ABSTRACT

This project is the design and implementation of an Interactive Parse-Tree Generator to be used as a teaching tool in selected computer science courses at Texas A&M University – Corpus Christi. The generator is available to both students and instructors via the Internet so as to provide maximal access. The data source for the generator is a C++ statement provided by the user through the keyboard. If the statement is syntactically correct, the output is a graphic display of the corresponding parse tree. If the statement is syntactically incorrect, the generator displays an error message.
# Table of Contents

Background and Rationale................................................................. 1

Narrative......................................................................................... 4

Environment.................................................................................... 11

Procedure......................................................................................... 12

  Overview of System Design.......................................................... 12

Graphical User Interface Development.......................................... 13

Implementation of the Lexical Analyzer......................................... 15

Implementation of the Parsing Sequence........................................ 19

  TreeArea Class............................................................................. 19

  Node Class.................................................................................. 20

  OddJob Class.............................................................................. 21

  Parser Class............................................................................... 23

  StatementParser Class............................................................... 25

  ExpressionParser Class............................................................. 38

Implementation of the Parse Tree Display..................................... 60

Development of the Parse-tree Generator Web Pages..................... 69

Results............................................................................................ 71

Selected Bibliography....................................................................... 72

Appendix A - Context-free Syntax of (C++)-- in BNF...................... 73

Appendix B – Context-free Grammar for this Project...................... 75

Appendix C - Data Dictionary......................................................... 77

Appendix D – Source Code.............................................................. 103
# List of Figures

Figure 1 Parse-tree Generator Home Page............................................................ 5
Figure 2 Parse-tree Generator Help Page............................................................... 5
Figure 3 Page Containing the Key to Parse-tree Symbols......................................... 6
Figure 4 Appearance of Generator Frame.............................................................. 7
Figure 5 Generator Frame with Assignment Statement Entered by the User.............. 9
Figure 6 Generator Frame with Parse Tree............................................................ 10
Figure 7 Flow of Control During Execution of Parse-tree Generator Program........... 12
Figure 8 Appearance of Parse-tree Generator Frame.............................................. 15
Figure 9 Interaction of Classes During Lexical Analysis......................................... 18
Figure 10 Interaction of Classes During Parsing Sequence.................................... 20
Figure 11 Communication among Classes at Beginning of Parsing Sequence............. 24
Figure 12 Communication among Classes During Call to parseOneStatement.......... 25
Figure 13 Communication among Classes During the First Step of Parsing an Assignment Statement............................................................... 27
Figure 14 Communication among Classes During the Second Step of Parsing an Assignment Statement............................................................... 30
Figure 15 Communication among Classes During the Second Step of Parsing an If Statement............................................................... 32
Figure 16 Communication among Classes During Parsing of an “else” with No Braces............................................................... 33
Figure 17 Communication among Classes During Parsing of an “else” with Braces............................................................... 34
Figure 18 Communication among Classes During Parsing of a While Statement..... 36
Figure 19 Communication among Classes During Statement Sequence Parsing...... 38
Figure 20 Communication among Classes During Parsing of Comparison with One Statement ................................................................. 42

Figure 21 Communication among Classes During Parsing of an Expression .......... 44

Figure 22 Communication among Classes During Parsing of an Expression Tail .... 46

Figure 23 Communication among Classes During Parsing of a Factor ................. 49

Figure 24 Communication among Classes During Parsing an Operand
   Without Parentheses ........................................................................ 51

Figure 25 Communication among Classes During Parsing an Operand
   With Parentheses ........................................................................... 52

Figure 26 Communication among Classes During Parsing a Factor Tail
   With an Addition Operator ................................................................ 54

Figure 27 Communication among Classes During Parsing a Factor Tail
   With a Right Parenthesis ................................................................ 56

Figure 28 Communication among Classes During Parsing an Operand Tail
   With Multiplication or Relational Operators ...................................... 58

Figure 29 Example of Overlapping Columns in Tree Display .......................... 68
List of Tables

