



Bridging Microsoft Oslo and Eclipse EMF

Seminar Modellgetriebene Softwareentwicklung
Abschlusspräsentation

Stanley Hillner

Microsoft Codename „Oslo“

- **Microsofts neuestes Werkzeug für MDSD**
- **Heißt jetzt „MS SQL Server Modeling Services“**

- **3 wesentliche Bestandteile**
 - ▶ Oslo Modeling Language „M“
 - ▶ Oslo Repository
 - ▶ „Quadrant“

- **Oslo verfolgt typische Ziele des MDSD**
 - ▶ Modelle nehmen im gesamten SW-Lebenszyklus zentrale Rolle ein
 - ▶ Steigerung der Produktivität
 - ▶ Steigerung der SW-Qualität

- **Direkte Anbindung an .NET Plattform**

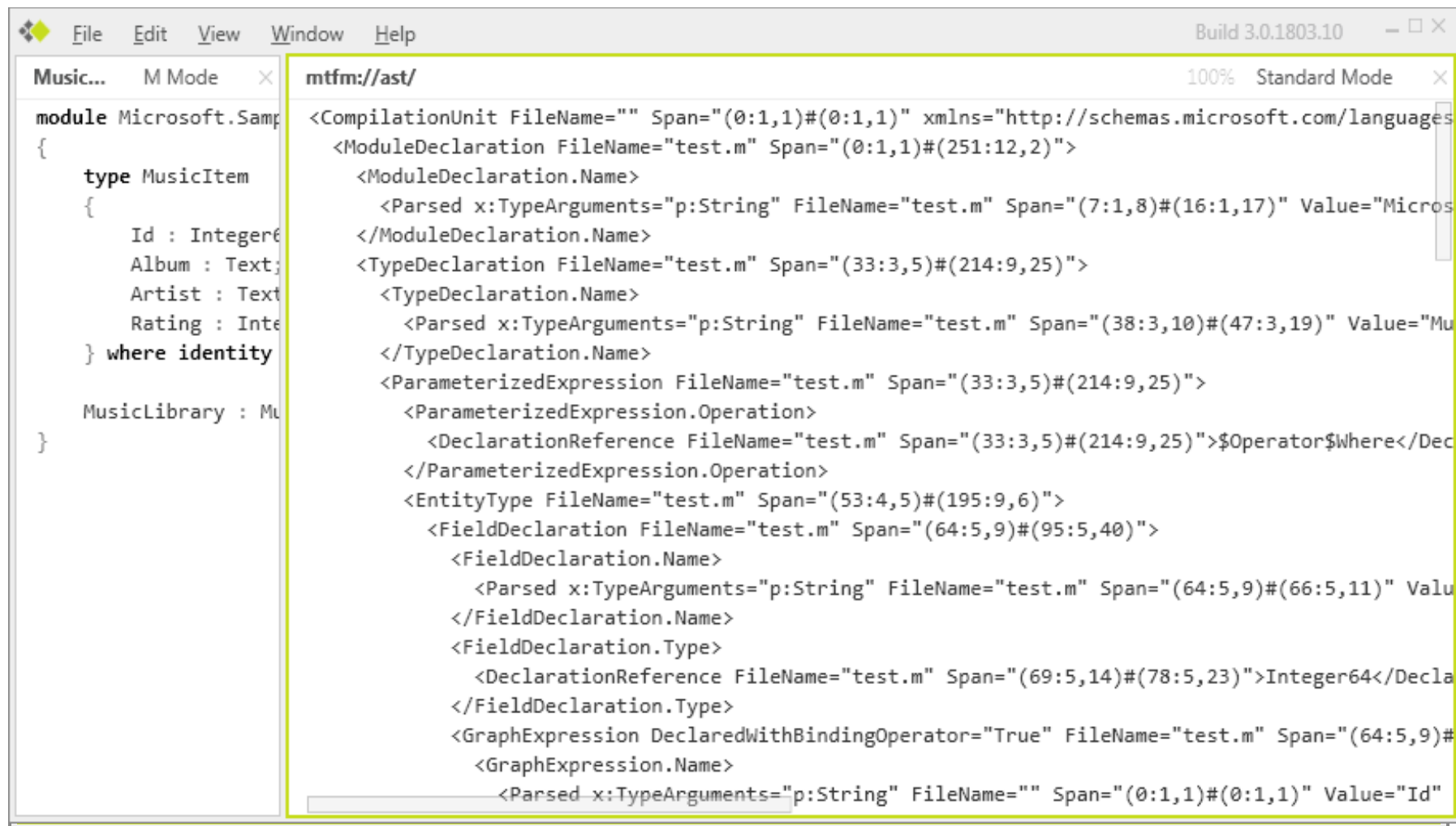
- **Generische Modellierungssprache (3 Bestandteile)**
 - ▶ MSchema: Definition von Datenschemata (Metamodelle)
 - ▶ MGraph: Definition von Dateninstanzen (Modelle)
 - ▶ MGrammar: Definition von Grammatiken für textuelle DSLs

- **Oslo stellt auch Editor für „M“ zur Verfügung („Intellipad“)**
 - ▶ Modellierung aller drei Sprachbestandteile
 - ▶ Verschiedene Sichten auf die Schemata, Modelle, Grammatiken, usw.

