Generation of Themed Portable Crosswords
A WordNet Implementation

Aoife Aherne
BA (Mod) CSLL
Final Year Project 2005
Supervisor: Dr. Carl Vogel
15th April 2005
Declaration

I hereby declare that this thesis is entirely my own work and that it has not been submitted as an exercise for a degree at any other university.

April 15, 2005

Aoife Aherne
Acknowledgements

Thank you to my classmates for four fun-filled years in CSLL, my parents for their considerable support and guidance and last but by no means least, my course director and project supervisor, Carl Vogel, for his direction and infectious enthusiasm over the past four years.
Person on a bus doing a crossword:

*Can you think of two personal pronouns?*

Innocent Bystander:

*Who, me?*
Contents

1 Introduction ........................................... 8
  1.1 Why Themed Crosswords? .......................... 8
  1.2 Why Printable, Portable Crosswords? ............ 10

2 Description of Graham 2003 ............................ 11
  2.1 Functionality ....................................... 11
  2.2 Resources ......................................... 11
  2.3 Random Symmetric Grid Generation ............... 11
  2.4 Database Queries ................................... 12
  2.5 Finding a Word Set to Satisfy the Grid .......... 12
  2.6 Communication Between Java and Prolog .......... 12
  2.7 Communication Between Server and Applet ........ 13
  2.8 Constructive Criticism .............................. 13

3 Technical Difficulties .................................. 15
  3.1 Introduction ....................................... 15
  3.2 Background ....................................... 15
  3.3 Memory Allocation Conflict ....................... 16
  3.4 The Linux Solution: Proposed by Developers of Prolog 17
  3.5 The Windows Solution: Proposed by Developers of Prolog 17
  3.6 Conclusion ......................................... 18

4 Project Goals .......................................... 19
  4.1 Introduction ....................................... 19
  4.2 Printable Versions of Crosswords ................. 19
    4.2.1 Why Are Printable Versions Required? .......... 19
    4.2.2 Proposed Implementation ....................... 19
  4.3 Integration of WordNet ............................. 20
    4.3.1 Introduction .................................. 20
    4.3.2 Why Integrate WordNet? ....................... 20
    4.3.3 Proposed Implementation ....................... 20
### CONTENTS

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>4.3.4</td>
<td>Why a MySQL version of WordNet2?</td>
<td>21</td>
</tr>
<tr>
<td>4.4</td>
<td>Restriction of Database Access</td>
<td>22</td>
</tr>
<tr>
<td>4.4.1</td>
<td>Why Restrict Database Access?</td>
<td>22</td>
</tr>
<tr>
<td>4.4.2</td>
<td>Proposed Implementation</td>
<td>22</td>
</tr>
<tr>
<td>4.5</td>
<td>Conclusion</td>
<td>22</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>5</td>
<td>WordNet - A Lexical Resource</td>
<td>24</td>
</tr>
<tr>
<td>5.1</td>
<td>Introduction</td>
<td>24</td>
</tr>
<tr>
<td>5.2</td>
<td>What is WordNet?</td>
<td>24</td>
</tr>
<tr>
<td>5.3</td>
<td>WordNet Statistics</td>
<td>25</td>
</tr>
<tr>
<td>5.4</td>
<td>Structural Aspects of WordNet</td>
<td>25</td>
</tr>
<tr>
<td>5.4.1</td>
<td>Synset</td>
<td>25</td>
</tr>
<tr>
<td>5.4.2</td>
<td>Organisation of the WordNet Hierarchy</td>
<td>27</td>
</tr>
<tr>
<td>5.5</td>
<td>Limitations of WordNet</td>
<td>28</td>
</tr>
<tr>
<td>5.6</td>
<td>Existing WordNet Driven Applications</td>
<td>29</td>
</tr>
<tr>
<td>5.7</td>
<td>Conclusion</td>
<td>30</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Section</th>
<th>Title</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>6</td>
<td>Integrating WordNet</td>
<td>31</td>
</tr>
<tr>
<td>6.1</td>
<td>Introduction</td>
<td>31</td>
</tr>
<tr>
<td>6.2</td>
<td>Software Required</td>
<td>31</td>
</tr>
<tr>
<td>6.3</td>
<td>Basic Algorithm</td>
<td>32</td>
</tr>
<tr>
<td>6.4</td>
<td>Algorithm Explained</td>
<td>32</td>
</tr>
<tr>
<td>6.4.1</td>
<td>An Iterative Problem</td>
<td>33</td>
</tr>
<tr>
<td>6.4.2</td>
<td>The Magic Number 5</td>
<td>33</td>
</tr>
<tr>
<td>6.5</td>
<td>Installation of the WordNet2 Database</td>
<td>34</td>
</tr>
<tr>
<td>6.6</td>
<td>Adaptation of WordNet2 Database</td>
<td>34</td>
</tr>
<tr>
<td>6.7</td>
<td>Interfacing WordNet2 MySQL Database &amp; Java</td>
<td>36</td>
</tr>
<tr>
<td>6.7.1</td>
<td>Java Data Structures Developed</td>
<td>36</td>
</tr>
<tr>
<td>6.7.2</td>
<td>Connecting to the Database</td>
<td>37</td>
</tr>
<tr>
<td>6.7.3</td>
<td>Retrieving the Words</td>
<td>38</td>
</tr>
<tr>
<td>6.7.4</td>
<td>Retrieving the Definitions</td>
<td>39</td>
</tr>
<tr>
<td>6.7.5</td>
<td>Removing Multiple Forms of Solution From A Clue</td>
<td>41</td>
</tr>
<tr>
<td>6.8</td>
<td>Grid Generation</td>
<td>42</td>
</tr>
<tr>
<td>6.8.1</td>
<td>Quality versus Quantity</td>
<td>42</td>
</tr>
<tr>
<td>6.9</td>
<td>Independent Analysis of Themed Crosswords</td>
<td>45</td>
</tr>
<tr>
<td>6.9.1</td>
<td>Interpretation Of Responses</td>
<td>46</td>
</tr>
<tr>
<td>6.9.2</td>
<td>Results</td>
<td>46</td>
</tr>
<tr>
<td>6.9.3</td>
<td>Discussion of Results</td>
<td>47</td>
</tr>
<tr>
<td>6.10</td>
<td>Conclusion</td>
<td>48</td>
</tr>
</tbody>
</table>
CONTENTS

7 Integration of the Printing Interface 49
  7.1 Introduction .............................................. 49
  7.2 Tools Required ........................................... 49
    7.2.1 \LaTeX ............................................. 49
    7.2.2 Source ............................................. 49
    7.2.3 Features .......................................... 50
    7.2.4 Packages .......................................... 50
    7.2.5 crosswrd.sty ....................................... 51
    7.2.6 Structure of the .tex file ......................... 51
  7.3 Conversion of \TeXFile to PDF File ....................... 52
  7.4 Integrating the production of PDF files with the current system 53
    7.4.1 Stage One - User Interaction ....................... 53
    7.4.2 Stage Two - Production of .tex file ................ 53
    7.4.3 Stage Three - Production of DVI and PDF files ...... 54
    7.4.4 Stage Four - Making the PDF File Available to the User. 56
  7.5 Sample Output ........................................... 57
    7.5.1 Solution .......................................... 57
    7.5.2 Blank ............................................. 58
  7.6 Problems Encountered .................................... 59
  7.7 Conclusion .............................................. 59

8 Restricting Database Access 60
  8.1 Introduction ............................................. 60
  8.2 Implementation ......................................... 60
  8.3 Basic Algorithm ........................................ 60
  8.4 Changes to the Database ................................ 60
  8.5 Changes Made to the MultiServerThread Class .......... 61
  8.6 The isValidPassword Method ............................. 62
  8.7 Adapting the Applet ..................................... 63
  8.8 Conclusion .............................................. 63

9 Provision of user Help File 64
  9.1 Introduction ............................................. 64
  9.2 Help File Contents .................................... 64
  9.3 Making Help File Available to the User ................ 65
  9.4 Conclusion .............................................. 65

10 Conclusion 66
  10.1 Further Study .......................................... 66
  10.2 Concluding Remarks .................................... 67
E Developer & Installation Guide 91
E.1 Introduction ........................................... 91
E.2 Packages - Installation & Configuration .................. 91
E.2.1 EasyPHP ..................................... 91
E.2.2 Java ........................................ 92
E.2.3 SICStus Prolog .................................. 92
E.2.4 Adobe Reader 5 .................................. 93
E.2.5 L\LaTeX .................................... 93
E.3 Setting of Classpaths .................................. 94

F User Help File 95
F.1 Introduction ........................................... 95
F.2 Crossword Generation .................................... 96
F.2.1 WordNet2 ..................................... 96
F.2.2 Websters Unabridged Dictionary .................. 96
F.2.3 User Database .................................... 97
F.3 Uploading Databases .................................. 98
F.3.1 Introduction .................................... 98
F.3.2 Text File Format .................................. 98
F.3.3 Database Format .................................. 98
F.4 Changing Passwords .................................. 99
F.4.1 How to? ...................................... 99
F.5 Creating Printable Versions of Crosswords .......... 100
F.5.1 Viewing Crosswords ................................ 100
# List of Figures

1.1 New York Time’s Themed Crossword .......................... 9
3.1 Java Virtual Machine Error Message ......................... 16
4.1 Hyponym & Hypernym Examples .............................. 21
4.2 System Structure & Data-flow ............................... 23
6.1 Wn_synset Prolog File Structure ............................. 34
6.2 Adaptation of Prolog File .................................. 35
6.3 Word Retrieval .................................................. 37
6.4 SQL Hypernym Query ......................................... 39
6.5 Definition Retrieval ........................................... 40
6.6 Grid Containing 1 & 2 Letter Words ....................... 43
6.7 Grid Free of 1 & 2 Letter Words ............................. 44
7.1 \texttt{\LaTeX} File 1st Section ............................... 52
7.2 \texttt{\LaTeX} File 3rd Section ............................... 52
7.3 \texttt{\LaTeX} File Clue Section ................................ 52
7.4 Java String Representation of \texttt{\LaTeX} Clue Section .... 55
7.5 DVI and PDF Production ..................................... 55
7.6 Solution Crossword ........................................... 57
7.7 Blank Crossword ............................................... 58
8.1 PASSWORD method in MultiServerThread .................... 62
8.2 isValidPassword method in DatabaseConnection class .... 62
8.3 Display of Password Dialogue ............................... 63
A.1 Introductory Screen ......................................... 69
A.2 A Completed Crossword ...................................... 70
A.3 Creating PDF File ............................................. 71
A.4 Uploading to Database ....................................... 72
List of Tables

5.1 WordNet Statistics ................................................. 25
5.2 WordNet Results for "sport" ................................. 26
6.1 Crossword Evaluation Results ............................. 47
Abstract

This report presents a computer program, developed as a language learning tool, that produces at random, crosswords based on a particular user stipulated theme. The crossword generator algorithm exploits a WordNet2 MySQL database to extract thematically related words and integrates a LATEX printing interface to facilitate the printing of the crosswords produced.
Chapter 1

Introduction

The content of this report describes how the piece of software responsible for the generation of themed portable crosswords came into being. Technical information and background information is provided on WordNet and LaTeX, two resources that are central to the project's implementation. Technical difficulties encountered are documented and attention is also paid to the application’s role in a real world language-learning environment. Ideas for future extensions to the application are elaborated upon.

1.1 Why Themed Crosswords?

The aim of the program is to accomplish by means of a computer program that which is currently achieved manually by those involved in the preparation of the New York Times’s themed crossword. An example of such a crossword is provided in Figure 1.1. Note that the “long” solutions are related to the theme indicated. Other solutions are not related to the theme.

Secondly, the acquisition of vocabulary by second language learners often adopts a themed approach, whereby vocabulary about a particular topic is presented to the learner. This can be viewed, for example, on the following website dedicated to second language learners of the French language: www.realfrench.net. In this instance, the site proposes a total of 59 crosswords, based on specific predetermined themes. These crosswords are compiled by hand. Ideally, a computer could accomplish this task. It is the aim of the program described in this report to achieve exactly that.
Figure 1.1: New York Time's Themed Crossword
1.2 Why Printable, Portable Crosswords?

