Data Access Layer Development for Interoperable GIS Solutions Using NHibernate Mapper

Obeada Khalid Ahmed, Asst. Prof. Dr. Sefer Kurnaz

Electrical and Computer Engineering
Electrical and Computer Engineering
obeadaobeada258@gmail.com; sefer.kurnaz@altinbas.edu.tr

Abstract— This paper presents the NHibernate Mapper, an object - to - relation database mapping tool that has the ability to automatically cross the schema and database library to provide suggestions on the correspondence and match the database provided by this NHibernate Object Mapper framework. Suggested mapping can be accepted or rejected via the GUI, which can also define the mapping in a final form by hand. Because of their ability to support data access layer development through database schema analysis for DBM enabled, NHibernate can provide great support to GIS solutions. NHibernate mapper has the possibility, in particular with the Domain Class library generated by GIS solution to allow rapid data access laying development.

Keywords— NHibernate; GIS; Object Mapper framework; GUI

I. INTRODUCTION

Most advanced systems rely on their trustworthiness and standardized query language in order to store information in a database. Object – The most common languages of system programming are Java, C # and C++ programming - oriented languages. The problem known as "object" - relational impedance mismatch "poses traditional approaches in developing software for accessing databases of objects – oriented languages. This is because the object paradigm-oriented and relationship databases are based on different principles in different special differences: implementation and association. To address the problem, the tools for mapping objects (ORM) were developed. NHibernate, Oracle TopLink and the newly developed Microsoft Entity Framework (EF) are the most successful representatives of ORM instruments. The ORM tools produce automatic database access layers on the basis of the Agile software and the methodology for database development. Application developers are encouraged to consider objects on the data layer and their links to those tools. The ORM controls how objects and relationships are handled over time. It tracks and inserts, updates and deletes the required SQL statements automatically during committing.

NHibernate is an open source framework comprehensive and stable. It can be used for or for Java as well. NHibernate can be found in .NET or Java environments. Although only a few commercial tools are available that can map existing object models to existing object generation databases and map files from existing

databases. This is important when trying to make standard domain knowledge of the legacy database systems. These three components are enough to generate a complete data access layer from NHibernate.

The main purpose is to create the NHibernate ORM framework for objects – to– database mapping specifications. The correspondence between the GIS solution model and the database can be automatically determined semi - detailed. Corresponding attributes and columns of the database are established. The tool has a visual foreground, which shows the mapping of Levenshtein distance by heuristic data type. These suggestions have to be tapped by the user interaction to define final mapping definitions. The XML - based mapping file is also created using defined mapping definitions.

II. MOTIVATION AND RELATED WORK

We have developed GIS with NHibernateMapper for this work. It aims in particular to support rapid data access layer generation with the help of existing ORM solutions. Ontology offers a chance to define the importance of information and is mainly used to resolve the significance of the available data in different GIS communication. Ontology could, however, also be seen as a GIS development model. The concept of ontology can be regarded as domain objects that are mapped in classes orientated towards objects. We developed an object - mapping tool - a database in NHibernateMapper. It aims to map classes of GIS ontology to existing databases.

Data is the common problem between different images that has been tackled either from the algorithm or from the point of view of visualization. In this field, a lot of work has been done, mainly with a degree of automation. E. Rahm et al. provides a summary of the various approaches to mapping schemes and the generation of automated mappings. A user interaction scheme analysis and choice combination is presented in [6] and [10] for data mapping between two XML schemes. The business tool, such as Altova MapForce and Microsoft BizTalk Mapper, is used to map scenarios including Xml - to - Xml, Xml databases - to - Xml and DatabaseMaps. Commercial tools are available everywhere. However, it proved too generic and difficult to learn about a particular domain application.

III. NHIBERNATEMAPPER ARCHITECTURE

The NHibernateMapper file generation process is based on existing relation database and specific domain knowledge in application programming classes. Application programming classes are available. The tool uses open scheme definitions of NHibernate - mapping file name for generating the mapping file (nhibernate-mapping.xsd). The Dynamic Linked Library (DLL) classes must be used for the use of the NHibernate Mapper tool, which is the existing and the domain object. Database and Member class attributes for selected DLL are automatically extracted from NHibernateMapper.

