Laboratórios de Desenvolvimento de Software - LEI

Entity Framework

Ricardo Santos **Nuno Macedo** Ângelo Costa
nfm@estgf.ipp.pt

November 6, 2013

ESTGF | POLITÉCNICO DO PORTO

Object-relational mapping

- The *object-relational mapping* (ORM) is one of the biggest challenges when developing applications:
  - Class diagrams cannot be directly mapped to relational database schemes;
- There is an object-relational *impedance mismatch*:
  - E.g., how to map the notions of inheritance, interfaces or polymorphism into databases?

Object-relational mapping

- The developer should work at the business layer without being encumbered by database technicalities.
- Wide set of tools have been developed to manage ORM.

ADO.NET

- .NET technologies to access databases (ADO.NET) have been evolving:
  - Low level *SqlConnections*;
  - *DataSets*, higher-level but keep the database structure;
  - *LINQ to SQL*, allows managing databases as objects but relationship is still 1:1;
  - *Entity Framework*, the first "real" ORM tool in ADO.NET.

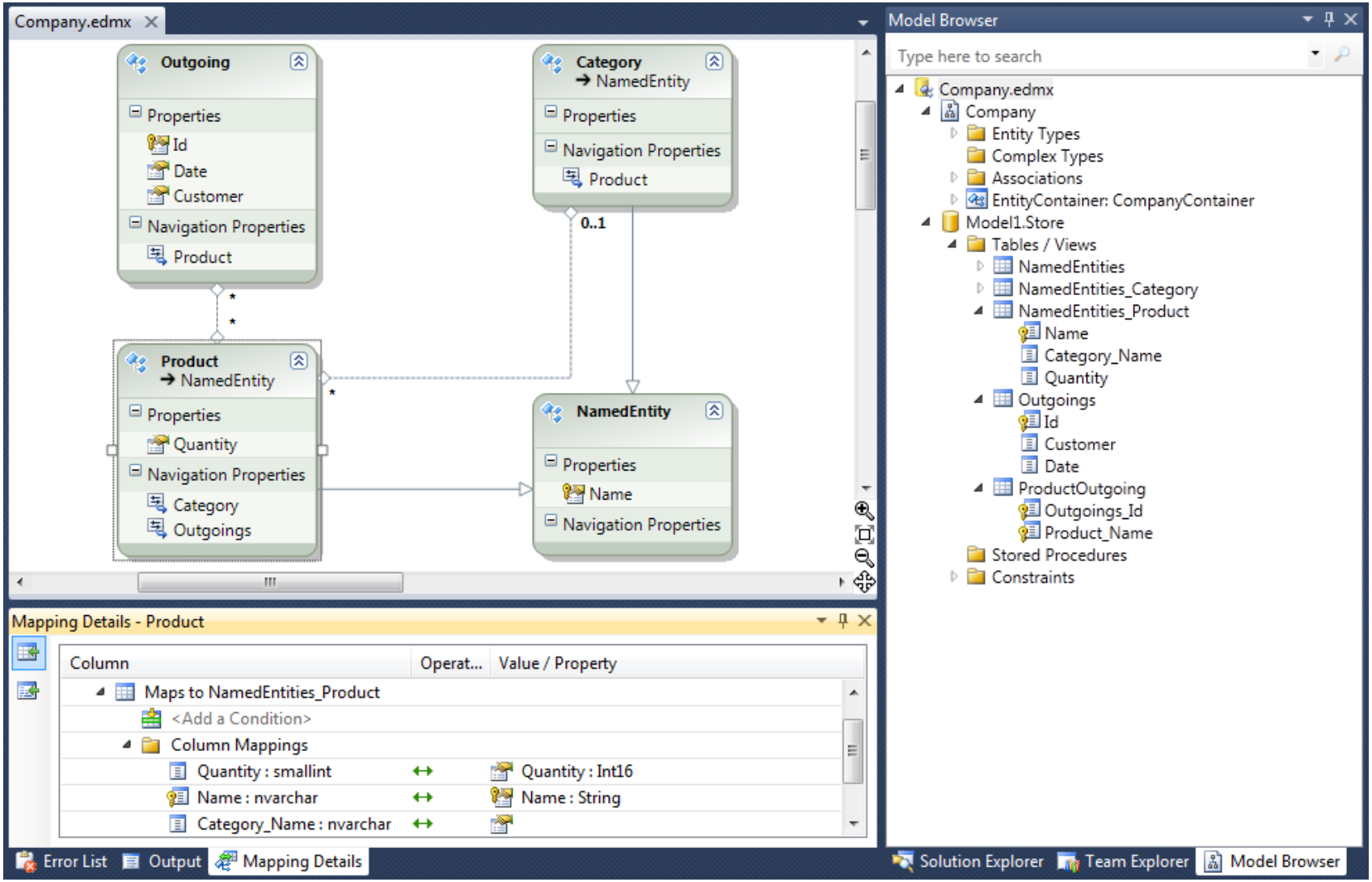
Entity Framework

- The main component of the Entity Framework is the *Entity Model*;
- It is comprised by:
  - The *conceptual model*, which represents the object model;
  - The *store model*, which represents the database structure;
  - The *mapping*, that connects the two models.

Entity Framework

- The object and database models are no longer related 1:1;
- E.g., an object may be mapped to multiple tables;
- They can be independently modified;
- E.g., entities names can be changed independently of the table names.

Example


Company.edmx
Outgoing
Properties
Id
Date
Customer
Navigation Properties
Product
Category
NamedEntity
Properties
Navigation Properties
Product
0..1
Product
NamedEntity
Properties
Quantity
Navigation Properties
Category
Outgoings
NamedEntity
Properties
Name
Navigation Properties
Model Browser
Type here to search
Company.edmx
Company
Entity Types
Complex Types
Associations
EntityContainer: CompanyContainer
Model1.Store
Tables / Views
NamedEntities
NamedEntities_Category
NamedEntities_Product
Name
Category_Name
Quantity