- **Beispiel für Modellierung in „M“ unter Nutzung von „Intellipad“**
 - ▶ Musikbibliothek
 - **Alben**
 - **Künstler**
 - **Bewertungen**

Microsoft Codename „Oslo“ – MSchema

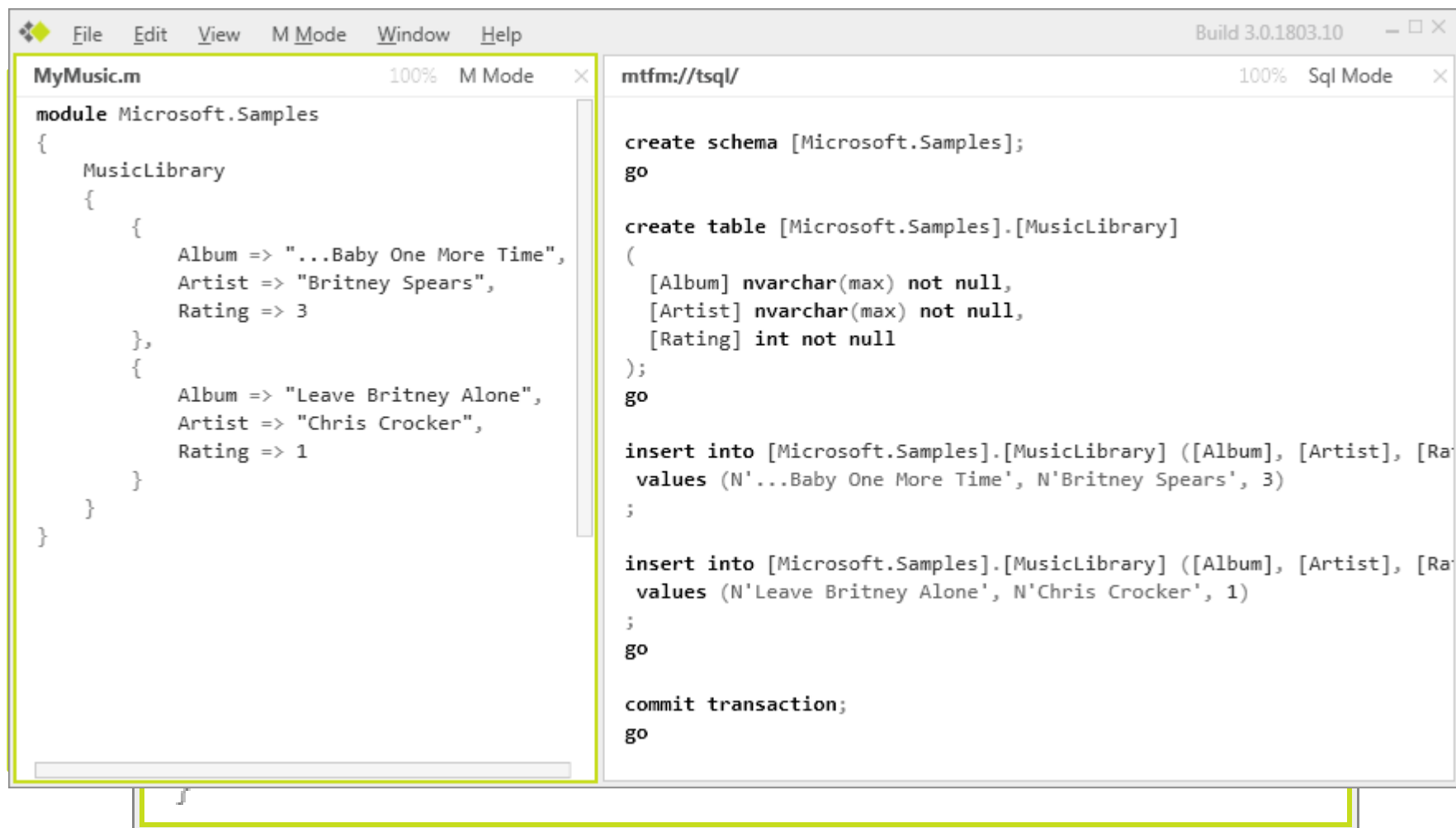
■ Definition des Schemas (Bibliothek, Alben, Interpreten, Ratings)



The image shows a screenshot of the Visual Studio IDE. On the left, a code editor displays F# code for a module named 'Microsoft.Samp'. The code defines a 'MusicItem' type with fields 'Id', 'Album', 'Artist', and 'Rating', and a 'where identity' clause. On the right, the 'mtfm://ast/' window shows the corresponding XML Abstract Syntax Tree (AST) for the code. The AST is a tree structure of XML elements representing the code's syntax, including declarations, type arguments, and expressions. The code in the AST is as follows:

```
<CompilationUnit FileName="" Span="(0:1,1)#(0:1,1)" xmlns="http://schemas.microsoft.com/languages
  <ModuleDeclaration FileName="test.m" Span="(0:1,1)#(251:12,2)">
    <ModuleDeclaration.Name>
      <Parsed x:TypeArguments="p:String" FileName="test.m" Span="(7:1,8)#(16:1,17)" Value="Micros
    </ModuleDeclaration.Name>
    <TypeDeclaration FileName="test.m" Span="(33:3,5)#(214:9,25)">
      <TypeDeclaration.Name>
        <Parsed x:TypeArguments="p:String" FileName="test.m" Span="(38:3,10)#(47:3,19)" Value="Mu
      </TypeDeclaration.Name>
      <ParameterizedExpression FileName="test.m" Span="(33:3,5)#(214:9,25)">
        <ParameterizedExpression.Operation>
          <DeclarationReference FileName="test.m" Span="(33:3,5)#(214:9,25)">$Operator$Where</Dec
        </ParameterizedExpression.Operation>
        <EntityType FileName="test.m" Span="(53:4,5)#(195:9,6)">
          <FieldDeclaration FileName="test.m" Span="(64:5,9)#(95:5,40)">
            <FieldDeclaration.Name>
              <Parsed x:TypeArguments="p:String" FileName="test.m" Span="(64:5,9)#(66:5,11)" Valu
            </FieldDeclaration.Name>
            <FieldDeclaration.Type>
              <DeclarationReference FileName="test.m" Span="(69:5,14)#(78:5,23)">Integer64</Decla
            </FieldDeclaration.Type>
            <GraphExpression DeclaredWithBindingOperator="True" FileName="test.m" Span="(64:5,9)#
              <GraphExpression.Name>
                <Parsed x:TypeArguments="p:String" FileName="" Span="(0:1,1)#(0:1,1)" Value="Id"
```

