

Telerik OpenAccess ORM Q2 2009

Daten und Objekte im Verbund

O/R-Mapper dienen dem objektorientierten Zugriff auf relationale Datenbanken. Telerik bietet einen ausgereiften und leistungsfähigen O/R-Mapper für .NET an. **Andreas Maslo**

Auf einen Blick

Inhalt

Telerik OpenAccess ORM ist ein professioneller, objektrelationaler Mapper für das .NET Framework, der das Forward- und Reverse-Mapping unterstützt und zusammen mit leistungsfähigen Assistenten und Dokumentationen den objektorientierten Zugriff auf relationale Datenbanken vereinfacht.

Plattform

Windows, .NET Framework

Technik/Anwendung

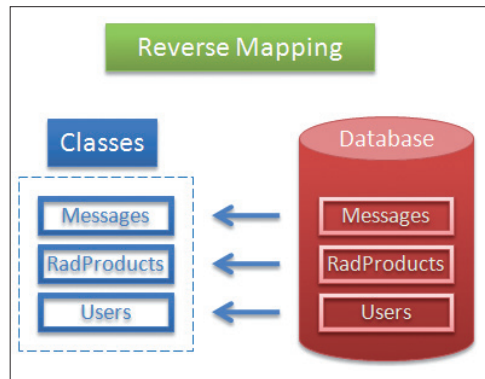
Datenbankentwicklung, Mapping

Autor

Dipl.-Ing. Andreas Maslo leitet das Ingenieurbüro IngES, das sich mit der Erstellung von EDV-Publikationen und der Softwareentwicklung befasst. Er ist als freier Journalist, EDV-Berater, Fachbuchautor und Redakteur der database pro tätig.

Frameworks für das O/R-Mapping haben die Aufgabe, die Hürden zwischen der objektorientierten Entwicklung und den relationalen Datenbanken zu überwinden und den Datenaustausch zwischen Objekten und Tabellen zu automatisieren. Dabei wird idealerweise der objektorientierte Quellcode von der Datenbank getrennt. Dies hat den Vorteil, dass sich Domänenentwickler ihren OO-Modellen und Datenbankentwickler den Mapping-beziehungsweise Abbildungsfunktionalitäten widmen können. Im einfachsten Fall werden dabei relationale Tabellen ein zu eins auf die entsprechenden Objekte abgebildet. In der objektorientierten Programmierung spielt jedoch die Vererbung eine wesentliche Rolle, die ihrerseits die objektrelationale Abbildung, das sogenannte Mapping, erschwert. In diesem Zusammenhang ist zu berücksichtigen, wie die Programmentwicklung erfolgt: Liegt eine Datenbank vor, die auf ein Domänenmodell abgebildet werden soll, oder ist der Ausgangspunkt der Programmentwicklung ein persistierendes OO-Modell (vergleiche Textkasten „Begriffe kurz erläutert“), das über eine neue Datenbank verwaltet werden soll? O/R-Mapper müssen dementsprechend flexibel nutzbar sein. Aktuelle O/R-Mapper unterscheiden sich im Funktionsumfang, bei den mitgelieferten Tools und auch in puncto Funktionalität erheblich. Sie sind mitunter speziell für bestimmte objektorientierte Sprachen und/oder Systemumgebungen verfügbar. Dies hat zur Folge, dass Programmportierungen von einem O/R-Mapper zu einem anderen und einhergehend damit von einer Systemplattform zu einer anderen derzeit kaum möglich sind.

O/R-Mapper müssen sich mit Objekten und Daten auseinandersetzen und damit auch das Problem der Identitäten berücksichtigen. In der objektorientierten Programmierung ist ein instanziiertes Objekt anhand der Speicheradresse eindeutig identifizierbar. Objekte mit unterschiedlicher Identität und unterschiedlicher Speicheradresse dürfen den gleichen Inhalt haben. In einer Datenbank und deren Tabellen dürfen entsprechend mehrere Zeilen den gleichen Inhalt haben. Um die

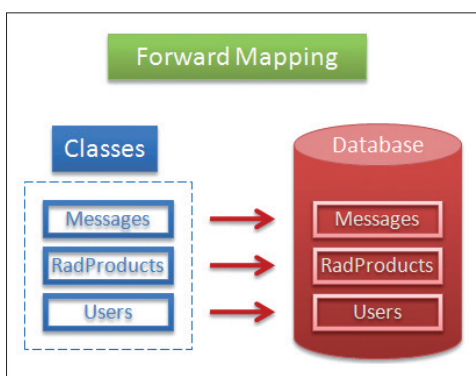


Beim Reverse-Mapping werden zu einer bestehenden Datenbank Klassen angelegt (Quelle: Telerik) (Bild 1)