The user has to manually select an object – a class orientated and a particular data table. The database schema and class attribute will then be automatically scanned and user mapping suggestions will be made and displayed. The applicant mappings for each column are created based on a string matching algorithm (column - to - mapping attribute) and on a comparison of the data type. The user can accept the proposed mail or reject it. Users can also use an interactive GUI, which can be used for finalizing mapping suggestions. Mapping can also be redefined manually in addition to the acceptance or rejection of proposed mapping. When the mapping definition is finally fulfilled, user can be saved in the NHibernate XML mapping file.
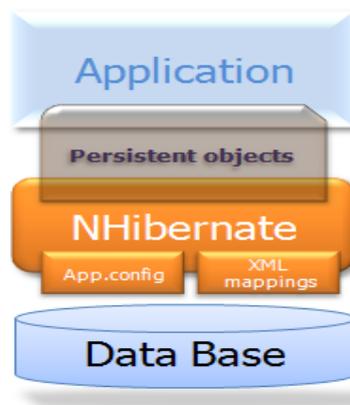

Figure1. Nhibernate Architecture

IV. IMPLEMENTATION DETAILS

Microsoft. NET Framework 4.0 is the complete implementation of NHibernateMapper in C#. This creates a class library called NHibernateMapper.dll. Remembering database and file management classes can be split into two main groups in all library programming classes. The class Dbinfo is a central class of the scheme for database recovery. It represents a database schema object model. Schema information, such as tables (DbTableInfo), columns (DbColumnIninformation), and restriction (DbConstraintInfo) is collected and connected within the DbInfo class.

```
private static ISessionFactory CreateNhSessionFactory()//(Assembly _assembly)
{
        var connStr = DBHelper.getDB_CNN(); //
        ConfigurationManager.ConnectionStrings["DB_CNN"].ConnectionString;
        try
        {
                return Fluently.Configure()
            .Database(MsSqlConfiguration.MsSql2008.ConnectionString(connStr))
                .Mappings(m=>m.FluentMappings.AddFromAssembly
                (Assembly.GetAssembly(typeof(ENTUserAccountMap)))
                .Conventions.AddFromAssemblyOf<PriamryKeyConvention>())
                    .BuildSessionFactory();
        }
         catch (FluentConfigurationException e)
        {
        throw new Exception("Connection Error:" + e.Message, e.InnerException);
        }
}
```

Figure 2. Example of database connection file

The configurable design supports any other DBMS that has a.NET data provider. Support could be introduced for a new DBMS in three steps:

- Defines three further queries to recover information on the database schema. The queries are part of the DbInfoQueries class. (GetTables, GetColumns and GetConstraints)
- The DBMS Data Provider (in DLL form) offered by the DBMS Provider. We've already got MySql in MySQL and Npgsql.dll in PostgreSQL. Using Data.dll.
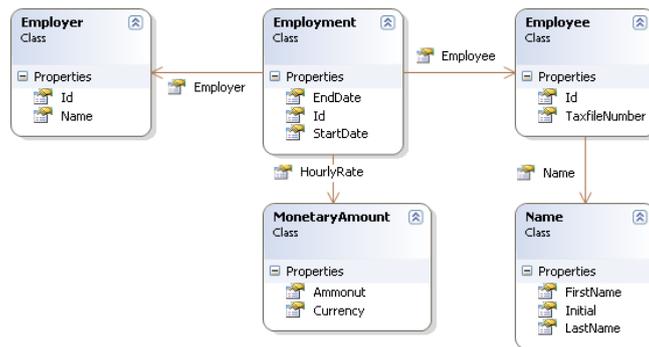- Change the settings file to include an added declaration from the data provider.

Figure 3. Diagram of Nhibernate schema relationships

For the build - up of file manipulation classes the XML mapping file object model and system file storage are responsible. Class XmlMapElementInfo manages mapping a column of a database with an attribute of that class. The central class of this part is XmlMapFileGenerator. It connects all mapped items to an entire mapping file and saves them to an XML file. The architecture of new heuristic algorithms is sufficiently flexible, as are the "Levenshtein Distance" and the "Correspondence Type."

