APPLICATION OF MACHINE LEARNING FOR EXTRACTING PROGRAMMING LANGUAGE CONSTRUCTS FROM 4GL LEGACY CODE

W. S. A. Ilakshini C. Subasinghe

138237A

University of Moratuwa, Sri Lanka. Electronic Theses & Dissertations www.lib.mrt.ac.lk

Degree of Master of Science

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2015
APPLICATION OF MACHINE LEARNING FOR EXTRACTING PROGRAMMING LANGUAGE CONSTRUCTS FROM 4GL LEGACY CODE

W. S. A. Ilakshini C. Subasinghe

138237A

Dissertation submitted in partial fulfilment of the requirements for the degree Master of Science in Computer Science specializing in Software Architecture

Department of Computer Science and Engineering

University of Moratuwa

Sri Lanka

May 2015
DECLARATION OF THE CANDIDATE & SUPERVISOR

I declare that this is my own work and this dissertation does not incorporate without acknowledgement any material previously submitted for a Degree or Diploma in any other University or institute of higher learning and to the best of my knowledge and belief it does not contain any material previously published or written by another person except where the acknowledgement is made in the text.

Also, I hereby grant to University of Moratuwa the non-exclusive right to reproduce and distribute my thesis/dissertation, in whole or in part in print, electronic or other medium. I retain the right to use this content in whole or part in future works (such as articles or books).

Signature: ...................................................    Date: .........................................

Name: W. S. A. Ilakshini C. Subasinghe

The above candidate has carried out research for the Masters Dissertation under my supervision.

Signature: ...................................................    Date: .........................................

Name of Supervisor: Dr. Amal Shehan Perera
ACKNOWLEDGEMENTS

I most gratefully acknowledge my appreciation to Dr. Shehan Perera for his acceptance, guidance and encouragement as my supervisor for this research, without which this work would not have been successful. I further extend my appreciation to Dr. Malaka Walpola and all my lecturers at the department for their encouragement and support to complete the research accurately in a timely manner.

I sincerely acknowledge the support my colleagues and seniors at work have provided, especially Mr. Mifraz Marzoon for his invaluable expertise as well as my friends who motivated me to complete the research.

Last but not least, I extend my deepest gratitude towards my parents and family for their tolerance and undivided love and support throughout the program.
ABSTRACT

With the progression and innovations of the Information Technology industry, computer systems have become not only a part of an organization but the heart of it that drives their daily routines and manages and tracks the entire business process for most enterprises and for decades Advanced Business Languages (ABL) have been evolving to provide successful economic solutions to drive these businesses. Progress 4GL (Fourth Generation Language) is one such Advanced Business Language where organizations have developed entire business process on for 30 years. However, with the advancement of Free and Open Sourced Software providing business solutions, some organizations using these legacy systems are looking for means of migration. Even though proprietary service providers exists for the migration process, organizations with decades old data are reluctant to use them for both cost and security reasons. Yet, in house development is also costly since ABL experts are very few and would require much time and effort to complete the process.

This research project is focused on a solution to develop such expert system that can interpret progress 4GL code to aid not only enterprises with migration but also engineers to learn and understand the language logic with ease. With the use of the Machine Learning technologies where research concerning modelling human thinking into machines are popular, this thesis provides a Proof of Concept for a methodology in which, an expert system can be created to read 4GL code, analyse the code, understand and infer the code logic and output the workflow in a graphical Flow Chart format. The prototype is run through several training 4GL programs to evaluate the implementation of the proposed theory. Current application proves to be successful for code with simple syntax and leaves room for further improvements to the system that can be enhanced to process 4GL’s many complex and evolving constructs and also the possibility of translating to a different language.

Keywords: Expert Systems, Natural Language Processing, CLIPSJNI, Progress 4GL, mxGraph, Java-ML, Proparse
# TABLE OF CONTENTS

Declaration of the Candidate & Supervisor ............................................................. iii
Acknowledgements .................................................................................................. iv
Abstract ................................................................................................................... v
Table of contents ..................................................................................................... vi
List of figures .......................................................................................................... ix
List of tables ........................................................................................................... xii
List of abbreviations ............................................................................................. xiii
List of appendices .................................................................................................. xiv

1. Introduction ....................................................................................................... 1
   1.1. Thesis Statement ......................................................................................... 2
   1.2. Thesis Overview ......................................................................................... 2

