Least modification principle for case-based reasoning: a software project planning experience

Jae Kyu Lee, Nobok Lee
2006-6-7
Presenter: Dongwon Kang
Contents

- Introduction
- Background
- Overview: CBR approach for software project planning
- Case representation
- Case retrieval / reuse
- Performance evaluation
- Related work
- Conclusion
- Discussion
Introduction

- Software project planning is a complex process
  - Require field knowledge and experience
    • Case Based Reasoning (CBR) is widely used
- Modification of a past case is hard to automate
  - Software project plan vary according to
    • Software development methodology
    • Style of the project manager

Goal: Reduce modification effort of project planners

• Suggest Least Modification Principle (LMP)
Background

- **Case Based Reasoning (CBR)**
  - Problem solving method by utilizing similar past cases
  - Steps for CBR
    - **Case representation**
      - Decide factors for describing case contents
      - Decide appropriate structure for effective storage and retrieval
    - **Case retrieval**
      - Decide how to measure similarity
    - **Case reuse**
      - Modify the selected case to adjust to the new requirements
    - **Case revision**
      - Evaluate the case solution resulting from applying solution and repair the solution
    - **Case retain**
      - Integrate the case in the structure of the case base
Overview: CBR approach for software project planning

1. Define a New Project
2. Retrieve the case that minimize modification
3. Modify the case for the new project
4. Enroll the new project plan

Case representation
- Project specific information
- Case index
- Past cases
- Modification effort
- Modification rules
- Add a new rule, if necessary
- Case base

Case retrieval
- Selected past case
- A new project plan

Case reuse
- Case representation
- Case index
- Case base

Case retain
- Add a new case
- Interactive modification
- Project specific information

Software Engineering Lab, KAIST
Case representation (1/3)

- **Elements of case representation**
  - Project activity network
    - Consist of Phases / Activities / Tasks
      - Phases fixed as Analysis, Design, Construction, Testing, Installation
    - Example

```
• Analysis
  • Systems Investigation
    • Investigate current system data
    • Identify current system problems
  • Requirement Definition
    • Define functional requirements
    • Define external system interface
```

← Phase
← Activity
← Task
## Elements of case representation (Cont’d)

- Factors related to the shape of project network

<table>
<thead>
<tr>
<th>Category</th>
<th>Factors</th>
<th>Alternative values</th>
<th>Category</th>
<th>Factors</th>
<th>Alternative values</th>
</tr>
</thead>
<tbody>
<tr>
<td>Adopted technology</td>
<td>Analysis of current system</td>
<td>Necessary/Unnecessary</td>
<td>Staff expertise</td>
<td>Application domain</td>
<td>Logistics / Banking / HRM / C3I / Medicine</td>
</tr>
<tr>
<td></td>
<td>Business process reengineering</td>
<td>Necessary/Unnecessary</td>
<td></td>
<td>Development methodology</td>
<td>Structured/OO/CBD</td>
</tr>
<tr>
<td></td>
<td>Structuredness of requirement</td>
<td>Structured/Unstructured</td>
<td></td>
<td>OS platform</td>
<td>UNIX, Guardian, MVS</td>
</tr>
<tr>
<td></td>
<td>Maturity of technology</td>
<td>Proven / New</td>
<td>(experienced / not-experienced)</td>
<td>Programming Language</td>
<td>C / Cobol / Java / MVC++</td>
</tr>
<tr>
<td></td>
<td>Data migration</td>
<td>Necessary/Unnecessary</td>
<td></td>
<td>CASE tool</td>
<td>COOL:Gen, ROSE, System Architect</td>
</tr>
<tr>
<td></td>
<td>System architecture</td>
<td>Terminal-Host / Web-server / Client-Server</td>
<td></td>
<td>Middleware</td>
<td>Inprise / Weblogic / Tmax / Tuxedo / Top-end</td>
</tr>
<tr>
<td></td>
<td>Development methodology</td>
<td>Structured/OO/CBD</td>
<td></td>
<td>DBMS</td>
<td>Oracle / Informix / Sybase / Tandem</td>
</tr>
<tr>
<td></td>
<td>Focus of prototyping</td>
<td>UI/Performance/Feasibility/Unnecessary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Case tool</td>
<td>Necessary/Unnecessary</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Building reusable component</td>
<td>Necessary/Unnecessary</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
Case representation (3/3)

