To my parents, Martha and Charles Hollingsworth,  
my wife, Janet Vician Hollingsworth,  
and children, Emma Jean and Max Wilson.
ACKNOWLEDGMENTS

Sincere thanks goes to Bruce Weide, my adviser. Without your guidance, patience and seemingly limitless interest in my research, I would have made too many “wrong turns” and not enough “right turns.” I also wish to thank Stuart Zweben and Neelam Soundararajan, the other members of my reading committee, for their suggested clarifications and constructive comments. Your comments and suggestions helped to make this dissertation be a more complete document.

I am also grateful to current and former members of the Reusable Software Research Group at Ohio State. Thanks goes to Bill Ogden, Stephen Edwards, Doug Harms, Wayne Heym, and Murali Sitaraman. It has been a pleasure to work with you along the way. Special thanks goes to Mike Stovsky for helping me navigate through this last requirement for a Ph.D.

Finally, I am indebted to Janet, my wife, for allowing me to take on this endeavor. Furthermore, without your encouragement and backing, I would not have been able to finish what I started. I am also grateful to Emma and Max, my children, for their patience and understanding, especially while I worked endless hours on my “black book.”
VITA

November 7, 1958 ........................................... Born — Kokomo, Indiana

1982............................................................ Bachelor of Science
Indiana University
Bloomington, Indiana

1984............................................................ Master of Science
Purdue University
West Lafayette, Indiana

1984 - 1986 ................................................... Software Design Engineer
Texas Instruments
Dallas, Texas

1986 - Present............................................... Research Assistant / 
Teaching Assistant
The Ohio State University
Columbus, Ohio

1987 - Present............................................... Owner
Hollingsworth Solutions
Columbus, Ohio

1991 - Present............................................... Principal Research Scientist
Battelle Memorial Institute
Columbus, Ohio
PUBLICATIONS


FIELDS OF STUDY

Major Field: Computer and Information Science

Minor Fields: Software Engineering, Artificial Intelligence, Semantics of Distributed Computing, Compiler Construction Techniques, and Programming Language Design
# TABLE OF CONTENTS

ACKNOWLEDGMENTS .................................................................................. iii  
VITA ........................................................................................................ iv  
LIST OF FIGURES .................................................................................. xi
LIST OF TABLES ..................................................................................... xiii

CHAPTER I  Introduction ........................................................................ 1  
  1.1 The Thesis .................................................................................. 2
  1.2 Definition of a Software Discipline .............................................. 2
  1.3 A Discipline for Constructing High-Quality Components .......... 4   
    1.3.1 The Discipline’s Goal and Properties .................................. 4
    1.3.2 The Discipline’s Principles .................................................. 5
    1.3.3 Requiring the Discipline to Be Practical ............................ 5
    1.3.4 Related Work .................................................................. 9
  1.4 Outline of Dissertation ............................................................... 10

CHAPTER II  First Principles for Constructing Components in Ada .......... 11  
  2.1 Component Interface Principles ............................................... 11   
    2.1.1 The Unit of Modularity ...................................................... 12
    2.1.2 Location of Abstract State ................................................ 15
    2.1.3 Variable Finalization ....................................................... 17
    2.1.4 Variable Initialization ....................................................... 22
    2.1.5 Testing Preconditions ....................................................... 25
    2.1.6 Limited Private Types ...................................................... 28
    2.1.7 Data Movement Operation ............................................... 33
    2.1.8 Operations as Procedures ............................................... 35
    2.1.9 Package Finalization ....................................................... 40
    2.1.10 Package Initialization ..................................................... 41
    2.1.11 Conceptual Type Parameters ......................................... 45
    2.1.12 Conceptual Type’s Standard Operations ......................... 46
LIST OF FIGURES

Figure 1 — A Software System as a Collection of Components .................2
Figure 2 — Reverse Queue ...........................................................6
Figure 3 — Area of Reasoning for Certifying Reverse Queue ....................7
Figure 4 — Area of Reasoning Because of a Leaky Abstraction ...............8
Figure 5 — Linked List Representation for Queue..............................17
Figure 6 — Initial Configuration for a Queue Variable .........................22
Figure 7 — Items Needed When Reasoning About a Client ....................29
Figure 8 — Array Representation of a Bounded Queue .........................29
Figure 9 — Singly Linked List Representation of a Bounded Queue ..........30
Figure 10 — Non-local Reasoning Required by Bounded Queue ............32
Figure 11 — Multiple Implementations of Queue_Template ..................59
Figure 12 — Multiple Implementations of Queue_Template in Ada ..........60
Figure 13 — Client Implementation .................................................61
Figure 14 — Component Implementation .........................................66
Figure 15 — Items Examined to Locally Certify Composability .............77
Figure 16 — Area of Reasoning for Local Certification of Correctness ......80
Figure 17 — A Client of Components Having Abstract Module-Level State ..83
Figure 18 — Items Required to Certify Correctness When Abstract Module-
Level State is Present .............................................................84
Figure 19 — Items Examined to Locally Certify Reusability and
Understandability ..................................................................86
Figure 20 — Bootstrapping From Raw Ada .......................................128
Figure 21 — Simulated Pointers Using Parallel Arrays .........................142
Figure 22 — Free List of Individual Pieces of Storage .........................153
Figure 23 — Free List of Linked Structures ....................................154
Figure 24 — Simulated Pointers for Doubly Linked Structure ..............155
Figure 25 — Two Positions in Their Own Cycle ................................159
Figure 26 — Two Positions Linked by One Cycle.................................159
Figure 27 — Cycle Containing More Than Two Positions.....................160
Figure 28 — One Position Removed From a Cycle Using Transpose_After .160
Figure 29 — One Position Removed From a Cycle Using
Transpose_Before ..............................................................................160
Figure 30 — Splitting a Cycle into Two.............................................161
Figure 31 — Joining Two Cycles .....................................................161
Figure 32 — Cyclic Linked Structure With Three Positions ..........168
Figure 33 — Circular Doubly Linked List Implementation..............169
Figure 34 — Representation Using a Shadow List.........................170
LIST OF TABLES

Table 1 — Component Interface Principles (Section 2.1) Followed by Published Component Libraries and/or Guidelines..............175