■ Definition von Dateninstanzen (konform zum Schema)



The screenshot displays two windows from the Microsoft Codename Oslo MGraph application. The left window, titled 'MyMusic.m', shows a module definition in M Mode. The right window, titled 'mtfm://tsql/', shows the corresponding SQL script in Sql Mode. The SQL script creates a schema, a table, and inserts two data rows.

```
MyMusic.m 100% M Mode x
module Microsoft.Samples
{
  MusicLibrary
  {
    {
      Album => "...Baby One More Time",
      Artist => "Britney Spears",
      Rating => 3
    },
    {
      Album => "Leave Britney Alone",
      Artist => "Chris Crocker",
      Rating => 1
    }
  }
}

mtfm://tsql/ 100% Sql Mode x
create schema [Microsoft.Samples];
go

create table [Microsoft.Samples].[MusicLibrary]
(
  [Album] nvarchar(max) not null,
  [Artist] nvarchar(max) not null,
  [Rating] int not null
);
go

insert into [Microsoft.Samples].[MusicLibrary] ([Album], [Artist], [Ra
values (N'...Baby One More Time', N'Britney Spears', 3)
;

insert into [Microsoft.Samples].[MusicLibrary] ([Album], [Artist], [Ra
values (N'Leave Britney Alone', N'Chris Crocker', 1)
;
go

commit transaction;
go
```

Microsoft Codename „Oslo“ – MGrammar

The screenshot displays the Microsoft Codename „Oslo“ MGrammar IDE interface. It features a menu bar with File, Edit, View, DSL, Window, and Help. The main workspace is divided into several panes:

- MusicLibraryLanguage.mg***: Shows the DSL grammar definition in DSL Grammar Mode.

```
module Microsoft.Samples
{
    import Language;
```
- untitled3* ryLanguage.mg Mode**: Shows the input text.

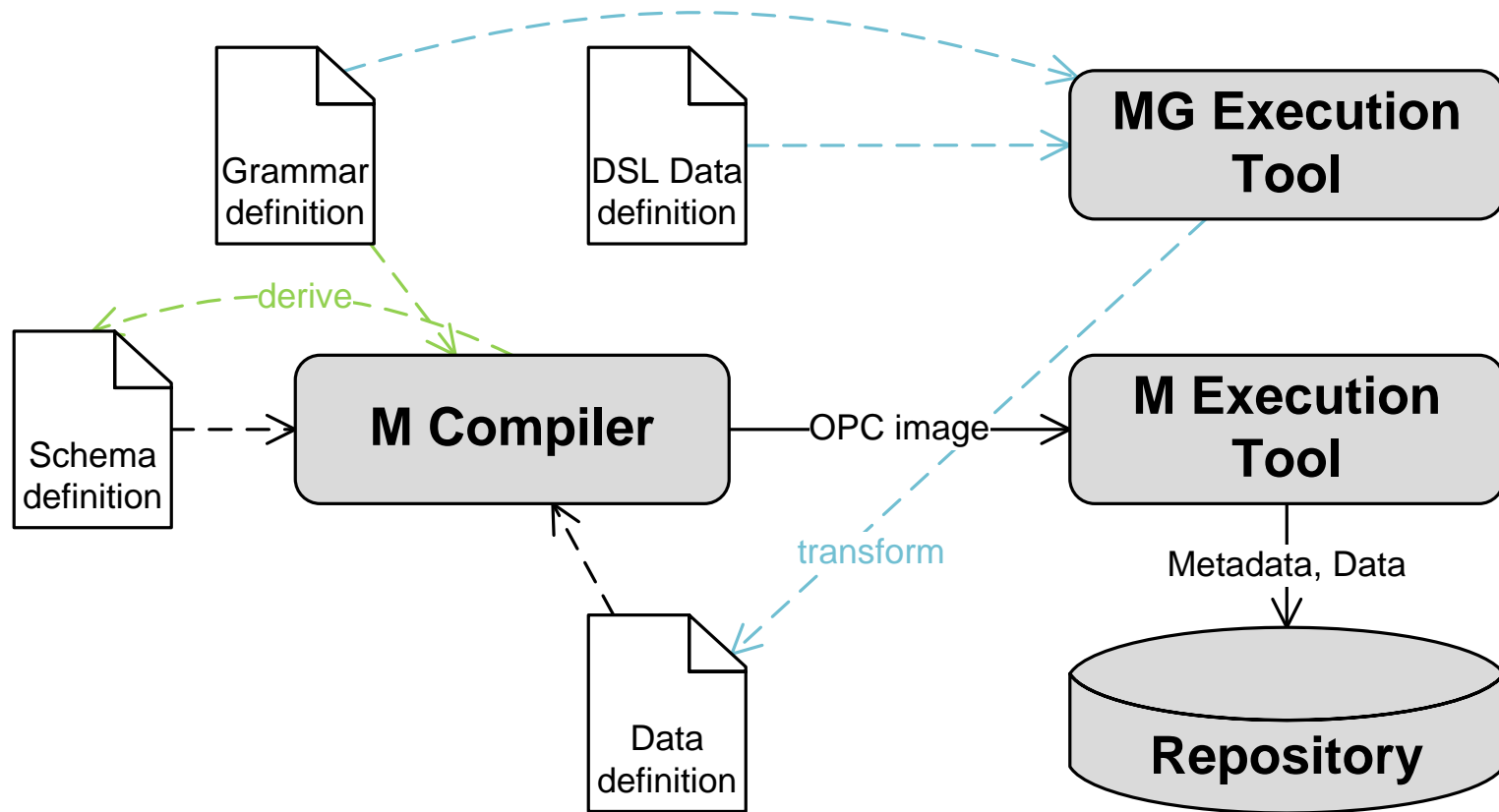
```
"Four" by "Led Zeppelin" is awesome!
"Back in Black" by "AC/DC" is so so.
```
- MusicLibrar... .Grammar Mode**: Shows the generated grammar code.

```
language MusicLibraryLanguage
{
    syntax Main = s:Statement;
    syntax Statement = album;

    @{Classification["Keyw
    token Rating1 = "terr
    @{Classification["Keyw
    token Rating2 = "so so
    @{Classification["Keyw
    token Rating3 = "awesc
```
- untitled3* je.mg Output Mode**: Shows the output of the grammar.

```
MusicLibrary{
{
    Album => "\"Four\"",
    Artist => "\"Led Zeppelin\"",
    Rating => 3
},
{
    Album => "\"Back in Black\"",
    Artist => "\"AC/DC\"",
    Rating => 2
}
}
```
- errors**: An empty pane for error messages.