Convenience is the word of choice. Should this program be employed as a language learning tool it is necessary to have a printing interface available to the primary user (in this case the lecturer/teacher). This allows the distribution of a hard copy to each student - 30 computers would not be necessary. Furthermore, even if computer resources were unrestrained, given that the crosswords are produced at random, for any given theme entered multiple crosswords will be produced - hence a hard distributable copy is required - ideally with an option of printing the crossword with or without it’s solution displayed.

Furthermore, not only can crosswords be printed, they can also be saved to the user’s system, providing them with a virtual “soft-copy” that they may consult repeatedly.
Chapter 2

Description of Graham 2003

2.1 Functionality

User interaction with the program is controlled via a web user interface, written in HTML and containing a Java applet that displays the user instructions, options and eventually the crossword puzzle itself.

2.2 Resources

The database exploited by Graham is “Webster’s Unabridged Dictionary”. This electronic dictionary is very dated and is lacking in modern day words and expressions. Dictionary information is recorded in a MySQL database, stored on macneill, a college server and using a JDBC driver is queried through the implementation of methods inherited from the Java.sql package. In addition to Websters Unabridged Dictionary the user may also upload their own dictionary in text file format containing a word and it’s respective definition on the following line.

2.3 Random Symmetric Grid Generation

This aspect of the project was implemented entirely in Java and functions impressively. The user stipulates the grid size and level of difficulty desired. These parameters determine the nature of the grid generated, with the level of difficulty determining the ratio of black to white squares in the grid. The WordSpace data structure developed by Graham encodes word lengths required by the grid and their position in the grid.
2.4 Database Queries

Only words of a specific length are sought from the database. A “theme” search algorithm is employed such that all words in the dictionary that contain the target theme word in their definition or as head words are selected. Clues of the words included in the grid’s solution set are retrieved from the database separately. Several Java classes were developed by Graham to represent information relating to the solution set and characteristics of the words retrieved from the database. These include the WordData and Solution classes.

2.5 Finding a Word Set to Satisfy the Grid

Prolog, due to its theorem proving capabilities, was deemed a suitable tool for this task. Attention was paid to the approach described in Berghel and YI (1989) for solving crossword puzzles, reducing the search space and the avoidance of naive backtracking by ordering the word slots and eliminating at an early stage words that are unlikely to fit into the grid. Similarly, when a word poses a problem - all of its inter-linking words are removed and re-assessed without the program needlessly backtracking through all of the solutions found thus far.

2.6 Communication Between Java and Prolog

The jasper package, a member of the Java class hierarchy controls communication between the principal application, written in Java and the ancillary Prolog application used to find a solution set of words for the grid generated. Firstly, Java writes a Prolog file “crossword.pl” containing the words in a format that can be interpreted by Prolog as well a linear encoding of the word slots and relationships between same. A SICStus Prolog object is initiated which in turn calls the newly generated Prolog file and a Prolog predicate file “functions.pl” that contains the necessary Prolog code for interrogating “crossword.pl”. Should the Prolog compiler succeed in generating a solution set this is communicated via the SICStus object to Java which in turn queries the database for the definitions associated with the words selected for the grid before communicating everything to the Java applet for display on screen.
2.7 Communication Between Server and Applet

Communication between applet and server and hence user and program is facilitated by MultiServer and MultiServerThread, Java classes inherited from the Java library and exploited by Graham to initiate crossword generation and database interrogation.

2.8 Constructive Criticism

The project is very well coded and presented. The user interface provided is more than adequate. Information pertinent to crossword generation and crossword solving are intuitive and easy-to-use. However, more explicit instruction could be provided particularly if this tool was to be used in a classroom environment, as suggested by Graham. This is particularly true of information relating to the option of the user supplying their own database in textfile format.

Secondly, on a more practical issue, users cannot ever retrieve or save a copy of the crossword to complete at a later date — they are obliged to finish the crossword in one sitting. An ideal solution to this problem would be the automatic generation of printable versions of the crossword at the users request. In a classroom environment this would be particularly useful, allowing an entire class to work on a specific crossword.

Thirdly, in Graham (2003) the solution to the clue very often appears in the clue text. Graham removes exact occurences of the solution using the removeAnswerFromClue() method in the Solution class, but has not dealt with the solution in it’s plural, past particible or present particible forms.

A security issue, again, with respect to potential use as a language learning tool in a classroom environment is raised with access to the database. Graham did not restrict database access. Access must be restricted such that only those with sufficient privileges (e.g.”root” /”teacher”) may add/modify databases on the system. Those with insufficient privileges (e.g ”student”) should only be able to generate and solve crossword puzzles.

Graham made exceptional advances in improving the execution time of the program through modification of the Prolog code. Despite the provision of a viable algorithm, the crosswords produced are very loosely related and
at times totally unrelated to the theme specified. The fault does not lie within the algorithm employed, it lies with the dictionary database it exploits. Due to the nature of the dictionary it was necessary, in order that enough words be returned to create a crossword, to select all words and definitions within which the string representation of the target word could be found. Hence, given the theme ”sport”, words like ”transport” and indeed clues containing ”transport” would be selected. Similarly the target theme ”cat” returns words and clues containing the substring ”cat”. Dictionary size also contributes greatly to the overall result. The incorporation of a WordNet database, a lexical resource more similar to a thesaurus than to a standard dictionary, would go a long way towards solving this problem. Characteristics of WordNet are described in Chapters 5 and 6.
Chapter 3

Technical Difficulties

3.1 Introduction

The aim of this section is to outline the technical difficulties encountered when attempting to install and run Graham (2003) on Linux systems, as described in that report. Although a definite solution is not found, information here should be of use to anyone wishing to extend this project further or indeed attempt to resolve the platform incompatibility problem currently experienced. Technical specifications detailed in this section were sourced from Sun (2005) and SICStus (2004).

3.2 Background

The Java Virtual Machine, is an abstract computing machine that is a central component of the Java platform. Its role is to manipulate system memory using its instruction set. The JVM is in this sense independent of the host operating system.

Jasper is the linking package between Java and SICStus Prolog, allowing the creation of SICStus objects and their processing by SICStus Prolog, independently of the Java application. The SICStus Prolog runtime requires memory space in the lower 256MB of the address range. SICStus objects will not initialise if insufficient memory is available in this location. Furthermore, methods cannot be invoked on already initialised SICStus objects if there is not enough memory available to carry out the task.
3.3 Memory Allocation Conflict

Errors were produced when the application attempted to invoke the `Load` method on an existing SICStus object. The program aborted, with the JVM producing a log file detailing the threads, or method sequences, that led to abort and the current level of memory usage within the JVM. An example of such an outburst is included in Figure 3.1.

An unexpected exception has been detected in native code outside the VM.
Unexpected Signal : 11 occurred at PC=0x40614C5A
Function=Java.se.sics.jasper.SICStus.spLoad+0x2A
Library=/usr/local/stow/sicstus-3.11.2/lib/libspnative.so

Current Java thread:
at se.sics.jasper.SICStus.spLoad(Native Method)
at se.sics.jasper.SICStus.load(SICStus.Java:940)
  - locked <0x44c79b70> (a se.sics.jasper.SICStus)
at CallP.fillGrid(CallP.Java:46)
at MultiServerThread.run(MultiServerThread.Java:240)

Heap at VM Abort:

<table>
<thead>
<tr>
<th>Generation</th>
<th>Total</th>
<th>Used</th>
<th>0x44780000, 0x44820000, 0x44c600000</th>
</tr>
</thead>
<tbody>
<tr>
<td>Eden space</td>
<td>512K, 53%</td>
<td>447c4000, 44800000, 448000000</td>
<td></td>
</tr>
<tr>
<td>From space</td>
<td>64K, 99%</td>
<td>448000000, 44800000, 448100000</td>
<td></td>
</tr>
<tr>
<td>To space</td>
<td>64K, 0%</td>
<td>448100000, 448200000</td>
<td></td>
</tr>
<tr>
<td>Tenured</td>
<td>1408K, 36%</td>
<td>44c600000, 44dc00000, 487800000</td>
<td></td>
</tr>
<tr>
<td>Compacted</td>
<td>4096K, 51%</td>
<td>487800000, 4898ea98, 4898ec00, 48b800000</td>
<td></td>
</tr>
</tbody>
</table>

Figure 3.1: Java Virtual Machine Error Message

As can be seen in the above error, the “from space” which forms part of the Eden space is full to 99% of capacity. The Eden Space is primarily
used for the storing of “new” objects. In this instance, insufficient memory would appear to be available. The error “name” Unexpected Signal 11 would appear to be causing a lot of people problems, given the number of internet forums dedicated to this yet unresolved issue.

3.4 The Linux Solution: Proposed by Developers of Prolog

SICStus would appear to be aware of this problem and had reportedly fixed the bug from version 3.10 onwards. The application was running on SICStus 3.11. SICStus proposed a temporary solution which involved the execution of a perl script to dedicate 256MB of space to SICStus in the lower address range simply by typing:

- spld -more-memory

Despite numerous attempts and correspondence with system’s administrators this attempt was unsuccessful.

3.5 The Windows Solution: Proposed by Developers of Prolog

This problem is not restricted to Linux. SICStus experiences similar problems with the JVM on Windows machines. However, this can be overcome by “re-basing” the JVM on the Windows machine. “Re-basing” increases the size of the initial JVM heap to such a great extent that it will not fit in the lower address range and will hence leave this section of memory available to SICStus. SICStus proposed this solution and it is available for viewing at www.sics.se/perm/sicstus/rebase_jvm.html

This solution works! Given the heart-ache experienced and the startling lack of crosswords produced it was decided to run with implementing the project on a laptop running Windows XP. Following the installation of all necessary packages and the configuration of classpaths the application ran smoothly and crosswords were generated. See Appendix E for details on how to successfully configure classpaths and install all relevant packages.
3.6 Conclusion

The problems posed by the Linux/Java Virtual Machine/SICStus memory allocation conflict proved excessively time consuming. Despite numerous communications with SICStus themselves no solution was found. There was no choice but to go ahead with the implementation of the project on a laptop running Windows. This worked impressively. In the future, SICStus, Linux, Java or indeed another individual may come up with a workable solution that will render this kind of multi-language application compatible with both Windows and Linux platforms.

In conclusion, the problems encountered, despite their obvious annoyance, provided me with a valuable insight into the workings of Linux, Windows, the JVM and SICStus that I would not have otherwise acquired.
Chapter 4

Project Goals

4.1 Introduction

This section details the goals set out for my project, how it is proposed to implement them and why a particular implementation was chosen. As stated in an earlier section, Graham (2003) is a very comprehensive program with few limitations. There is however scope for improvement and the addition of extensions to further enhance the crossword solving experience.

4.2 Printable Versions of Crosswords

4.2.1 Why Are Printable Versions Required?

Reasons for the inclusion of a printing interface are described in Section 1.2. In summary, printable versions are required for the following reasons:

- Convenience, in a Language Learning Environment
- Saving of crosswords for completion at a later stage
- Applet interactivity prevents printing of the webpage itself

4.2.2 Proposed Implementation

From the outset, the goal was to produce “portable document format” or PDF versions of the generated crosswords. The reason for this is threefold:

1. High quality output - professional look
2. Availability of \LaTeX, a document preparation system to create PDF files

3. Availability of a crossword “style file” within \LaTeX to facilitate the printing of crosswords

4.3 Integration of WordNet

4.3.1 Introduction

Clearly, as described in Section 2.8, Graham did not succeed in producing themed crosswords. The fault does not lie with the algorithm employed but with the resource exploited. To this end, it was decided that in order to best improve the situation a resource different to an electronic dictionary was required.

4.3.2 Why Integrate WordNet?

WordNet is a semantic web whose structure mirrors current psycholinguistic models of how lexical information is stored in the human brain. This project was initiated and developed by the Cognitive Science Laboratory, Princeton University, New Jersey. The structure of WordNet is closer to that of a thesaurus than to that of a traditional dictionary. Hence meaning searches as opposed to word searches are carried out on WordNet. This allows the retrieval of words related to a particular theme.

Secondly, the source code for this linguistic tool is freely available. In addition numerous other developers have made available a variety of interfaces to WordNet including a MySQL database version, generated from the Prolog source code. These projects can be sourced from WordNet2 (2004).

