

# “Introducing ‘Oslo’ – Microsoft’s Modeling Framework” Shy Cohen

May 22, 2009

[MS Techdays 2009] Summary “Introducing ‘Oslo’ Microsoft’s Modeling Framework” Shy Cohen

I added some personal interpretations!

## So what’s modeling Anyway?

A **model** is a simplification to address complexity, showing structure and providing just enough info for a given purpose. It is captured in a concrete form that allows easy sharing.

**Modeling** is the art of creating a model. (Duh :))

## Software modeling

Software modeling is trending toward executable data. It is the logical next step in the evolution going from assembler over C to C#. Where C# added a virtual machine and intermediate language to the stack of compilers, we now want to add our own language and runtime to it.

Why? Since the introduction of **Domain Driven Development** we are seeing the advantages of implemented objects that are structured in a way very close to the problem domain. Wouldn’t it be great if we could express business logic in a language that is close to the problem domain and then just compile the language? This would make code very flexible, transparent, readable – This is idea of the Domain Specific Languages.

## Everybody is doing it!

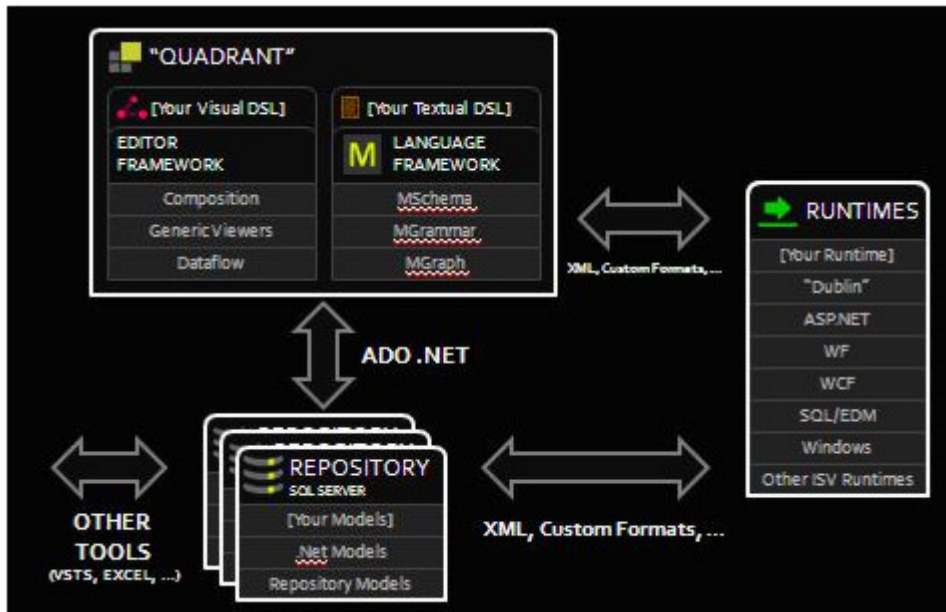
If you have a look at the latest product releases, you see model popping up in one way or another. CAML in MS Office SharePoint, X++ in MS Dynamics, XLANG in MS BizTalk, ...

## So what’s the problem?

Everybody is doing it in his own way! Every specific tool is a one-off implementation: different file formats (xml versus binary formats), inconsistent data formats (INI versus XML), no common storage mechanism (file versus DB ), ...

The solution is Oslo The idea is to provide a modeling platform composed of:

- A Modeling language: called M
  - Define data types
  - Define data instances
  - Describe DSL-s that compile text into data instances
- A SQL-based repository for the models:
  - The repository is the heart of Oslo and is a SQL Server 2008 DB
- A visual data manipulation tool called Quadrant
  - Is not yet released



Important to remember is that during its history the codename Oslo has changed from meaning. Started as a code-name for the umbrella of VS 10, BizTalk 6, BitTalks Services 1, System Center 4 and .NET 4, it finally became the codename for the modeling platform.

To define model MS provided us with Intellipad. It is the emacs for Windows. Besides other things it can be used as textual DSL to create model definitions. An example:

```

File Edit M Mode Window Help
Model.m 100% MMode x
module VideoGameStore
{
    type Product
    {
        ProductId : Integer32 = AutoNumber();
        ProductName : Text where value.Count() < 250;
        Description : Text#250;
        ReleaseDate : Date?;
    } where identity ProductId, unique ProductName;

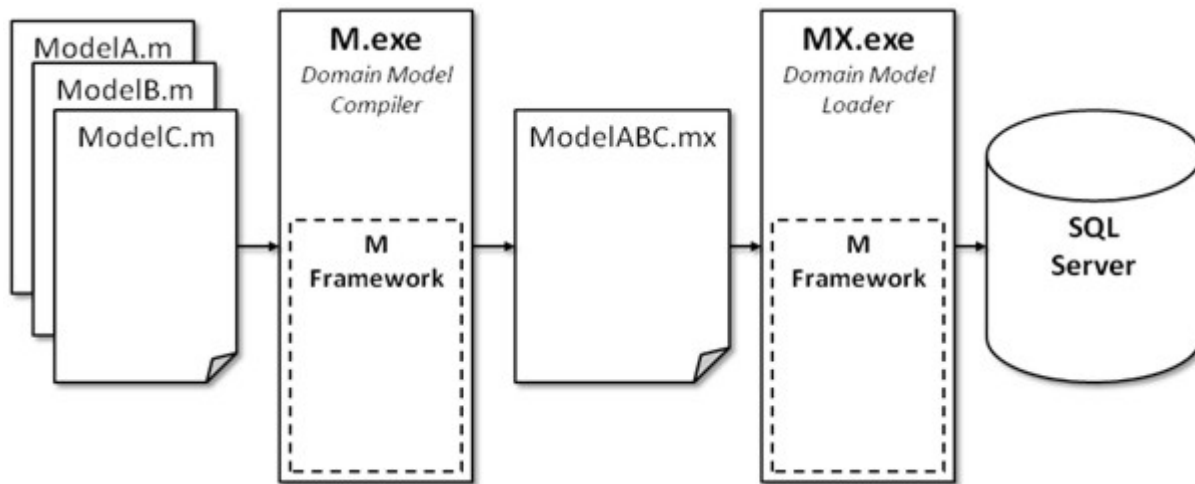
    Products : Product*;

    type Game : Product
    {
        Genre : Text#100;
        Developer : Text#150;
        Publisher : Text#150;
    };

    Games : Game*;
}

