Interactive scalable condensation of reverse engineered UML class diagrams for software comprehension
Osman, M.H.B.

Citation

Version: Not Applicable (or Unknown)
License: Leiden University Non-exclusive license
Downloaded from: https://hdl.handle.net/1887/32210

Note: To cite this publication please use the final published version (if applicable).
The handle http://hdl.handle.net/1887/32210 holds various files of this Leiden University dissertation.

**Author:** Osman, Mohd Hafeez Bin  
**Title:** Interactive scalable condensation of reverse engineered UML class diagrams for software comprehension  
**Issue Date:** 2015-03-10
In this Appendix, we list the candidate projects for our case studies (Table A.1). Item mark with * are the selected case studies.

Table A.1: List of Case Study Candidates

<table>
<thead>
<tr>
<th>No</th>
<th>Project</th>
<th>Source</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>ArgoUML*</td>
<td><a href="http://sourceforge.net/projects/argouml/">http://sourceforge.net/projects/argouml/</a></td>
</tr>
<tr>
<td>2</td>
<td>Bookszenbooks</td>
<td><a href="http://code.google.com/p/bookszenbooks/">http://code.google.com/p/bookszenbooks/</a></td>
</tr>
<tr>
<td>3</td>
<td>cmpt371t1</td>
<td><a href="http://code.google.com/p/cmpt371t1/">http://code.google.com/p/cmpt371t1/</a></td>
</tr>
<tr>
<td>4</td>
<td>Concurrentadt</td>
<td><a href="http://code.google.com/p/concurrentadt/">http://code.google.com/p/concurrentadt/</a></td>
</tr>
<tr>
<td>5</td>
<td>CrimsonPortal</td>
<td><a href="http://code.google.com/p/crimsonportal/">http://code.google.com/p/crimsonportal/</a></td>
</tr>
<tr>
<td>7</td>
<td>DocDoku</td>
<td><a href="http://code.google.com/p/docdoku/">http://code.google.com/p/docdoku/</a></td>
</tr>
<tr>
<td>8</td>
<td>driving-bc</td>
<td><a href="http://code.google.com/p/driving-bc/">http://code.google.com/p/driving-bc/</a></td>
</tr>
<tr>
<td>9</td>
<td>DRMJ-Webshop</td>
<td><a href="http://code.google.com/p/drmj-webshop/">http://code.google.com/p/drmj-webshop/</a></td>
</tr>
<tr>
<td>11</td>
<td>Epydoc</td>
<td><a href="http://epydoc.sourceforge.net/api/epydoc-module.html">http://epydoc.sourceforge.net/api/epydoc-module.html</a></td>
</tr>
<tr>
<td>12</td>
<td>europa-ps</td>
<td><a href="http://code.google.com/p/europa-ps/">http://code.google.com/p/europa-ps/</a></td>
</tr>
<tr>
<td>14</td>
<td>Fuge</td>
<td><a href="http://fuge.sourceforge.net/dev/index.php">http://fuge.sourceforge.net/dev/index.php</a></td>
</tr>
<tr>
<td>15</td>
<td>gelsanalyzer</td>
<td><a href="http://code.google.com/p/gelsanalyzer/downloads/detail?name=MVC%20Class%20Diagram%20Iteration2.png&amp;can=2&amp;q=">http://code.google.com/p/gelsanalyzer/downloads/detail?name=MVC%20Class%20Diagram%20Iteration2.png&amp;can=2&amp;q=</a></td>
</tr>
<tr>
<td>Case Study Candidates</td>
<td>URL</td>
<td></td>
</tr>
<tr>
<td>----------------------</td>
<td>-----</td>
<td></td>
</tr>
<tr>
<td>16 GNetWatch</td>
<td><a href="http://gnetwatch.sourceforge.net/doc.html">http://gnetwatch.sourceforge.net/doc.html</a></td>
<td></td>
</tr>
<tr>
<td>17 Google-voice-java</td>
<td><a href="http://code.google.com/p/google-voice-java/">http://code.google.com/p/google-voice-java/</a></td>
<td></td>
</tr>
<tr>
<td>18 Gotogate</td>
<td><a href="http://gotogate.googlecode.com/svn/trunk/">http://gotogate.googlecode.com/svn/trunk/</a></td>
<td></td>
</tr>
<tr>
<td>19 gwavmerger</td>
<td><a href="http://gwavmerger.sourceforge.net/gwm-design/design.html">http://gwavmerger.sourceforge.net/gwm-design/design.html</a></td>
<td></td>
</tr>
<tr>
<td>20 gwt-portlets*</td>
<td><a href="http://code.google.com/p/gwtportlets/">http://code.google.com/p/gwtportlets/</a></td>
<td></td>
</tr>
<tr>
<td>21 gwtuml</td>
<td><a href="http://code.google.com/p/gwtuml/">http://code.google.com/p/gwtuml/</a></td>
<td></td>
</tr>
<tr>
<td>22 httpdbase4j</td>
<td><a href="http://httpdbase4j.berlios.de/">http://httpdbase4j.berlios.de/</a></td>
