

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.

# Integration of Database Programming and Query Languages for Distributed Object Bases

Markus Kirchberg

A dissertation presented in partial fulfilment of the requirements for the degree of  
Doctor of Philosophy in Information Systems at Massey University

**Supervisor:**

Prof. Dr. Klaus-Dieter Schewe

**Co-Supervisor:**

Associate Prof. Dr. Ray Kemp

**Internal Examiner:**

Dr. Alexei Tretiakov

**Australasia Examiner:**

Prof. Dr. Xiaofang Zhou

**International Examiner:**

Prof. Dr. Bernhard Thalheim

**Date of Examination:**

July 13, 2007

# Abstract

Object-oriented programming is considered advantageous to other programming paradigms. It is believed to provide more flexibility, simplify development and maintenance, better model real-world situations, and is widely popular in large-scale software engineering. The idea behind object-oriented programming is that a program is composed of objects that act on each other. While programming languages mainly focus on processing, data storage and data sharing only play a minor role. Database management systems, on the contrary, have been built to support large bodies of application programs that share data. In today's database marketplace, relational and object-relational database systems are dominant. Object-oriented database systems were originally deemed to replace relational systems because of their better fit with object-oriented programming. However, high switching costs, the inclusion of object-oriented features in relational database systems, and the emergence of object-relational mappers have made relational systems successfully defend their dominance in the DB marketplace. Nevertheless, during the last few years, object-oriented database systems have established themselves as a complement to relational and object-relational systems. They have found their place as embeddable persistence solutions in devices, on clients, in packaged software, in real-time control systems etc.

In order to utilise the combined power of programming languages and SQL-like database query languages, research has focused on the embedded and integrated approaches. While embedded approaches are more popular, they suffer from the impedance mismatch, which refers to an inadequate or excessive ability of one system to accommodate input from another. This situation worsens when considering the numerous conceptual and technical difficulties that are often encountered when a relational database system is being used by a program that is written in an object-oriented language.

Research presented in this thesis addresses the integration of programming languages, database query languages, object-oriented programming and traditional database concepts. We investigate the stack-based approach to database programming and querying. The major objectives of our research are to:

- Develop a database architecture that is suitable to integrate database programming languages, object-oriented programming and more traditional database system features;
- Propose an intermediate-level database programming and querying language that combines the advantages of object-oriented programming and database querying languages. Before such a powerful language design can be achieved, a number of conceptual and technical challenges have to be investigated;
- Define a suitable run-time environment, which permits an efficient and effective evaluation of such an integrated language in a distributed database environment; and
- Provide proof of concept implementations.



# Acknowledgement

I would like to thank *Klaus-Dieter Schewe* for giving me the opportunity to pursue an academic career under his supervision. This thesis would not have been possible without his invaluable input, constant support and understanding.

My thanks also goes to *Ray Kemp* who kindly agreed to act as co-supervisor of this thesis.

Thanks to *Massey University* and the *Department of Information Systems* for their financial support.

Special thanks to *Sebastian Link*, *Sven Hartmann* and *Faizal Riaz-ud-Din* for their support, useful discussions and their friendship.

I am grateful to my wife *Gowri* for her love, encouragement and understanding. It is her support that gave me the strength to complete this work.

Finally, I would like to thank my parents *Sigrud* and *Dieter Kirchberg*, who have always been there for me and supported me in every way possible.



# Table of Contents

|   |           |
|---|-----------|
| <b>1 Introduction</b>   | <b>9</b>  |
| 1.1 Database Systems and Programming Languages . . . . .              | 10        |
| 1.1.1 The Impedance Mismatch . . . . .                                | 14        |
| Objects and RDBMSs. . . . .   | 15        |
| Objects and Object-Relational Databases. . . . .                      | 15        |
| Objects and Pure Object Databases. . . . .                            | 15        |
| 1.1.2 On the Integration of Programming and Query Languages . . . . . | 16        |
| 1.1.3 DBPL vs. Conventional Programming Languages . . . . .           | 18        |
| 1.2 Contributions . . . . .   | 19        |
| 1.3 Assumptions . . . . .   | 21        |
| 1.4 Outline . . . . .   | 22        |
| <b>2 A Review of DBPLs and Related Concepts</b>                       | <b>23</b> |
| 2.1 The Stack-Based Approach . . . . .                                | 23        |
| 2.2 Overview of DB Programming Environments . . . . .                 | 25        |
| 2.2.1 The Database Programming Language DBPL . . . . .                | 25        |
| 2.2.2 The O <sub>2</sub> Object Database System . . . . .             | 26        |
| 2.2.3 The Object Base Management System TIGUKAT . . . . .             | 26        |
| 2.2.4 The Parallel Database Programming Language FAD . . . . .        | 27        |
| 2.2.5 Additional Relevant Research Results on DBPLs . . . . .         | 28        |
| <b>3 An Integrated Approach to DB Programming and Querying</b>        | <b>31</b> |
| 3.1 A Distributed Object-Oriented Database System . . . . .           | 31        |
| 3.1.1 Architecture Overview . . . . .                                 | 32        |
| 3.1.2 Properties of the OODM . . . . .                                | 36        |
| Notes on the Choice of the Data Model. . . . .                        | 37        |
| 3.1.3 Fragmentation . . . . .   | 37        |
| 3.1.4 Linguistic Reflection . . . . .                                 | 38        |
| 3.1.5 A Simple University Application . . . . .                       | 39        |
| The Global OODM Database Schema. . . . .                              | 39        |
| Distributing the University Database. . . . .                         | 43        |
| 3.1.6 A Note on the Contribution of the Proposed ODBS . . . . .       | 50        |
| 3.2 Processing User Requests . . . . .                                | 50        |