At the programming level their object model is presented by mapping, which is generated from the schemo definition file. The schema definition of the developer permits the validation and storage within the XML file of the mapping file. The whole mapping process of file generation can be compiled in the programming classes described above with a code snippet. As already mentioned, the tool is intended to

implement it in GIS solutions in the interoperable field. With the existing GIS database we tested NHibernateMapper in the PostgreSQL.

V. CONCLUSIONS

A group of scheme mapping tools is NHibernateMapper. This tool has been developed to specifically map data obtained through existing databases to domain model classes. Ontology can be used for defining the domain classes as a starting point. The mapping specifications for a semicolon based on the NHibernate ORM were also developed for an automatic object. Ontologies offer a uniform mechanism for defining the significance of data and are an appropriate way to define a domain model. The ontology concepts can be seen as onto maps of the existing relationship databases by representing ontology as an object - oriented class as a domain object. For application in the area of Geographical Information Systems NHibernateMapper is mainly developed. It can be combined with an existing instrument that transforms concept ontology into object - oriented classes. Consequently, by using existing ORM tools, NHibernateMapper should support data layer access generation. The use of semi - automatic mapping and the tool [4] described for interoperable GIS solutions can make a significant contribution to quickly developing the data access layers.

The configurable design of the NHibernateMapper provides an intuitive new DBMS. NHibernateMapper currently supports only hibernate mapping. The enhancement plan assumes that the Microsoft Entity Framework can be added to other popular ORM map definitions. The tool is also intended for inclusion in the entire generator solution of the WebGIS application source code.

REFERENCES

[1] Adya, J. Blakeley, S. Melnik and S. Muralidhar, "Anatomy of the ADO.NET Entity Framework". In Chan, C. Y. Ooi B. C. & Zhou, A. (eds), Proceedings of the ACM SIGMOD International Conference on Management of Data, June 11-14, Beijing, China, 2007, pp. 877-888.

[2] P. Castro, S. Melnik and A. Adya, "ADO.NET entity framework: raising the level of abstraction in data programming," In Chan, C. Y. Ooi B. C. & Zhou, A. (eds), Proceedings of the ACM SIGMOD International Conference on Management of Data, Beijing, China, 2007, pp. 1070-1072.

[3] S. Ambler, Agile Database Techniques, Wiley, 2003.

[4] A. Stanimirović, M. Bogdanović and Stoimenov, "Data Access Layer Generation for Interoperable GIS Environments," 12th AGILE International Conference on Geographic Information Science, 02-05 June, Hannover, Germany, ISSN 2073-8013, 2009

[5] E. Rahm, P. A. Bernstein, A survey of approaches to automatic schema mapping, The VLDB Journal, Springer Verlag, 10: 334-350, 2001.

[6] H. Su, H. Kuno and E. A. Rudensteinern, "Automating the Transformation of XML Documents," Proc Workshop on Web Information and Data Management, 2001.

[7] L. L. Yan, R. J. Miller, L. M. Haas, and R. Fagin, "Data-driven understanding and refinement of schema mappings." In SIGMOD Conference, 2001.

[8] Altova Mapforce, http://www.altova.com/products_mapforce.html

[9] M. A. Goulde, Microsoft's BizTalk Framework adds messaging to XML, E-Business Strategies & Solutions, 1999, pp.10-14.

[10] S. Bossung, H. Stoeckle, J. Grundy, R. Amor and J. Hosking, "Automated Data Mapping Specification via Schema Heuristics and User Interaction," 19th International Conference on Automatic Software Engineering (ASE'04), 2004, pp. 208—217.

[11] C. Bauer, G. King, Java Persistence with Hibernate, Manning Publications, 2006.

[12] OracleTopLink, http://www.oracle.com/technology/products/ias/toplink

[13] F. T. Fonseca, M. J. Egenhofer, "Ontology-Driven Geographic Information System," 7th ACM Symposium on Advances in Geographic Information System, Kansas City, 1999.

[14] M. Bogdanović, A. Stanimirović and L. Stoimenov, "Ginis MvcGen – a Prototype of WebGIS Application Source Code Generator," ICEST 2010, Ohrid, Makedonija, 2010, pp. 307 – 310.