

Copyright is owned by the Author of the thesis. Permission is given for a copy to be downloaded by an individual for the purpose of research and private study only. The thesis may not be reproduced elsewhere without the permission of the Author.

A Direct Manipulation Object-Oriented Environment to Support Methodology-Independent CASE Tools

A thesis presented in partial fulfilment of the requirements for the
degree of Master of Science in Computer Science at Massey University

Bei Zhong

1996

Abstract

The aim of the thesis is research into application of direct-manipulable OO graphical environments to the development of methodology-independent CASE tools. In this thesis, a Methodology-Independent Graphical OO CASE Environment (MIGOCE) is proposed.

MIGOCE consists of three parts: OO Notation Workshop, OO Notation Repository and Universal OO Diagramming Tool. OO Notation Workshop is an OO graphical editor which is used to design existing and new notations; OO Notation Repository is a notation database that stores different notations designed by the notation workshop; Universal OO Diagramming Tool is an upper-CASE graphical environment, by which a user can draw arbitrary OO diagrams of different methodologies. The MIGOCE database management system provides OO notation sets management, OOA/OOD diagrams management and OO repository management for data integrity and sharing.

MIGOCE has three outstanding characteristics: Methodology-independence, Directly-manipulable graphical environment and Easily-expanded program structure. MIGOCE is completely methodology-independent. It not only supports existing OO methodologies, but also supports users' own notation designs. It provides support for mixing, updating existing methodologies or defining new ones. It typically allows the user to switch quickly different OO notation sets supported by corresponding methodologies for designing diagrams. Direct manipulation interfaces of MIGOCE enable it more flexible and distinctive. The user can easily add, delete, edit or show notation shapes, and get the system feedback very quick on the screen. The MIGOCE system itself is programmed using object-oriented programming language — C++. Its program structure enable the functions of itself easy to be modified and expanded.

Although MIGOCE is a prototype, it provides a new way to develop the real methodology-independent OO CASE environment. So far, the way and style taken by MIGOCE have not been found in OO CASE literatures. This system gives a complete possibility of implementing a methodology-independent OO CASE tool and shows distinct effectiveness of such a tool in practice.

Acknowledgments

I would like to express my grateful thanks to my first supervisor Dr. Daniela Mehandjiska-Stavreva for her supervision, valuable guidance, and encouragement throughout my study. I deeply appreciate her patience in reading and constructively criticising the manuscript of this thesis.

Special thanks go to my second supervisor Dr. Chris Phillips for his helpful advises on my work, and his patience in modifying my manuscript.

My deep appreciation is also extended to all staff and postgraduate students in the Department of Computer Science for their assistance and friendship.

Lastly, I would especially like to express my gratitude to my family for their continued support and encouragement.

Table of Contents

Abstract

Acknowledgments

Table of Contents

Chapter 1. Introduction	1
1.1. Object-Oriented Development	1
1.2. Overview of CASE Tools.....	4
1.3. A Methodology-Independent Graphical CASE Environment	6
1.4. Overview of the Thesis	8
Chapter 2. Software Development Approaches	9
2.1. Functional Approach	9
2.2. Object-Oriented Approach	10
2.2.1. Key Concepts	11
2.2.2. Existing OO Methodologies	13
Chapter 3. CASE Tools.....	29
3.1. Life cycle of software system development.....	30
3.1.1. "Waterfall" Life Cycle.....	30
3.1.2. OO Life Cycle	31
3.2. CASE Tools Categories	33
3.3. OO CASE Tools	35
3.3.1. Background	35
3.3.2. Types of OO CASE Tools.....	36
3.4. Evaluation of the Current CASE Tools	37
3.4.1. Advantages	37
3.4.2. CASE Tools Problems.....	38
3.5. CASE Tools Development Research Strategies	40
3.5.1. Federated CASE Environment.....	40
3.5.2. CASE Shell.....	42
3.6. Rationale Behind MIGOCE Development	44
3.6.1. Motivation	44
3.6.2. MIGOCE Overall Architecture	46

Chapter 4. Task Analysis and User Interface Design	51
4.1. User Interface Design of OO CASE Tools	52
4.1.1. The Role of Task Analysis	52
4.1.2. User Interface Styles	54
4.1.3. Overview of Techniques for Task Analysis	55
4.2. Task Knowledge Structures and Knowledge Analysis of Tasks	56
4.2.1. Task Knowledge Structures	56
4.2.2. Knowledge Analysis of Tasks	59
4.3. A User Task Model	60
Chapter 5. The Static and Dynamic Object Models of MIGOCE	65
5.1. Modeling OO Notation Workshop	65
5.1.1. OO Notation Workshop Specifications	65
5.1.2. The Class Hierarchy of the Basic Shapes	66
5.1.3. The Static Object Model of OO Notation Workshop	69
5.1.4. The Main Behaviours of OONW	71
5.2. Modeling Universal OO Diagramming Tool	75
5.2.1. UOODT Specifications	75
5.2.2. The Static Object Model of UOODT	76
5.2.3. Object Diagrams in UOODT	77
Chapter 6. OO Notation Workshop	80
6.1. GUI of OO Notation Workshop	80
6.1.1. Users of OO Notation Workshop	80
6.1.2. Task Analysis and Task Model	80
6.1.3. Visual Components of OO Notation Workshop	85
6.2. Designing the OO Notation	92
6.2.1. Selecting the Basic Drawing Tools	92
6.2.2. Creating Basic Shapes and Special Basic Shapes	94
6.2.3. Editing or Modifying Basic Shapes	94
6.2.4. Viewing and Editing OO Notation Shapes	95
6.3. OO Notation Examples	97
6.3.1. The Notation of the Booch Methodology	97
6.3.2. The Notation Examples of Other OO Methodologies	101
Chapter 7. Universal OO Diagramming Tool	104
7.1. Overview of Universal OO Diagramming Tool	104
7.1.1. GUI of UOODT	104
7.1.2. The Main Functions of UOODT	107

7.2. Using UOODT in Object-Oriented Development	110
7.2.1. Selecting an OO Methodology	110
7.2.2. Drawing and Editing OO Diagrams	110
7.3. Examples of Object-Oriented Development	112
7.4. Summary	120
Chapter 8. Conclusion	121
8.1. Overview of MIGOCE	121
8.2. The Characteristics of MIGOCE	122
8.3. Further work	124
References	125
Appendix Program Listing	139
Glossary	198