

# A Comparative Analysis of Object-relational mappings for .NET Framework

Kuldeep A. Hule<sup>1</sup>, Zakir M. Shaikh<sup>2</sup>

<sup>1</sup>M.E. (CSE) Department of Computer Science & Engineering  
Nagesh Karajagi Orchid College of Engg. & Technology  
Solapur University, Solapur, Maharashtra, India

<sup>2</sup>Assistant Professor in Department of Computer Science & Engineering  
Nagesh Karajagi Orchid College of Engg. & Technology  
Solapur University, Solapur, Maharashtra, India

## ABSTRACT

*Object relational mapping from relational databases has become topic to be dealt with in the recent decade. Object Relational Mapping is a layer which is responsible to bridge the gap between object-oriented programs and relational database. Lot of new Object Relational Mappers have been developed which includes forward engineering as well as reverse engineering approach for developing the object code. In this paper we are proposing the method of reverse engineering for developing object code file. Also we have compared the existing object relational mapping tools with our one.*

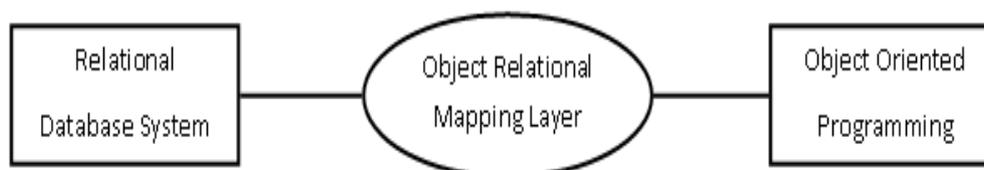
**Keywords**—Object relational mapping, relational database, impedance mismatch, entity, Business Object

## 1. INTRODUCTION

Information Technology (IT) systems need persistent storage (unless, of course, the systems are really trivial). This persistent storage usually comes in the form of a relational database, such as Oracle, Microsoft Structured Query LANGAUGE (SQL) Server, IBM DB, and likewise. There are several alternatives to RDBMSs; for instance, object-oriented databases (OODBs) e.g. db4o, persistent object stores (Prevaler, XML databases etc.). However, at least until now, all these alternatives haven't really caught and the fact is that Relational Database Management Systems (RDBMSs) remain pretty much ubiquitous. Now a days, Information Technology (IT) companies are using the object oriented programming languages like C#, Java, VB, etc. Relational databases [9] and object-oriented programming languages are based upon distinct paradigms, with technical conceptual and cultural incompatibilities. The set of those incompatibilities is commonly referred to as the object-relational impedance mismatch problem [2]. Object-relational mapping (ORM) frameworks address the impedance problem of software implementation level [3], providing the developer with ways to declare how each technical incompatibility should be treated [5]. ORM tools brings to the same level relational resources, such as data querying and object-oriented resources, such as inheritance, polymorphism, enabling to explore the synergy between those constructs [4]. Object relational mapping (ORM) in computer software is a programming technique for converting data between incompatible type systems in databases and object-oriented programming languages. This creates, in effect, “virtual object databases” which can be used from within the programming language [1] In other words, Object Relational Mapping hides the underlying tables, SQL etc., inside the RDBMS. Due to the popularity of relational data storage with its broad installation base and the use of object-oriented concepts for software development, application objects inevitably need to be stored into (object) relational databases [6].

### 1.1 Object Relational Mapping

Object Relational Mapping is a layer that is responsible to bridge the gap between object-oriented programs and relational database. In fact which provides the objects with persistent services which means that ability to read data from and write data to, and delete data from data sources. Figure1 illustrates the concept of object relational mapping:

