

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.

**Associative Access  
in Persistent Object Stores**

**A thesis presented in partial fulfilment  
of the requirements for the degree of**

**Master of Information Sciences  
in Information Systems**

**at Massey University  
Palmerston North, New Zealand**

**Weena Nusdin  
2004**



## ABSTRACT

*The overall aim of the thesis is to study associative access in a Persistent Object Store (POS) providing necessary object storage and retrieval capabilities to an Object Oriented Database System (OODBS) (Delis, Kanitkar & Kollios, 1998 cited in Kirchberg & Tretiakov, 2002).*

*Associative access in an OODBS often includes navigational access to referenced or referencing objects of the object being accessed (Kim, Kim, & Dale, 1989). The thesis reviews several existing approaches proposed to support associative and navigational access in an OODBS. It was found that the existing approaches proposed for associative access could not perform well when queries involve multiple paths or inheritance hierarchies.*

*The thesis studies how associative access can be supported in a POS regardless of paths or inheritance hierarchies involved with a query. The thesis proposes extensions to a model of a POS such that approaches that are proposed for navigational access can be used to support associative access in the extended POS. The extensions include (1) approaches to cluster storage objects in a POS on their storage classes or values of attributes, and (2) approaches to distinguish references between storage objects in a POS based on criteria such as reference types – inheritance and association, storage classes of referenced storage objects or referencing storage objects, and reference names.*

*The thesis implements Matrix-Index Coding (MIC) approach with the extended POS by several coding techniques. The implementation demonstrates that (1) a model of a POS extended by proposed extensions is capable of supporting associative access in an OODBS and (2) the MIC implemented with the extended POS can support a query that requires associative access in an OODBS and involves multiple paths or inheritance hierarchies. The implementation also provides proof of the concepts suggested by Kirchberg & Tretiakov (2002) that (1) the MIC can be made independent from a coding technique, and (2) data compression techniques should be considered as appropriate alternatives to implement the MIC because they could reduce the storage size required.*



## ACKNOWLEDGEMENTS

I would like to thank Alexei Tretiakov, my supervisor, for his patience, guidance and suggestions during this thesis. I am also thankful to Markus Kirchberg for his guidance and suggestions through the early stages of work. Thanks to Klaus-Dieter Schewe for his kindness in allowing me to attend his paper, 157.794 Object Oriented Databases. Thanks to Roland Kaschek for his comments during the thesis.

Thanks to staff of International Student Support at Massey University, Palmerston North for their assistance. Special thanks go to Susan Flynn for her assistance and the arrangement of financial support during my thesis.

While I was undertaking the thesis, a number of friends I met them in New Zealand have given me lots of helps and encouragement, which has helped me pass through difficult times. Special thanks go to the following friends.

- Marie Hau for being my good, helpful friend and flatmate,
- Yuen Xie for our discussions about data access approaches used in relational database systems.
- Jayson Speer for his suggestions and encouragement during the beginning of learning a C++ programming language,
- Lin Shi for his suggestions during the thesis,
- Rebecca Freeman for her assistance in proof-reading my draft thesis,
- Nanthaporn Chitchai for always coming to visit and listen to my troubles,
- Weerawate Utto for helping me out of depressing times and especially for smiling tomatoes from his experiment,
- Angkana Noisuwan, Duljira Sukboonyasatit and Chanapha Sawatdeenaruenat for the laughs, encouragement and assistance they have given,

- Somsaowanuch Chamusri and Kuephan Klankaradi for their kindness for taking turns to accompany me and giving me a ride from my office to my flat during the night for the last three months of my thesis, and
- Thiengtham's family, Kanittha Watanakeeree, Wanwadee Wongmongkol, Piyarat Piyaket and Duangrat Thongphak for good times, assistance and encouragement.

Thanks to my friends from Kasetsart University in Thailand for their encouragement and understanding. Special thanks go to Nuanjan Suntornkiti, without whom this year would have been much harder and longer.