Microsoft Codename „Oslo“ – Modellierungsprozess



Microsoft Codename „Oslo“ – Quadrant

The screenshot displays the Microsoft Codename Oslo software interface, version 3.0.1803.10. The interface is divided into several panes:

- MusicLibrary - Collection (Table):** A table showing a list of music items. The row for "Back in Black" by AC/DC is highlighted.
- MusicLibrary - Collection (Tree Master/Detail):** A tree view showing the "MusicLibrary (6)" folder expanded, with item 4 highlighted. The detail pane shows the selected item's properties: Album "Back in Black", Artist "AC/DC", and Rating 2.
- 1 Explorer:** A file explorer showing a project structure with folders like Catalog, Microsoft.Repository.Loader, Microsoft.Samples, and Microsoft.Uml2. The "MusicLibrary" folder is highlighted.
- Workspaces - Default Workspace (Inspector):** An inspector pane showing properties for the "Default Workspace", including Name, ShowGridlines, LastActiveSessionId, LeftOffset, and References.

Id	Album	Artist	Rating
1	Leave Britney Alone	Chris Crocker	1
2	...Baby One More Time	Britney Spears	3
3	"Bad"	"Michael Jackson"	1
4	"Back in Black"	"AC/DC"	2
5	"Four"	"Led Zeppelin"	3
6	Thriller	Michael Jackson	2

Workspaces - Default Workspace Inspector

- Properties
 - Name: Default Workspace
 - ShowGridlines:
 - LastActiveSessionId: 4
 - LeftOffset: 212,172100544958
- References
 - Users: 1
 - Workpads: (4)
- Workpad
 - Workpad Query: Not Applicable
 - Selection Query: Not Applicable
 - Viewer Configuration: Not Applicable
 - No active workpad.

Eclipse Modeling Framework

- **Open-Source Java-Framework zur Erstellung von Anwendungen**
- **Arbeitet mit strukturierten Modellen**

- **Wesentliche Bestandteile**
 - ▶ EMF Core → Ecore als Metametamodell (MOF-Implementierung)
 - ▶ EMF.Edit → Erstellung von Editoren für Metamodelle
 - ▶ EMF.Codegen → Generierung von Code für (Meta)Modelle

- **EMF ist leicht und schnell zu erlernen**
 - ▶ Trotzdem sehr mächtig

- **Große Akzeptanz und Verbreitung**
 - ▶ Ist Open-Source
 - ▶ Vielzahl an Werkzeugen vorhanden, für Aufgaben des MDSD

