Change Propagation for Assessing Design Quality of Software Architectures

Lane Department of Computer Science, West Virginia University

Ali Mili
College of Computing Sciences
New Jersey Institute of Technology

C. Fuhrman
Department of Software and IT Engineering
Ecole de technologie supérieure

Supported by the NSF through ITR program, and by NASA through Office of Safety and Mission Assurance (OSMA) Software Assurance Research Program (SARP) managed through the NASA IV&V Facility, Fairmont, West Virginia

8 November 2005 WICS A 2005, Pittsburgh PA.
Outline

- Introduction
- Change Propagation Probabilities (CP)
- Methodology and Rationale
- Case Studies
- Comparing CP w.r.t. other OO Metrics
- Conclusion and Future Work
Introduction

- The study of Quantitative Assessment of software architectures is gaining importance due to its role in assessing the quality of architecture enhancements.
- With the increasing emphasis on design patterns, the traditional practice of ad-hoc software construction is slowly shifting towards pattern-oriented development.
Introduction

- In this paper, we show that change propagation probability (CP) is helpful and effective in assessing the design quality enhancements of software architectures.
- We study two different architectures (one that employs patterns versus one that does not) for the same application.
- We analyze and compare change propagation metric with respect to other coupling-based metrics.
Outline

- Introduction
- Change Propagation Probabilities (CP)
- Methodology and Rationale
- Case Studies
- Comparing CP w.r.t. other OO Metrics
- Conclusion and Future Work
Change Propagation Probability matrix

$CP = [cp_{ij}]$

cp_{ij} is the conditional probability that a change in $C_i$ due to corrective/ perfective maintenance requires a change in $C_j$ while maintaining the overall function of a system $S$

$$cp_{ij} = P([C_j] \neq [C_j'] \mid [C_i] \neq [C_i'] \land [S] = [S'])$$
Change Propagation: Usage

<table>
<thead>
<tr>
<th></th>
<th>C1</th>
<th>C2</th>
<th>C3</th>
<th>C4</th>
<th>C5</th>
<th>C6</th>
<th>C7</th>
<th>C8</th>
<th>C9</th>
<th>C10</th>
</tr>
</thead>
<tbody>
<tr>
<td>A</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>C1</td>
<td>1</td>
<td>0.11</td>
<td>0.15</td>
<td>0.3</td>
<td>0.76</td>
<td>0.73</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C2</td>
<td>0.37</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.52</td>
<td>0</td>
</tr>
<tr>
<td>C3</td>
<td>0.03</td>
<td>0.32</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C4</td>
<td>0.03</td>
<td>0.19</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C5</td>
<td>0</td>
<td>0.67</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C6</td>
<td>0</td>
<td>0.28</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C7</td>
<td>0.32</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C8</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>C9</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>C10</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
</tr>
</tbody>
</table>

Component B (does it have to change?)

Probability that B changes given that A does

Component A (where change originates)

alarm!!!
Change Propagation: Usage

<table>
<thead>
<tr>
<th>Component A (where change originates)</th>
<th>Probability that B changes given that A does</th>
<th>Component B (does it have to change?)</th>
</tr>
</thead>
<tbody>
<tr>
<td>C1</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>C2</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>C3</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>C4</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>C5</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>C6</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>C7</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>C8</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>C9</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
<tr>
<td>C10</td>
<td>![Image]</td>
<td>![Image]</td>
</tr>
</tbody>
</table>

8 November 2005 WICSA 2005, Pittsburgh PA.
Estimating Change Propagation

- $c_{pij}$ is estimated by

$$c_{pij} = \frac{1}{|V_i|} \sum_{v \in V_i} \pi_{ij}^{v}$$
Change Propagation Coefficient (CPC)

- The Change Propagation Coefficient $CPC$ is a scalar that reflects the potential of an architecture to insulate its components from each other’s changes.
- The idealistic change propagation coefficient corresponds to an identity matrix $I$.
- At the other extreme, the worst possible $CPC$ is one for which all cells of the CP matrix are 1s.

$$CPC = \sum_{i} \sum_{j \neq i} cp_{ij} \frac{1}{N^2 - N}$$
Outline

- Introduction
- Change Propagation Probabilities (CP)
- Methodology and Rationale
- Case Studies
- Comparing CP w.r.t. other OO Metrics
- Conclusion and Future Work
Methodology and Rationale

- Enhance an architecture with a design pattern.
- Compute CP metric on both architectures before and after enhancement.
- Compute object oriented metrics like CBO, RFC, MPC,.. etc on these architectures.
- Analyze and compare results.
Outline

- Introduction
- Change Propagation Probabilities (CP)
- Methodology and Rationale
- Case Studies
- Comparing CP w.r.t. other OO Metrics
- Conclusion and Future Work
Case Study-Job Application
Class Diagram before applying pattern
Case Study-Job Application
Class Diagram after applying strategy pattern
CP of Job Application before applying strategy pattern.

CPC = 0.18
CP of Job Application after applying strategy pattern

CPC = 0.11
Case Study- Colleague States
Class Diagram before applying mediator pattern
Case Study- Colleague States
Class Diagram after applying mediator pattern
CP for Colleague States before applying the mediator pattern

CPC = 0.11
CP for Colleague States after applying the mediator pattern

CPC = 0.05
Outline

- Introduction
- Change Propagation Probabilities (CP)
- Methodology and Rationale
- Case Studies
- Comparing CP w.r.t. other OO Metrics
- Conclusion and Future Work
CBO metric for the two case studies
RFC metric for the two case studies

![Graph showing RFC metric for two case studies](image-url)
MPC metric for the two case studies
Outline

- Introduction
- Change Propagation Probabilities (CP)
- Methodology and Rationale
- Case Studies
- Comparing CP w.r.t. other OO Metrics
- Conclusion and Future Work
Conclusion

- In this paper we presented
  - the applicability of change propagation (CP) in assessing design quality of software architectures.
  - The different perspective provided by CP with respect to other OO-metrics.
- This study is conducted as part of a larger project exploring a wide range of architecture-level attributes, including Error Propagation Probabilities, and Requirements Propagation Probabilities
Future Work

- We plan to
  - Study larger case studies to validate applicability of change propagation CP.
  - Automate the steps of the methodology.
  - Add more architectural attributes, other than change propagation, such as error propagation, coupling and cohesion, diagonality, ……etc in the methodology.
WMC and MCC metrics for case study - Colleague States
WMC and MCC metrics for case study-Job Application