- Steps for deriving factors in this work
  - Review the literature
    - Software effort estimation model
    - Software productivity model
    - Software process model
  - Screen factors based on the experts’ opinion
    - Select factors with 80% agreement
  - Consider factors based on the 31 real cases
  - Derive factors relevant to the modification rules by the author
    - Screen out factors not influencing the shape of project network
Modification rules

- Rules to transform past cases into another (new) case for each factor
  - Derived 69 rules from logical relationship and experience
    - Add/ delete/ replace operation for activities and tasks
  - Assumption : factors are mutually independent

<table>
<thead>
<tr>
<th>Rule [F6; Client/Server-to-Web/Server]</th>
</tr>
</thead>
<tbody>
<tr>
<td>New: System_architecture = Web/Server</td>
</tr>
<tr>
<td>Past: System_architecture = Client/Server</td>
</tr>
<tr>
<td>Action:</td>
</tr>
<tr>
<td>ADD_TASK    Design_web_page TO Preliminary_Design</td>
</tr>
<tr>
<td>ADD_TASK    Define_web_page_design_standards TO Detail_Design</td>
</tr>
<tr>
<td>ADD_TASK    Design_security_system_architecture TO Detail_Design</td>
</tr>
<tr>
<td>DELETE_TASK Design_client_modules TO Detail_Design</td>
</tr>
<tr>
<td>DELETE_TASK Code_and_compile_client_programs FROM Construction_and_Unit_Testing</td>
</tr>
<tr>
<td>DELETE_TASK Select_client_programs_distribution_tool FROM Construction_and_Unit_Testing</td>
</tr>
<tr>
<td>ADD_TASK    Install_web_application_server TO Construction_and_Unit_Testing</td>
</tr>
<tr>
<td>ADD_TASK    Code_and_compile_web_application_programs TO Construction_and_Unit_Testing</td>
</tr>
<tr>
<td>DELETE_TASK Distribute_client_programs FROM Installation_and_Handover</td>
</tr>
<tr>
<td>DELETE_TASK Install_client_programs FROM Installation_and_Handover</td>
</tr>
</tbody>
</table>
Case retrieval / reuse (2/3)

- Least Modification Principle
  - Used to select the most similar case
    - Retrieve the case minimizing manual modification effort
    - Modification effort is defined by modification rules derived by authors
      - By counting modification actions in the applied modification rules
        - If a value in the new project exists in the rules
        - By averaging modification efforts in the rules related to the factor
          - If a value in the new project doesn’t exist in the rules
Case retrieval / reuse (3/3)

- Least Modification Principle (Cont’d)
  - Example

<table>
<thead>
<tr>
<th>Rule [F₆; Client/Server-to-Web/Server]</th>
</tr>
</thead>
<tbody>
<tr>
<td>New: System_architecture = Web/Server</td>
</tr>
<tr>
<td>Past: System_architecture = Client/Server</td>
</tr>
<tr>
<td>Action:</td>
</tr>
<tr>
<td>ADD_TASK Design_web_page TO Preliminary_Design</td>
</tr>
<tr>
<td>ADD_TASK Define_web_page_design_standards TO Detail_Design</td>
</tr>
<tr>
<td>ADD_TASK Design_security_system_architecture TO Detail_Design</td>
</tr>
<tr>
<td>DELETE_TASK Design_client_modules TO Detail_Design</td>
</tr>
<tr>
<td>DELETE_TASK Code_and_compile_client_programs FROM Construction_and_Unit_Testing</td>
</tr>
<tr>
<td>DELETE_TASK Select_client_programs_distribution_tool FROM Construction_and_Unit_Testing</td>
</tr>
<tr>
<td>ADD_TASK Install_web_application_server TO Construction_and_Unit_Testing</td>
</tr>
<tr>
<td>ADD_TASK Code_and_compile_web_application_programs TO Construction_and_Unit_Testing</td>
</tr>
<tr>
<td>DELETE_TASK Distribute_client_programs FROM Installation_and_Handover</td>
</tr>
<tr>
<td>DELETE_TASK Install_client_programs FROM Installation_and_Handover</td>
</tr>
</tbody>
</table>

Modification effort for F₆ = 10

If the value of F₆ is not in the rule set, modification effort for F₆ is 27 (average value from six rules)
Performance evaluation

Evaluation method

– Compare LMP, Factor Matching Principle (FMP), and Actual Minimum by applying 31 real cases

• LMP
  – Applied at the factor level (average modification effort)