---

|          |  |           |
|----------|--|-----------|
| <b>4</b> | <b>The Database Programming and Query Language iDBPQL</b>      | <b>57</b> |
| 4.1      | Language Overview . . . . .                                    | 57        |
| 4.2      | Basic Language Concepts . . . . .                              | 58        |
| 4.2.1    | Challenges . . . . .   | 59        |
| 4.2.2    | Conventions . . . . .  | 60        |
| 4.2.3    | Literals, Names and Other Rudimentary Language Elements . .    | 61        |
|          | Literals. . . . .  | 61        |
|          | Identifiers. . . . .   | 61        |
|          | Comments. . . . .  | 61        |
|          | Names. . . . .   | 61        |
| 4.2.4    | Types and Values . . . . .                                     | 62        |
|          | Primitive Types. . . . .                                       | 62        |
|          | The Record Type. . . . .                                       | 63        |
|          | Type Definitions. . . . .                                      | 67        |
|          | Type Parameters. . . . .                                       | 68        |
|          | Collection Types. . . . .                                      | 69        |
|          | NULLable Types. . . . .  | 71        |
|          | Value Initialisation and Default Values. . . . .               | 72        |
|          | Sub-typing and Type Conversion. . . . .                        | 73        |
|          | Variables. . . . .   | 74        |
|          | Default Type Operations. . . . .                               | 74        |
|          | Type Definitions for the University Application. . . . .       | 75        |
| 4.2.5    | Classes and Objects . . . . .                                  | 78        |
|          | Structure of a Class. . . . .                                  | 78        |
|          | Variables and Reference-Types. . . . .                         | 80        |
|          | Methods. . . . .   | 81        |
|          | (Multiple) Inheritance. . . . .                                | 82        |
|          | Variables, Types, Objects, and Classes. . . . .                | 88        |
|          | The FINAL and the ABSTRACT Modifiers. . . . .                  | 88        |
|          | The UNION-Type. . . . .  | 89        |
|          | Special Pre-Defined Classes. . . . .                           | 91        |
|          | Class-Collections. . . . .                                     | 93        |
|          | Constraints. . . . .   | 94        |
|          | Database Schemata and Classes. . . . .                         | 95        |
|          | Run-Time MetaData Catalogue Entries. . . . .                   | 96        |
|          | Persistence. . . . .   | 97        |
|          | Class Definitions for the University Application. . . . .      | 98        |
| 4.3      | Evaluation Plans . . . . .                                     | 100       |
| 4.3.1    | Challenges . . . . .   | 100       |
| 4.3.2    | Components of Evaluation Plans . . . . .                       | 101       |
| 4.3.3    | Evaluation Blocks and Their Properties . . . . .               | 102       |
| 4.3.4    | Statements . . . . .   | 104       |
|          | Assignment Statements. . . . .                                 | 104       |
|          | Control Flow Statements. . . . .                               | 105       |
|          | Type Operation Invocation, Method Calls and Object Creation. . | 107       |
| 4.3.5    | Expressions . . . . .  | 108       |