Eclipse Modeling Framework – Komponenten

EMF Core

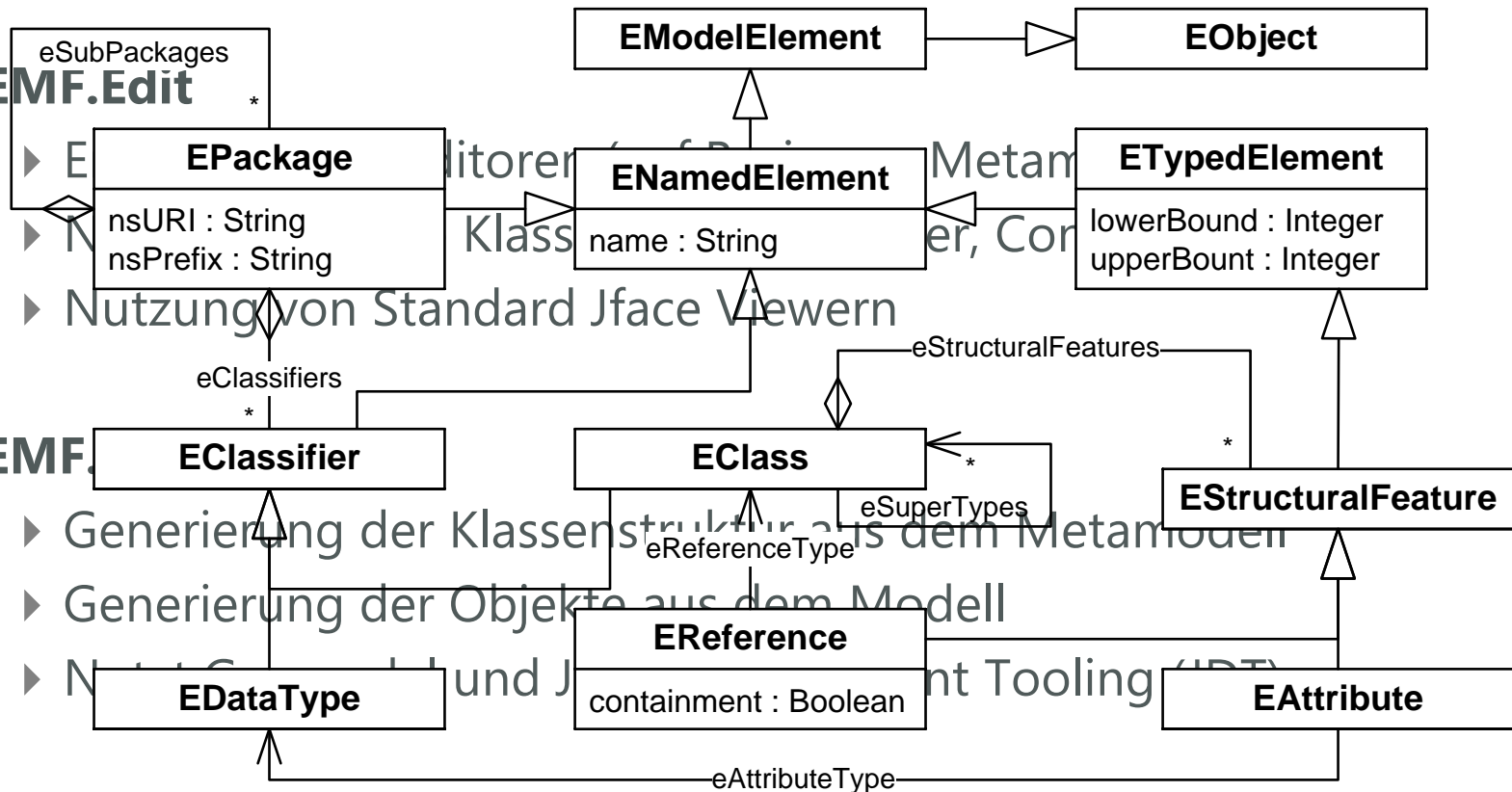
- ▶ Ecore als Metametamodell
- ▶ Implementiert MOF-Spezifikation der OMG
- ▶ Aufbau siehe Diagramm

EMF.Edit

- ▶ EPackage
- ▶ Nutzung von Standard Jface Viewern

EMF

- ▶ Generierung der Klassenstruktur aus dem Metamodell
- ▶ Generierung der Objekte aus dem Modell
- ▶ Nutzung von JFace und JDT



Eclipse Modeling Framework – (Meta)Modellierung

■ Erstellung eines Metamodells (konform zu Ecore)

- ▶ Nutzung des Ecore model editor, graphisch, aus XSD, Rose-Modell

The screenshot shows the Eclipse IDE with the Ecore model editor for `MusicLibrary.ecore`. The editor displays the following grammar:

```
grammar org.eclipse.xtext.common.Terminals
import "platform:/resource/Oslo-EMF/model/MusicLibrary.ecore"
platform:/resource/MusicLibrary/MyMusic.osl
MusicLibrary Lib(MyMusicItems += MusicItem)*;
    MusicItem 0
MusicItem : Album = STRING "by" Artist = STRING "is" Rating = Rating (Crock);
    MusicItem 1
enum Rating : terrible|awful="1"
so_so="2"
awesome="3";
```

The Properties window on the right shows the instance values for the `MusicItem` class:

Property	Value
Album	Leave Britney Alone
Artist	Chris Crock
Id	1
Rating	1

■ Erzeugung einer DSL z.B. durch Grammatik in XText

Mögliche Vorteile einer Oslo-EMF Brücke

■ EMF ist bereits Hub für Bridging von MDSD Tools

- ▶ Bereits viele Brücken vorhanden (Visio, Aris, MetaEdit, DSL Tools, GME, usw.)
- ▶ Verwendung existierender Brücken zur Verringerung des Aufwandes (transitive Brücken)
- ▶ Dadurch höhere Effizienz und Qualität

■ Bereitstellung von Funktionalität

- ▶ EMF ist Open-Source → große Community → Viele Tools
- ▶ Passendes Werkzeug für (fast) jede Aufgabe
- ▶ Oslo hat auch Vorzüge (Repository, Quadrant)