• Factor Matching Principle (FMP)
  – Select the cases matching to factors of the new case in the most times

• Actual Minimum
  – From modification rules
    • LMP with value level (counting the number of modification actions)
### Result

<table>
<thead>
<tr>
<th>Test Case</th>
<th>LM</th>
<th>ME*</th>
<th>MRE(%)</th>
<th>FM</th>
<th>Selected Cases</th>
<th>ME*</th>
<th>MRE(%)</th>
<th>Actual Minimum</th>
</tr>
</thead>
<tbody>
<tr>
<td>Case N1</td>
<td>Case 22</td>
<td>50</td>
<td>0</td>
<td>Case 7(53), 19(75), 9(74), 29(50)</td>
<td>67</td>
<td>34</td>
<td>Case 22</td>
<td></td>
</tr>
<tr>
<td>Case N2</td>
<td>Case 29</td>
<td>50</td>
<td>0</td>
<td>Case 29(50), 19(86), 8(64), 4(101), 38(99)</td>
<td>80</td>
<td>60</td>
<td>Case 29</td>
<td></td>
</tr>
<tr>
<td>Case N3</td>
<td>Case 11</td>
<td>45</td>
<td>0</td>
<td>Case 1(45), 31(85)</td>
<td>65</td>
<td>44</td>
<td>Case 11</td>
<td></td>
</tr>
<tr>
<td>Case N4</td>
<td>Case 25</td>
<td>48</td>
<td>0</td>
<td>Case 10(74), 2(101)</td>
<td>88</td>
<td>83</td>
<td>Case 25</td>
<td></td>
</tr>
<tr>
<td>Case N5</td>
<td>Case 9</td>
<td>59</td>
<td>0</td>
<td>Case 9(59), 3(75)</td>
<td>66</td>
<td>12</td>
<td>Case 9</td>
<td></td>
</tr>
<tr>
<td>Case N6</td>
<td>Case 20</td>
<td>48</td>
<td>0</td>
<td>Case 13(54)</td>
<td>54</td>
<td>13</td>
<td>Case 20</td>
<td></td>
</tr>
<tr>
<td>Case N7</td>
<td>Case 24, Case 22</td>
<td>41</td>
<td>8</td>
<td>Case 24, Case 22(38), 19(48)</td>
<td>45</td>
<td>13</td>
<td>Case 22</td>
<td></td>
</tr>
<tr>
<td>Case N8</td>
<td>Case 17</td>
<td>43</td>
<td>13</td>
<td>Case 17(43), 22(38), 12(51), 15(56), 2(64), 106(73)</td>
<td>56</td>
<td>47</td>
<td>Case 29</td>
<td></td>
</tr>
<tr>
<td>Case N9</td>
<td>Case 25</td>
<td>38</td>
<td>0</td>
<td>Case 9(59), 14(57), 4(76), 1(74), 37(99)</td>
<td>73</td>
<td>92</td>
<td>Case 25</td>
<td></td>
</tr>
<tr>
<td>Case N10</td>
<td>Case 20</td>
<td>59</td>
<td>28</td>
<td>Case 20(59), 6(64), 15(61), 16(54)</td>
<td>60</td>
<td>30</td>
<td>Case 19</td>
<td></td>
</tr>
<tr>
<td>Case N11</td>
<td>Case 3</td>
<td>47</td>
<td>0</td>
<td>Case 7(70)</td>
<td>70</td>
<td>49</td>
<td>Case 3</td>
<td></td>
</tr>
<tr>
<td>Case N12</td>
<td>Case 20</td>
<td>32</td>
<td>0</td>
<td>Case 8(51)</td>
<td>51</td>
<td>59</td>
<td>Case 20</td>
<td></td>
</tr>
<tr>
<td>Case N13</td>
<td>Case 22</td>
<td>33</td>
<td>0</td>
<td>Case 19(42), 32(55)</td>
<td>49</td>
<td>48</td>
<td>Case 22</td>
<td></td>
</tr>
<tr>
<td>Case N14</td>
<td>Case 16</td>
<td>63</td>
<td>0</td>
<td>Case 9(57), 7(70), 13(75)</td>