4.3.3 Proposed Implementation

The WordNet hierarchy forges links between words that are interchangeable in a given context. These words point to the same synset. Those words that are synonymous are grouped into synsets which represent the concept described by their shared meaning. Not only is there a relationship between words in a particular synset there exists numerous links between the synsets themselves. Of particular pertinence is the hyponym and hypernym hierarchy. A hyponym synset of a particular synset contains words that are more specific with respect to an underlying theme than the original synset. A
CHAPTER 4. PROJECT GOALS

hypernym synset of a particular synset contains words that are less specific with respect to an underlying theme than the original synset.

<table>
<thead>
<tr>
<th>Word = 'dog'</th>
<th>Gloss: a member of the genus Canis;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hyponyms of 'dog':</td>
<td>pooch, doggy</td>
</tr>
<tr>
<td>Gloss: Informal terms for dogs</td>
<td></td>
</tr>
</tbody>
</table>

Figure 4.1: Hyponym & Hypernym Examples

<table>
<thead>
<tr>
<th>Hypernyms of 'dog':</th>
<th>canine, canid</th>
</tr>
</thead>
<tbody>
<tr>
<td>Gloss: fissiped mammals with nonretractile claws [...]</td>
<td></td>
</tr>
</tbody>
</table>

Hence an algorithm exploiting this data structure would ensure that only words directly relevant to the theme be returned to the program. A more detailed discussion of WordNet follows in Chapter 5.

4.3.4 Why a MySQL version of WordNet2?

There are many versions of WordNet available: Windows, Linus, Prolog versions etc. Other individuals have developed a WordNet2 Java interface. This is a series of Java classes that can be used to query a WordNet database. Given that the Crossword Generator is coded almost exclusively in Java it would seem logical to use the Java interface available to query the WordNet database. However, this approach was rejected on the following grounds:

- Employing the Java interface would result in the importation of a number of classes - many of whom are not required by the Crossword Generator

- The methods within the Java interface do not correspond exactly to the requirements of the algorithm described in Section 6.3. Methods would require modification.

For these reasons, it was decided that the inclusion of a WordNet2 MySQL database and the provision of suitable methods to query this MySQL database constituted a more efficient means of integrating WordNet with the existing system.
4.4 Restriction of Database Access

4.4.1 Why Restrict Database Access?

At present, anybody can upload files to the server using the “upload database” option. This could result in the uploading of corrupt files and lead to server down-time.

4.4.2 Proposed Implementation

An improvement to the current system would be to password protect the database uploading feature. This would be particularly useful in a classroom environment whereby only the teacher could upload information to the program. Students would merely have rights to create and solve themed crosswords.

Ideally, upon selecting the upload database option the user should be asked to submit a password. If this password matches the password stored in the database access is granted, otherwise access is refused. The password should be stored in the database and not in the Java file as changing the password would require the re-compilation of the Java file it is in. Hence the most efficient solution is the storing of the password in a table in the database.

4.5 Conclusion

Figure 4.2 summarises both the structure of the proposed system as well as the flow of data between each of it’s components. Communication between components is fundamental to the success of the project and in Figure 4.2 is indicated by means of directional arrows.
Figure 4.2: System Structure & Data-flow
Chapter 5

WordNet - A Lexical Resource

5.1 Introduction

In this section background information is provided on the nature of WordNet. Next more detailed information is provided on the WordNet hierarchical structures. Limitations of WordNet as a lexical resource are discussed with reference to Vogel and Devitt (2004) and Devitt (2004). Reference is made to other applications that employ WordNet in their implementation. Finally, the reasons for WordNet’s inclusion in the project are outlined.

5.2 What is WordNet?

WordNet is, according to its founders, an "on-line lexical database". WordNet is hand-coded. WordNet is much more than an electronic dictionary and indeed it’s structure distinguishes it from established lexical resources such as The Oxford English Dictionary.

WordNet employs a model of classifying words that aims to imitate how the lexicon is organised in the human brain. This model emerged from psycholinguistic studies carried out over the past number of years. Hence possible uses of WordNet extend way beyond searching for a definition for a particular word or collocation. In fact the inverse of this is the case: one can search for a word or words associated with a given definition.
5.3 WordNet Statistics

Table 5.1 outlines the size of the WordNet2 database and the number of relationships that exist between words and meanings. Note that each *unique string* is considered as a single word entry. Table 5.1 is sourced from WordNet2 (2004).

<table>
<thead>
<tr>
<th>Part of Speech</th>
<th>Unique Strings</th>
<th>Synsets</th>
<th>Total Word-Sense Pairs</th>
</tr>
</thead>
<tbody>
<tr>
<td>Noun</td>
<td>114648</td>
<td>79689</td>
<td>141690</td>
</tr>
<tr>
<td>Verb</td>
<td>11306</td>
<td>13508</td>
<td>24632</td>
</tr>
<tr>
<td>Adjective</td>
<td>21436</td>
<td>18563</td>
<td>31015</td>
</tr>
<tr>
<td>Adverb</td>
<td>4669</td>
<td>3664</td>
<td>5808</td>
</tr>
<tr>
<td>Totals</td>
<td>152059</td>
<td>115424</td>
<td>203145</td>
</tr>
</tbody>
</table>

Table 5.1: WordNet Statistics

5.4 Structural Aspects of WordNet

5.4.1 Synset

As stated in the final sentence of the introduction it is possible to query WordNet for words associated with a particular definition. This is due to the fact that central to the WordNet model is the principle of synonymy. Synonymy is the semantic relation that holds between two words that can (in a given context) express the same meaning. Those words that are synonymous are grouped into synsets which represent the concept described by their meaning. A Synset is hence a representation of a concept that contains a set of words that are interchangeable in some context. However as indicated in Fellbaum (1998), Pg 5.

"These synonym sets do not explain what the concepts are, they merely signify that the concepts exist"

Given that words are grouped according to meaning/concept, multiple entries can exist for any given word. In essence, for every definition/meaning of a particular word or collocation there will exist a synset for that definition in WordNet. Consider the following query:
MySQL queries of this type will be explained in more detail at a later stage. In short this query searches for the word *sport* in the WordNet database and all of its associated definitions, indicating the part of speech that is assigned to the word in each instance. The results returned are as follows:

<table>
<thead>
<tr>
<th>Word</th>
<th>Definition</th>
<th>POS</th>
</tr>
</thead>
<tbody>
<tr>
<td>sport</td>
<td>an active diversion...</td>
<td>n</td>
</tr>
<tr>
<td>sport</td>
<td>occupation of athletes who compete for pay</td>
<td>n</td>
</tr>
<tr>
<td>sport</td>
<td>verbal wit</td>
<td>n</td>
</tr>
<tr>
<td>sport</td>
<td>temporary summer resident of inland Maine</td>
<td>n</td>
</tr>
<tr>
<td>sport</td>
<td>organism with characteristics resulting from chromosomal alteration</td>
<td>n</td>
</tr>
<tr>
<td>sport</td>
<td>someone who engages in sports</td>
<td>n</td>
</tr>
<tr>
<td>sport</td>
<td>play boisterously</td>
<td>v</td>
</tr>
<tr>
<td>sport</td>
<td>wear/display in an ostentatious manner</td>
<td>v</td>
</tr>
</tbody>
</table>

Table 5.2: WordNet Results for ”sport”

As can be observed in Table 1 WordNet contains eight separate entries for sport, each corresponding to a particular usage of the word. Each definition is identified by a unique numerical id. This allows the assignment of many words to the same synset and furthermore allows for the setting of relationships between synsets.
5.4.2 Organisation of the WordNet Hierarchy

The constituent synsets of WordNet are inter-linked by relations. These relations are either semantic or lexical in nature. Lexical relations hold between word forms, semantic relations hold between word meanings. All of these relations are reflexive. What follows is a brief outline of each relation that exists within the WordNet structure taken from WordNet2 (2004, pg 3).

- **Synonymy & Antonymy**

  The *synonymy* relation facilitates the grouping of words that are interchangeable in a particular context into the same synset.

  *Antonymy* is a lexical relation that concerns itself with adjectives and adverbs, positing antonymous adjectives or adverbs in opposition to each other. Examples of this relation include:

  - *hot* is an antonym of *cold*
  - *fast* is an antonym of *slow*

- **Hyponymy & Hypernymy**

  These relations organise nouns and verbs into a hierarchical structure through which individual synsets are related to each other. The hypernym relation is defined as follows:

  “The *generic* term used to designate a whole class of specific instances. *Y* is a hypernym of *X* if *X* is a (kind of) *Y*.”

  e.g. “canine” is a hypernym of “dog”

  The hyponym relation can be described thus:

  “The *specific* term used to designate a member of a class. *X* is a hyponym of *Y* if *X* is a (kind of) *Y*.”

  e.g. “pooch” is a hyponym of “dog”

  In simpler terms: hyponyms are more specific than the base word, hypernyms are more general than the base word.
• Meronymy & Holonymy

The meronym relation is defined as:

"The name of a constituent part of, the substance of, or a member of something. X is a meronym of Y if X is a part of Y”

The following example illustrates the meronym relation:

- Given: X = wheel, Y = car
- X is a meronym of Y as a wheel is a part of a car.

The inverse of the meronym relation is the holonym relation, defined as follows:

"The name of the whole of which the meronym names a part. Y is a holonym of X if X is a part of Y.”

This example describes the nature of the holonym relation:

- Given: X = wheel, Y = car
- Y is a holonym of X as wheel is a part of a car.

• Entailment

A verb X entails Y if X cannot be done unless Y is, or has been, done

An example of this relation is the following:

- X = to breath out - exhale
- Y = to breath in - inhale
- Exhale entails inhale:- in order to exhale one must have inhaled

5.5 Limitations of WordNet

• Language

As WordNet was developed in America, American English is used in its implementation. American geography, state names and figures from American history feature strongly in the database. Furthermore American expressions are also employed in describing synset concepts.
• Coverage

WordNet is a hand-coded lexical database. Hence, some concepts have been developed in greater detail than others. This is highlighted in Devitt (2004). The example given to illustrate this phenomenon is the following:

• Drogheda: site of 16th century battle

• Limerick: poem or a port city

Given that both Drogheda and Limerick are Irish towns of similar size, it is unusual that Drogheda not be considered a city like its south-western counterpart.

5.6 Existing WordNet Driven Applications

Given that use of WordNet is cost-free and that its source code is well documented many researchers have adopted this tool for use in a variety of areas including:

• Disambiguation of Meaning

• Semantic Tagging

• Information Retrieval

• Conceptual Identification

• Machine Translation

• Document Classification

In addition, EuroWordNet, a European project based on WordNet, has adopted WordNet’s basic structure and is developing WordNets for several European languages. At present this resource is not under general release and a fee is paid depending on the number of synsets requested.

WordNet developers are currently developing READER, a system designed to aid language learners in the acquisition of vocabulary. This system is at present unavailable.
5.7 Conclusion

WordNet is a well established and highly sophisticated database. Despite the fact that it has been developed “by hand” topic coverage is relatively good. It’s inclusion in other natural language applications gives an indication of it’s popularity in the language technology community. WordNet’s hierarchical structure lends itself to numerous search algorithms for the extraction of information it contains. Most importantly searches for thematically related words are possible.
Chapter 6

Integrating WordNet

6.1 Introduction

In this chapter the algorithms employed in the integration of a WordNet2 MySQL database are explained in detail. Theoretical issues that arose are discussed and changes made to the inherited system are outlined. Modifications made to the generation of random symmetric grids are described and feedback received on the themed crosswords is presented.

6.2 Software Required

The following packages were required to integrate WordNet to the existing system:

- WordNet2 MySQL database

  This database was sourced from AndroidTechnologies (2005) and was developed from Prolog files used in the implementation of the Prolog version of WordNet. It is a freely available tool.

- Java.sql

  Java.sql is a class within the Java hierarchy that facilitates communication between a Java program and a MySQL database. Queries can be executed on the database and the result set can be returned to the Java program for interpretation.

- Java.io
Java.io, a constituent class of the Java hierarchy makes possible “input” and “output” operations. This is of particular importance when working with server technology as information must be passed from, in this case, the program/applet to the server and from the server to the program/applet.

