CSE 736: ADVANCED TOPICS IN DATABASE SYSTEMS

DATABASE SEMINAR REPORT

SENTHIL KUMAR GURUSAMY

Compiling Mappings to Bridge Applications and Databases

- *Sergey Melnik, Atul Adya and Philip A. Bernstein*

Anatomy of the ADO .NET Entity Framework

- *Atul Adya, Jose A. Blakeley, Sergey Melnik, S. Muralidhar and the ADO .NET Team*

Overview

The first paper explains Entity Data Model of ADO .NET framework, states the challenges associated with the model and justifies the proposed solutions to the challenges encountered. The second paper describes the entire ADO .NET Framework, detailing the specific functions of each component in the framework. The motivation to such a framework is the convenience and popularity of Object Relation Mapping (ORM) enterprise systems.

The Entity Data Model (EDM) uses a declarative language to specify the Object to Relation mappings. The mappings are specified as individual mapping fragments. The declarative language to specify the fragment is very simple and similar to SQL statements with added semantics. It is also robust and eliminates loss of data during any of the object to relational or vice-versa transformations. Introduction of the concept of bidirectional views is also a novel work proposed by this paper. In this model, the object oriented application view of the data is translated into a bidirectional conceptual relational view which can be queried using SQL semantics on application programming variables. The specified Mappings are compiled into a view. Mapping specifications will not always guarantee data roundtripping. Application developers cannot be entrusted with this additional task of ensuring data roundtripping criterion. For the purpose of generating a consistent view, the relational data is divided into partitions based on the mapping fragments. Each partition is compiled using relevant mapping fragment. The compiled parts are regrouped and merged into a view. The algorithms defined for each functions take of ensuring data integrity for any specified mapping.

The Authors evaluated the model based on correctness, performance and efficiency giving more important to correctness. The correctness of the proposed algorithms was verified by the product test team using a automated suite and by a tool developed by the Product Verification Group at Microsoft Research in addition to proving them theoretically. The runtime time of the entire process was determined as O(n log n). So the Compile time stated to be in terms of seconds when *n* exceeded a few dozens. This time is taken during the start of the application and not during each query hit. Furthermore, this speeds up the response time of the application. So it was not considered a serious performance issue.

The paper explaining the ADO .NET framework is an industrial paper with more details on specifications of the different components in the framework and their functions. The framework has graphical tools for specifying mappings and modeling entity scheme which is used in compiling for generating the view. The programming layers, BI/Reporting interfaces, web services interact directly with the Entity framework. Entity framework is the first of its kind to allow reporting interface on entity data. It supports three different programming schemes viz. Entity SQL, CRUD on entity objects and LINQ. The EDM relies on relational data supplied from data stores using the Store Specific providers available for each type of relational data storage. The concept of Update Views and Query Views drive the bidirectional view to serve up-to-date and consistent from the data stores amidst of concurrent connections. The relation data is updated using Update translation techniques on entity data. The EDM has its own metadata and services for serving the framework.

Detailed Comments

The first paper compliments the second paper with a detailed description of Entity framework the most important sub component of the ADO .NET framework. The specification of object to relational mapping using the declarative mapping language was well explained with explicit and sufficient examples. The semantics of entity SQL and LINQ was illustrated but not enumerated sufficiently. The simplicity of LINQ was shown. The surfacing of Common Language Runtime(CLR) objects through the Object services of the framework from the view whenever needed is paragon of flexibility and compatibility with diverse programming layers. The mapping compilation problem was clearly delineated and technique of partitioning and regrouping exploiting the inner, was proposed the solution with very clear explanations.

The papers failed to provide any quantitative results. This may be due to the confidential nature of the results which could to be used to make future changes or improvements to the framework or to obscure an potential issue which they plan to address in future versions. The papers lack clarity in providing clear demarcation between the bidirectional view and CLR objects. It also fails to address the scope and lifetime of the CLR objects in the framework. The papers did not explain the real advantage of LINQ, the optimizations the framework carries out on LINQ queries alone. This is a valuable solution to the most problem of performance degradation of faced by ORM frameworks. This is the extra mile that makes ADO .NET superior and survivable.

The questions asked during the presentations mainly concentrated on the comparison of framework with other similar frameworks and the capability of the declarative mapping specification - as to how the language could be used to specify a complex entity schema. They were also questions on the relevance between Entity Objects and the generated Bidirectional View which is a complicated abstraction hard to digest in a short period of time even with the necessary explanations.