<td>67</td>
<td>6</td>
<td>Case 9</td>
<td></td>
</tr>
<tr>
<td>Case N15</td>
<td>Case 21</td>
<td>40</td>
<td>0</td>
<td>Case 8(56), 17(75)</td>
<td>66</td>
<td>65</td>
<td>Case 21</td>
<td></td>
</tr>
<tr>
<td>Case N16</td>
<td>Case 19</td>
<td>43</td>
<td>0</td>
<td>Case 19(43)</td>
<td>43</td>
<td>0</td>
<td>Case 19</td>
<td></td>
</tr>
<tr>
<td>Case N17</td>
<td>Case 8</td>
<td>43</td>
<td>0</td>
<td>Case 15(75)</td>
<td>75</td>
<td>74</td>
<td>Case 8</td>
<td></td>
</tr>
<tr>
<td>Case N18</td>
<td>Case 31</td>
<td>51</td>
<td>0</td>
<td>Case 3(51)</td>
<td>51</td>
<td>0</td>
<td>Case 31</td>
<td></td>
</tr>
<tr>
<td>Case N19</td>
<td>Case 13</td>
<td>22</td>
<td>0</td>
<td>Case 16(45)</td>
<td>43</td>
<td>95</td>
<td>Case 13</td>
<td></td>
</tr>
<tr>
<td>Case N20</td>
<td>Case 12</td>
<td>32</td>
<td>0</td>
<td>Case 12(32), 6(48), 10(59)</td>
<td>46</td>
<td>44</td>
<td>Case 12</td>
<td></td>
</tr>
<tr>
<td>Case N21</td>
<td>Case 15</td>
<td>40</td>
<td>0</td>
<td>Case 15(40)</td>
<td>40</td>
<td>0</td>
<td>Case 15</td>
<td></td>
</tr>
<tr>
<td>Case N22</td>
<td>Case 7</td>
<td>38</td>
<td>0</td>
<td>Case 7(38)</td>
<td>38</td>
<td>0</td>
<td>Case 7</td>
<td></td>
</tr>
<tr>
<td>Case N23</td>
<td>Case 31</td>
<td>42</td>
<td>0</td>
<td>Case 3(42), 26(59), 17(50), 24(96)</td>
<td>62</td>
<td>48</td>
<td>Case 31</td>
<td></td>
</tr>
<tr>
<td>Case N24</td>
<td>Case 7</td>
<td>43</td>
<td>0</td>
<td>Case 20(70)</td>
<td>70</td>
<td>63</td>
<td>Case 7</td>
<td></td>
</tr>
<tr>
<td>Case N25</td>
<td>Case 9</td>
<td>38</td>
<td>0</td>
<td>Case 9(38), 27(61)</td>
<td>50</td>
<td>32</td>
<td>Case 9</td>
<td></td>
</tr>
<tr>
<td>Case N26</td>
<td>Case 30</td>
<td>67</td>
<td>8</td>
<td>Case 30(67)</td>
<td>67</td>
<td>8</td>
<td>Case 27</td>
<td></td>
</tr>
<tr>
<td>Case N27</td>
<td>Case 9</td>
<td>49</td>
<td>0</td>
<td>Case 25(61)</td>
<td>61</td>
<td>24</td>
<td>Case 9</td>
<td></td>
</tr>
<tr>
<td>Case N28</td>
<td>Case 1</td>
<td>57</td>
<td>0</td>
<td>Case 24(70)</td>
<td>70</td>
<td>23</td>
<td>Case 1</td>
<td></td>
</tr>
<tr>
<td>Case N29</td>
<td>Case 8</td>
<td>38</td>
<td>0</td>
<td>Case 3(30)</td>
<td>50</td>
<td>32</td>
<td>Case 8</td>
<td></td>
</tr>
<tr>
<td>Case N30</td>
<td>Case 3</td>
<td>44</td>
<td>0</td>
<td>Case 26(67), 28(90)</td>
<td>79</td>
<td>80</td>
<td>Case 3</td>
<td></td>
</tr>
<tr>
<td>Case N31</td>
<td>Case 18</td>
<td>51</td>
<td>21</td>
<td>Case 18(51), 17(65), 29(56), 8(74), 9(99), 4(89), 2(84), 3(100)</td>
<td>77</td>
<td>83</td>
<td>Case 23</td>
<td></td>
</tr>
</tbody>
</table>