■ Wechsel ermöglichen

- ▶ EMF→Oslo
- ▶ Oslo→EMF

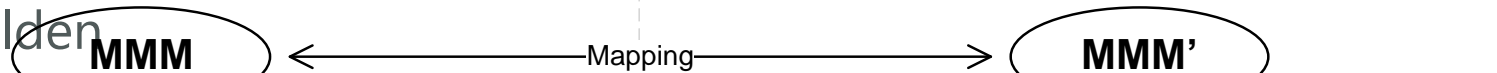
Das M3-level based Bridging pattern

- Setzt Vorhandensein einer Metamodell-Hierarchie voraus

- **Mapping auf M3-Ebene**

- ▶ Konstrukte des einen auf Konstrukte des anderen Werkzeugraums abbilden

M3

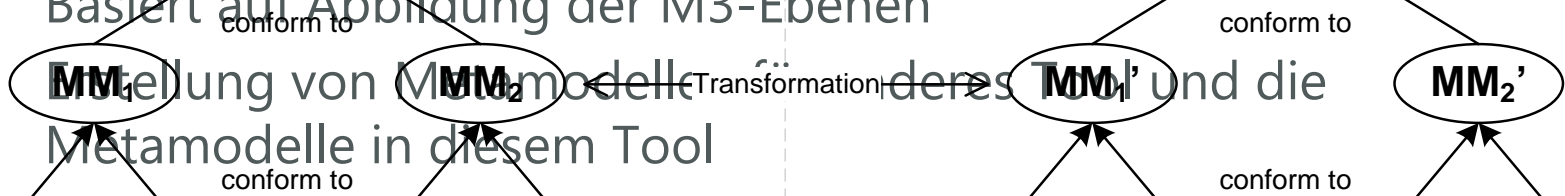


- **Transformation auf M2-Ebene**

- ▶ Basiert auf Abbildung der M3-Ebenen

M2

- ▶ Erstellung von Metamodellen MM_1 und MM_2 aus MMM und die Metamodelle in diesem Tool



