Framework
Information Based
Java Software
Architecture Recovery

Xiaocong Li, Li Zhang, Ning Ge
Beihang University, Beijing, China
2017-12-04
OUTLINE

- Introduction
- Problem Statement
- Our Solution
- Experiment & Result
- Conclusion & Future Works
INTRODUCTION
· Program comprehension is a major cost factor
· Lack of documents or the documents are out of date
· Architecture recovery methods are used
· Hierarchical clustering algorithm is used to divide source files into different modules
· The more features two entities have in common, the more likely they are in the same module
HIERARCHICAL CLUSTERING ALGORITHMS

- Examples
  - Single Linkage Algorithm (SLA)
  - Complete Linkage Algorithm (CLA)
  - Weighted Combined Algorithm (WCA)
  - scaLable InforMation BOttleneck (LIMBO)

- We want to improve the accuracy using framework information
  - Spring
  - Struts
  - Hibernate
  - MyBatis
HIERARCHICAL CLUSTERING ALGORITHMS

- Feature selection
- Similarity calculation
- Entities mergence
- Repetition
Use the Spring MVC as an example
# PROBLEM STATEMENT

The diagram illustrates the relationships between different components of a system, including entities, models, controllers, and HTTP utilities. The table below summarizes the occurrences of specific annotations in each component:

<table>
<thead>
<tr>
<th>Component</th>
<th>Log</th>
<th>HttpUtil</th>
<th>@Entity</th>
<th>@Controller</th>
<th>@Deprecated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Model2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Controller1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Controller2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

The diagram shows that the Log and HttpUtil are commonly used across different components, while the @Entity and @Controller annotations are more specific to certain components. The @Deprecated annotation is only present in Model1 and Controller1.
PROBLEM STATEMENT

<table>
<thead>
<tr>
<th></th>
<th>Log</th>
<th>HttpUtil</th>
<th>@Entity</th>
<th>@Controller</th>
<th>@Deprecated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Model2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Controller1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Controller2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
## PROBLEM STATEMENT

<table>
<thead>
<tr>
<th></th>
<th>Log</th>
<th>HttpSession</th>
<th>@Entity</th>
<th>@Controller</th>
<th>@Deprecated</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1</td>
<td>1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Model2</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Controller1</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>Controller2</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>
## Problem Statement

### Model-Controller Flow

<table>
<thead>
<tr>
<th></th>
<th>Model1</th>
<th>Model2</th>
<th>Controller1</th>
<th>Controller2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1</td>
<td>-</td>
<td>0.75</td>
<td>0.5</td>
<td>1.25</td>
</tr>
<tr>
<td>Model2</td>
<td>-</td>
<td>-</td>
<td>1.25</td>
<td>0.5</td>
</tr>
<tr>
<td>Controller1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.75</td>
</tr>
<tr>
<td>Controller2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
PROBLEM STATEMENT

- Module1
  - Model1
  - Controller1
- Module2
  - Model2
  - Controller2
PROBLEM STATEMENT

- Module1
  - Model1
  - Controller1

- Module2
  - Model2
  - Controller2
PROBLEM STATEMENT

- Module1
  - Model1
  - Controller1
- Module2
  - Model2
  - Controller2

- Model layer
  - Model1
  - Model2
- Controller layer
  - Controller1
  - Controller2
PROBLEM STATEMENT

- Module1
  - Model1
  - Controller1

- Module2
  - Model2
  - Controller2

- Model layer
  - Model1
  - Model2

- Controller layer
  - Controller1
  - Controller2
Use framework information

OUR SOLUTION
FRAMEWORK INFORMATION

- Frameworks
  - Spring
  - Struts
  - Hibernate
  - MyBatis

- Annotations: like @Entity

- Interfaces

- Super class

- Class name key words
FRAMEWORK INFORMATION

- Original Feature Vector
  \[ v_i = (v_i^1, v_i^2, ..., v_i^n) \]

- Framework Feature Weight
  \[ w_s = (w_s^1, w_s^2, ..., w_s^n) \]

- Weighted Feature Vector
  \[ wv_i = (w_s^1 v_i^1, w_s^2 v_i^2, ..., w_s^n v_i^n) \]
## USE FRAMEWORK INFORMATION