I also would like to thank my colleagues, especially Thra Boondechanun, at the Bureau of Flight Safety Standards, Department of Civil Aviation of Thailand for their invaluable assistance in taking care of my jobs and responsibilities during my study leave. Special thanks go to Jutharat Nakhsewi for the kindness, patience, and wonderful support he has always given me throughout my study.

I wish to express my deep gratitude to my parents for their endless patience and love, and wonderful support they have given me. Without them, I would never have completed this work.

Finally, I would like to express my sincere thanks to the New Zealand Government for their financial support during my stay in New Zealand, and to the Thai Government for granting my study leave.

Weena Nudin

February 2004

# TABLE OF CONTENTS

<b>ABSTRACT</b>	<b>i</b>
<b>ACKNOWLEDGEMENTS</b>	<b>iii</b>
<b>TABLE OF CONTENTS</b>	<b>v</b>
<b>LIST OF FIGURES</b>	<b>ix</b>
<b>LIST OF TABLES</b>	<b>xiii</b>
<b>1. INTRODUCTION</b>	<b>1</b>
1.1 Database Systems	1
1.2 The Functionality of a DBMS	3
1.3 An Object Oriented Database System (OODBS)	5
1.4 Providing Data Access in an OODBS	7
1.5 A Persistent Object Store (POS)	10
1.6 Thesis Motivations	11
1.7 Thesis Objectives	12
1.8 Structure of the Thesis	14
<b>2. RELATIONAL AND OBJECT ORIENTED DATA MODELS</b>	<b>17</b>
2.1 The Concept of a RDM	17
2.2 The concept of an OODM	21
2.3 Differences between RDM and OODM concepts	27
2.4 Summary	30
<b>3. OBJECT ORIENTED DATABASE SYSTEMS (OODBSs)</b>	<b>33</b>
3.1 ORION	33
3.2 O <sub>2</sub>	35



3.3	Link Objects in a Query Integrated System (LOQIS)	36
3.4	A Multi-Level Architecture for Distributed Object Bases	39
3.5	Summary	41
<b>4.</b>	<b>EXISTING DATA ACCESS APPROACHES IN AN OODBS</b>	<b>43</b>
4.1	Queries Requiring Associative Access in an OODBS	44
4.2	Approaches Supporting Queries Requiring Associative Access in an OODBS	50
4.2.1	Approaches Supporting Queries that require only Associative Access	51
4.2.2	Approaches Supporting Queries Made against Logical Objects of Classes in an Inheritance Hierarchy	51
4.2.3	Approaches Supporting Queries Made against Logical Objects of Classes in a Class-Attribute Hierarchy	65
4.2.4	Approaches Supporting Queries Made against Logical Objects of Classes in Inheritance or Class-Attribute Hierarchies	71
4.3	Approaches Supporting Navigational Access in an OODBS	73
4.3.1	Modifications of Join Indices	73
4.3.2	Object Skeletons	78
4.3.3	Reference Pointer Approaches	80
4.3.4	In-memory Calculation Approaches	82
4.4	Summary	90
<b>5.</b>	<b>DATA COMPRESSION TECHNIQUES</b>	<b>93</b>
5.1	Fundamentals of Data Compression	93
5.2	Common Measures of Data Compression	95
5.2.1	Redundancy	95
5.2.2	Average Message Length	95
5.2.3	Compression Ratio	96
5.3	Data Compression Techniques	96
5.3.1	Semantic-Dependent Data Compression Techniques	97
5.3.2	General-Purpose Data Compression Techniques	97
5.4	A Start/Stop Data Compression Technique	101
5.5	Summary	104