Unterscheidung der Tabellenzeilen zu erlauben, müssen O/R-Mapper diese mit einem automatisch vergebenen Primärschlüssel versehen, der auch im Domänenmodell Verwendung findet. O/R-Frameworks haben ferner zu berücksichtigen, dass mitunter mehrere Instanzen der Daten vorgehalten oder auch für die Parallelbearbeitung durch mehrere Benutzer verfügbar gemacht werden müssen. Objekte müssen überwacht und im Rahmen von Transaktionen berücksichtigt werden, um die korrekte Datenaktualisierung sicherzustellen. Der objektorientierte Datenzugriff und die Datenabfragen müssen auf optimierte Datenbankzugriffe umgeleitet werden. Ferner muss intern sichergestellt sein, dass Objekte sinnvoll geladen und wieder entladen werden, und zwar so, dass sie den Speicher effektiv nutzen. Dabei gilt es zu beachten, dass Objekte immer in der Gesamtheit geladen werden, anders als dies beim Laden von Teiltabellen über SQL-Anweisungen möglich ist.

Ist der relationale Datenbankzugriff sehr leicht nachvollziehbar, ergibt sich beim Einsatz von O/R-Mappern eine gesteigerte Komplexität, da nunmehr alle relationalen Datenbankinhalte über „dazwischengeschaltete“ Objekte verfügbar gemacht werden. Dies erleichtert zwar den Zugriff auf der objektorientierten Sprachebene, verbirgt aber die eigentlichen Datenbankinhalte. Damit sind einige der Besonderheiten des O/R-Mapping angeführt. Detaillierte Informationen zum O/R-Mapping finden Sie in [1]. In jedem Fall müssen Sie sich in die O/R-Mapper einarbeiten. Hilfreich ist es, wenn Sie nicht nur das O/R-Mapper-Tool, sondern zu- ▶

Beim Forward-Mapping werden Tabellen über bestehende Klassen angelegt (Quelle: Telerik) (Bild 2)

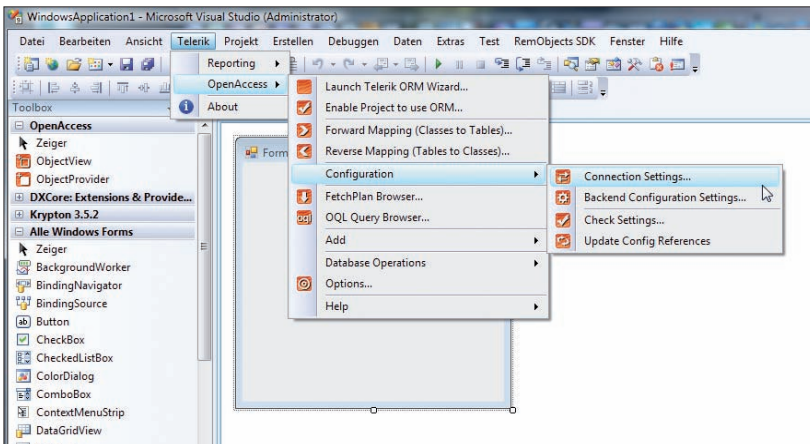


sammen mit dem O/R-Mapper auch ausführliche Informationen, Beispielprojekte sowie die Dokumentationen zu den verwendeten Programmierschnittstellen erhalten.

Express & Co.

Mit Telerik OpenAccess ORM ist ein praxiserprobter O/R-Mapper für das .NET Framework verfügbar, den Sie im Verbund mit Visual Studio 2005 und 2008 sowie den .NET-Programmiersprachen einsetzen (zum Beispiel C#, Visual Basic .NET) und der sich in die Entwicklungsumgebungen von Visual Studio integriert [2]. Er

Erweiterte Menüfunktionen zum ORM-Tool von Telerik (Bild 3)



bietet eine Vielzahl von Assistenten, die Sie bei der Programmentwicklung interaktiv unterstützen. Ferner erhalten Sie erweiterte Programmierschnittstellen (APIs) für den vereinfachten Datenzugriff sowie eine spezielle Abfragesprache für Objekte (OQL = Object Query Language). Telerik OpenAccess ORM ist in zwei Produktvarianten erhältlich. Die Express-Edition ist kostenfrei erhältlich und bietet die volle Funktionalität, ist aber ausschließlich mit kostenfrei verfügbaren Datenbanken einsetzbar. Dazu gehö-

ren die Express-Editionen von SQL Server 2005 beziehungsweise 2008 sowie MySQL 5.x, Firebird Server 2.x sowie Oracle Database 10g Express Edition. Die kommerzielle Variante von OpenAccess ORM unterstützt zusätzlich kommerzielle Datenbanken, wie SQL Server 2000, 2005 und 2008, Advantage Database Server 8.1, SQL Anywhere Server 10.x sowie Oracle Server in den Versionen 9.2, 10.x und 11.x. Für spätere Versionen soll auch die Unterstützung für SQL Server CE, VistaDB, SQLite, Postgres und DB2 ergänzt werden. Für den ersten Einstieg ist es sinnvoll, das Produkt ausgiebig in der Express-Edition im Verbund mit einer der unterstützten, frei verfügbaren Datenbanken zu testen.

