Comparing the Decompositions Produced by Software Clustering Algorithms using Similarity Measurements

2001 IEEE International Conference on Software Maintenance (ICSM'01).

Brian S. Mitchell & Spiros Mancoridis
Math & Computer Science, Drexel University
Motivation

Using module dependencies when determining the similarity between two decompositions is a good idea...
Clustering the Structure of a System (1)

Given the structure of a system...
Clustering the Structure of a System (2)

The goal is to partition the system structure graph into clusters...

The clusters should represent the subsystems.
But how do we know that the clustering result is good?
Ways to Evaluate Software Clustering Results...

Given a software clustering result, we can:

- Assess it against a mental model
- Assess it against a benchmark standard

Techniques:

- Subjective Opinions
- Similarity Measurements
Example: How “Similar” are these Decompositions?

Blue Edges: Similarity still the same...

Green Edges: Similarity still the same...

Red Edges: Not as similar...

Conclusions: Once we add the red edges the similarity between $P_A$ and $P_B$ decreases
Observations

- Edges are important for determining the similarity between decompositions.
- Existing measurements don’t consider edges:
  - Precision / Recall (similarity)
  - MoJo (distance)

**Our idea:** Use the edges to determine similarity.
Research Objectives

- Create new similarity measurements that use dependencies (edges)
  - **EdgeSim** (similarity)
  - **MeCl** (distance)
- Evaluate the new similarity measurements against MoJo & Precision/Recall
- Use similarity measurements to support evaluation of software clustering results (see our WCRE’01 paper)
Example: How “Similar” are these Decompositions? 

Add Blue Edges: PR, MoJo, MeCl & EdgeSim unchanged.

Add Green Edges: PR, MoJo, MeCl & EdgeSim unchanged.

Add Red Edges: PR, MoJo unchanged. EdgeSim, MeCl reduced.
Definitions

**Internal/ Intra-Edge:** Edge within a cluster

**External/ Inter-Edge:** Edge between two clusters
EdgeSim Example

MDG

\[ \begin{align*}
P_A &= \begin{array}{c}
  a \\
  b \\
  c \\
  d \\
  e \\
  f \\
  g \\
  h \\
  i \\
  j \\
  k \\
  l
\end{array} \\
\end{align*} \]

\[ \begin{align*}
P_B &= \begin{array}{c}
  a \\
  b \\
  c \\
  d \\
  e \\
  f \\
  g \\
  h \\
  i \\
  j \\
  k \\
  l
\end{array} \\
\end{align*} \]
EdgeSim Example

Step 1:
Find Common Inter-
and Intra-Edges
### EdgeSim Example

#### MDG

![MDG Diagram](image)

#### Common Edge Weight

\[
\text{Common Edge Weight} = \frac{10}{19} = 53\%
\]

#### Total Edge Weight (\(P_A\))

#### Total Edge Weight (\(P_B\))
MeCl Example

MDG

P_A

P_B
MeCl Example
(A→B)
MeCl Example
($A_1 \cap B_1$)
MeCl Example
($A_2 \cap B_1$)
MeCl Example
\((A_1 \cap B_2)\)
MeCl Example
($A_2 \cap B_2$)
MeCl Example
($A_3 \cap B_2$)
MeCl Example
(A → B)
MeCl Example (A→B)

Newly Introduced Inter-Edges
MeCl Example
(B→A)
MeCl Example (B→A)
MeCl Example (B→A)

Newly Introduced Inter-Edges

Drexel University Software Engineering Research Group (SERG)
http://serg.mcs.drexel.edu
**MeCl Calculation**

**Inter-Edges Introduced**

MeCl(A → B):
( \{b,e\}, \{e,c\}, \{g,h\}, \{f,h\} )

MeCl(B → A):
( \{e,i\}, \{h,j\}, \{b,f\}, \{c,f\}, \{h,e\} )

\[
\text{MeCl} = 1 - \frac{\max_w(M_{A\rightarrow B}, M_{B\rightarrow A})}{\text{Total Edge Weight}}
\]

\[
\text{MeCl} = 1 - \frac{5}{19} = 73.7\%
\]
Similarity Measurement Recap

MoJ(\(P_1\)) = MoJ(\(P_2\)) = 87.5%

PR(\(P_1\)) = PR(\(P_2\)) = P:84.6%, R:68.7%, AVG\(_{PR}\) = 76.7%

Conclusion... \(P_1\) is equally similar to \(P_2\)
Similarity Measurement Recap

**A₁:**

- EdgeSim(P₁) = 77.8%
- MeCl(P₁) = 88.9%

**B₁:**

**P₁**

**A₂:**

- EdgeSim(P₂) = 58.3%
- MeCl(P₂) = 66.7%

**B₂:**

**P₂**

<table>
<thead>
<tr>
<th></th>
<th>M1</th>
<th>M2</th>
<th>M3</th>
<th>M4</th>
<th>M5</th>
<th>M6</th>
<th>M7</th>
<th>M8</th>
</tr>
</thead>
<tbody>
<tr>
<td>A₁</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>B₁</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Conclusion... P₁ is more similar than P₂
Summary: EdgeSim & MeCl

EdgeSim:
- Rewards clustering algorithms for preserving the edge types
- Penalizes clustering algorithms for changing the edge types

MeCl:
- Rewards the clustering algorithm for creating cohesive “subclusters”
Special Modules

**Omnipresent Modules:**
“Strong” Connection to other Modules

**Library Modules:**
Always used by other modules, never use other modules

**Isomorphic Modules:**
Modules equally connected to other subsystems
Special Modules

Special Treatment of Special Modules helps to determine the Similarity

Omnipresent Modules: Removed

Library Modules: Removed

Isomorphic Modules: Replicated
Case Study Overview

Source Code

```c
void main()
{
    printf("hello");
}
```

Clustering Algorithms

Clustered Result

![Clustered Result Diagram]

Similarity Evaluation Tool

- Precision/Recall
- MoJo
- EdgeSim
- MeCl

Similarity Analysis

Average, Variance, etc. based on 100 clustering runs...
(4950 Evaluations)
Case Study Observations

- All similarity measurements exhibit consistent behavior for the systems studied:
  
  For all systems examined:
  If $\text{MeCl}(S_A) < \text{MeCl}(S_B)$ then $\text{MoJo}(S_A) < \text{MoJo}(S_B)$,
  $\text{PR}(S_A) < \text{PR}(S_B)$, and $\text{EdgeSim}(S_A) < \text{EdgeSim}(S_B)$

- Removal of “special” modules improved all similarity measurements

- Treating isomorphic modules specially only improved similarity slightly

- EdgeSim and MeCl produced higher and less variable similarity values than Precision/Recall and MoJo
Questions

Special Thanks To:
- AT&T Research
- Sun Microsystems
- DARPA
- NSF
- US Army