<b>6. THE EXTENDED POS</b>	<b>105</b>
6.1 Extending a Model of a POS	106
6.2 Clustering Storage Objects	108
6.2.1 A Storage Class	108
6.2.2 Mapping Logical Objects to Storage Objects	110
6.2.3 Approaches to Cluster Storage Objects on Storage Classes	114
6.2.4 Approaches to Cluster Storage Objects on Values of Attributes	117
6.3 Distinguishing References in a POS	120
6.3.1 Circumstances to Distinguish References in a POS	121
6.3.2 Approaches to Distinguish References in a POS	124
6.4 Summary	126
<b>7. PERFORMANCE OF MIC WITH THE EXTENDED POS</b>	<b>131</b>
7.1 The Cost Performance of MIC to Support Associative Access in the Extended POS	133
7.2 The Additional Storage Size of the MIC to Support Associative Access in the Extended POS	138
7.3 Summary	148
<b>8. THE IMPLEMENTATION OF MIC WITH THE EXTENDED POS</b>	<b>149</b>
8.1 A Query Used in the Implementation	150
8.2 Implementation Details	152
8.2.1 Mapping Logical Objects to Storage Objects in the Implementation	153
8.2.2 The Extensions of a Model of a POS in the Implementation	157
8.2.3 Access Steps in the Implementation	158
8.3 Implementation Results	159
8.4 Summary	163
<b>9. CONCLUSION</b>	<b>165</b>
9.1 Conclusion	165
9.1.1 A Review of Existing Data Access Approaches in an OODBS	167
9.1.2 The Extended POS	169
9.1.3 The Implementation of the MIC	171

9.2 Future Research	172
<b>REFERENCES</b>	<b>175</b>
<b>APPENDIX A: AN EXAMPLE RELATIONAL DATA MODEL (RDM)</b>	<b>181</b>
<b>APPENDIX B: AN EXAMPLE OBJECT ORIENTED DATA MODEL (OODM)</b>	<b>187</b>
<b>APPENDIX C: C++ SOURCE CODES FOR THE IMPLEMENTATION OF MIC</b>	<b>193</b>
<b>APPENDIX D: INPUT DATA</b>	<b>253</b>
<b>APPENDIX E: IMPLEMENTED MIC</b>	<b>257</b>
<b>APPENDIX F: RESULTS OF USING MIC TO SUPPORT A QUERY Q</b>	<b>277</b>

## LIST OF FIGURES

Figure 1.1	Three-level architecture of database systems (based on Abiteboul et al., 1995, p.5)	2
Figure 3.1	ORION Architecture (Kim, Ballou et al., 1989, p.253)	34
Figure 3.2	The O <sub>2</sub> Object Manager (Banchilhon et al., 1992, p. 356)	35
Figure 3.3	An Approach to Map Logical Objects to Atoms in LOQIS Proposed by Subieta (1994a; 1994b)	37
Figure 3.4	An Approach to Map Logical Objects to Atoms in LOQIS Proposed by Jodlowski (2002)	38
Figure 3.5	Architecture for Distributed Object Bases (Kirchberg et al., 2003, p.2)	40
Figure 4.1	An Example Inheritance Hierarchy	52
Figure 4.2	Structure of a B+ tree (Ramakrishnan & Gehrke, 2003, p.345)	53
Figure 4.3	Structure of a Non-Leaf Node of a B+ Tree (Kirchberg, 2003, p.32)	54
Figure 4.4	Structure of a Leaf Node of a B+ Tree (Kirchberg, 2003, p.33)	54
Figure 4.5	An Example B+ Tree (Ooi & Tan, 2001, p.14)	55
Figure 4.6	A Non-Leaf Node of a SC Index (Kim, Kim et al., 1989, p.377)	57
Figure 4.7	A Leaf Node of a SC Index (Kim, Kim et al., 1989, p.377)	57
Figure 4.8	A Leaf Node of a CH Index (Kim, Kim et al., 1989, p.377)	59
Figure 4.9	H-trees (Low et al., 1992, p.136)	61
Figure 4.10	A hcC-tree (Sreenath & Seshadri, 1994, p.206)	62
Figure 4.11	An Inheritance Hierarchy (Ramaswamy & Kanellakis, 1995, p.144)	63
Figure 4.12	An Example CD Index (Ramaswamy & Kanellakis, 1995, p.144)	64
Figure 4.13	A Class-Attribute Hierarchy	66
Figure 4.14	Example Paths	66
Figure 4.15	An Example of a Join Index (JI)	67
Figure 4.16	Structure of a Leaf Node of a Path Index	70
Figure 4.17	An Example Access Support Relation	70