Table 1 Token Types Assigned to Token Objects........................................ 17
Table 2 Classes and Methods Involved in Lexical Analysis.......................... 18
Table 3 Attributes of a Node Object....................................................... 21
Table 4 Node Types Assigned to Tree Nodes............................................ 22
Table 5 Methods Called from Parser Class............................................... 26
Table 6 Methods Called from StatementParser During the First Step of Parsing an Assignment Statement.................................................. 28
Table 7 Methods Called from StatementParser During the Second Step of Parsing an Assignment Statement.......................................... 31
Table 8 Methods Called from StatementParser During the Second Step of Parsing an If Statement.............................................................. 32
Table 9 Methods Called From StatementParser During Parsing and “else” with No Braces................................................................. 33
Table 10 Methods Called From StatementParser During Parsing and “else” with Braces............................................................... 35
Table 11 Methods Called from StatementParser During the Second Step of Parsing a While Statement................................................... 36
Table 12 Methods Called from StatementParser During Statement Sequence Parsing................................................................. 39
Table 13 Methods Called from ExpressionParser During Parsing of Comparison with One Statement.................................................. 43
Table 14 Methods Called from ExpressionParser During Parsing of an Expression................................................................. 45
Table 15 Methods Called from ExpressionParser During Parsing of an Expression Tail................................................................. 47
Table 16 Methods Called from ExpressionParser During Parsing of a Factor........... 50
Table 17 Methods Called from ExpressionParser During Parsing of an Operand without Parentheses ......................................................... 52
Table 18 Methods Called from ExpressionParser During Parsing of an Operand with Parentheses ......................................................... 53
Table 19 Methods Called from ExpressionParser During Parsing of a Factor Tail with an Addition Operator ......................................................... 55
Table 20 Methods Called from ExpressionParser During Parsing of a Factor Tail with a Right Parenthesis ......................................................... 57
Table 21 Methods Called from ExpressionParser During Parsing of an Operand Tail with Multiplication or Relational Operators ......................... 59
Table 22 Position of Children of Assignment Statement ................................................. 62
Table 23 Position of Children of Simple If Statement ................................................. 63
Table 24 Position of Children of If Statement with Braces ........................................ 64
Table 25 Position of Children of Simple If-else Statement ........................................ 64
Table 26 Position of Children of Complex If-else Statement ..................................... 65
Table 27 Position of Children of Simple While Statement ......................................... 66
Table 28 Position of Children of While Statement with Braces ................................. 67
BACKGROUND AND RATIONALE

An important aspect of any computer science curriculum is learning and understanding parse trees. This is especially true in courses such as Compiler Construction, Programming Languages, and System Software. A thorough comprehension of parse trees, their structure, and their purpose allows the student to more easily fathom such mysteries as language development and language interpretation. At present, most of the teaching of parse trees at Texas A&M University – Corpus Christi is the draw-it-on-the-whiteboard-during-lecture approach. While this method works well, there are some drawbacks. For example, if students wish to practice with additional parse trees after lecture, they have no immediate way of knowing if their answers are correct. This project will offer an interactive approach that provides students with immediate feedback. It also provides the faculty with an additional teaching tool that may be used to enhance the presentation of the subject.

A considerable amount of educational research has been done which shows how the use of multimedia in teaching improves retention and understanding of the material being taught. Several of these studies generally attribute the improvement to Paivio’s Dual Coding Theory (Becker and Dwyer 1998; Najjar 1995; Najjar 1996; Najjar 1998). According to this theory, people tend to process information by one of two relatively independent methods. One method tends to process information more readily if the information is in a verbal form, such as text or audio. The other method tends to process information more readily if the information is in a non-verbal form, such as pictures. Even though the two methods are generally independent, studies have shown that when information is presented in both forms, learning is enhanced (Najjar 1995).
Additional research has shown that, when used for a suitable learning task, multimedia presentations of the material to be learned have a strong influence on how well the information is assimilated (Najjar 1998). If the information to be learned is spatial in nature, pictures appear to communicate the information more effectively. When pictures are used in conjunction with textual presentations, students can more easily establish conceptual connections between the verbal and graphical information. Since a parse tree is so easily represented as a two-dimensional structure, it lends itself to a spatial display. Showing the graphical representation of a parse tree with its corresponding C++ statement in a multimedia setting should enhance the student's ability to understand the concept of parse trees.

The type of student involved in learning a specific task has an effect on the efficacy of educational multimedia (Najjar 1998). Students with limited prior knowledge of a subject tend to benefit from a multimedia presentation of the material to be learned. Adults tend to benefit more from the use of educational multimedia than do children. This project is targeted for adult students in a university environment. Many of these students will have little or no prior exposure to the concept of parse trees.

Other parse-tree generators, both commercial and non-commercial, have been produced. The obvious disadvantage of the commercially available generators is the expense, particularly if it is to be made available to a large number of people. For example, SandStone Technology, Inc. in La Jolla, California, offers their Visual Parse ++ program for a minimum of $495 per copy. The one non-commercial generator found on the World Wide Web (Umrigar 1997) offers several parsing methods, although the author admits there are several "bugs" in the program. The author also states the parser may not
perform as expected on all platforms. The parse-tree generator component of this graduate project was built “in-house” and tailored to meet the needs of the Computing and Math Sciences Department at Texas A&M University – Corpus Christi.