### 6.3 Basic Algorithm

1. User submits a valid theme
2. MySQL base query for primary synsets
3. MySQL query for hyponym synsets of each synset in 2
4. Iterate 3 five times
5. MySQL query for hypernym synsets of each synset in 2
6. Iterate 5 five times
7. Selecting a word of appropriate length from each synset returned by steps 3 & 5

### 6.4 Algorithm Explained

- **A Valid Theme**

As described in the Section 4.3.3, the semantic relations pertinent to the retrieval of words belonging to a particular theme are the “Hyponym” and “Hypernym” relations. These relations can only be applied to singular common nouns and verbs. You cannot for example search for a hyponym of the preposition “at” nor the pronoun “she”. Hence the user must submit either a singular common noun or a verb to the program for the algorithm to function correctly.

- **Primary Synsets**

The term “primary” synsets is attributed to the synsets in which the subject word occurs. This is the starting point of future queries.

- **Hyponym & Hypernym Synsets**
In Section 5.4.2 the terms Hyponym and Hypernym are discussed in detail. In summary a hyponym synset of a base synset contains words that are more “specifically” related to a particular topic that those words in the base synset. Similarly, a hypernym synset of a given synset will contain words that are more “generally” related to a particular theme than those in the initial synset.

Using the primary synsets as an initial starting point, steps 3 and 5 search for hyponym and hypernym synsets of the primary synsets. The first iteration of step 3 and 5 queries for hyponym and hypernym synsets of the primary synsets returned in step 2. However successive iterations of steps 3 and 5 use the last set of hyponym/hypernym synsets returned as their starting point. This results in the creation of chains of related hyponym and hypernym synsets.

• Words of Appropriate Length

This algorithm must satisfy two separate constraints. Firstly, the words returned must be thematically related to each other. Secondly, the words returned must be able to satisfy the constraints of the random grid generated. That is to say, only words whose length corresponds to the requirements of the grid are accepted.

6.4.1 An Iterative Problem

The number of synsets returned by a particular query cannot be determined in advance. This computational problem was solved in the following manner. Each query is executed once only, however the result set returned is iterated through to determine its length. Next the data structure size is modified to accommodate the result set. Finally the result set is iterated through a second time and the synsets returned are stored in the data structure.

6.4.2 The Magic Number 5

As can be observed in the algorithm, the searches for hyponym and hypernym synsets are iterated five times each. This is to ensure that all words associated with the submitted theme are returned. The approach adopted follows statistics related to the WordNet 1.7 hierarchy presented by Vogel and Devitt (2004). Of particular pertinence is the following:

• 78.5% of synsets are “leaf-synsets”
• Maximum distance from any synset to a leaf synset is 5
• Minimum distance from any synset to a leaf synset is 2

Leaf synsets are situated at the extremes of the WordNet hierarchy. In simple terms, no other synset inherits from a leaf synset: there are no hyponym synsets of leaf synsets. 78.5% of synsets constitutes a very large majority of synsets. Of even greater importance is the fact that the maximum distance from any synset to a leaf synset is five synsets. Hence by iterating the hyponym and hypernym searches five times, full coverage is ensured.

6.5 Installation of the WordNet2 Database

The MySQL version of WordNet2 was sourced from Android Technologies(2004). A single .sql file contained all of the information required to install the database. This .sql file was uploaded to the MySQL database running on the EasyPHP system using the LOAD DATA LOCAL INFILE command for uploading files containing series of MySQL statements. Due to its considerable size (39.1 MB) the creation of the database took approximately 20mins.

6.6 Adaptation of WordNet2 Database

The WordNet2 database did not provide all the information that the crossword generator requires to function effectively. In particular, in wn_synset - the principal table in the WordNet2 database - there is no field that stores the length of the word in the row. The easiest way to solve this problem was to regenerate the table from a personally modified version of the Prolog file associated with the table. Figure 6.1 contains a sample of the file’s structure:

```
s(100001740,1,'entity',n,1,11).
s(100002056,1,'thing',n,12,0).
s(100002342,1,'anything',n,1,0).
s(100002452,1,'something',n,1,0).
```

Figure 6.1: Wn_synset Prolog File Structure
CHAPTER 6. INTEGRATING WORDNET

The entire Prolog file required re-formatting. The majority of this re-formatting was carried out using the “find-replace” procedure in the Emacs editor. Parentheses, periods and the leading ‘s’ were removed followed by the replacement of the separating commas by a blank space. The resulting data was saved and passed to a Prolog file that parsed each line using the StringTokenizer class (part of the Java hierarchy). An extra ‘token’ was appended to each string. This token is word length. Finally, each of the tokens was placed into a MySQL INSERT statement and written to file. An excerpt of the code is contained in Figure 6.2.

```java
while(s != null){
    st = new StringTokenizer(s, " ");
    synset_id = st.nextToken();
    w_num = st.nextToken();
    word = st.nextToken();
    ss_type = st.nextToken();
    sense_num = st.nextToken();
    tag_count = st.nextToken();
    length = word.length();

    output = "INSERT INTO wn_synset VALUES (" + synset_id + "," + w_num + "," + word + "," + ss_type + "," + sense_num + "," + tag_count + "," + length + ");");
    pw.println(output);
    s=br.readLine();
}
```

Figure 6.2: Adaptation of Prolog File

The existing wn_synset table was removed, it’s definition was extended to include the word length field and the file containing the INSERT statements was uploaded to this new table. Once again, the LOAD DATA LOCAL INFILE command was exploited to achieve this.
6.7 Interfacing WordNet2 MySQL Database & Java

With an adapted WordNet2 database in place, various methods were required to query the database in the manner outlined by the algorithm and in keeping with the inherited program’s general logical sequence.

6.7.1 Java Data Structures Developed

Given that the synset is of great significance in WordNet and of great significance to the application, a Synset class was written describing a synset object. This object has two attributes:

- Synset_Id – int
  - The unique identifier of a synset
- Gloss – String
  - The concept associated with a synset

As the WordNet2 database consists of synset queries rather than word based queries, a distinction is made during the program’s execution to take into account which lexical resource has been selected. The words returned from each query are recorded in two different arrays.

- WordData 2D array (possibleWords) containing all possible words
- Synset 2D array (wnWords) containing all possible words and their synset ID numbers

The WordData array is used by the application to write a Prolog file that is subsequently queried in order to produce a solution set of words that satisfy the grid generated. The Synset array acts as a reference array. When the solution set of words has been determined, it is necessary to query the database for their respective definitions. As words occur in numerous synsets not keeping the synset ID number with the word could result in the retrieval of definitions that are not associated with the theme of the crossword.
6.7.2 Connecting to the Database

Database connection is managed by the FillingGrid class, written by Graham and extended in this project to include the interrogation of a WordNet database. A FillingGrid object is created within the MultiServerThread class. After a grid has been generated, the database is queried for words of suitable length. This is achieved by invoking the writeProlog method on the FillingGrid object. Embedded in this method is another of FillingGrid’s member methods: getWords(String subject). This is the method responsible for connecting to the database and retrieving the words required. As stated in the Section 6.7.1, the search algorithm employed for querying a WordNet database differs from that which queries standard electronic dictionaries. The code in Figure 6.3 illustrates the steps taken in querying and retrieving words from the database. The words returned are recorded in two data structures as described in Section 6.7.1.

```java
if (dicAddress.equals(WORDNET2)){
    System.out.println("Recognises its a wordNet Dictionary");
    for(int j=MIN WORD LENGTH; j<=MAX WORD LENGTH; j++){
        //go through all word lengths required
        if((wordLengths[j] == true) &&
            (possibleWords[j][0].getWord().equals(NO WORD YET))){
            //Querying database for length j words
            temp = database.getWNwords(subject,j);
        }
        //make sure there are words returned
        if(temp.length>=1){
            posWords = new WordData[temp.length];
            for(int k=0; k<temp.length; k++){
                word = temp[k].getWord();
                posWords[k] = new WordData(word);
            }
            //record data in possibleWords and wnWords
            possibleWords[j]=posWords; wnWords[j]=temp;
        }
    }
}
```

Figure 6.3: Word Retrieval
6.7.3 Retrieving the Words

As can be seen in Figure 6.3, the method responsible for returning words of a particular length, pertaining to a particular theme, is `getWNwords(subject, j)`. This is a member method of the DatabaseConnection class, developed by Graham and extended in this project to include methods that query a WordNet database. The `getWNwords` method contains several embedded methods that carry out steps 2 to 6 of the algorithm. These methods are:

- **public Synset[] getPrimarySynsets(String subject)**
  - Base query, returning a synset array of all the synsets containing the word represented by the String “subject”

- **public Synset[] getHypo(Synset[] id)**
  - Returns an array of synsets that are hyponyms of the array of synsets represented by the Synset array “id”

- **public Synset[] getHyper(Synset[] id)**
  - Returns an array of synsets that are hypernym synsets of the array of synsets represented by the Synset array “id”

- **public Synset[] getWord(Synset[] id, int len)**
  - Returns an array of synsets objects which contain a word of length “len” and a synset ID number. One word is selected per synset in the array “id”. The array “id” is a concatenation of all the arrays returned by successive calls of `getPrimarySynsets`, `getHypo` and `getHyper` methods.

In keeping with the algorithm, each time `getWNwords` is called, `getPrimarySynsets` is invoked once, `getHyper` and `getHypo` are called five times each and finally, the `getWord` method is called once. Within each of these methods objects from the Java.sql package are employed to send SQL queries to the database are created. A separate SQL query is executed for each of the methods above. These queries are dynamic - they vary depending on the subject and word length requested.

The code excerpt in Figure 6.4 is an example of an SQL query taken from the `getHyper` method. Note that the number of Synsets returned by the query is recorded. This is to facilitate the creation of a Synset array whose size corresponds to the size of the result set, thus avoiding array out
for(int i = 0; i<id.length; i++){
   //prepare the statement:
   get_hyper = connection.prepareStatement
   ("select synset_id_2 from wn_hypernym where synset_id_1 = "+id[i].getSynsetId()+";";);
   //assign result set to be equal to whats returned by DB
   hyperRS = get_hyper.executeQuery();
   //count number of synsets returned and store in numSyns
   while(hyperRS.next()){
      synset = hyperRS.getInt("synset_id_2");
      tnumSyns++;}
}

Figure 6.4: SQL Hypernym Query

of bounds exceptions. When all words have been returned a Prolog file is
written containing these words. Next the Prolog file is queried by functions
in the file functions.pl and should a solution set of words be found, they are
communicated to the application. Further information regarding the Prolog
powered solution retrieval can be found in Graham (2003).

6.7.4 Retrieving the Definitions

The storeSolution method in the FillingGrid class records the words that
make up the solution set as well as their positions in the grid. It also re-
trieves the definitions associated with each of the words in the result set from
the database. Given that this method involves querying the database, it was
necessary to re-write it incorporating WordNet search methods.

The solution set, as well as being recorded in the WordSpace words array,
is also recorded in the Synset wnSolutionSet array. wnSolutionSet includes
both the word and the synset id, referenced from the wnWords synset array
which was populated in the getWords method, thus conserving the link to
the appropriate definition. Next the database is queried for each word’s def-
inition.

