>
<th>Time</th>
<th>MSLOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>ActiveMQ</td>
<td>Message Broker</td>
<td>20</td>
<td>8/04-12/05</td>
<td>3.4</td>
</tr>
<tr>
<td>Cassandra</td>
<td>Distributed DBMS</td>
<td>127</td>
<td>9/09-9/13</td>
<td>22.0</td>
</tr>
<tr>
<td>Chukwa</td>
<td>Data Monitoring</td>
<td>7</td>
<td>5/09-2/14</td>
<td>2.2</td>
</tr>
<tr>
<td>Hadoop</td>
<td>Data Processing</td>
<td>63</td>
<td>4/06-8/13</td>
<td>30.0</td>
</tr>
<tr>
<td>Ivy</td>
<td>Dependency Manager</td>
<td>20</td>
<td>12/07-2/14</td>
<td>0.4</td>
</tr>
<tr>
<td>JackRabbit</td>
<td>Content Repository</td>
<td>97</td>
<td>8/04-2/14</td>
<td>34.0</td>
</tr>
<tr>
<td>Jena</td>
<td>Semantic Web Framework</td>
<td>7</td>
<td>6/12-9/13</td>
<td>2.7</td>
</tr>
<tr>
<td>JSPWiki</td>
<td>Wiki Engine</td>
<td>54</td>
<td>10/07-3/14</td>
<td>1.2</td>
</tr>
<tr>
<td>Log4j</td>
<td>Logging</td>
<td>41</td>
<td>01/01-06/14</td>
<td>2.4</td>
</tr>
<tr>
<td>Lucene</td>
<td>Search Engine</td>
<td>21</td>
<td>12/10-1/14</td>
<td>5.1</td>
</tr>
<tr>
<td>Mina</td>
<td>Network Framework</td>
<td>40</td>
<td>11/06-11/12</td>
<td>2.3</td>
</tr>
<tr>
<td>PDFBox</td>
<td>PDF Library</td>
<td>17</td>
<td>2/08-3/14</td>
<td>2.7</td>
</tr>
<tr>
<td>Struts</td>
<td>Web Apps</td>
<td>36</td>
<td>3/00-2/14</td>
<td>6.7</td>
</tr>
<tr>
<td>Xerces</td>
<td>XML Library</td>
<td>22</td>
<td>3/03-11/09</td>
<td>2.3</td>
</tr>
</tbody>
</table>
Recovery Techniques

- **PKG** – package structure recovery

- **ACDC*** – algorithm for comprehension-driven clustering

- **ARC**** – architecture recovery using concerns

* V. Tzerpos et al., *ACDC: an algorithm for comprehension-driven clustering*, In Working Conference on Reverse Engineering (WCRE), 2000

** J. Garcia et al., *Enhancing architectural recovery using concerns*, In International Conference on Automated Software Engineering (ASE), 2011
RQ1 – How Architecture Changes

- **Different views** of the system architecture **complement** each other

Value unit is percentage
Lower numbers mean more change

On average, architecture changes range from 15-25%
RQ2- System vs. Component Level

- **Architecture changes** occur **within software components** even when the system’s overall architectural structure remains relatively stable.

Architectural similarity between minor versions of “Ivy”

In ARC view, architecture changes more than 80% within components.
RQ3 – When Significant Change Occurs

- Dramatic architecture change can occur across minor versions of a software system

Minimum a2a values between minor versions

Architecture Similarity

Architecture changes more than 50%
Summary

• The **largest empirical study** on architecture change in long-lived software systems using ARCADE, a novel **automated workbench** for software architecture recovery and analysis

• Important findings
  – FOSS versioning is **not an accurate indicator** of architecture change
  – Our study points to the **significance** of a **semantics-based** architectural perspective

• Future work
  – Leverage ARCADE to enable **prediction of architecture change**
  – Catalogue of **architectural smells and patterns indicating decay**
• ARCADE is available on request

• Two other ongoing projects related of ARCADE
  – ARCADE Runner: GUI front end for ARCADE
  – ATLAS: Automated Tailorable Large-scale Analysis of Software

• Contact
  – General questions: ducmle@usc.edu
  – Arcade Runner: dlink@usc.edu
  – ATLAS: pbehnamg@usc.edu