Reverse- und Forward-Mapping

Prinzipiell bietet sich der Einsatz von O/R-Mappern mit objektorientierten Programmiersprachen an. Allerdings sollten Sie den verwendeten O/R-Mapper vor dem Praxiseinsatz zunächst im Detail auf Ihre Anforderungen hin prüfen. Telerik OpenAccess ORM unterstützt Sie beim funktionalen Test, bietet Demos und detaillierte Informationen sowie Testanwendungen für Windows Forms und ASP.NET an. Das Tool selbst empfiehlt sich als einer der leistungsfähigsten O/R-Mapper für das .NET Framework. Hilfreich ist dabei insbesondere die Integration in die Entwicklungsumgebung von Visual Studio sowie die funktionale Einbindung über Tool-spezifische Assistenten. Diese unterstützen Sie nicht nur beim Reverse-, sondern auch beim Forward-Mapping. Beim Reverse-Mapping ist eine bestehende Datenbank der Ausgang für ein neues Software-Projekt. In diesem Fall werden persistente Klassen über die existierende Datenbank generiert, um anschließend über diese Klassen direkt auf die Datenbank zuzugreifen (Bild 1).

Begriffe kurz erläutert

Persistenz: Der Begriff Persistenz bezeichnet in der Informatik die Fähigkeit, Daten oder auch Objekte in nicht flüchtigen Speichermedien, zu denen Dateisysteme, aber auch Datenbanken gehören, zu speichern.

Objekt: In der objektorientierten Programmierung (OOP, Object-oriented Programming) ist ein Objekt ein Datentyp, der über eine Klasse als Objekttyp definiert wird. Die Klasse definiert eine Schnittstelle (Interface), die den Objektzugriff regelt. Die Schnittstelle besteht aus Methoden, Ereignissen und Eigenschaften (Attribute). Per Klassendefinition und durch Herstellung eines Objektbezugs wird ein Objekt angelegt. Bestimmte Ausprägungen eines Objekttyps werden Instanzen genannt.

Entität: Bei der Datenmodellierung wird ein eindeutig zu bestimmendes Informationsobjekt als Entität (Entity) bezeichnet. Eine Entität wird einem Entitätstyp zugeordnet; Entitäten stellen bestimmte Ausprägungen eines Entitätstyps dar. Entitäten mit gleichem Entitätstyp werden anhand eines identifizierenden Attributs oder anhand einer Attributkombination unterschieden, die ihrerseits als Identifikator (ID = Kennung) bezeichnet wird. Wird ein Entitätstyp in einer relationalen Datenbank gespeichert, wird aus der ID der Schlüssel (Key). Es gilt:

Objekttyp = Entitätstyp
+ zugehörige Methoden

Entity-Relationship-Modell: Entitäten können mit sich selbst und mit anderen Entitäten in Beziehung stehen (Relationship). Das in der Regel grafische Abbild der Beziehungen wird im Rahmen der Datenmodellierung als Entity-Relationship-Modell bezeichnet. Über das Datenmodell kann eine relationale Datenbank erstellt werden, wobei Entitätstypen als Tabellen in der Datenbank umgesetzt und einzelne Entitäten als Tabelleneintrag (Datensatz) gesichert werden.

Objektorientierte Programmierung: Bei der objektorientierten Programmierung (OOP) handelt es sich um ein Paradigma, bei dem im Verbund stehende Daten und Funktionen über Objekte zusammengefasst beziehungsweise gekapselt werden.

Alternativ lassen sich basierend auf bestehenden Klassen Tabellen in einer relationalen Datenbank anlegen. Klassen werden dabei wie gewohnt definiert und per Forward-Mapping in entsprechende Datenbanktabellen und -spalten umgewandelt. Nach der Abbildung greifen Sie über die Klassen auf die zugeordnete Datenbank zu und persistieren die Objekte in der Datenbank (Bild 2).