2. Background ....................................................................................................... 4
   2.1. 4GL Application Environments .................................................................. 4
   2.2. 4GL Conversion and Migration ................................................................... 6
   2.3. 4GL Reverse Engineering ......................................................................... 10
   2.4. 4GL and Machine Learning ..................................................................... 12
   2.5. Systems that Think Rationally .................................................................. 13
       2.5.1. Google Translate ............................................................................... 14
       2.5.2. Moses: Statistical Machine Translation System .................................. 14
       2.5.3. Google Prediction API ....................................................................... 15
       2.5.4. Creating New Language Translation for a Rulebase ....................... 16
   2.6. Example Based Machine Translation (EBMT) .......................................... 16
   2.7. Rule Based and Expert Systems ................................................................. 17
   2.8. Intelligent Compilers ................................................................................ 18
  3.1. Deep Learning - How to Create a Mind? ................................................. 22
  3.2. Progress 4GL .......................................................................................... 25
  3.3. Java & Machine Learning ....................................................................... 27
  3.4. Tools & Techniques ............................................................................... 28
    3.4.1. NetBeans IDE 8.0 ......................................................................... 28
    3.4.2. Proparse ....................................................................................... 29
    3.4.3. CLIPSJNI - Clips Java Native Interface ....................................... 29
    3.4.4. Java-ML ....................................................................................... 32
    3.4.5. MongoDB .................................................................................... 33
    3.4.6. ANTLR ......................................................................................... 34
    3.4.7. mxGraph – JGraphX .................................................................... 35
  4. System Implementation and Evaluation ...................................................... 36
    4.1. System Overview ............................................................................... 36
    4.2. Proparse Configuration & Syntax Tree Generation ............................ 38
    4.3. CLIPS Integration & Evaluation ......................................................... 41
    4.4. Java-ML Integration & Classification ................................................. 47
      4.4.1. Dataset Generation ..................................................................... 48
      4.4.2. Classification with Java-ML ......................................................... 50
      4.4.3. Evaluation of the Classification for Label Prediction ................... 51
      4.4.4. Other tools for Classification ....................................................... 53
    4.5. Interface ............................................................................................. 55
    4.6. Output Display .................................................................................... 56
4.7. Limitations of the Current System & Future Work ...........................................59

5. Conclusion ..............................................................................................................60

5.1. Problem with legacy systems and migration .....................................................60

5.2. Application of Machine Learning for Extracting Programming Language
Con structs from 4GL Legacy Code ..........................................................................61

5.3. Summary ..........................................................................................................63

References ..................................................................................................................65

Appendix A: 4GL Program Code & OUTPUT ..........................................................73

Appendix B: Classification Output ............................................................................92

- Rules.NNge (WEKA) ............................................................................................92
- Bayes.NaiveBayes (WEKA) ..................................................................................94
- Trees.J48 (WEKA) ................................................................................................96
- KNearestNeighbour (Java-ML) ...........................................................................98
LIST OF FIGURES