<td></td>
</tr>
<tr>
<td>23 Jalli</td>
<td><a href="http://jalli.berlios.de/">http://jalli.berlios.de/</a></td>
<td></td>
</tr>
<tr>
<td>ral Network Library</td>
<td></td>
<td></td>
</tr>
<tr>
<td>26 JavaClient*</td>
<td><a href="http://java-player.sourceforge.net/">http://java-player.sourceforge.net/</a></td>
<td></td>
</tr>
<tr>
<td>27 JGAP*</td>
<td><a href="https://sourceforge.net/project/screenshots.php?group_id=11618">https://sourceforge.net/project/screenshots.php?group_id=11618</a></td>
<td></td>
</tr>
<tr>
<td>28 jjack</td>
<td><a href="http://jjack.berlios.de/">http://jjack.berlios.de/</a></td>
<td></td>
</tr>
<tr>
<td>29 jpmc*</td>
<td><a href="http://jpmc.sourceforge.net/diagrams.html">http://jpmc.sourceforge.net/diagrams.html</a></td>
<td></td>
</tr>
<tr>
<td>30 Jvending</td>
<td><a href="http://jvending.sourceforge.net/">http://jvending.sourceforge.net/</a></td>
<td></td>
</tr>
<tr>
<td>31 Krank</td>
<td><a href="http://code.google.com/p/krank/">http://code.google.com/p/krank/</a></td>
<td></td>
</tr>
<tr>
<td>32 Mandragora Project</td>
<td><a href="http://mandragora.sourceforge.net/referenceguide/how-to-extend.html">http://mandragora.sourceforge.net/referenceguide/how-to-extend.html</a></td>
<td></td>
</tr>
<tr>
<td>34 Maze-Solver*</td>
<td><a href="http://code.google.com/p/maze-solver/wiki/MazeModelDoc">http://code.google.com/p/maze-solver/wiki/MazeModelDoc</a></td>
<td></td>
</tr>
<tr>
<td>35 monopolj</td>
<td><a href="http://code.google.com/p/monopolj/">http://code.google.com/p/monopolj/</a></td>
<td></td>
</tr>
<tr>
<td>36 MyDas</td>
<td><a href="http://code.google.com/p/mydas/">http://code.google.com/p/mydas/</a></td>
<td></td>
</tr>
<tr>
<td>39 Neuroph*</td>
<td><a href="http://neuroph.sourceforge.net/">http://neuroph.sourceforge.net/</a></td>
<td></td>
</tr>
<tr>
<td>40 nmt-cs326-g5</td>
<td><a href="https://code.google.com/p/nmt-cs326-g5/">https://code.google.com/p/nmt-cs326-g5/</a></td>
<td></td>
</tr>
<tr>
<td>42 Nymp</td>
<td><a href="http://code.google.com/p/nymph/">http://code.google.com/p/nymph/</a></td>
<td></td>
</tr>
<tr>
<td>43 ObjectLabKit</td>
<td><a href="http://objectlabkit.sourceforge.net/apidocs/net/objectlab/kit/datecalc/jdk/CalendarPeriodCountCal-culator.html">http://objectlabkit.sourceforge.net/apidocs/net/objectlab/kit/datecalc/jdk/CalendarPeriodCountCal-culator.html</a></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Product</td>
<td>URL</td>
</tr>
<tr>
<td>---</td>
<td>--------------------</td>
<td>----------------------------------------------------------------------</td>
</tr>
<tr>
<td>45</td>
<td>OpenMeeting</td>
<td><a href="http://code.google.com/p/openmeetings/">http://code.google.com/p/openmeetings/</a></td>
</tr>
<tr>
<td>46</td>
<td>pacpounder</td>
<td><a href="https://code.google.com/p/pacpounder/">https://code.google.com/p/pacpounder/</a></td>
</tr>
<tr>
<td>47</td>
<td>Pendulim</td>
<td><a href="http://code.google.com/p/pendulim/">http://code.google.com/p/pendulim/</a></td>
</tr>
<tr>
<td>49</td>
<td>Primitive Collections</td>
<td><a href="http://pcj.sourceforge.net/docs/guide/pcj-guide.html">http://pcj.sourceforge.net/docs/guide/pcj-guide.html</a></td>
</tr>
<tr>
<td>50</td>
<td>Qt OIC Container 3.5</td>
<td><a href="http://qtioccontainer.sourceforge.net/uml.html">http://qtioccontainer.sourceforge.net/uml.html</a></td>
</tr>
<tr>
<td>51</td>
<td>RandyLoops</td>
<td><a href="http://randyloops.sourceforge.net/">http://randyloops.sourceforge.net/</a></td>
</tr>
<tr>
<td>52</td>
<td>rpcstruts</td>
<td><a href="https://code.google.com/p/rcpstruts/">https://code.google.com/p/rcpstruts/</a></td>
</tr>
<tr>
<td>56</td>
<td>wro4J*</td>
<td><a href="http://code.google.com/p/wro4j/">http://code.google.com/p/wro4j/</a></td>
</tr>
<tr>
<td>57</td>
<td>Xuml-compiler*</td>
<td><a href="http://code.google.com/p/xuml-compiler/">http://code.google.com/p/xuml-compiler/</a></td>
</tr>
</tbody>
</table>
List of Figures