ME: Actual modification effort from the selected case
MRE: Mean of Relative Errors \( \sum (\text{Measured-Min})/\text{Min} \)
Performance evaluation (2/3)

- Result (Cont’d)
Performance evaluation (3/3)

- Statistical evaluation using paired t-test and Wilcoxon test
  - Hypothesis 1: LMP outperforms FM
    - Proved at the 1% level of significance
  - Hypothesis 2: Performance of LMP is as good as the actual minimum
    - Proved at the 1% level of significance

- Paired t-test
  - Method for comparing two experiments performed for the same group
- Wilcoxon test
  - Alternative method for paired t-test
    - When number of samples are not enough
    - When differences are not guaranteed for normal distribution
Related work

- **Project planning area**
  - Adaptable process model [Pressman, 2001]
    - Support process task set according to the degree of rigor
      - Causal / structured / strict / quick reaction
    - Not handle the weight between factors

- **Software effort estimation area**
  - Examining the feasibility of a case-based reasoning model for software effort estimation [Mukhopadhyay et al., 1992]
    - Use sum of square differences in five factors as similarity measure
Conclusion

- Least Modification principle
  - Method for estimating modification effort in software project planning
    - Consider reduction of project planners’ effort rather than syntactic distance of the traditional factor matching
      - Outperform traditional factor matching method
Discussion (1/4)

- Limitations
  - Unfair evaluation
    - Comparison with ‘real’ actual minimum is required
  - Immoderate assumption about weighting operators
    - Add/ delete/ replace operators may require different workload
    - Effort for modification of activities and tasks may be different
  - Additional effort for rule generation
    - A case from new domain should be compared with cases from each factors

- Future work
  - Consider other elements consisting the project network
    - Precedence among activities and tasks
    - Schedule and human resource allocation
Discussion (2/4)

- ADD 프로젝트: 국방 소프트웨어 프로세스 태일러링 연구
  - 프로세스를 컴포넌트화하고 이를 이용한 자동화된 테일러링 기법 지원에 관한 연구
  - Research goal:
    • 1단계: 컴포넌트 기반 소프트웨어 프로세스 태일러링 기술 확보
    • 2단계: 프로세스 기반의 일정 및 비용 예측 기술 확보
    • 3단계: 실제 개발사례에 적용하여 엔진 효과 검증 및 개선
Discussion (3/4)

* 1단계 수행 결과물을 이용한 프로세스 테일러링 과정

- Knowledge base
  - Project specific information
    - Project Manager
    - Interactive modification

- Process component base
  - Define a process for the new project
  - Retrieve the most similar process
  - Modify the case automatically for the new project
  - Manual tailoring guided by tailoring rules
  - Enroll the new process and knowledge

- Process component search engine
  - Similar process
  - Process components for tailoring
  - Tailoring knowledge
    - Guideline rules for manual tailoring
  - Tailoring knowledge and guideline rules for manual tailoring

- Knowledge base
  - New process

Software Engineering Lab, KAIST
Discussion (4/4)

1단계: 컴포넌트 기반 소프트웨어 프로세스 기초 기술 확보
- 1차년도: 프로세스 추론 기반 구조 완성
  - 사례 수집 및 분류 체계 개발
  - 유사성 인덱스 개발
- 2차년도: 과거 프로세스 지식화 기법 및 추론 기법 향상
  - 프로세스 컴포넌트 정의
  - 프로세스 컴포넌트 및 지식 베이스 구축
  - 프로세스 검색 엔진 구현
  - 유사 프로세스 컴포넌트 및 지식 추론엔진 구현
- 3차년도: 프로세스 수정 및 재사용 기반구조 완성
  - 시제 재구성 알고리즘 및 모듈 개발
  - 프로세스 테일러링 지원 방법론 개발
  - 프로세스 테일러링 지원도구 개발
Similarity measures

\[ \text{SIM}(C_o, C_i) = \sum_{j=1}^{n} [w_jD_j(f_{ij}, f_{oj})] \]

- \( \text{SIM}(C_o, C_i) \): Similarity between a new case \( C_o \) and a past case \( C_i, i=1..m \)
- \( w_j \): weight of factor \( j, j=1..n \)
- \( D_j(x,y) \): Distance function between the value \( x \) and \( y \) in factor \( j \)

- Tversky’s ordinary matching function

\[
\begin{align*}
D(f_{ij}, f_{oj}) = & \begin{cases} 
1 & \text{if } f_{ij} \neq f_{oj} \\
0 & \text{if } f_{ij} = f_{oj}
\end{cases}
\end{align*}
\]

- Nearest-neighbor matching function
  - Measure semantic distance considering domain knowledge
    - Modification effort as a measure of distance