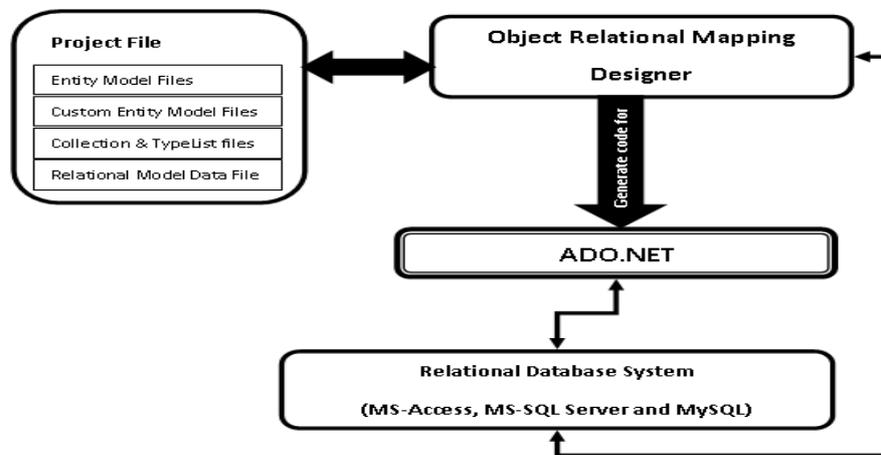


**Figure 1:** Object Relational Mapping System

In other words, object relational mapping is a technology that is employed to bridge the impedance mismatch between object- oriented programs and relational database. It tries to eliminate the duplication of data and maintenance cost and susceptibility to error(s) associated with it [8].

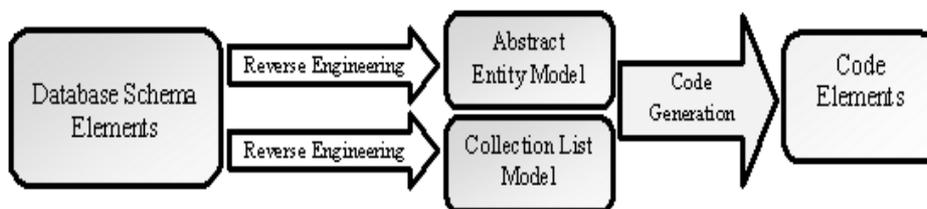
**2. MATERIALS AND SYSTEM ARCHITECTURE**

On the development station, we used a system with a Pentium Dual-core 2.70Ghz Processor, 2GB RAM, 32-bit System Architecture, and windows 8 Operating System. Object Relational Mapping has been implemented entirely in C# using Microsoft .NET Framework 4.0. It provides by default a wealth of frameworks, API's (Application Programming Interface), standards implementations, and libraries that foster reusability and reduces development time. The objective of any objet relational mapping framework/API is to relieve the programmers from the hard work of manual SQL statement writing and manipulation, & allow them to focus on manipulating data structure from the problem domain [1]. In this model we have aimed at providing optimum scalability and reusability. Object-Relational Mapping, or O/R Mapping is the general term for the concept of creating mappings between tables and their fields, and Object Oriented classes and their fields to be able to represent at runtime an Entity Definition Instance in the form of a table row in an Object Oriented program via a class instance and vice versa. The management logic is necessary to read an Entity Definition Instance (which is the data) from the database into an instance of an instance of a class and storing it back into the database, together with the entity definitions, is called an O/R Mapper framework. In the Object Relational mapping designer, these are called Target Frameworks[7]. ORM Designer will generate the code for ADO.NET Framework. Via this generated code library a developer can manipulate data in the database, using classes and their methods. We also had provided the collection, customized views of the database data. Figure 2 shows the system architecture:



**Figure 2:** System Architecture

The overall architecture for the system utilizes structures that represent tables, databases and connections to these databases. Database- First or Database-driven approach is used for this architecture which works as reverse engineered an Abstract Entity Model from a set of database schema elements [7]. The idea behind Database-First is that the Database Schema Elements, e.g. tables, views, foreign key constraints and like, were once created using an abstract entity model and to look at the database schema elements, one can reverse engineer them to an Abstract Entity Model which is roughly the same, or represents the same, as the database schema elements are projection results of the Abstract Entity Model[7]. Figure 3 shows the schematic view of Database First model for the above architecture