---

---

|  |            |
|--|------------|
| Simple Expressions. . . . .                                  | 108        |
| Parenthesised Expressions. . . . .                           | 108        |
| Assignment Expressions. . . . .                              | 108        |
| Type Operation Invocation, Method Calls and Object Creation. | 108        |
| Renaming Expressions. . . . .                                | 108        |
| Boolean Expressions. . . . .                                 | 109        |
| Query Expressions. . . . .                                   | 110        |
| Cast Expressions. . . . .                                    | 114        |
| The SUPER Expression and Keywords THIS and SUPER. . . . .    | 114        |
| 4.4 Simultaneous Processing . . . . .                        | 117        |
| 4.4.1 Implicit Inter-Transaction Concurrency . . . . .       | 120        |
| 4.4.2 Support for Explicit Concurrency . . . . .             | 120        |
| 4.4.3 Implications . . . . .                                 | 124        |
| 4.5 Examples . . . . .                                       | 125        |
| <b>5 On the Implementation of iDBPQL</b>                     | <b>131</b> |
| 5.1 MetaData Catalogues and Evaluation Plans . . . . .       | 131        |
| 5.1.1 Challenges . . . . .                                   | 132        |
| 5.1.2 MetaData Entries and Associated Information . . . . .  | 132        |
| Representing Type Information. . . . .                       | 134        |
| Representing Class Information. . . . .                      | 135        |
| Representing Descriptors. . . . .                            | 137        |
| Representing Other Attributes. . . . .                       | 137        |
| Inheritance Relations. . . . .                               | 138        |
| 5.1.3 The Representation of Objects and Values . . . . .     | 138        |
| 5.1.4 The Representation of Evaluation Plans . . . . .       | 141        |
| 5.1.5 Overview of Annotations . . . . .                      | 144        |
| 5.2 DBS Component Interface Definitions . . . . .            | 145        |
| 5.2.1 A Persistent Object Store . . . . .                    | 145        |
| An Object Model for POS. . . . .                             | 146        |
| Access Methods. . . . .                                      | 150        |
| The Architecture of POS. . . . .                             | 150        |
| The Service Interface. . . . .                               | 151        |
| POS as a Platform for iDBPQL. . . . .                        | 159        |
| 5.2.2 A Multi-Level Transaction Management System . . . . .  | 162        |
| The Service Interface. . . . .                               | 165        |
| 5.2.3 The Remote Communication Module . . . . .              | 166        |
| The Database Agent Communication Language (DBACL). . . . .   | 166        |
| 5.3 The Execution of Evaluation Plans . . . . .              | 167        |
| 5.3.1 Challenges . . . . .                                   | 168        |
| 5.3.2 The Run-Time Environment . . . . .                     | 169        |
| The REE Stack Area. . . . .                                  | 171        |
| The Environment Stack (ES). . . . .                          | 172        |
| The Result Stack (RS). . . . .                               | 174        |
| Operations on Stacks and Queues. . . . .                     | 175        |
| Initialising Result Queues. . . . .                          | 178        |

---

---

|   |            |
|---|------------|
| Binding Behaviour Names to Evaluation Plans. . . . .                  | 179        |
| 5.3.3 The SYSTEMcall, POScall and TMScall Primitives . . . . .        | 179        |
| The SYSTEMcall Primitive. . . . .                                     | 179        |
| The POScall Primitive. . . . .  | 180        |
| The TMScall Primitive. . . . .  | 180        |
| 5.3.4 Overview of the Evaluation Process . . . . .                    | 180        |
| Machine Instructions. . . . .   | 180        |
| Unnesting Objects and Values. . . . .                                 | 181        |
| The Main Evaluation Routine. . . . .                                  | 183        |
| 5.3.5 Maintaining the Run-Time Environment . . . . .                  | 186        |
| 5.3.6 Evaluating Individual Statements and Expressions . . . . .      | 187        |
| Simple Expressions: Literals and Names. . . . .                       | 187        |
| Object Identifiers and Stacks. . . . .                                | 187        |
| Expressions with Unary and Binary Operators. . . . .                  | 188        |
| Boolean Expressions. . . . .  | 190        |
| Renaming Expressions. . . . .   | 194        |
| Accessing Data Objects. . . . .                                       | 195        |
| Accessing Persistent Data: Beyond Direct Access. . . . .              | 199        |
| More Query Expressions. . . . .                                       | 201        |
| Controlling the Flow of Serial Data Processing. . . . .               | 209        |
| Invocation of Behaviours. . . . .                                     | 213        |
| Object Creation and Assignments. . . . .                              | 214        |
| Cast Expressions, SUPER and THIS. . . . .                             | 215        |
| 5.3.7 Evaluating Statements and Blocks of Statements . . . . .        | 216        |
| Blocks of Statements. . . . .   | 216        |
| Simple Transactions. . . . .  | 217        |
| Atomic Statement Blocks. . . . .                                      | 221        |
| 5.3.8 Processing Evaluation Plans . . . . .                           | 222        |
| 5.3.9 Simultaneous Evaluation of Statements and Expressions . . . . . | 222        |
| 5.3.10 Distributed Processing of Evaluation Plans . . . . .           | 223        |
| Migrating Objects. . . . .  | 223        |
| Processing Evaluation Plans on a Remote Node. . . . .                 | 224        |
| Distributed Transactions. . . . .                                     | 224        |
| 5.4 Optimisation of the Evaluation Process . . . . .                  | 226        |
| <b>6 Proof of Concept</b>   | <b>229</b> |
| 6.1 Simultaneous Stack-Based Abstract Machines . . . . .              | 229        |
| 6.2 The iDBPQL Prototype System . . . . .                             | 231        |
| <b>7 Summary</b>  | <b>239</b> |
| 7.1 Main Contribution . . . . .                                       | 239        |
| 7.2 Future Plans and Open Problems . . . . .                          | 241        |
| <b>Bibliography</b>   | <b>245</b> |