Outgoings
Id
Customer
Date
ProductOutgoing
Outgoings_Id
Product_Name
Stored Procedures
Constraints
Mapping Details - Product
Column
Operat...
Value / Property
Maps to NamedEntities_Product
<Add a Condition>
Column Mappings
Quantity : smallint
Quantity : Int16
Name : nvarchar
Name : String
Category_Name : nvarchar
Error List
Output
Mapping Details
Solution Explorer
Team Explorer
Model Browser

Querying data

- There are a number of techniques to query data in the Entity Framework;
- *LINQ to Entities* is the most standard one;
- Queries are performed under the standard LINQ syntax;
- The entity model defines an object context:

```
CompanyContainer context = new CompanyContainer();
```

Querying data

```
CompanyContainer context = new CompanyContainer();
var qry = from p in x.Outgoings
          where p.Customer == "x"
          select p;
```

Changing data

- The Entity Framework keeps track of the data changes;
- A `context.SaveChanges()` command propagates changes to the database;

Changing data

- Create:

```
Product p = new Product();
p.Name = "new_name";
...

context.Products.AddObject(p);
context.SaveChanges();
```

Changing data

- Update:

```
var cs = from c in context.Products
         where c.ID == 1
         select c;
Product p = cs.FirstOrDefault();

p.Name = "new_name";
context.SaveChanges();
```

Changing data

- Navigating is also easy (joins are abstracted):

```
var cs = ...;
Product p = cs.FirstOrDefault();

foreach (Category in = p.Categories) {
    c.Name = ... ;
}
```

- *Lazy loading*: one database access for each iteration:
  - May be a *performance* problem.

Changing data

- Data can also be eagerly loaded:

```
var cs = from c in context.Products
                                .Include("Categories")
         ...
Product p = cs.FirstOrDefault();

foreach (Category in = p.Categories) {
    c.Name = ... ;
}
```

- All `Categories` are loaded at the first query.

Bibliography


N. Randolph, D. Gardner, M. Minutillo and C. Anderson
Professional Visual Studio 2010.
Wrox, 2010.


Nagel, C., Evjen, B., Glynn, J., Watson, K. and Skinner, M.
Professional C# 4 and .NET 4.
Wrox, 2010.