The code excerpt in Figure 6.5 is taken from the storeSolution method
and illustrates, the retrieval of definitions and also the transfer of information from the wnSolutionSet array to the words array. The `getSynsetGloss` is a member method of the DatabaseConnection class. It’s function, is to retrieve the definitions associated with each of the synset id’s passed to it in the form of a Synset array. This method returns another Synset array, with each object in the array containing a synset id number and it’s associated definition.

```java
System.out.println("Getting wordNet definitions");
//gets definition for each word in wnSolutionSet
temp = database.getSynsetGloss(wnSolutionSet);

//copies defintion/clue num from wnSolutionSet to words[]
for(int j=0; j<wordspace_end; j++){
target = words[j].getWord();

for(int k=0; k<wnSolutionSet.length; k++){
    //iterating through wnSolutionSet to find corresponding word
    if(wnSolutionSet[k].getWord().equalsIgnoreCase(target)){
        cible = wnSolutionSet[k].getSynsetId();

        //iterating through definitions/synset id's
        for(int l = 0; l<temp.length; l++){
            //if both synset id's match have defintion
            if(temp[l].getSynsetId()==cible){
                //clue number and definition transferred to words[]
                words[j].setNumClues(1);
                words[j].setClue(0,temp[l].getWord());
                break;
            }
        }
    }else{
        break;
    }
}
```

Figure 6.5: Definition Retrieval
6.7.5 Removing Multiple Forms of Solution From A Clue

As stated in Section 2.8, Graham does not successfully remove all forms of the solution from the clue. To overcome this problem, it was decided to extend the `removeAnswerFromClue` method, a member method of the `Solution` class. Use was made of the `replaceAll(string this, string that)` method available in the Java `String` class. This method replaces occurrences of the string “this” with the string “that”. The aim was to remove the solution whether it occurred at the end or in the middle of a sentence, in its plural, past participle, present participle, adverbial and superlative forms. Graham (2003) successfully removes occurrences of the solution in all parts of the clue, but does not deal with other morphological forms of the solution. An outline of the extension to the `removeAnswerFromClue` method is given below and involves searching for representative strings:

- Solution in middle of sentence
  - whitespace + answer + whitespace

- Solution at the end of a sentence
  - whitespace + answer + "," or ",!" or ",?" or ",:" or ",;"

The occurrences of plurals, present and participle are found in a similar manner and are sought both in the middle or at the end of the sentence:

- Plural Forms in the middle of a sentence
  - whitespace + answer + "s" + whitespace

- Past & Present Particibles
  - whitespace + answer + "ed" or "d" or "ing" + whitespace

- Adverbial & Superlative Forms
  - whitespace + answer + "ly" or "ily" or "er" or "est" + whitespace

This method clearly does not take into account those words whose plural or other forms do not follow general morphological rules. In many instances exceptional cases result in a complete change to the word itself - (e.g. eat : ate) thus ensuring that the actual solution would not appear in the clue text, therefore coverage of exceptional cases is in many instances not required.
6.8 Grid Generation

Graham (2003) implemented random symmetric grid generation. However, many of the grids contained a large proportion of one and two letter words. This characteristic does not lend itself to the production of themed crosswords. Very few, if any, one or two letter words “belong” to a theme. Prepositions like “to”, “in” and “at” are not covered by the hypernym and hyponym semantic relations and are hence not going to occur in any theme. The problem could be solved in one of two ways:

- Solution A: 1 & 2 letter words in WordNet available to all grids
  - Quality of crossword dependant on random grid generated

- Solution B: Generation of random grids - but only those without 1 & 2 letter words
  - High-quality themed crosswords up to grid size 6/7
  - Server fatigue resulting in larger gridsizes not functioning correctly
  - Higher failure rate - fewer crosswords generated when requested

6.8.1 Quality versus Quantity

The decision boiled down to whether the quality of the crosswords was of greater importance than the number of them produced. Given that the goal was to create themed crosswords, the second solution was chosen. Furthermore, allowing one and two letter words appear in the grid had an inconsistent bearing on the result - some crosswords contained more words related to the theme than “filler” words whereas it was more frequently the case that there were more filler words than thematically related words. This is evident from the following sample crossword:
Themed Crossword: Earth

Figure 6.6: Grid Containing 1 & 2 Letter Words
At least with the second solution, when a crossword is generated a high quality result is guaranteed. This can be observed in the next sample crossword:

**Themed Crossword: Sport**

![Crossword Grid]

**ACROSS**
1. a person or animal that is markedly unusual or deformed (5)
2. the pursuit and killing or capture of wild animals regarded as a sport (4)
3. a headlong plunge into water (4)
4. a living thing that has (or can develop) the ability to act or function independently (5)

**DOWN**
6. a brief swim in water (3)
7. the act of ***** as a sport (3)
8. a person trained to compete in sports (7)
9. a humorous play on words; 'I do it for the ***** of it'; ‘his constant punning irritated her’ (3)
10. a message whose ingenuity or verbal skill or incongruity has the power to evoke laughter (3)

Figure 6.7: Grid Free of 1 & 2 Letter Words
6.9 Independent Analysis of Themed Crosswords

In order to determine whether or not the system actually succeeds in creating themed crosswords an experiment was carried out. In this experiment a total of twenty individuals, both male and female, spread between the ages of 18 and 60 were presented with a completed crossword puzzle produced by the program. Their task was to “Guess the Theme”. There was no time limit set for this task, but participants were asked to follow their “gut” instinct. Participants could respond in sentence form, single word or a couple of words that they felt best described the theme of the crossword. A total of eight crosswords were presented to the participants. The following themes were covered:

- Sport
- Water
- Music
- Pick
- Match
- Food
- Animal
- Colour

The themes were selected so as to provide a varying level of abstraction and hence difficulty. Of the themes selected sport, water, pick, match and colour can be interpreted as a noun or a verb. Food, animal and music are all nouns. It would be expected that the first group pose more problems than the second given the extra degree of complexity. Furthermore, given that pick is primarily employed as a verb the crossword generated contains numerous verbal synonyms and is hence more difficult to identify it’s theme. The crosswords presented to the participants can be viewed in Appendix A.

Participants were asked to state whether or not they ever studied linguistics and if so in what capacity. After the experiment, the subjects noted which crossword they found most difficult and which they found easiest.
6.9.1 Interpretation Of Responses

In order to accurately analyse results a sliding scale was employed. It was decided that responses that included the exact word entered be considered as positive results. Words that could be considered as being part of the same domain were also accepted but accorded less importance. An example of a word/phrase belonging to a particular domain is: “athletic skill”, a member of the “sport” domain. Domain membership was verified using WordNet’s hyponym and hypernym semantic relations. Needless to say responses that were outside of the subject domain were considered as being negative results.

6.9.2 Results

• Crossword 1: Sport
  – Sample of Participant Responses:
    - sport, activity, athletic skill, actions

• Crossword 2: Water
  – Sample of Participant Responses:
    - liquid, water, bodies of water, flowing, rivers & lakes, nature

• Crossword 3: Music
  – Sample of Participant Responses:
    - music, musical forms/terminology, work

• Crossword 4: Pick
  – Sample of Participant Responses:
    - actions, directions, hobbies, making decisions, sorry haven’t a clue

• Crossword 5: Match
  – Sample of Participant Responses:
    - being influenced by others, ranking of respect, recognition, fitting in, dunno, changing form, imitate

• Crossword 6: Food
  – Sample of Participant Responses:
- food, food & drink, eating & drinking

- Crossword 7: Animal
  - Sample of Participant Responses:
    - birds, insects & lizards, animals, flying creatures, small animals, birds, insects of the animal kingdom, feathers & fins

- Crossword 8: Colour
  - Sample of Participant Responses:
    - colour, colouring, in colour, colour & mediums used by artists, painting

- Overall Results

A total of 20 individuals participated in the evaluation process. Statistical results for each of the crosswords are detailed in Table 6.1.

<table>
<thead>
<tr>
<th>Crossword</th>
<th>Correct</th>
<th>Domain</th>
<th>Incorrect</th>
</tr>
</thead>
<tbody>
<tr>
<td>Crossword 1</td>
<td>70%</td>
<td>20%</td>
<td>10%</td>
</tr>
<tr>
<td>Crossword 2</td>
<td>30%</td>
<td>70%</td>
<td>0%</td>
</tr>
<tr>
<td>Crossword 3</td>
<td>90%</td>
<td>10%</td>
<td>0%</td>
</tr>
<tr>
<td>Crossword 4</td>
<td>0%</td>
<td>10%</td>
<td>90%</td>
</tr>
<tr>
<td>Crossword 5</td>
<td>0%</td>
<td>10%</td>
<td>90%</td>
</tr>
<tr>
<td>Crossword 6</td>
<td>90%</td>
<td>10%</td>
<td>0%</td>
</tr>
<tr>
<td>Crossword 7</td>
<td>60%</td>
<td>40%</td>
<td>0%</td>
</tr>
<tr>
<td>Crossword 8</td>
<td>70%</td>
<td>20%</td>
<td>10%</td>
</tr>
</tbody>
</table>

Table 6.1: Crossword Evaluation Results

6.9.3 Discussion of Results

The results presented in Table 6.1 are largely predictable.

Themes were largely successfully chosen by the participants for crosswords 1, 2, 3, 6, 7 and 8 - only 10% of responses to crosswords 1 and 8 were entirely unrelated to the theme.

Crosswords 4 and 5 scored poorly given the high level of abstraction associated with the theme words selected: pick and match. Although the
participants were not informed as to the syntactic category that the theme words belonged to, the responses returned for crosswords’ 4 and 5, although outlandish, were largely verbal.

Whether the participant had or does study linguistics had no bearing on the overall results. However, upon completion of the evaluation, the participants were given the correct theme for each crossword. In general, participants “understood” why a particular theme could be associated with a crossword, but linguists were naturally quicker to recognise the themes relevance to the crossword.

6.10 Conclusion

Through the integration of WordNet, the crossword generator now successfully generates “themed” crosswords. The findings of the crossword evaluation survey confirm this fact. The algorithm employed for the extraction of thematically related words ensures that all relevant words are returned to the program for consideration. Adaptation of the WordNet2 MySQL database contributes to this process. Another distinct advantage of the current system is that it opens door to easily integrating EuroWordNet and it’s Inter-Lingual-Index, which will allow the production of multi-lingual themed crosswords, where the clues could be presented in one language (e.g. French), with the solutions expected in the target language (e.g. English).

Careful theme selection produces interesting results. Conventional themes, like for example ”language”, ”country”, ”food”, ”sport” produce consistently good crosswords, indicating that in a language learning environment, crosswords can be reliably produced for “popular” themes. More “original” themes, like for example “pick” or “match”, provide highly-interesting, abstract crosswords which require a greater level of lateral thinking to determine their theme. These could be described as ”verbally synonymous” crosswords, where light is shed on the relationship between verbs and expressions as opposed to noun-noun relationships.

A sample of crosswords produced by the system can be found in Appendix C and D.
Chapter 7
Integration of the Printing Interface

7.1 Introduction

In this section the tools and algorithms employed in the production and development of the “Printable Portable Crosswords” extension are explained and discussed in detail. Furthermore, pitfalls and problems encountered during the development of this feature are also discussed.

7.2 Tools Required

7.2.1 \LaTeX

\LaTeX is a document preparation system for high-quality type-setting, based on Donald E Knuth’s “\TeX” type-setting language. For Knuth, TeX is

“intended for the creation of beautiful books - and especially for books that contain a lot of mathematics”

Here the intention is to create beautiful crosswords, in keeping with Knuth's tradition! Leslie Lamport is the founding father of \LaTeX, however \LaTeX is now developed by the \LaTeX3 Project.

7.2.2 Source

The Windows version of \LaTeX, MikTeX takes approximately 90mins to download and install and is available at: http://www.miktex.org/. \LaTeX documentation can be sourced at (LaTeX, 2004).
7.2.3 Features

With \LaTeX{} the content and layout are specified separately. This allows for the production of a standard “template” and the “plugging in” of relevant content as required. This is particularly useful for the crossword generator as that way standard characteristics of the crossword are described in advance:

- Consists of a grid
- Set of numbered clues
- Clues refer to a particular cell in the grid
- Clues are either across or down
- Solutions have a particular length
- Solutions are linked to a clue number
- Solutions have a particular starting point on the grid (row & column number)
- Those cells that do not contain a letter are black
- Grid can be displayed with or without its solution

\LaTeX{} is particularly suited to the production of scientific reports and mathematical papers. This is due to its adaptability with respect to the display of mathematical formulae and other figures. It would also be very difficult, or nearly impossible, to communicate to another documentation system, such as Microsoft Word, the various parameters associated with crossword generation: gridsize, position of black/white squares, direction of clues and words in the grid.

For these reasons and the fact that \LaTeX{} is freely available on-line it was decided that \LaTeX{} be employed in this project.

7.2.4 Packages

\LaTeX{} has and continues to be updated by people around the world. Where particular features do not exist in the \LaTeX{} class hierarchy, classes can be developed to encapsulate what is missing. This is the case with the crossword class file. The author of crosswrd.sty, as distributed with the current version of MiKTeX, is Frank Mittelbach. Mittelbach extended Brian Hamilton Kelly’s version of the .sty file to ensure its compatibility with \LaTeX{}2e, the latest version of \LaTeX{}. 
7.2.5 crosswrd.sty

The structure of this file allows for the specification of many parameters that are related to the crossword:

- Grid Size
- Display Solution?