**Figure 3:** Database-First Approach

**3. COMPARATIVE ANALYSIS OF ORM WITH EXISTING ORM TOOLS**

There is plenty of work done on Object Relational Mapping (ORM). There may be various platforms and frameworks available for Object Relational Mapping. There are many Object Relational Mapping (ORM) tools for .NET Framework and we have compared some of them which will be considered when we develop our Object Relational Mapping tool. Following table shows the comparative analysis of our system with existing ORM tools with our system (Highlighted in gray color):

**Table 1:** Comparative Analysis of Object Relational Mapping with Existing ORM tools

Parameter	LINQ-to-SQL (formerly "DLINQ")	NHibernate	LLBLGen Pro	Object Relational Mapping Tool*
Company	Microsoft, USA	Open Source	Solution Design	-
Website	<a href="http://msdn.microsoft.com/en-in/library/bb397926.aspx">http://msdn.microsoft.com/en-in/library/bb397926.aspx</a>	<a href="http://www.nhforge.org/">http://www.nhforge.org/</a>	<a href="http://www.llblgen.com/">http://www.llblgen.com/</a>	-
Price	Free	Free	Standard-EUR 299 , Standard Plus-EUR 398.	-
First RTM Release	November 2007	October 2005	January 2003	June 2014
Current Version	.NET 4.5.1 (June 2013)	3.3.3.4001(August 2013)	4.2(July 2014)	1.0.0(June 2014)
<b>Requirements</b>				
Supported Database Management Systems (DBMS)	SQL Server 2000,2005 and 2008 plus SQLCE	MSSQL Server, DB2, Ingress Postgres, MySQL, Oracle, Sybase, Firebird, SQLite.	MS Access, MS- SQL Server, Oracle, MySQL, PostgreSQL, Firebird , IBM DB2 , Sybase Adaptive Server Enterprise (ASE), Sybase iAnywhere(ASA)	MS Access, MS- SQL Server, MySQL.
SQL Server 2008 data types	Y	Y	Y	Y
Minimum required .NET Version	3.5	2.0	4.0 or higher	4.0
Supported .NET Languages	VB 9.0,C# 3.0	All	C# or VB.NET	C#4.0
<b>Scope</b>				
Use in Windows Desktop Apps	Y	Y	Y	Y
Use in web applications	Y	Y	Y	Y
<b>Mapping Techniques</b>				
Internal Data Access API	ADO.NET	JDBC	LLBLGen Pro Runtime Library	ADO.NET
ORM Base Technique	Reflection+ Runtime Code Generation	Reflection+ Runtime Code Generation	Reflection+ Runtime Code Generation	Reflection+ Runtime Code Generation
Forward Engineering Mapping/ Code First	Y (Runtime only)	N	Y	N
Reverse Engineering Mapping/ Database First	Y	Y	Y	Y
Default naming of generated BO Classes	Table Name	Table Name	Table Name	Table Name
Default Naming of generated BO Collections	Table Name + "s" ("s" get be switched off)	Table Name	User Defined	Table Name, Custom +random number ( 0 to 500)
Join, Grouping, Ordering & Projection	Y (Partially)	Y	Y	Y
Parameter	LINQ-to-SQL (formerly "DLINQ")	NHibernate	LLBLGen Pro	Object Relational Mapping Tool*
<b>Supported Mappings</b>				
Persisting fields and/or properties	Fields	Properties	Properties	Properties
Support for Nullables	Y	Y	Y	Y
Mapping 1 class-> 1 table	Y	Y	Y	Y
Mapping 1class->n table	N	Y	Y	Y
Mapping n classes-> n table	N	Y	Y	Y
Support collection types	EntitySet	IList, ISet, IDictionary, ICollection	EntitySet, Enumerable<T>, IList, IDictionary, ICollection	DataTable, Entity, Enumerable<T>, IList, IDictionary, ICollection
XML-based Mapping	Y	Y	Y	N
<b>Data Query</b>				

