ACS-3402

INVC..nms...18/2/02

PERSISTENCE FRAMEWORK FOR MULTIPLE LEGACY DATABASES OF PATIENT INFORMATION SYSTEMS

Khor Chin Heong

A project report submitted to the Faculty of Computer Science and Information Technology, University of Malaya, in partial fulfilment of the requirements for the degree of Master of Computer Science.

2 JUNE 2000

Perpustakaan Universiti Malaya
A510573623

Abstract

The use of object-oriented technologies has led to a gap between object-oriented applications and legacy relational databases. A persistence framework can fill this gap by providing services to convert and store objects in relational databases and other forms of storage. A persistent object lasts beyond the execution of the application and can be retrieved at a later time. This project involves the development of an object persistence framework in the Java language to work with different storage mechanisms, while concentrating on transparency and reusability aspects. The persistence framework is made up of reusable and extendable sets of classes that provide services for persistence objects such as for translation of objects to records to be saved in a certain type of patient relational database and translation of records to objects when retrieving from the database. It supports storage in relational databases, flat files, e-mail servers, and the ObjectStore object database. The framework was found to be successful in providing basic persistence services while maintaining transparency.

Acknowledgements

I wish to thank my project supervisor, Dr. Lee Sai Peck, along with Mr. Thin Siew Khim and Mr. Liu Hong Song for their useful input and feedback during the development of this project.

Table of Contents

1. Introduction 1

1.1 Background 1

1.2 Motivation 2

1.3 Problem Statement 2

1.4 Objectives 3

1.5 Results 3

1.6 Chapter Organization 3

2. Literature Review 5

2.1 The Relational Model 5

2.1.1 Limitations Of Relational Databases 6

2.2 The Object Model 7

2.2.1 Object Oriented Databases 7

2.2.2 Objects And Relational Databases 8

2.3 Mapping Inheritance 9

2.4 Mapping Relationships 12

2.5 Frameworks 14

2.6 Persistence Frameworks 16

2.6.1 Design Of A Persistence Framework 17

2.6.2 Existing Persistence Framework Products 21

2.7 Application In The Healthcare Industry 22

3. Architecture 25

4. Analysis 28

4.1 Use Case Model 28

4.2 Analysis Model 30

4.2.1 Object Mapper 30

4.2.2 Broker 30

4.2.3 Relational Data 40

4.2.4 Persistence Mechanism 41

5. Design 42

5.1 OIDObject And DataObject 43

5.2 Relational Data: TableObj, Record, And Column 43

5.3 Persistence Mechanism: FlatFile, Email, JDBC, And ObjectStore 45

5.4 Broker: Persistencebroker And Supporting Classes 48

5.4.1 Saving 51

5.4.2 Restoring 51

5.4.3 Deleting 53

5.4.4 Querying And Updating 54

5.4.5 Persistence Mechanisms And Transactions 55

5.5 Framework Facade 56

6. Implementation 58

7. Results And Evaluation 59
7.1 Test Cases And Results 59
7.2 Evaluation 60

8. Conclusion 62
8.1 Strengths Of The Persistence Framework 62
8.2 Limitations Of The Persistence Framework 63
8.3 Comparison With Other Persistence Frameworks 63
8.4 Future Work 64

9. References 66

A. Results 68
A.1 Class JDBCMechanism 68
A.2 Class FlatFileMechanism 71
A.3 Class EmailMechanism 80
A.4 Class ObjectStoreMechanism 80

B. User Guide 81
B.1 Creating The Broker 81
B.2 Connecting To A Data Source 81
B.3 Saving Objects 83
B.4 Restoring Objects 83
B.5 Deleting Objects 83
B.6 Updating Objects 84
B.7 Queries 84
B.8 Transactions 85
B.9 Disconnecting 87
B.10 Example Program 87

List of Figures

Figure 2.1 Class Hierarchy to be Mapped 10

Figure 2.2 Three Different Class Inheritance Mappings 10

Figure 2.3 Mapping Relationships 12

Figure 2.4 Design of an Object Layer Persistence Framework 18

Figure 2.5 Design of an Object Layer+Broker Persistence Framework 20

Figure 3.1 Layered Architecture (Adapted from [6]) 25

Figure 4.1 Use Case Model of the Persistence Framework 29

Figure 4.2 Analysis Model of the Persistence Framework 31

Figure 4.3 Class Hierarchy to be Mapped 32

Figure 4.4 Mapped Class Hierarchy 32

Figure 4.5 Sequence Diagram for the Connect/Disconnect Use Cases 33

Figure 4.6 Object Graph to be Saved. B is The Persistent Root Object and Shaded Boxes Represent All Objects Reachable by it 34

Figure 4.7 Sequence Diagram for the Save Use Case 35

Figure 4.8 Sequence Diagram for the Restore Use Case 36

Figure 4.9 Sequence Diagram for the Delete Use Case 37

Figure 4.10 Sequence Diagram for the Query Use Case 39

Figure 4.11 Sequence Diagram for the Update Use Case 40

Figure 5.1 Design Model of the Persistence Framework 42

Figure 5.2 Base Objects Class Diagram 43

Figure 5.3 Relational Data Class Diagram 44

Figure 5.4 Persistence Mechanism Class Diagram 45

Figure 5.5 Persistence Broker Class Diagram 49

Figure 5.6 Object Mapping Process 50

Figure 5.7 Activity Diagram for Persistencebroker::save(),deepSave() 52

Figure 5.8 Activity Diagram for Persistencebroker::restore(),deepRestore() 53

Figure 5.9 Activity Diagram for Persistencebroker::delete(),deepDelete() 54

Figure 5.10 Reuse Example of the Persistence Framework 57

List of Tables

Table 2.1	Comparison Between Different Mappings	11
Table 2.2	Survey of Solutions to the Legacy System Problem	23
Table 3.1	Deployment Configurations for Persistence (Adapted from [6])	27
Table 7.1	Test Cases for the Persistence Framework	60