```

Models are compiled and then deployed. This double step will become one step in the near future.



Once deployed in SQL Server they can be accessed like any other data in the DB. This is how to model will appear in SQL Server once it is deployed:

Model.m 100% MMode x

```

module VideoGameStore
{
    type Product
    {
        ProductId : Integer;
        ProductName : Text;
        Description : Text;
        ReleaseDate : Date;
    } where identity Product

    Products : Product*;

    type Game : Product
    {
        Genre : Text#100;
        Developer : Text#150;
        Publisher : Text#150;
    };

    Games : Game*;
}

```

mtfm://reachsql/ 100% SqlMode x

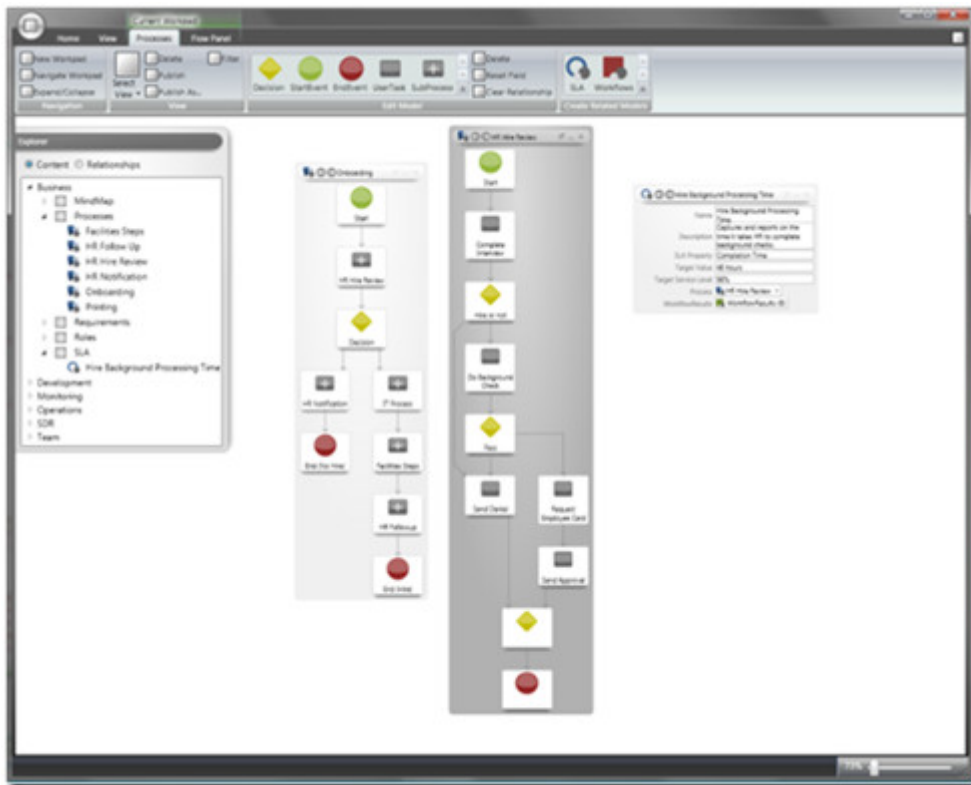
```

create table [VideoGameStore].[Products]
(
    [ProductId] int not null identity,
    [Description] nvarchar(250) not null,
    [ProductName] nvarchar(249) not null,
    [ReleaseDate] date null,
    constraint [PK_Products] primary key clustered ([ProductId]),
    constraint [Unique_Products_ProductName] unique ([ProductName])
);
go

create table [VideoGameStore].[Games]
(
    [ProductId] int not null identity,
    [Description] nvarchar(250) not null,
    [Developer] nvarchar(150) not null,
    [Genre] nvarchar(100) not null,
    [ProductName] nvarchar(249) not null,
    [Publisher] nvarchar(150) not null,
    [ReleaseDate] date null,
    constraint [PK_Games] primary key clustered ([ProductId]),
    constraint [Unique_Games_ProductName] unique ([ProductName])
);

```

Quadrant is the graphical DSL tool and will look like the screenshot below (sorry it is nearly unreadable).



Once defined, the models can be hosted in several runtimes, amongst others .NET applications.