- **Transformation auf M1-Ebene**

M1

- ▶ Konkrete Instanzen (Modelle) werden abhängig vom Metamodell transformiert

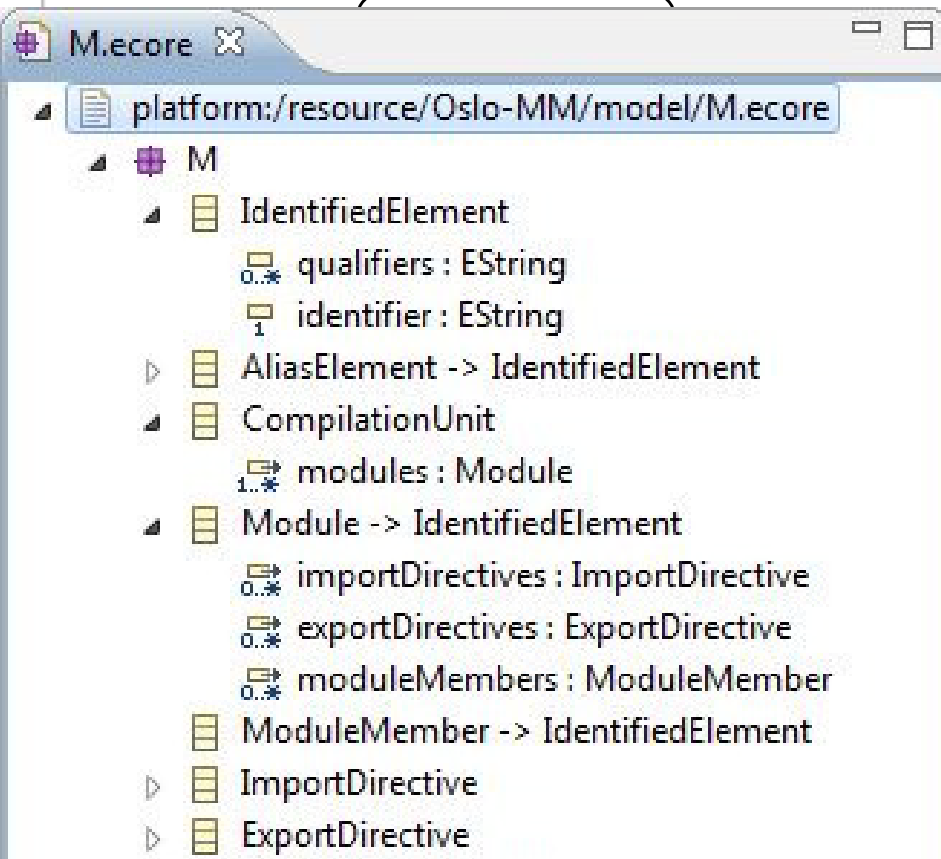


- **Problem:**

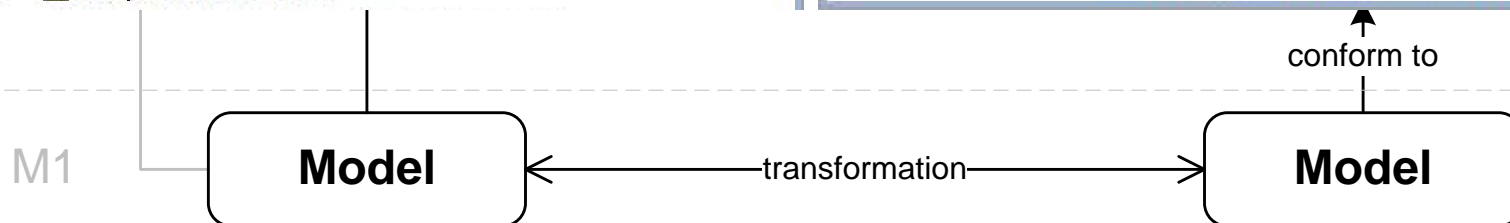
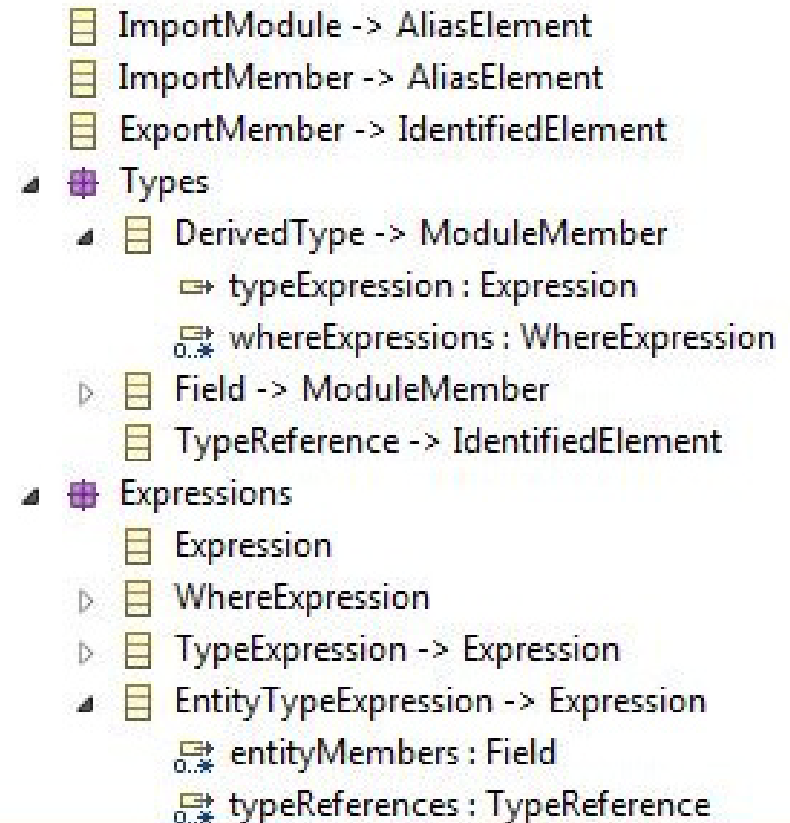
- ▶ Metamodelle bzw. Modelle in generischem Format

Bridging-Ansatz (M3B)