---

---

|          |  |            |
|----------|--|------------|
| <b>A</b> | <b>The Syntax of iDBPQL</b>                                | <b>255</b> |
| A.1      | The Lexical Syntax of iDBPQL . . . . .                     | 255        |
| A.2      | The Syntax of MetaData Catalogue Entries . . . . .         | 256        |
| A.2.1    | Syntax of DBS MetaData Units . . . . .                     | 256        |
| A.2.2    | Syntax of Run-Time MetaData Units . . . . .                | 256        |
| A.2.3    | Common Syntax of Type-System-Related Definitions . . . . . | 256        |
| A.2.4    | Common Syntax of iDBPQL Types . . . . .                    | 257        |
| A.3      | The Remaining Syntax of iDBPQL . . . . .                   | 258        |
| A.3.1    | Syntax of Evaluation Units . . . . .                       | 258        |
| A.3.2    | Syntax of Evaluation Blocks . . . . .                      | 258        |
| A.3.3    | Syntax of Statements . . . . .                             | 258        |
| A.3.4    | Syntax of Expressions . . . . .                            | 259        |
| A.3.5    | Identifiers, Labels, Values and More . . . . .             | 259        |
| <b>B</b> | <b>The <i>Parentage</i> Database Example</b>               | <b>261</b> |

# List of Figures

|  |     |
|--|-----|
| 2.1 Relationship Between Data Models and SBQL's Abstract Storage Model ([131, Figure 1]). . . . .  | 23  |
| 3.1 Architecture of the Distributed Object-Oriented Database System. . . .   | 33  |
| 3.2 HERM Diagram of the University Database. . . . .   | 40  |
| 3.3 Relationship between User Requests, Data Models and iDBPQL. . . . .  | 51  |
| 4.1 The Type System of iDBPQL. . . . .   | 71  |
| 4.2 Union Types, Sub-typing and Inheritance ([53, Figure 1]). . . . .  | 91  |
| 5.1 Sample Evaluation Graph for the <code>verifyEnrolment</code> Method. . . . .   | 143 |
| 5.2 Overview of Embedded References Between Instances of Classes of the <code>University<sub>CC</sub></code> Schema Fragment as Considered in Example 5.3. . . . . | 149 |
| 5.3 Architecture of the Persistent Object Store. . . . .   | 151 |
| 5.4 Architecture of the Transaction Management System. . . . .   | 163 |
| 5.5 Local Heap with Embedded POS and RCM Shared Memory Areas. . . .  | 170 |
| 5.6 Logical View of the Environment and Result Stacks. . . . .   | 173 |
| 5.7 Logical View of the Effects of <code>evalMultiES ( )</code> Evaluation Procedures on ES. . . . .   | 184 |
| 5.8 Overview of the Evaluation Process of the <code>verifyEnrolment</code> Method from Figure 5.1. . . . .   | 186 |
| 6.1 Overview of a Stack-Based Abstract Machine with Distribution Capabilities. . . . .   | 230 |
| 6.2 Overview of a Local Stack-Based Abstract Machine. . . . .  | 231 |
| 6.3 Usage Diagram for the iDBPQL Prototype. . . . .  | 234 |
| 6.4 Evaluation of the <i>HelloWorld</i> Example. . . . .   | 235 |
| 6.5 iDBPQL and Internal Representations of the <i>Parentage</i> Database. . . .  | 236 |
| 6.6 Executing Requests on the <i>Parentage</i> Database. . . . .   | 237 |

# List of Tables

|   |     |
|---|-----|
| 4.1 The Primitive Types of iDBPQL. . . . .  | 62  |
| 4.2 Primitive Types and Their Supported Operations. . . . .                                   | 64  |
| 5.1 Modifier Flags and Their Interpretation. . . . .  | 135 |
| 5.2 Internal Object Structure of Instances of Classes Presented in Example<br>4.12. . . . .   | 140 |
| 5.3 (Binary) Path-Operators that are Supported by POS. . . . .                                | 152 |
| 5.4 Type Information Mapping to Machine Instructions. . . . .                                 | 181 |
| 5.5 Overview of Machine Instructions for Binary Expressions without Side<br>Effects. . . . .  | 189 |
| 5.6 Overview of Machine Instructions for Assignment Expressions with Side<br>Effects. . . . . | 190 |
| 5.7 Overview of Additional Boolean Expressions and their Machine Instruc-<br>tions. . . . .   | 191 |