Figure 4.18	Structure of a NIX (Hua & Tripathy, 1994, p.510)	72
Figure 4.19	a)-b) Schema Paths	75
Figure 4.20	A Complete Join Index Hierarchy	76
Figure 4.21	a) – b) Partial Join Index Hierarchies	77
Figure 4.22	A Triple-Node Hierarchy Supporting Tri (1, 5, T) and Tri ( $\perp$ , 1, 5) (Luk & Fu, 1998, p.8)	78
Figure 4.23	The Framework of Object Skeletons (Hua & Tripathy, 1994, p. 511)	79
Figure 4.24	Ring Structure (Subieta, 1994b, p.12)	81
Figure 4.25	Spider Structure (Subieta, 1994b, p.12)	81
Figure 4.26	Organisation of Backward Pointers (Subieta, 1994b, p.12)	82
Figure 4.27	An Example Reference Graph	84
Figure 4.29	Example Expander and Linker Tables	87
Figure 4.30	A Reference Matrix	88
Figure 4.31	SICF Codes Calculated by MIC	89
Figure 6.1	An Example of Mapping Logical Objects to Storage Objects	112
Figure 6.2	Storage Class Objects (SCOs)	115
Figure 6.3	Storage Class Flags (SCFs)	116
Figure 6.4	a)-c) Value Objects (VOs)	118
Figure 6.5	a)-c) Value Flags (VFs)	119
Figure 6.6	a)-c) Example Circumstances to Distinguish References	123
Figure 6.7	Storing References and Storage Objects Separately	124
Figure 6.8	Reference Flags (RFs)	125
Figure 7.1	The Increasing Number of Sharing Storage Objects in the Extended POS	135
Figure 7.2	The Decreasing Number of Sharing or Branching Storage Objects in the Extended POS	136
Figure 7.3	Changes in a Model of a POS extended by Clustering Storage Objects by SCO or VO Approaches	141
Figure 7.4	Additional Storage in the Extended POS caused by Flag Approaches	143
Figure 7.5	Additional Storage in the Extended POS caused by Storing Several Sets of References	145

Figure 8.1	Paths involved with a Query Q	151
Figure 8.2	Inheritance Hierarchies involved with a Query Q	152
Figure 8.3	A Comparison of the Number of Bits required for the Implemented MIC with the Extended POS for each Storage Object	162
Figure 8.4	A Comparison of the Number of Bits required for the Implemented MIC with the Extended POS for each Storage Object (Sorted by the Number of Bits required When the MIC is Implemented with No Coding Technique)	163
Figure C.1	Modules	203
Figure F.1	Results of Using MIC Implemented with the Extended POS to Support a Query Q	277



## LIST OF TABLES

Table 2.1	Differences between RDM and OODB concepts	27
Table 4.1	Differences between B, B+ and B* trees	55
Table 5.1	Algorithm of a Start/Stop data compression technique ( $\langle m_i \rangle$ representing a string of $m_i$ bits) (Pigeon, 2001)	102
Table 5.2	Example codewords encoded by a Start/Sop coding technique when $\{m_1, m_2, \dots, m_k\} = \{0, 3, 2, 0\}$ (Pigeon, 2001)	103
Table 7.1	Extensions proposed to a model of a POS	131
Table 7.2	Additional storage required for the MIC implemented with a POS extended by SCOs and SCFs approaches	147
Table 7.3	Additional storage required for the MIC implemented with a POS extended by VO and VF approaches	147
Table 7.4	Additional storage required for the MIC implemented with a POS extended by distinguishing references in a POS	148
Table D.1	Storage Objects Identifiers (OIDs) and (Value/Storage Class) flags	254
Table D.2	References and (reference) flags	255
Table E.1	MIC implemented with no coding technique	257
Table E.2	MIC implemented with a SICF coding technique	262
Table E.3	MIC implemented with a Start/Stop coding technique	267