<table>
<thead>
<tr>
<th>Model1</th>
<th>Model2</th>
<th>Controller1</th>
<th>Controller2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Log</td>
<td>Log</td>
<td>Log</td>
<td>Log</td>
</tr>
<tr>
<td>HttpUtil</td>
<td>HttpUtil</td>
<td>HttpUtil</td>
<td>HttpUtil</td>
</tr>
<tr>
<td>@Entity</td>
<td>@Entity</td>
<td>@Controller</td>
<td>@Controller</td>
</tr>
<tr>
<td>@Deprecated</td>
<td>@Deprecated</td>
<td>@Deprecated</td>
<td>@Deprecated</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td></td>
<td>Log</td>
<td>HttpUtil</td>
<td>@Entity</td>
</tr>
<tr>
<td>----------------</td>
<td>------</td>
<td>----------</td>
<td>---------</td>
</tr>
<tr>
<td>Model1</td>
<td>1</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Model2</td>
<td>0</td>
<td>0</td>
<td>4</td>
</tr>
<tr>
<td>Controller1</td>
<td>1</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>Controller2</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
</tbody>
</table>
USE FRAMEWORK INFORMATION

<table>
<thead>
<tr>
<th></th>
<th>Model1</th>
<th>Model2</th>
<th>Controller1</th>
<th>Controller2</th>
</tr>
</thead>
<tbody>
<tr>
<td>Model1</td>
<td>-</td>
<td>0.75</td>
<td>2.00</td>
<td>2.75</td>
</tr>
<tr>
<td>Model2</td>
<td>-</td>
<td>-</td>
<td>2.75</td>
<td>2.00</td>
</tr>
<tr>
<td>Controller1</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>0.75</td>
</tr>
<tr>
<td>Controller2</td>
<td>-</td>
<td>-</td>
<td>-</td>
<td>-</td>
</tr>
</tbody>
</table>
PROBLEM STATEMENT

- Model layer
  - Model1
  - Model2

- Controller layer
  - Controller1
  - Controller2
PROBLEM STATEMENT

- Model layer
  - Model1
  - Model2

- Controller layer
  - Controller1
  - Controller2
EXPERIMENT AND RESULT
EXPERIMENT DESIGN

- Select projects using the Spring Framework on GitHub
- Select framework specific features
- Run the algorithm with different feature weights
- Compare results with original architecture
<table>
<thead>
<tr>
<th>Project</th>
<th>Modules</th>
<th>File Number</th>
<th>LOC</th>
</tr>
</thead>
<tbody>
<tr>
<td>act</td>
<td>6</td>
<td>39</td>
<td>5158</td>
</tr>
<tr>
<td>cms</td>
<td>5</td>
<td>40</td>
<td>4264</td>
</tr>
<tr>
<td>gen</td>
<td>5</td>
<td>18</td>
<td>2005</td>
</tr>
<tr>
<td>oa</td>
<td>4</td>
<td>16</td>
<td>1669</td>
</tr>
<tr>
<td>sys</td>
<td>8</td>
<td>37</td>
<td>5025</td>
</tr>
<tr>
<td>SpringBlog</td>
<td>9</td>
<td>45</td>
<td>2100</td>
</tr>
</tbody>
</table>
SELECTED FEATURES

- Usage of @Controller Annotation
- Usage of @RestController Annotation
- Usage of @Service Annotation
- Usage of @Entity Annotation
- Implementation of Serializable Interface
- Implementation of JpaRepository Interface
ACCURACY CALCULATION

- Initial architecture
  \[ M_{\text{init}} = \{M_{\text{init}}^1, M_{\text{init}}^2, \ldots, M_{\text{init}}^n\} \]

- Recovered architecture
  \[ M_{\text{new}} = \{M_{\text{new}}^1, M_{\text{new}}^2, \ldots, M_{\text{new}}^n\} \]

- Accuracy
  \[ \text{Similarity}[M_{\text{init}}, M_{\text{new}}] = \frac{\sum_{i=0}^{n} |M_{\text{init}}^i \cap M_{\text{new}}^i|}{\sum_{i=0}^{n} |M_{\text{init}}^i|} \]
RESULT

ACCURACY

WEIGHT OF DOMAIN FEATURES

act  cms  gen  oa  sys  SpringBlog
RESULT

ACCURACY

WEIGHT OF DOMAIN FEATURES

act  gen  oa  sys  SpringBlog
RESULT

ACCURACY vs WEIGHT OF DOMAIN FEATURES

- act
- cms
- gen
- oa
- sys
- SpringBlog
RESULT

![Graph showing accuracy vs weight of domain features]

**Accuracy**

- 0.4
- 0.45
- 0.5
- 0.55
- 0.6
- 0.65
- 0.7
- 0.75
- 0.8

**Weight of Domain Features**

- 0
- 1
- 10
- 100

**CMS**
CONCLUSION AND FUTURE WORKS
CONCLUSION & FUTURE WORKS

・Framework information can be used to improve architecture recovery algorithm

・Future works
  ・More features should be used
  ・Recommend the weight using machine learning
  ・Apply the method to other systems
  ・Large-scale system should be tested
Q&A