The Discriminating Object Model

Thesis by

Clara Murdaca

for the Degree of

Doctor of Philosophy

2010

Certificate of Authorship/Originality

I certify that the work in this thesis has not previously been submitted for a degree nor has it been submitted as part of requirements for a degree except as fully acknowledged within the text.

I also certify that the thesis has been written by me. Any help that I have received in my research work and the preparation of the thesis itself has been acknowledged. In addition, I certify that all information sources and literature used are indicated in the thesis.

Signed by

Production Note:
Signature removed prior to publication.

Clara Murdaca

Acknowledgements

I would like to thank my family, close friends and the university for the continued support they have provided me throughout this thesis project.

Abstract

This dissertation shows how to combine the expressive power of object-oriented programming with the efficiency of relational databases and query languages. Relational databases store and manipulate large amounts of simple data efficiently. The economies of scale they are able to achieve have yet to be significantly rivalled by alternate approaches to database storage. However relational databases are not good at storing complex objects and their associated relationships. Object-oriented programming languages are good at organising computation in a way that is both flexible and maintainable but support of persistent data is a problem. Research into combining the two has been ongoing over the past three decades. One such research direction has been the development of object-relational mapping tools. They link object-oriented programming and relational database storage, mapping the concepts from the programming language to the relational database. The majority of these mapping tools however are quite complex and require a knowledgeable user.

The pattern calculus is a new foundation for computation. It has been implemented in the programming language bondi. It supports the most popular programming styles within a single small system. In particular it supports generic pattern-matching functions and has a simple representation of temporary data. It is a powerful calculus which has not previously been tied to relational data. The Discriminating Object Model (DOM), developed in this thesis, is a data model that is suitable for both object-oriented programming and the organisation of data in relational databases. It aligns the notion of discriminators from relational databases with type variables from the programming language. Objects are mapped to records with additional fields. The DOM has been implemented in a database extension to the programming language bondi called dbbondi. Data manipulation is performed as a combination of object-oriented methods and database queries. They are executed in the environment in which their evaluation is optimal, thereby taking advantage of the efficiencies of both environments.

Contents

Certificate of Authorship/Originality

Acknowledgements i

Abstract ii

1 Introduction 1

2 Object-Orientation and Databases 7

2.1 Introduction 7

2.2 Object-Orientation 10

2.2.1 Key Features 10

2.2.2 Type Problems 13

2.2.3 Persistent Objects 15

2.3 Database Storage 16

2.3.1 Relational Databases 16

2.3.2 Object-Oriented Databases 17

2.4 Impedance Mismatch 19

2.5 Solutions 20

2.5.1 Object-Relational Databases 20

2.5.2 Object-Relational Mapping 20

2.5.2.1 NHibernate 21

2.5.2.2 SimpleORM 21

2.6 XML and Relational Databases 22

2.7 Conclusion 22

3 The Programming Language 24

3.1 Introduction 24

3.2 The Machinery 27

3.3 Programming Language Store 29

3.4 Functional Features 35
3.5 Pattern-Matching Functions 36
3.6 Four Kinds of Polymorphism 37
3.7 Imperative Features 38
3.8 Object-Oriented Features 39
3.8.1 Objects 41
3.8.2 Method Invocation 44
3.8.3 Type Problems of Object-Orientation 45
3.9 Programming Language Syntax 48
3.9.1 Shell Actions 48
3.9.2 Examples 49
3.10 Future Work 52
3.11 Conclusion 53

4 The Discriminating Object Model **55**
4.1 Introduction 55
4.2 The Object Model 56
4.3 The Data Model 59
4.4 Aligning the Data and Object Models 61
4.5 Conclusion 64

5 DOM Implementation **65**
5.1 Introduction 65
5.2 Open Database Connectivity 66
5.3 The Database Connection 66
5.4 The dbbondi Store 67
5.5 Mapping Classes to Tables 68
5.6 Typed Objects to Records 73
5.7 Records to Typed Objects 77
5.8 SQL Operations 80
5.8.1 where Clause 80
5.8.2 sqlselect 81
5.8.3 sqlupdate 83
5.9 dbbondi Functions 85
5.10 Database Logs 85
5.11 Conclusion 86

6 dbbondi 87
6.1 Introduction 87
6.2 Enhancements and Development 87
6.2.1 New Compiler Functions 89
6.2.2 Enhanced Compiler Functions 91
6.2.3 New Programming Language Syntax 91
6.3 Five Main Features 92
6.3.1 Class to Table Mappings 92
6.3.2 Object to Record Mappings 93
6.3.3 Record to Object Mappings 94
6.3.4 Bulk Data Loads 94
6.3.5 Functions as Embedded Queries 95
6.3.5.1 sqlselect 95
6.3.5.2 sqlupdate 96
6.3.5.3 sqlmap 97
6.4 Process Map 97
6.5 Conclusion 99
7 Test Suite 100
7.1 Introduction 100
7.2 Test Program Suite 102
7.2.1 Test Program 1 102
7.2.2 Test Program 2 104
7.2.3 Test Program 3 106
7.2.4 Test Program 4 107
7.2.5 Test Program 5 108
7.2.6 Test Program 6 109
7.2.7 Test Program 7 112
7.2.8 Test Program 8 114
7.2.9 Test Program 9 115
7.3 Conclusion 117
8 Conclusions 119
9 Future Work 123
9.1 Introduction 123
9.2 Built-in dbbondi Types 124