There are also parameters that contain information relevant to the puzzle’s clues and their respective solutions:

- Number
- Direction
- Column Number
- Row Number
- Solution
- Text
- Length of Solution

To gain access to this resource the .tex file or “source file” must contain a reference to this particular package. This is achieved by the following command:

- \usepackage{crosswrd}

7.2.6 Structure of the .tex file

Given that layout and content are treated separately in LaTeX the generation of a suitable .tex file is trivial. There are three main sections to the .tex file.

The first section contains information with regard to the classes sourced in the document, height of text, size of margins:
CHAPTER 7. INTEGRATION OF THE PRINTING INTERFACE 52

\documentclass{article}
\usepackage{crosswd}
\pagestyle{empty}
\setlength\topmargin{-1in}
\setlength\textheight{10.7in}
\begin{document}

Figure 7.1: \TeX{} File 1st Section

The third section contains tags that ”close” the file correctly:

\end{crossword}
\end{document}
\endinput

Figure 7.2: \TeX{} File 3rd Section

In between these two sections commands relating to the nature of the clues and solutions are stored. These have a set standard format, only the information contained within its parameters is subject to change.

\begin{crossword}{GRIDSIZE}{SHOW ANSWERS Y or N?}
\begin{crossword}{5}{Y}
\clue{clue num}{A/D}{col}{row}{ans}{clue}{ans length}
\clue{1}{A}{2}{1}{ARIA}{an elaborate song for solo voice}{4}
\end{crossword}
\end{crossword}

Figure 7.3: \TeX{} File Clue Section

7.3 Conversion of \TeX{}File to PDF File

The \LaTeX{} environment contains various tools one of which facilitates the conversion of .tex files to .dvi files. Assuming system classpath variables have been set appropriately it should be possible to execute the following command line argument and thus create a .dvi file. The name of the file should be substituted for “filename”.

- latex filename
The next step in the process is to convert this file to PDF. Again this requires a simple command line invocation of:

- pdflatex filename

Note, on Linux systems the appropriate tool is 'dvipdf' and not 'pdflatex'

7.4 Integrating the production of PDF files with the current system

As can be easily remarked from the previous sections the production of a PDF file requires that several criteria be met and that several sequential stages take place.

7.4.1 Stage One - User Interaction

Files are only generated at the users request and to the users requirements. Via a drop down menu and “Create PDF” button on the applet the user can indicate whether she/he requires a blank crossword grid or a completed crossword to be created with the appropriate clues. This interaction is controlled by methods within the Crossword class.

The user is asked to submit a filename for their crossword. The .tex .dvi and .pdf files generated during the process described below will be saved as filename.tex, filename.dvi and filename.pdf respectively.

7.4.2 Stage Two - Production of .tex file

The only dynamic aspect of any .tex file that the system is going to produce will be the first and second sections of the .tex file. In section three the information can be hard coded from the outset.

Whether the user has requested a blank or complete solution to the puzzle is recorded at runtime by the action listener assigned to the print format choice menu on the applet. The method concerned is the print method found in the Crossword class.

- printFormat.getSelectedItem()
Similarly, information pertaining to grid size is also available at runtime and can be accessed as follows:

- `gridSize.getSelectedItem()`

This information, together with the size of the grid is then inserted into the following string:

```java
pre = "
documentclass{article}
usepackage{crosswr}
pagestyle{empty}
setlength
topmargin{-1in}
setlength
textheight{10.7in}
begin{document}
begin{crossword}{"+gridSize+"}{"+blank+"};
```

Clue and solution data is stored in the “solution” data structure. This data structure stores all information pertaining to the crossword solution and can be very easily accessed from within the program at runtime.

A for-loop loops through the contents of the solution array. For each element in the array the required information is extracted using methods from the Solution class and the information is appended to the string allClues:

The text relating to sections one and three is stored as two separate strings. Finally each of the three strings are sent to the server so that they maybe printed to a file, named by the user, using the FileWriter and PrintWriter objects supplied as part of Java.io. File writing and conversion is carried out server-side and is as such detailed in the `MultiServerThread` class.

### 7.4.3 Stage Three - Production of DVI and PDF files

The methods used to produce DVI and PDF files are similar. Java Runtime and Process objects are used to access command line features available on the system. The execution of commands independent of the current program is possible but can be constrained by available system memory.

The only distinction made when initiating the PDF file conversion is:

- `cmd = ”pdflatex usercw”;

for(int x = 0; x < totalNum; x++){
    current = solution[x];
    number = current.getNumber();
    if(current.isAcross()){
        direction = "A";
    }
    else{
        direction = "D";
    }
    col = current.getFirstSquare().getCol()+1;
    row = current.getFirstSquare().getRow()+1;
    answer = current.getAnswer().toUpperCase();
    clue = current.getClue().replaceAll("'","\"\";
    .
    length = current.getLength();
    allClues+="\"clue\"{\"+number+\"\"+direction+\"
    \"+col+\"\"+row+\"\"+answer+\"\"+clue+\"\"+length+\"\"};
}

Figure 7.4: Java String Representation of \TeX Clue Section

String cmd= "latex filename"
Runtime r= Runtime.getRuntime();
//check for free memory
int freemp= (int) (100*r.freeMemory()/r.totalMemory());
System.out.println (freemp + "% of JVM memory is free.");
//create process object
Process p=r.exec(cmd);

Figure 7.5: DVI and PDF Production
7.4.4 Stage Four - Making the PDF File Available to the User.

Due to security constraints it is exceptionally difficult to pass a file from the server to the user via an applet. However, as can be experienced on millions of sites on the internet files can be made available for downloading from any site.

This aspect of the program exploits, once again, the Runtime objects found in the Java.io package. In this instance upon clicking the “View PDF” button the user is prompted to enter the filename including the .pdf extension. Next runtime objects are created and the following command is executed:

- acrord32 filename.pdf

The above command causes the Adobe Acrobat Reader program to open and display the file specified in a new window. The user can then save or print a copy as desired.
7.5 Sample Output

7.5.1 Solution

Themed Crossword: Sport

\[
\begin{array}{cccccc}
\text{ACROSS} & \text{DOWN} \\
1 & 6 & \text{a person or animal that is markedly unusual or deformed (5)} & \text{a brief swim in water (3)} \\
5 & 2 & \text{the pursuit and killing or capture of wild animals regarded as a sport (4)} & \text{the act of **** as a sport (3)} \\
6 & 3 & \text{a headlong plunge into water (4)} & \text{a person trained to compete in sports (7)} \\
8 & 7 & \text{a living thing that has (or can develop) the ability to act or function independently (5)} & \text{a humorous play on words; ‘I do it for the **** of it’; ‘his constant punning irritated her’ (3)} \\
& 4 & \text{a message whose ingenuity or verbal skill or incongruity has the power to evoke laughter (3)} & \text{a brief swim in water (3)} \\
\end{array}
\]

Figure 7.6: Solution Crossword
7.5.2 Blank

Themed Crossword: match

ACROSS
4 imitate uncritically and in every aspect; "Her little brother ******** her behavior" (3)
5 compete for something; engage in a contest; measure oneself against others (3)

DOWN
1 a member of a widespread secret fraternal order pledged to mutual assistance and brotherly love (5)
2 be sufficient; be adequate, either in quality or quantity; ‘A few words would answer’; ‘This car suits my purpose well’; ‘Will $100 do?’; ‘A ‘B’ grade doesn’t suffice to get me into medical school’; ‘Nothing else will ********’ (5)
3 a friction match with a large head that will stay alight in the wind (5)

Figure 7.7: Blank Crossword
7.6 Problems Encountered

The primary issue here concerned the escaping of illegal characters. Although this very easily rectified, it proved troublesome during debugging as DVI and PDF files were produced with half of the text flying off the page and no apparent explanation. The problem characters are:

- &
- $
- \%
- \#
- -
- {
- }

These characters can be escaped by preceding each occurrence of them by a \. However, the backslash in Java is a special character, hence given that Java is writing the file to the server each of these characters must be preceded by TWO backslashes so that the Java interpreter will correctly print, for example, \& in the .tex file so that the \LaTeX interpreter will successfully print &.

\begin{verbatim}
Java \\\n\LaTeX\\
\end{verbatim}

7.7 Conclusion

The printing interface developed is intuitive and easy to use. A great degree of optionality is afforded to the user. Crosswords are available with and without solution, the user submits a filename of their choice and can choose to print or save it to their own machine. The user has access to all crosswords they saved to the server.
Chapter 8

Restricting Database Access

8.1 Introduction

The aim of this section is to provide a detailed account of how the restriction of access to the database is implemented. Furthermore information will be provided on how the super-user can change the password if required.

8.2 Implementation

This aspect of the program involved making changes to the user interface, the addition of a password table to the dictionary database and the inclusion of additional methods in the MultiServerThread class and DatabaseConnection class.

8.3 Basic Algorithm

- Prompt user for password
- Check password against password in database
- Grant access if password is valid
- Otherwise refuse access

8.4 Changes to the Database

Another table, named PASSWORDS was added to the dictionary database. This table has two fields
CHAPTER 8. RESTRICTING DATABASE ACCESS

- User
- Password

The following MySQL command was used to insert the table:

- CREATE TABLE PASSWORDS(user VARCHAR(20) NOT NULL, password VARCHAR(8) NOT NULL)

This table forms part of the dictionary database and can be accessed at all times during the execution of the program.

8.5 Changes Made to the MultiServerThread Class

The purpose of this Java file is to control communications between the program and the database. Socket objects are used to connect to the server, by means of a thread, which in turn connects to the database. These objects are initialised when the program starts and are available for its duration. The run method controls the flow of communication between the program, server and database. Several commands may be sent to this method so that actions may be carried out. The actions included by Graham (2003) are the following:

- Crossword Creation
- Uploading of Database (Text File Format)
- Uploading of Database (SQL format)

Another action was required namely a “password” method. This method communicates to the applet whether or not the password is valid. Within this method a DatabaseConnection object, dictionaryDatabase, which encapsulates the characteristics of the connection to the dictionary database, is employed. The boolean method isValidPassword(String s), a member of the DatabaseConnection Class is invoked on this object. The result returned is recorded and is subsequently passed in String form to the applet.
8.6 The isValidPassword Method

This method, a member of the DatabaseConnection class, is invoked by the MultiServerThread class. In isValidPassword a MySQL query is executed to extract the password from the database and the equals method from Java’s String class is used to compare the string returned by the MySQL query. The return type of this method is boolean, hence true if the passwords match and false if they are different. This value is communicated to MultiServerThread.
8.7 Adapting the Applet

The system developed by Graham (2003) for the uploading of data to the database involves the user selecting “other” in the drop down menu of available dictionary databases. Instead of being initially asked as to the nature of the database due to be uploaded, the user is firstly asked for a password. This password is passed to MultiServerThread and MultiServerThread responds to the applet in String form. A positive response results in the continuation of the database uploading process, a negative response results in the display of a “Password Invalid” dialogue box.

```java
password_entered = newDictionaryPane.showInputDialog("Please enter password:");
// call multiserver
out.println(MultiServerThread.PASSWORD);
// submit password
out.println(password_entered);
// check response
valid_pass = in.readLine();
if(valid_pass.equals("YES")){
    form = newDictionaryPane.showInputDialog("Is the new dictionary in the form of a
database or a file?");
    .
    .
    .
```

Figure 8.3: Display of Password Dialogue

8.8 Conclusion

The password entry procedure is coherent. The user is informed if they have not entered a valid password. Furthermore changing the password is a trivial matter as the super-user may alter it periodically in the dictionary database itself, which is accessible when the program is not running. This implies that recompilation of the Java source code is not necessary upon changing a password as this implementation ensures that the password is independent of the program.
Chapter 9

Provision of user Help File

9.1 Introduction

This section describes the content of the user help file and how it is made available to the user.

9.2 Help File Contents

There are three main sections to the help file:

1. Crossword Generation
2. Uploading Databases
3. Changing Passwords
4. Viewing & Printing Crosswords

These sections cover issues that are likely to arise during the use of the program. The “Crossword Generation” section describes the kinds of crosswords that can be developed using each of the three database resources available: WordNet2, Websters Unabridged Dictionary, a User Database. In each instance information is provided with respect to:

- Theme Selection
- Improving Success Rates
The “Uploading Databases” chapter describes the nature of the text file that the user may wish to upload, underlining several required structural properties. Attention is also paid to stipulating the exact location of the text file or database that is to be uploaded.