Name of Container	DataContext	Session	DataAdapter, Self Services	DataAdapter
Query Language	LINQ SQL	HQL, SQL, LINQ	SQL, LINQ SQL, HQL	SQL
SQL-Support	Y	Y	Y	Y
User provided ADO.NET Connections	Y	Y	Y	Y
Lazy Loading/Deferred Loading/Delayed Loading	Y, default	Y, default	Y	Y, default
Eager Loading/Immediate Loading/Demand Loading	Y	Y	Y	Y
<b>Data Update</b>				
Change Tracking (single object)	Y	Y	Y	Y
Change Tracking (collection)	Y	Y	Y	Y
Save an individual object	N	Y	Y	Y
Transaction Supported	Y	Y	Y	Y
Explicit Transactions required for all write operations	N	Y	Y	N
<b>Extensibility</b>				
Custom Code Generation	Y	Y	Y	Y
<b>Stored Procedure Support</b>				
Support for loading data	Y	Y	Y	Y(Planned)
Support for updating data	Y	Y	Y	Y(Planned)
Support for deleting data	Y	Y	Y	Y(Planned)
Support for Inserting data	Y	Y	Y	Y(Planned)

#### 4. CONCLUSION

The aim of our “object relational mapping system is to reduce the impact of impendence mismatch problem.” This work is focuses onto to the same one. This system will also provide code reduction and reusability facility through providing the rich set queries and simple ADO.NET code functionality.

#### REFERENCES

- [1] “An Object Relational Mapping Technique for Java Framework”- *International Journal of Engineering Science Invention*- Vol. 2 Issue 6 | June. 2013 | PP.01-09 - Ogheneovo, Edward Erhieyovwe, Asagba, Prince Oghenekaro, Ogini, Nicholas Oluwole.
- [2] S. Amber (2003), *Agile database techniques: effective strategies for the agile software developer*, USA: John Wiley and Sons, <http://www.agiledata.org/>, Retrieved April 20, 2011
- [3] S. W. Amber (2006), *Mapping objects to relational databases: O/R mapping in detail*, <http://www.ambysoft.com/mappingObjectsTut.html>.
- [4] L. DeMichiel and M. Keith (2006), JSR 220: Enterprise JavaBeans TM, version 3.0. Java Persistence API, <http://jcp.org/aboutJava/communityprocess/final/jsr220/index.html>
- [5] C. Bauer and G. King (2005), *Hibernate in action*, manning publishers Co. Boston/Massachusetts, pp. 1-25.
- [6] S. Philippi (2005), *Model driven generation and testing of object-relational mappings*, *Journal of Systems and Software*, 77, pp. 193-207.
- [7] LLBLGen Pro document:- <http://www.llblgen.com/pages/overview.aspx>
- [8] A. Keller and C. Keene (1993), *Persistence software: bridging object-Oriented programming and relational databases*, In *Proceedings of the ACM SIGMOD Int“ 1 Conference on Management of Data*, pp. 540-541.
- [9] Codd, E. F. (1970). *A Relational Model of Data for large Shared Data Banks*. *Communications of the ACM*, Vol. 13, No. 6, pp. 377-387.
- [10] Comparison Parameter taken from *Comparison of Object Relational Mapping Tools for the .NET Framework* By Dr. Holger Schwichtenberg



**Mr. Kuldeep Anil Hule:** received B.E degree in Computer Science and Engineering in 2012 from SOLAPUR University, Solapur, Maharashtra, India and pursuing the M.E. degree in Computer Science and Engineering in Nagesh Karajagi Orchid College of Engg. & Technology, Solapur, India. He is doing his dissertation work under the guidance of Mr. Shaikh Z. M., Assistant Professor at Nagesh Karajagi Orchid College of Engg. & Technology, Solapur, Maharashtra, India.



**Mr. Zakir Mujeeb Shaikh:** received BE degree in Computer Science and Engineering in 2000 from Dr. Baba Saheb Ambedkar University, Aurangabad, Maharashtra, India and ME Computer from BVU Pune, Maharashtra, India in 2004. He is presently working as an Assistant Professor in Nagesh Karajagi Orchid College of Engineering and Technology, Solapur, Maharashtra, India. His research area includes Image Processing, Database and software framework.