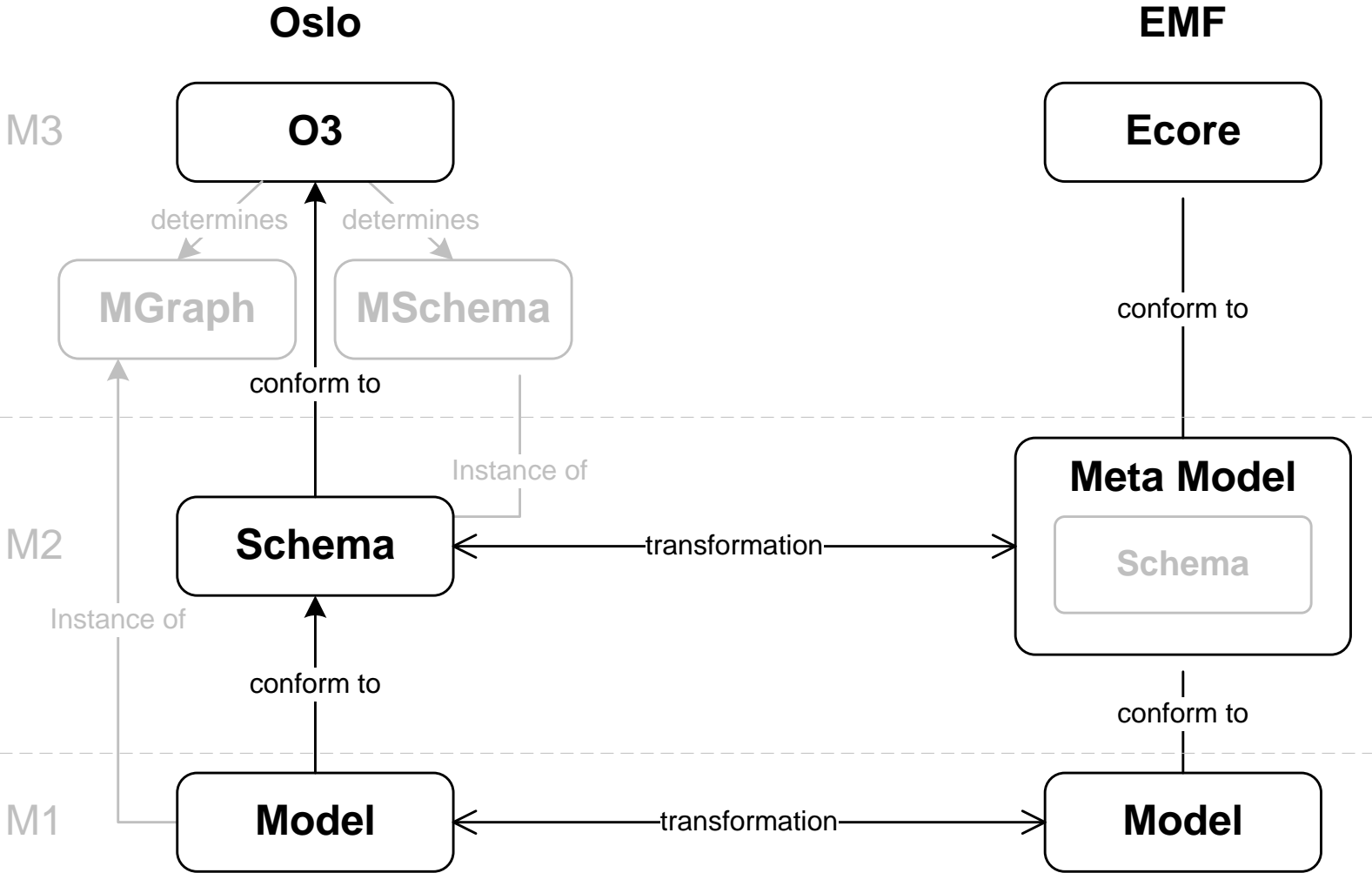
Oslo



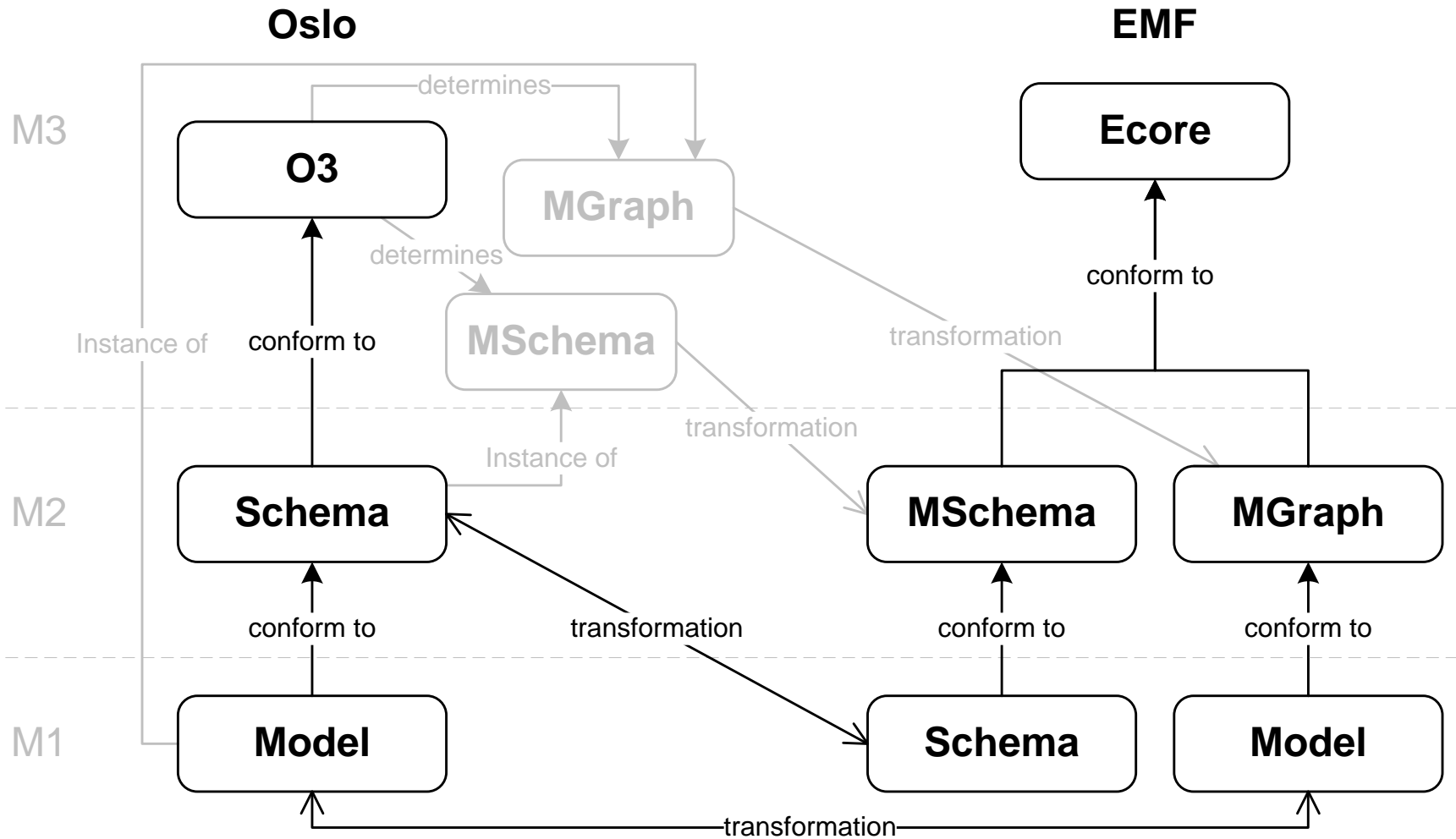
EMF



Bridging-Ansatz (M2B)



Bridging-Ansatz (Vertikale Transformationen)



Fragen?