Visual-Studio-Integration

Um die Einarbeitung in einen O/R-Mapper zu erleichtern, ist es sinnvoll, auch die funktionale Unterstützung innerhalb der verwendeten Benutzerumgebung für die Programmentwicklung zu vereinfachen. Dies realisiert Telerik durch die direkte Anbindung an die Entwicklungsumgebung von Visual Studio 2005 und 2008. Direkte Unterstützungen für andere .NET-Entwicklungsumgebungen, wie SharpDevelop, sind derzeit nicht verfügbar. Um demnach sinnvoll mit OpenAccess ORM arbeiten zu können, ist Visual Studio die Grundvoraussetzung. Bereits nach der Einrichtung des O/R-Mappers finden Sie im Menü *Telerik* den neuen Untermenüeintrag *OpenAccess* (Bild 3). Bevor die inaktivierte Menübefehle nutzbar sind, müssen Sie ein neues Projekt vom Typ *Windows Forms* oder *ASP.NET* anlegen. Anschließend rufen Sie zunächst den Menübefehl *Telerik/OpenAccess/Enable Project to use ORM* ab. Der Befehl sorgt dafür, dass Verweise auf die Komponenten *OpenAccess* und *OpenAccess.Query* dem Projekt hinzugefügt werden. Ferner bewirkt er, dass dem Projekt eine Klasse mit dem Namen *ObjectScopeProvider1* hinzugefügt wird, die selbst das Interface *IObjectScopeProvider* implementiert und die den Datenbankzugriff für eine dynamisch geladene Assembly verfügbar macht. Im Rahmen der ORM-Aktivierung legen Sie auch die zu verwendende Datenbank und deren Anbindung fest. Ob Sie im Projekt Forward- oder Reverse-Mapping nutzen wollen und in welcher Form die Abbildung im Detail vorzunehmen ist, legen Sie nun bereits per Assistent fest. Rufen Sie dazu den Menübefehl *Telerik/OpenAccess/Launch Telerik ORM Wizard* ab. In einem ersten Dialog geben Sie an, ob Sie eine existierende Datenbank für die Anlage von Klassen nutzen (*Reverse Mapping*) oder aber ausgehend von bestehenden Klassen Datenbanktabellen generieren wollen (*Forward Mapping*).

Abhängig von Ihrer Wahl gelangen Sie in weitere Dialoge. Über diese werden Assistenten verfügbar gemacht, über die Sie je nach Mapping-Variante spezialisierte Assistenten abrufen können. Über die Dialoge wählen Sie die beim Forward-Mapping zu berücksichtigenden vorhandenen Klassen aus (Bild 4) oder selektieren

Datenbanken und deren Tabellen für das Reverse-Mapping. Prinzipiell können Sie beide Mapping-Varianten auch in einem einzelnen Projekt kombinieren. Nach Bedarf lassen sich erweiterte Optionen und Datenanbindungsfunktionen im Dialog komfortabel variieren. Basierend auf den Abbildungsverfahren werden Klassen generiert, mit denen der spätere Datenbankzugriff erfolgt. OpenAccess ORM macht die Fähigkeit für Persistenz für die .NET-Datentypen verfügbar.

Einstieg und erweiterte Leistungsmerkmale

OpenAccess ist komplex. Für einen vereinfachten Einstieg sorgt ein 250-seitiges Dokument für Einsteiger, das im Internet unter der Adresse [3] zur Verfügung steht. Darin werden die Entwicklungsumgebung und die OpenAccess-Verwendung im Detail erläutert. Sie finden hier Informationen zur Vererbung (Inheritance), zu Transaktionen (Transactions), zum Datenbankzugriff sowie zur Optimierung von Anwendungen, die OpenAccess nutzen. Spezielle Funktionen können Sie sich direkt im Internet mitsamt erweiterten Leistungsmerkmalen online demonstrieren lassen [4]. database pro wird in einer späteren Ausgabe ein gesondertes Praxisprojekt vorstellen, das OpenAccess ORM im Verbund mit der Telerik-Reporting-Komponente zeigt.

Objektorientierter Datenzugriff

Setzen Sie O/R-Mapping-Frameworks wie OpenAccess ORM richtig ein, sorgen Sie fortan für eine effizientere Softwareentwicklung, speziell im .NET Framework. Sie erleichtern nicht nur die Verwendung unterschiedlicher Datenbanken, sondern sorgen für eine einheitliche, objektorientierte Codierung von Quelltexten. Die Nutzung eines Tools eines kommerziellen Anbieters hat den Vorteil, dass Sie gegenüber O/R-Mappern aus dem Open-Source-Bereich besser mit Zusatzinformationen, Beispielen und Dokumentationen versorgt werden. Sie teilen Anregungen für künftige Entwicklungen nicht nur dem Hersteller mit, sondern tauschen sich auch über Foren mit anderen Produktanwendern aus. Telerik informiert über geplante Neuerungen und richtet diese konkret auf die Anforderungen aus, die sich in der Praxis ergeben. Projekte sind damit zukunftssicher. [am]

- [1] Objektrelationales Mapping; www.agiledata.org/essays/mappingObjects.html
 [2] Telerik OpenAccess ORM; www.telerik.com/products/orm
 [3] Produkteinstieg; www.telerik.com/documents/OpenAccess-MadeEasy.pdf
 [4] Online-Demonstrationen; <http://demos.telerik.com/orm/examples/overview/defaultcs.aspx>

Unterfunktionen für das Forward-Mapping (Bild 4)