9.3 Reserved Words . . . 125
9.4 SQL functions . . . 126
9.5 Database Maintenance and Access Control . . . 126
9.6 Data Consistency and Data Integrity . . . 127
9.7 Complex Objects and Embedded Queries . . . 128
9.8 Multiple Inheritance . . . 131
9.9 Other Relationships . . . 131
9.10 Binding Mechanisms . . . 132
9.11 Conclusion . . . 132

Index **136**

Bibliography **136**

Publications **144**

Appendices **145**

A SQL **146**

A.1 SQL . . . 146
A.2 Data Definition . . . 147
A.2.1 Database Design . . . 150
A.2.2 Creating the Database . . . 154
A.3 Data Manipulation . . . 156
A.3.1 Inserting Data . . . 156
A.3.2 Conditional Where Clause . . . 157
A.3.3 Updating Data . . . 158
A.3.3.1 Mathematical Operations . . . 159
A.3.4 Deleting Data . . . 159
A.4 Data Retrieval . . . 160
A.4.1 Table Joins . . . 161

B dbbondi Syntax **163**

B.1 The dbbondi programming language syntax . . . 163
B.2 Shell Actions . . . 173
B.2.1 bondi Actions . . . 173
B.2.2 dbbondi Actions . . . 174
B.2.3 Compiler Mapping . . . 174
B.3 Shell Categories . . . 175

B.3.1 Term Declarations 175
B.3.2 Type and Datatype Declarations 176
B.3.3 Let Declarations 177
B.3.4 Class Declarations 177

C dbbondi Compiler Functions **178**
C.1 db_database Function 178
C.2 new table Function 178
C.3 db_term Function 179
C.4 db_filter Function 180
C.5 db_load Function 181
C.6 sqlmap Function 181
C.7 db_sqlop Function 182

D Process Flows **183**
D.1 Objects to Records 183
D.2 Extracting Single Records 184

E Properties and Clauses **185**
E.1 Create Table Properties 185
E.2 where Clause 185
E.3 set Clause 186

F Test Program Suite **187**
F.1 Test Program 1 187
F.1.1 dbbondi Code 187
F.1.2 bondi Code 189
F.1.3 Hibernate Code 191
F.1.4 MySQL Code 209
F.2 Test Program 2 211
F.2.1 dbbondi Code 211
F.2.2 Hibernate Code 212
F.2.3 MySQL Code 218
F.3 Test Program 3 219
F.3.1 dbbondi Code 219
F.3.2 Hibernate Code 220
F.3.3 MySQL Code 220
F.4 Test Program 4 222

F.4.1 dbbondi Code 222
F.4.2 Hibernate Code 223
F.4.3 MySQL Code 234
F.5 Test Program 5 236
F.5.1 dbbondi Code 236
F.5.2 Hibernate Code 240
F.5.3 MySQL Code 242
F.6 Test Program 6 246
F.6.1 dbbondi Code 246
F.6.2 Hibernate Code 247
F.6.3 MySQL Code 249
F.7 Test Program 7 249
F.7.1 dbbondi Code 249
F.7.2 Hibernate Code 250
F.7.3 MySQL Code 251
F.8 Test Program 8 252
F.8.1 dbbondi Code 252
F.8.2 Hibernate Code 254
F.8.3 MySQL Code 261
F.9 Test Program 9 263
F.9.1 dbbondi Code 263
F.9.2 Hibernate Code 264
F.9.3 MySQL Code 266

List of Tables

A.1 Sample Table 148
A.2 Example Entity Table 150
A.3 Example Staff Table 152
A.4 Example Student Table 152
A.5 Example PostGraduate Table 153
A.6 Example Research Table 153
A.7 Insert Method 1 156
A.8 Insert Method 2 157

List of Figures

2.1 Bruce's First Type Problem 13
2.2 Bruce's Second Type Problem 14
2.3 Bruce's Third Type Problem 14

3.1 Linear Terms 28
3.2 Common Terms 30
3.3 Reduction rules of pattern calculus 31
3.4 Pattern Matching and Case Matching 32
3.5 Evaluation 33
3.6 Stores 34
3.7 Simple Class Hierarchy 40
3.8 Bruce's First and Second Type Problem in bondi 46
3.9 Bruce's Third Type Problem in bondi 47

4.1 Class Hierarchy in dbbondi 57
4.2 Datatype Declarations in dbbondi 58
4.3 Function in dbbondi 59
4.4 Class Hierarchy to Tables 60

5.1 Object to Record 75
5.2 Term Variable into Query 78
5.3 sqlselect into SQL Query 82

A.1 Example University Database Schema 151