1.1 Thesis Roadmap ......................................................... 7

2.1 Integrated Model [174] .................................................. 14
2.2 Cognitive Design Elements for Software Exploration [156] ........... 15
2.3 The Software Development Life Cycle .............................. 16
2.4 Relationship between Forward Eng., Reverse Eng. and Other Related Terms [41] ................................................................. 16
2.5 Taxonomy of UML Version 2.4 [68] .................................... 18
2.6 Tours Online Class Diagram (Domain Analysis) .................... 19
2.7 Tours Online Class Diagram (Design Level) ....................... 20
2.8 Tours Online RE-CD (Code Level) .................................. 20
2.9 The Process of Supervised Machine Learning [98] .................. 24
2.10 ROC and Precision-Recall Curves under Class Skew ............. 29

3.1 Classes in Design vs Classes in Implementation .................... 40
3.2 ArgoUML Evolution in UML Diagrams and Number of Classes .... 44

4.1 Round-trip Engineering Experiment .................................. 53
4.2 Altova Reverse Engineered Package Diagram ....................... 56
4.3 Reverse Engineered Sequence Diagram using Altova .................. 56
4.4 Round-trip Test Result ............................................... 58
4.5 Example of Diagram on Aggregation Test ......................... 61
4.6 Number of Attributes and Methods .................................. 61
4.7 Bidirectional Relationship with Two Separate Links ............... 63