The “Changing passwords” section provides instruction on how to change the password that permits the user to upload databases.

Finally, the section on “Viewing & Printing” crosswords describes the steps necessary to produce a hard copy of the crossword generated.

9.3 Making Help File Available to the User

A “Help!” button was added to the interface. When this button is clicked the applet communicates with the “MultiServerThread” class. From here, use is made of the Java.io package. Runtime and Process objects are created. These allow commands, normally executed manually at a command prompt, to be executed automatically from within the application. The following command is executed on the Runtime object:

```
- acrord32 C:\Program Files\EasyPHP1-7\www\Crossword Generator\user.pdf
```

This results in the Adobe Acrobat Reader opening the user.pdf help file in a new window. This file can then be browsed while the program is executing. A copy of the user help file can be viewed in Appendix F.

9.4 Conclusion

The provision of a user help file enhances the system’s usability and will aid the user in creating more interesting crosswords. The feature is available to the user at all times.
Chapter 10

Conclusion

The crossword generator has been extended considerably. Of particular note is the successful integration of a WordNet2 MySQL database to facilitate the generation of themed crosswords. The algorithm employed is such that maximum coverage of words within the WordNet2 database is guaranteed. Furthermore, the algorithm is sufficiently generic to accommodate WordNets for other languages. The themed crosswords developed are of a consistently high-quality, as confirmed by independent analysis.

The inclusion of a printing interface has further enhanced the functionality of the program in a language learning environment. Crosswords may be printed and saved for future reference. Optionality is afforded to the user, in that they can choose to have the crosswords displayed with or without the solution.

Two new features that address security and usability include, controlling database access by means of a password and the provision of a user help file to deal with queries arising from the use of the program. These extensions contribute to the development of the program as a “complete package” that addresses all of the user’s requirements.

10.1 Further Study

An obvious extension of the current program would be the inclusion of other WordNets. Given that all WordNets are built on the same underlying structure, this aspect should be relatively easy to implement. Of greater difficulty, however, would be the integration of WordNet’s inter-lingual index, that links
synsets in one language with corresponding synsets in another language. This would render the generation of multi-lingual themed crosswords possible.

A more efficient algorithm for the generation of random symmetric grids free of one and two letter words would enhance the program’s execution time. At present, grids are fully complete before they are tested for one and two letter words. Ideally, grids should be tested at regular intervals during the grid building process to remove doomed grids as early as possible. Decreasing the number of black squares present in a grid will inevitably result in fewer grids containing one and two letter words.

An interesting study could involve the comparison of themed crosswords generated by the crossword generator and the New York Times themed crosswords. Not only would this highlight the quality or otherwise of the generated crosswords, it may provide an insight into which concepts require further development within WordNet.

Finally, overcoming the technical difficulties currently experienced with Linux operating systems the Java Virtual Machine and SICStus Prolog would make the program accessible to a wider audience.

10.2 Concluding Remarks

Overall, this project achieved exactly what it set out to achieve. Crosswords are themed, printable and portable. WordNet has been successfully incorporated into the system. The use of the program in a language learning environment is now a distinct possibility.
Bibliography


Appendix A

Screen Shots

Figure A.1: Introductory Screen
Figure A.2: A Completed Crossword
Figure A.3: Creating PDF File
Figure A.4: Uploading to Database
Appendix B
Evaluated Crosswords

Themed Crossword: 1

<table>
<thead>
<tr>
<th>ACROSS</th>
<th>DOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 a person or animal that is markedly unusual or deformed (5)</td>
<td>6 a brief swim in water (3)</td>
</tr>
<tr>
<td>5 the pursuit and killing or capture of wild animals regarded as a sport (4)</td>
<td>2 the act of rowing as a sport (3)</td>
</tr>
<tr>
<td>6 a headlong plunge into water (4)</td>
<td>3 a person trained to compete in sports (7)</td>
</tr>
<tr>
<td>8 a living thing that has (or can develop) the ability to act or function independently (5)</td>
<td>7 a humorous play on words; ‘I do it for the ______ of it’; ‘his constant punning irritated her’ (3)</td>
</tr>
<tr>
<td>4 a message whose ingenuity or verbal skill or incongruity has the power to evoke laughter (3)</td>
<td></td>
</tr>
</tbody>
</table>
Themed Crossword: 2

**ACROSS**
1. release (a liquid) in drops or small quantities; '********** sweat through the pores' (5)
2. the 4th largest of the Great Lakes; it is linked to the Hudson River by the New York State Barge Canal (4)
3. a river in central England that flows generally northeastward to join with the Ouse River and form the Humber (5)
4. a shallow salt lake in south central Australia about 35 feet below sea level; the largest lake in the country and the lowest point on the continent (4)
5. eliminate from the body; ‘Pass a kidney stone’ (5)
6. a river in northern England that flows east to the North Sea (4)

**DOWN**
1. eliminate (substances) from the body (5)
2. an assemblage of parts that is regarded as a single entity; ‘how big is that part compared to the whole?’; ‘the team is a ********’ (4)
3. a river in central England that flows generally northeastward to join with the Ouse River and form the Humber (5)
ACROSS
1  a musical work that has been created; ‘the composition is written in four movements’ (4)
5  a funeral lament sung with loud wailing (4)

DOWN
4  people descended from a common ancestor; ‘his family has lived in Massachusetts since the Mayflower’ (4)
2  a musical work that has been created; ‘the composition is written in four movements’ (5)
3  any communication that encodes a message; ‘signals from the boat suddenly stopped’ (4)
Themed Crossword: 4

**ACROSS**

1. take without the owner’s consent; ‘Someone stole my wallet on the train’; ‘This author stole entire paragraphs from my dissertation’ (5)

4. pick or gather berries; ‘We went ******** in the summer’ (5)

**DOWN**

1. express in words; ‘He said that he wanted to marry her’; ‘tell me what is bothering you’; ‘state your opinion’; ‘state your name’ (3)

2. to come or go into; ‘the boat entered an area of shallow marshes’ (5)

3. doing as one pleases or chooses; ‘if I had my ********’ (3)
ACROSS
4  imitate uncritically and in every aspect; ‘Her little brother **** her behavior’ (3)
5  compete for something; engage in a contest; measure oneself against others (3)

DOWN
1  a member of a widespread secret fraternal order pledged to mutual assistance and brotherly love (5)
2  be sufficient; be adequate, either in quality or quantity; ‘A few words would answer’; ‘This car suits my purpose well’; ‘Will $100 do?’; ‘A ’B’ grade doesn’t suffice to get me into medical school’; ‘Nothing else will **********’ (5)
3  a friction match with a large head that will stay alight in the wind (5)
Themed Crossword: 6

ACROSS
1 a diet that does not require chewing; advised for those with intestinal disorders (3)
5 seed of the annual grass Avena sativa (spoken of primarily in the plural as '******') (3)
7 (British) the dessert course of a meal ('******' is used informally) (3)
8 oval reproductive body of a fowl (especially a hen) used as food (3)

DOWN
4 lining of the stomach of a ruminant (especially a bovine) used as food (5)
6 a sweetened beverage of diluted fruit juice (3)
2 a sweet drink containing carbonated water and flavoring; ‘in New England they call sodas tonics’ (3)
3 a slice of meat cut from the fleshy part of an animal or large fish (5)
ACROSS
1 ten-armed oval-bodied cephalopod with narrow fins as long as the body and a large calcareous internal shell (6)
3 social or solitary hymenopterans typically having a slender body with the abdomen attached by a narrow stalk and having a formidable sting (4)
5 warm-blooded egg-laying vertebrates characterized by feathers and forelimbs modified as wings (4)
6 brilliantly colored pheasant of southern Asia (6)

DOWN
1 decapod having eyes on short stalks and a broad flattened carapace with a small abdomen folded under the thorax and pincers (4)
3 and of several small active brown birds of the northern hemisphere with short upright tails; they feed on insects (4)
2 any of various tailless stout-bodied amphibians with long hind limbs for leaping; semiaquatic and terrestrial species (4)
4 a tame parrot (4)
**ACROSS**

3  a blue dye obtained from plants or made synthetically (6)
4  a pure flat white with little reflectance (5)
6  turn golden (6)

**DOWN**

1  ******** paint used by an artist (3)
4  a blue-green that is one of the primary pigments (4)
2  have and exert influence or effect; ‘The artist’s ******** influenced the young painter’; ‘She ******** on her friends to support the political candidate’ (4)
5  a usually soluble substance for staining or coloring e.g. fabrics or hair (3)
Appendix C

5 Themed Crosswords: Blank

Themed Crossword: Country

ACROSS
1  the capital and largest city of Zambia (6)
3  a landlocked republic in northwestern Africa; achieved independence from France in 1960; ******** was a center of West African civilization for more than 4,000 years (4)
5  a country of southeastern Asia that extends southward along the Isthmus of Kra to the Malay peninsula; ‘Thailand is the official name of the former ******** ’ (4)
6  a Scandinavian kingdom in the eastern part of the Scandinavian Peninsula (6)

DOWN
1  a mountainous landlocked communist state in southeastern Asia; achieved independence from France in 1949 (4)
3  the capital of Maldives in the center of the islands (4)
2  the biblical name for ancient Syria (4)
4  a theocratic Islamic republic in the Middle East in western Asia; ******** was the core of the ancient empire that was known as Persia until 1935; rich in oil; involved in state-sponsored terrorism (4)
Themed Crossword: Language

ACROSS
1 the words of something written; ‘there were more than a thousand words of ******** ’; ‘they handed out the printed ******** of the mayor’s speech’; ‘he wants to reconstruct the original ******** ’ (4)
3 the language of the nomadic Lapp people in northern Scandinavia and the Kola Peninsula (4)
5 Kamarupan languages spoken in western Burma and Bangladesh and easternmost India (4)
6 a family of American Indian languages spoken by Mayan peoples (4)

DOWN
1 the dialect of Albanian spoken in southern Albania and in areas of Greece and Italy (4)
3 reading or glancing through quickly (4)
2 a branch of the Tai languages (4)
4 a dialect of the Chiwere language spoken by the ******** people (4)
Themed Crossword: Shape

ACROSS
1  a round shape formed by a series of concentric circles (5)
5  form a knot or bow in; '****** a necktie' (3)
6  a continuous portion of a circle (3)
8  form metals with a swage (5)

DOWN
4  an angular shape characterized by sharp turns in alternating directions (3)
7  form into the shape of a ******; ‘She cupped her hands’ (3)
2  a groove or furrow (especially one in soft earth caused by wheels) (3)
3  a part of a forked or branching shape; ‘he broke off one of the branches’; ‘they took the south fork’ (3)
Themed Crossword: Fruit

ACROSS

3  pod of the peanut vine containing usually 2 nuts or seeds; ‘groundnut’ and ‘monkey nut’ are British terms (6)

5  the small hard nutlet of a drupe or drupelet; the seed and the hard endocarp that surrounds it (6)

DOWN

3  any of various juicy purple- or green-skinned fruit of the genus Vitis; grow in clusters (5)

1  a several-seeded dehiscent fruit as e.g. of a leguminous plant (3)

4  the fruit or seed of a ***** plant (3)

2  dried plum (5)
Themed Crossword: City

**ACROSS**
1. a city in northwestern Iran; a place of pilgrimage for Shiite Muslims (3)
2. a city in the European part of Russia (3)
3. a port in southern Lebanon on the Mediterranean Sea; formerly a major Phoenician seaport famous for silks (3)
4. the former capital and 2nd largest city of Brazil; chief Brazilian port; famous as a tourist attraction (3)
5. the capital and largest city of Japan; the economic and cultural center of Japan (3)
6. a city in north central Morocco; religious center (3)

**DOWN**
4. a city in northeastern Egypt at the head of the Gulf of ******** and at the southern end of the ******** Canal (4)
2. a port in western Israel on the Mediterranean; incorporated into Tel Aviv in 1950 (4)
Appendix D