Figure 2.1: Next -generation application development and deployment: Source [7]...6
Figure 2.2: Automatic migration of Progress 4GL application to Java: Source [20]...9
Figure 2.3: ITOC Design Recovery Process: Source [22].................................10
Figure 2.4: Design Recovery Process of a Logistical Wholesale System: Source [23] ........................................................................................................11
Figure 2.5: Design Recovery Tasks: Source [23]..............................................11
Figure 2.6: Version Conflict in Migration – Source [25] ..................................12
Figure 2.7: Neural Network Model: Source [28] and [30].................................13
Figure 2.8: Word Alignment. Source [32].........................................................15
Figure 2.9: Moses Best English Output Sentence Model. Source [32].............15
Figure 2.10: Understanding of Example Based Machine Translation (EBMT) system to create translation system. Source [42]......................................................16
Figure 2.11: EBMT System Configuration. Source [35].................................17
Figure 2.12: Agents Acting on a Dynamic Environment: Source [44].........17
Figure 2.13: Strategic Decision Making to Arrive at a Solution: Source [44].....17
Figure 2.14: Roles of an Agent: Source [44].....................................................18
Figure 2.15: Architecture of the ConTraSt runtime environment and the abstract syntax: Source [49]......................................................................................20
Figure 3.1: AI’s Evolution – Source [61]............................................................23
Figure 3.2: AI System Categorization ...............................................................24
Figure 3.3: Deep Learning - Source [61]............................................................24
Figure 3.4: Progress Software History – Source [2].........................................26
Figure 3.5: Sample 4GL Code.........................................................................26
Figure 3.6: Research Categorization.................................................................27
Figure 3.7: CLIPS Construct – Source [68]......................................................31
Figure 3.8: CLIPS Dev Platform......................................................................31
Figure 3.9: Overview of the main algorithms included in Java-ML. The number of algorithms for each category is shown in parentheses. Source [77]...............32
Figure 3.10: Java-ML: KMeans integration to Java Code. Source [77].........32
Figure 3.11: Java-ML: Cross-validation experiment for specific dataset and classifier. Source [77]...
Figure 3.12: ANTLR Example – Source [63]...
Figure 3.13: Run ANTLR...
Figure 3.14: ANTLR GUI – Source [63]...
Figure 3.15: mxGraph Example - Source [67]...
Figure 4.1: 4GL Code Interpreter Workflow...
Figure 4.2: 4GL Code Interpreter with Tools & Technologies Used...
Figure 4.3: (a). Sample Data Dictionary (b.) PROPATH Settings ...
Figure 4.4: Progress Configurations for Proparse ...
Figure 4.5: Load Configuration Settings with Refactor Session ...
Figure 4.6: Generate Parser Tree for Progress 4GL Code ...
Figure 4.7: CLIPS Constructs. Source [68]...
Figure 4.8: Progress 4GL Variable Definition ...
Figure 4.9: CLIPS Template Definition for 4GL Variable Definition ...
Figure 4.10: Progress 4GL Variable Definition Complete Syntax - Source [69] ...
Figure 4.11: CLIPS Template Definitions ...
Figure 4.12: CLIPS Rule Definition for Progress 4GL Variable Definition ...
Figure 4.13: CLIPS Rule Definitions ...
Figure 4.14: CLIPSJNI Integration ...
Figure 4.15: CLIPS Evaluation ...
Figure 4.16: Sample Dataset ...
Figure 4.17: Sample Graphical View of Proparse Tree. Source [79] ...
Figure 4.18: Java-ML KNearestNeighbors ...
Figure 4.19: Class Labels in Training Dataset ...
Figure 4.20: Test dataset ...
Figure 4.21: RapidMiner - Java ML Classification. Source [82] ...
Figure 4.22: Weka - Java ML Example Source [83] ...
Figure 4.23: UI Interface 1 ...
Figure 4.24: UI Interface 2 - Input File ...
Figure 4.25: mxGraph Integration ...
Figure 4.26: mxGraph Generation Workflow ...
Figure 4.27: mxGraph Create Vertex.................................................................57
Figure 4.28: mxGraph Insert Edges.................................................................57
Figure 4.29: Sample Output (1)....................................................................58
Figure 4.30: Sample Output (2)....................................................................58
Figure 5.1: Knowledge Discovery Process Overview. Source [84]...............63
LIST OF TABLES

Table 4.1: Sample Dataset Attributes .................................................................51
Table 4.2: Classification Prediction Evaluation Results .....................................52
## LIST OF ABBREVIATIONS

<table>
<thead>
<tr>
<th>Abbreviation</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>4GL</td>
<td>Fourth Generation Languages</td>
</tr>
<tr>
<td>ABL</td>
<td>Advanced Business Language</td>
</tr>
<tr>
<td>AI</td>
<td>Artificial Intelligence</td>
</tr>
<tr>
<td>API</td>
<td>Application Program Interface</td>
</tr>
<tr>
<td>ANTLR</td>
<td>Another Tool for Language Recognition</td>
</tr>
<tr>
<td>CHUI</td>
<td>Character User Interface</td>
</tr>
<tr>
<td>CLIPS</td>
<td>C Language Integrated Production System – Expert System Dev Tool</td>
</tr>
<tr>
<td>CRUD</td>
<td>Create, Read, Update and Delete Operations</td>
</tr>
<tr>
<td>EBMT</td>
<td>Example Based Machine Translation</td>
</tr>
<tr>
<td>EGL</td>
<td>Enterprise Generation Language</td>
</tr>
<tr>
<td>GUI</td>
<td>Graphical User Interface</td>
</tr>
<tr>
<td>NLP</td>
<td>Natural Language Processing</td>
</tr>
<tr>
<td>PSC</td>
<td>Progress Software Corporation</td>
</tr>
<tr>
<td>RBMT</td>
<td>Rule Based Machine Translation</td>
</tr>
<tr>
<td>SDL</td>
<td>Specification and Description Language</td>
</tr>
<tr>
<td>WEKA</td>
<td>Weikato Environment for Knowledge Analysis</td>
</tr>
</tbody>
</table>
## LIST OF APPENDICES

<table>
<thead>
<tr>
<th>Appendix</th>
<th>Description</th>
</tr>
</thead>
<tbody>
<tr>
<td>Appendix – A</td>
<td>4GL Program Code &amp; Output</td>
</tr>
<tr>
<td>Appendix – B</td>
<td>Classification Output</td>
</tr>
</tbody>
</table>

University of Moratuwa, Sri Lanka. Electronic Theses & Dissertations
www.lib.mrt.ac.lk