5.1 Level of Detail Class Diagrams Preparation .......................... 74
5.2 Role of the Respondents ............................................. 76
5.3 Respondents Experience with Class Diagrams ........................ 76
5.4 Where did the Respondent Learn about UML .................................................. 77
5.5 Respondent’s skill on Class Diagram ............................................................... 78
5.6 Respondents Like or Dislike Source Code vs UML ......................................... 78
5.7 Programmers Like or Dislike Source Code vs UML ........................................ 79
5.8 Software Architects Like or Dislike Source Code vs UML ............................... 79
5.9 Software Designers Like or Dislike Source Code vs UML ................................. 79
5.10 Class Diagram Skill per Role ........................................................................... 79
5.11 Information of Attribute that Could be Left out .............................................. 81
5.12 Types of Operation that could be Excluded in Class Diagrams ......................... 81
5.13 Class Category ............................................................................................... 82
5.14 Types of Class that could not be Included in Class Diagrams .......................... 82
5.15 Class Role that could be Excluded in Class Diagrams ........................................ 83
5.16 Types of Information the Respondents Look for in Class Diagrams .................. 84
5.17 Information of Classes that could be Omitted ................................................... 85
5.18 Information of Operations that could be Omitted ............................................. 86
5.19 Important Criteria in a Class Diagram for Understanding a System .................... 87
5.20 The Type of Relationship in Class Diagrams that the Respondents Look at First ........................................................................................................... 87
5.21 The Features that a Tool Should have for Simplifying UML Class Diagrams .... 88

6.1 Role of the Respondents ..................................................................................... 99
6.2 Location of the Respondents ............................................................................ 99
6.3 Class Diagram Skill and Years of Experience .................................................... 100
6.4 Score Size Category (Question B1-B4) ............................................................ 101
6.5 Score Coupling Category (Question B5-B10) ..................................................... 102
6.6 Score Inheritance Category (Question B11-B13) .................................................. 103
6.7 Keywords to Include a Class in a Class Diagram .............................................. 103
6.8 Class Diagram A (ATM System) ....................................................................... 105
6.9 Respondents Selection of Classes that Should not be Included in an ATM System ........................................................................................................... 106
6.10 Class Diagram B (Library System) ................................................................. 107
6.11 Respondents Selection of Classes that should not be Included in a Library System ........................................................................................................... 108
6.12 Respondents Selection of Classes that should not be Included in a Pacman Game (Forward Design) ................................................................. 108
6.13 Pacman Game Forward Design (Class Diagram C) ........................................... 110
6.14 Reverse Engineered Pacman Game (Class Diagram D) ..................................... 111
6.15 Respondents Selection of Classes that should not be Included in a Pacman Game (Reverse Engineered Design) ................................................................. 112

7.1 Design Abstraction Process ............................................................................... 124
7.2 Average AUC Score for Every Dataset ............................................................ 128
7.3 AUC Score >= 0.60 ................................................. 130
7.4 Application of Random Forests Classification Algorithm. ........ 130

8.1 Overall Framework ............................................. 139

9.1 Overall Framework : Input, Process and Output .................... 159
9.2 Selection of the Candidate-Important Classes ....................... 160
9.3 The SAAbs Tool Displaying Ranking of Classes .................... 163
9.4 Textual Results of Classification ................................ 164
9.5 SAAbs tool Viewing Class+Package Diagram in Different Level of Abstraction ....................................................... 166
9.6 Highlighting of Less Important Classes ............................ 167
9.7 Greyscale Coloring: Less Important Classes with Darker Shades of Gray. 167

10.1 Flow of the Experiment ......................................... 177
10.2 Distribution of the Respondents ................................ 177
10.3 Respondents’ Experience with Class Diagram .................... 178
10.4 Skills of Understanding Class Diagram .......................... 178
10.5 The Understandability of Condensed Class Diagram ............ 179
10.6 The Respondents’ Role vs. Understandability Score ............ 180
10.7 The Respondents’ Experience vs. Understandability Score .... 180
10.8 Choices of Class Diagrams ...................................... 181
10.9 The Respondents’ Experience vs. Choice of Class Diagram .... 182
10.10Level of Abstraction of RE-CD for Software Comprehension ...... 183
10.11The Rating of the Usefulness of SAAbs Tool ..................... 183
10.12The Respondent’s Role vs the Tool’s Features .................. 184
10.13The Tool’s Features vs the Respondent’s Experience .......... 185
10.14The Respondent’s Skill vs the Tool’s Features .................. 185
10.15The Limitations of the SAAbs Framework and Tool ............. 186
# List of Tables