5 Themed Crosswords: Solution

Themed Crossword: Country

<table>
<thead>
<tr>
<th>ACROSS</th>
<th></th>
<th>DOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>the capital and largest city of Zambia (6)</td>
<td>1 a mountainous landlocked communist state in southeastern Asia; achieved independence from France in 1949 (4)</td>
</tr>
<tr>
<td>3</td>
<td>a landlocked republic in northwestern Africa; achieved independence from France in 1960; ******** was a center of West African civilization for more than 4,000 years (4)</td>
<td>3 the capital of Maldives in the center of the islands (4)</td>
</tr>
<tr>
<td>5</td>
<td>a country of southeastern Asia that extends southward along the Isthmus of Kra to the Malay peninsula; ‘Thailand is the official name of the former ******** ’ (4)</td>
<td>2 the biblical name for ancient Syria (4)</td>
</tr>
<tr>
<td>6</td>
<td>a Scandinavian kingdom in the eastern part of the Scandinavian Peninsula (6)</td>
<td>4 a theocratic islamic republic in the Middle East in western Asia; ******** was the core of the ancient empire that was known as Persia until 1935; rich in oil; involved in state-sponsored terrorism (4)</td>
</tr>
</tbody>
</table>
# Themed Crossword: Language

<table>
<thead>
<tr>
<th>ACROSS</th>
<th>DOWN</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>1</strong> the words of something written; ‘there were more than a thousand words of ******** ’; ‘they handed out the printed ******** of the mayor’s speech’; ‘he wants to reconstruct the original ******** ’ (4)</td>
<td><strong>1</strong> the dialect of Albanian spoken in southern Albania and in areas of Greece and Italy (4)</td>
</tr>
<tr>
<td><strong>3</strong> the language of the nomadic Lapp people in northern Scandinavia and the Kola Peninsula (4)</td>
<td><strong>3</strong> reading or glancing through quickly (4)</td>
</tr>
<tr>
<td><strong>5</strong> Kamarupan languages spoken in western Burma and Bangladesh and easternmost India (4)</td>
<td><strong>2</strong> a branch of the Tai languages (4)</td>
</tr>
<tr>
<td><strong>6</strong> a family of American Indian languages spoken by Mayan peoples (4)</td>
<td><strong>4</strong> a dialect of the Chiwere language spoken by the ******** people (4)</td>
</tr>
</tbody>
</table>
ACROSS
1  a round shape formed by a series of concentric circles (5)
5  form a knot or bow in; ‘****** a necktie’ (3)
6  a continuous portion of a circle (3)
8  form metals with a swage (5)

DOWN
4  an angular shape characterized by sharp turns in alternating directions (3)
7  form into the shape of a ****** ; ‘She cupped her hands’ (3)
2  a groove or furrow (especially one in soft earth caused by wheels) (3)
3  a part of a forked or branching shape; ‘he broke off one of the branches’; ‘they took the south fork’ (3)
Themed Crossword: Fruit

**ACROSS**
3 pod of the peanut vine containing usually 2 nuts or seeds; ‘groundnut’ and ‘monkey nut’ are British terms (6)
5 the small hard nutlet of a drupe or drupelet; the seed and the hard endocarp that surrounds it (6)

**DOWN**
3 any of various juicy purple- or green-skinned fruit of the genus Vitis; grow in clusters (5)
1 a several-seeded dehiscent fruit as e.g. of a leguminous plant (3)
4 the fruit or seed of a ***** plant (3)
2 dried plum (5)
Themed Crossword: City

ACROSS
1 a city in northwestern Iran; a place of pilgrimage for Shiite Muslims (3)
3 a city in the European part of Russia (3)
4 a port in southern Lebanon on the Mediterranean Sea; formerly a major Phoenician seaport famous for silks (3)
5 the former capital and 2nd largest city of Brazil; chief Brazilian port; famous as a tourist attraction (3)
6 the capital and largest city of Japan; the economic and cultural center of Japan (3)
7 a city in north central Morocco; religious center (3)

DOWN
4 a city in northeastern Egypt at the head of the Gulf of ******** and at the southern end of the ******** Canal (4)
2 a port in western Israel on the Mediterranean; incorporated into Tel Aviv in 1950 (4)
Appendix E

Developer & Installation Guide

E.1 Introduction

This appendix is dedicated to those who may wish to extend this project or run this project on their own computer. Information regarding the installation of all packages required for the successful execution of the program is provided. Common errors and pitfalls are also given.

E.2 Packages - Installation & Configuration

The program currently runs on a Windows Platform. See chapter 3 for further information on the project’s compatibility with Linux platforms. The information provided here is sufficient to get the program up and running but is by no means exhaustive.

E.2.1 EasyPHP

EasyPHP is a software package designed for Windows. This package comprises of an Apache Server, a MySQL database and full PHP language support. Of relevance to this project is the Apache Server and the MySQL database. The MySQL database hosts all databases exploited by the crossword generator. The Apache server facilitates communication between the crossword generator, the database and the web interface.

• Source

EasyPHP can be downloaded from the EasyPHP website EasyPHP (2004). Installation documentation is provided. Of note is the location of the web di-
directory in which all the program files should be saved: C:/Program Files/EasyPHP1-7/www. The server name is localhost. Further and more detailed information can be found in the httpd.conf file which comes with the software.

### E.2.2 Java

The principal programming language used in the development of the crossword generator is Java. The Java installation comprises of a Java Virtual Machine technology and a Java Runtime Environment both of which facilitate the compilation and execution of Java programs. A windows version of Java must be installed.

- **Source**

  The version of Java used in the development of this project is: Java(TM) 2 Runtime Environment, Standard Edition (build 1.4.2_02b03). This is freely available from Sun (2005). The full Java Development Kit is required. Note that changing to a different version of Java (e.g. Java 5) could have a detrimental effect on the crossword generator. See the Section E.2.3 for further details.

### E.2.3 SICStus Prolog

SICStus Prolog is a Prolog development system. Its role in the project is fundamental. Prolog is responsible for finding the solution set of words to satisfy the grid generated. See Graham (2003)) for further information. Communication between SICStus and Java is achieved using the Java.Jasper package. Jasper is a bi-directional interface between Java and SICStus.

- **Compatibility**

  At present Jasper only works under the following system configurations:

  - **Windows:**
    - Java Development Kit 1.3.1
    - Java Development Kit 1.2.2 & Java Development Kit 1.4.X are expected to work
  - **Linux**
    - Java Development Kit 1.3.1
– Java Development Kit 1.4.1 is expected to work
– Java Development Kit 1.5

As described in 3 serious difficulties arose during attempts to run the crossword generator on Linux systems using both Java 1.5 and Java 1.4.2.

• Source

SICStus Prolog is available for downloading from the SICStus Prolog website (SICStus, 2004).

**E.2.4 Adobe Reader 5**

The Adobe Reader is free software that allows users to view, print and search PDF files. (PDF: Portable Document Format). Its role in the project is to facilitate the viewing of printable versions of the crossword generated as well as the user help file.

• Source

The Adobe Reader is available from the Adobe site (Adobe, 2005). The version used in this project is version 5. Using more recent versions of Adobe will not adversely effect the crossword generator.

**E.2.5 \LaTeX**

\LaTeX is a document preparation system. Its features are described in detail in Section 7.2.1. \LaTeX facilitates the development of PDF files of the crossword generated by the applet. This tool is available from the \LaTeX site (\LaTeX, 2004).

• Source

A windows version of \LaTeX is downloadable @ http://www.miktex.org/. The installation must include the `crosswd.sty` file. This file is central to the creation of dvi and PDF files.
E.3 Setting of Classpaths

In order for each of the above packages to function effectively, classpaths pointing to certain aspects of each the packages must be set. The procedure to set classpaths for each of the packages on Windows XP follows.

- Start
- Control Panel
- System icon
- Advanced tab
- Environment Variables button
- System Variables frame
- Click on 'classpath'
- Click Edit
- Append each of the following strings separated by a semi-colon to any existing strings:
  - C:\texmf\miktex\bin
  - C:\j2sdk1.4.2.02\bin
  - C:\Program Files\SICStus Prolog 3.12.0\bin
  - C:\Program Files\SICStus Prolog 3.12.0\se\sics\jasper
  - C:\Program Files\Adobe\Acrobat 5.0\Reader
- Restart the system in order for the changes to take effect

Note that the classpaths above refer to the versions of the packages used in the current configuration and should be used as a guide with respect to future installations/configurations. Note also that \LaTeX{} generally sets its environmental variables during installation but that they only take effect after restarting the system.
Appendix F

User Help File

F.1 Introduction

This guide aims to maximise the users use of the Crossword Generator. Attention is paid to all functional aspects of the program.
F.2 Crossword Generation

F.2.1 WordNet2

Introduction
The WordNet2 database is a lexical database for the English language. This resource is was selected for the convenient way in which words related to a specific theme is returned to the user.

Theme Selection
Due to relational aspects of the database only singular nouns and verbs may be entered as a theme - other parts of speech will result in no crosswords being generated.

Improving Success Rates
1. If the information panel indicates that there was not enough words in the database to fill the grid, try reducing the size of the grid.
2. If the information panel indicates that no solution could be found for the grid, try increasing the difficulty level, as this reduces the number of words that appear in the grid and reduces the number of interlocking solutions, hence making the problem of finding a solution set easier to solve.

F.2.2 Websters Unabridged Dictionary

Introduction
This dictionary dates from the early 20th century and hence contains many old and obsolete expressions. It’s most impressive function is the production of random crosswords that are not related to a particular theme.

Theme Selection
It is advisable to use the WordNet2 database for the generation of themed crosswords. Leaving the subject field blank will result in the production of a non-themed crossword.
F.2.3 User Database

Introduction

These are databases added to the program by users. The database names generally describe their contents.

Exploitation of Database

Themed crosswords cannot be generated for user databases. User databases create general crosswords. The entire database is queried for words whose length correspond with the requirements of the grid. The subject field should be left blank - no input is required.

Improving Success Rates

Whether a crossword is generated or not depends on the size of the database and the size of the grid. Larger databases will stand a greater chance of producing enough words for a 7x7 grid than smaller databases.

1. If the information panel indicates that there was not enough words in the database to fill the grid, try reducing the size of the grid.

2. Alternatively try adding more words to the database

3. If the information panel indicates that no solution could be found for the grid, try increasing the difficulty level, as this reduces the number of words that appear in the grid and reduces the number of interlocking solutions, hence making the problem of finding a solution set easier to solve.
F.3 Uploading Databases

F.3.1 Introduction

Databases can be uploaded to the program while the program is running. These databases are then available as a resource to the program. Particular attention should be paid to their content and size. Results will be poor if the database is small and contains little variety of word length. Uploaded databases can be accessed from the drop down database menu until they are removed from the system.

F.3.2 Text File Format

Structure of text file

The text file contains the information that will be interpreted by the program. This information is converted to a MySql database. Each line of the text file contains a separate piece of information. The first line of the file will contain a word, the next line in the file will contain that word’s definition. Note that for words with multiple definitions there must be a separate word-definition pair for each definition.

Location of text file

The user is prompted for the location of the text file. The explicit path must be provided. That means including drive name, folder name and the name of the file. For example

- C:/Database/My Database Files/frenchdatabase.txt

Failing to enter the exact location of the file will result in an error message being displayed and the process will come to an end.

F.3.3 Database Format

Entire databases can be passed to the system. The location of the database must be given explicitly. Field names are altered automatically to suit the system.
F.4 Changing Passwords

F.4.1 How to?

Passwords can only be changed by the superuser - that is an individual who has full access privileges to the MySql database running on the server. This database is independent of the system. The following MySQL command, when run in the MySQL environment, updates the password:

- UPDATE password IN PASSWORDS WHERE user = 'teacher';
F.5 Creating Printable Versions of Crosswords

Two options are available:

- Print Crossword with solution
- Print Crossword with blank grid

These options can be selected from the drop down menu on the interface. Confirm option choice by clicking on "Create PDF". Submit a suitable filename and keep for further reference.

F.5.1 Viewing Crosswords

Crosswords generated and saved to server can be viewed by clicking on the "View PDF" button. Next the user is prompted for the filename of the crossword desired. Remember to include the .pdf extension. If the filename is animal the complete filename is animal.pdf

Adobe Acrobat Reader is required to view the crosswords.