1.1 Research Methods used in this Research .......................... 6  
2.1 Definitions of Program Comprehension .......................... 12  
2.2 The nine classification algorithms ............................ 26  
2.3 Confusion Matrix or Contingency Table ....................... 27  
2.4 Common Performance Measures and Terms ..................... 27  
3.1 List of Case Studies ........................................ 35  
3.2 Levels of Detail in UML models ............................ 36  
3.3 UML Diagram Usage ........................................ 37  
3.4 Classes in Design versus Classes in Implementation .......... 40  
3.5 LoD and Forward/Reverse Class Diagram ....................... 41  
3.6 Add, Remove and Modify of UML Diagrams in ArgoUML Project .... 43  
3.7 List of UML Diagrams used in ArgoUML Project ............... 44  
4.1 List of Evaluated CASE Tools .................................. 51  
4.2 Supported UML Diagrams for Reverse Engineering .............. 55  
4.3 Supported Programming Language for Reverse Engineering .... 57  
4.4 Additional Types of Input Format ............................ 59  
4.5 Class Relationship Test Result ............................. 60  
4.6 Relationship Correctness .................................... 62  
5.1 Level of Detail Description .................................... 73  
5.2 Information on Set A and Set B .............................. 74  
5.3 Detailed Explanation Part C .................................. 75  
5.4 Keywords on Types of Information to Understand a System .... 84  
6.1 The Chosen Software Design Metrics [180] ..................... 95
6.2 Answers Multiple Choices Questions ........................................ 96
6.3 Description of the Class Diagrams Used in the Questions .......... 97
6.4 Choices of Answers for Question 4 ........................................ 97
6.5 Choices of Answers for Question 6 ........................................ 98
6.6 Total of Responses ............................................................... 98
6.7 Score-System Metrics - Question B1-B13 ............................... 101
6.8 The Preferences between Class Diagram A (C1), B (C2) and C (C3) .... 109
6.9 The Preference between Class Diagram C and D ......................... 113
6.10 Overall Score for Software Design Metrics ............................. 114
7.1 List of Class Diagram Metrics .................................................. 122
7.2 List of Case Study ............................................................... 123
7.3 Data Preparation Steps ......................................................... 125
7.4 Predictor Sets ................................................................. 126
7.5 Univariate Predictor Performance (Information Gain) ................. 128
7.6 Results for Predictor set C .................................................... 129
8.1 List of Case Study ............................................................... 140
8.2 The Top List of Common Words in Class Diagrams .................... 144
8.3 The Top List of UnCommon Words in Class Diagrams ................ 145
8.4 Information Gain Results for Text Predictors .......................... 149
8.5 Classification Algorithms Performance on Predictors Sets (AUC Score ≥ 0.60) ................................................................. 149
8.6 Random Forests Result for Predictors Sets ............................. 151
8.7 Classification Result from Text Predictor (T) ............................. 153
9.1 The List of Prediction Features .............................................. 162
10.1 Type of Class Diagram used in this study .............................. 175
11.1 Summary of Background Research Findings ............................ 190
A.1 List of Case Study Candidates ............................................. 201
Bibliography


[98] S.B. Kotsiantis. Supervised machine learning: A review of classification tech-
Applications in Computer Engineering: Real Word AI Systems with Applications in
eHealth, HCI, Information Retrieval and Pervasive Technologies, pages 3–24, Amster-
and 205).

[99] A. Lake and C.R. Cook. Use of factor analysis to develop OOP software complexity
(cited on pages 122 and 162).

[100] M. Lanza and S. Ducasse. Polymetric views - A lightweight visual approach
to reverse engineering. IEEE Transactions on Software Engineering, 29(9):782–795,

(cited on page 26).

on page 74).

[103] T.C. Lethbridge, J. Singer, and A. Forward. How software engineers use docu-
(cited on pages 3 and 48).


In Proceedings of the Seventh ACM SIGKDD International Conference on Knowledge
ACM. (cited on page 138).

development page 141).

[107] W. Maalej, R. Tiarks, T. Roehm, and R. Koschke. On the comprehension of pro-
gram comprehension. ACM Transactions on Software Engineering and Methodology

tomatic clustering to produce high-level system organizations of source code.


classification. In AAAI-98 workshop on learning for text categorization, volume 752,


