

Eclipse Modeling Framework mehr als nur modellieren

Felix Suda, M.Sc.

mgm tp

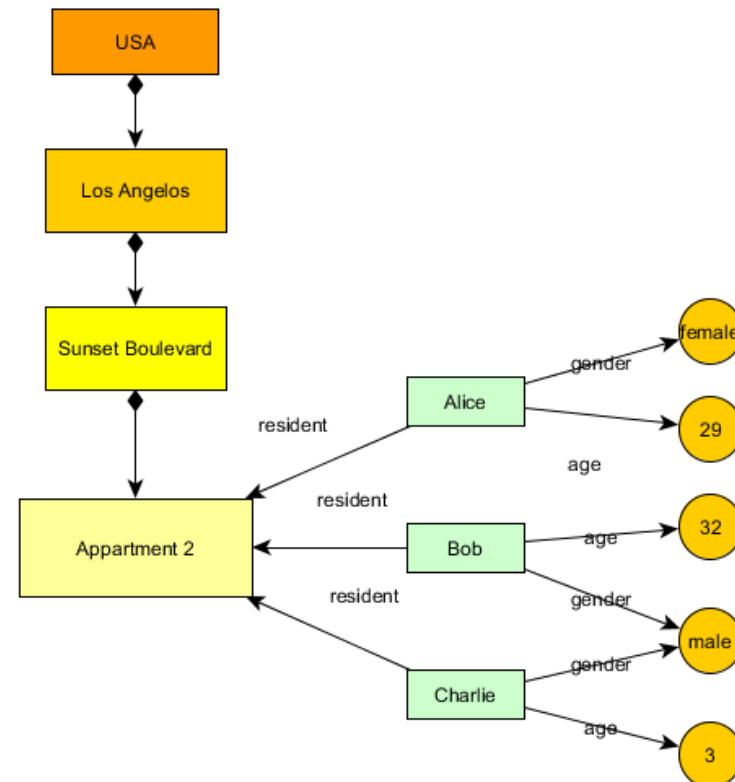
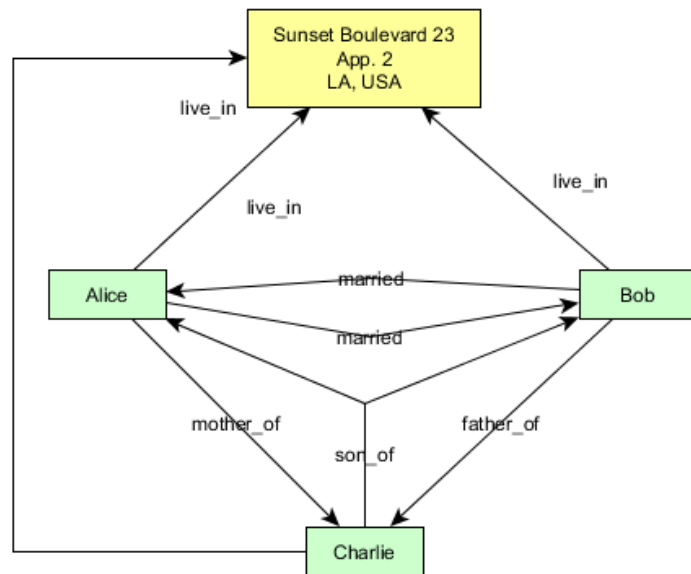
- [[felix.suda@](mailto:felix.suda@mgm-tp.com) | [https://](https://mgm-tp.com)] mgm-tp.com
 - Mittelständiges Unternehmen mit 18 Niederlassungen in 6 Ländern
 - IT-Projekte im eCommerce und eGovernment-Sektor
 - A12 Low Code Plattform (-> geta12.com)
 - Möglichkeiten für Praktika, Abschlussarbeiten und weitere Zusammenarbeit
- Bis Ende 2020 bei ScopeSet Technology
 - Entwicklung von modellbasierten Software-Tools im Bereich Automobil-, Luft- sowie Raumfahrtindustrie
 - MDSD getriebene Arbeitsweise
 - Entwicklungs- und Laufzeitplattform meist Technologien der Eclipse Foundation

Agenda

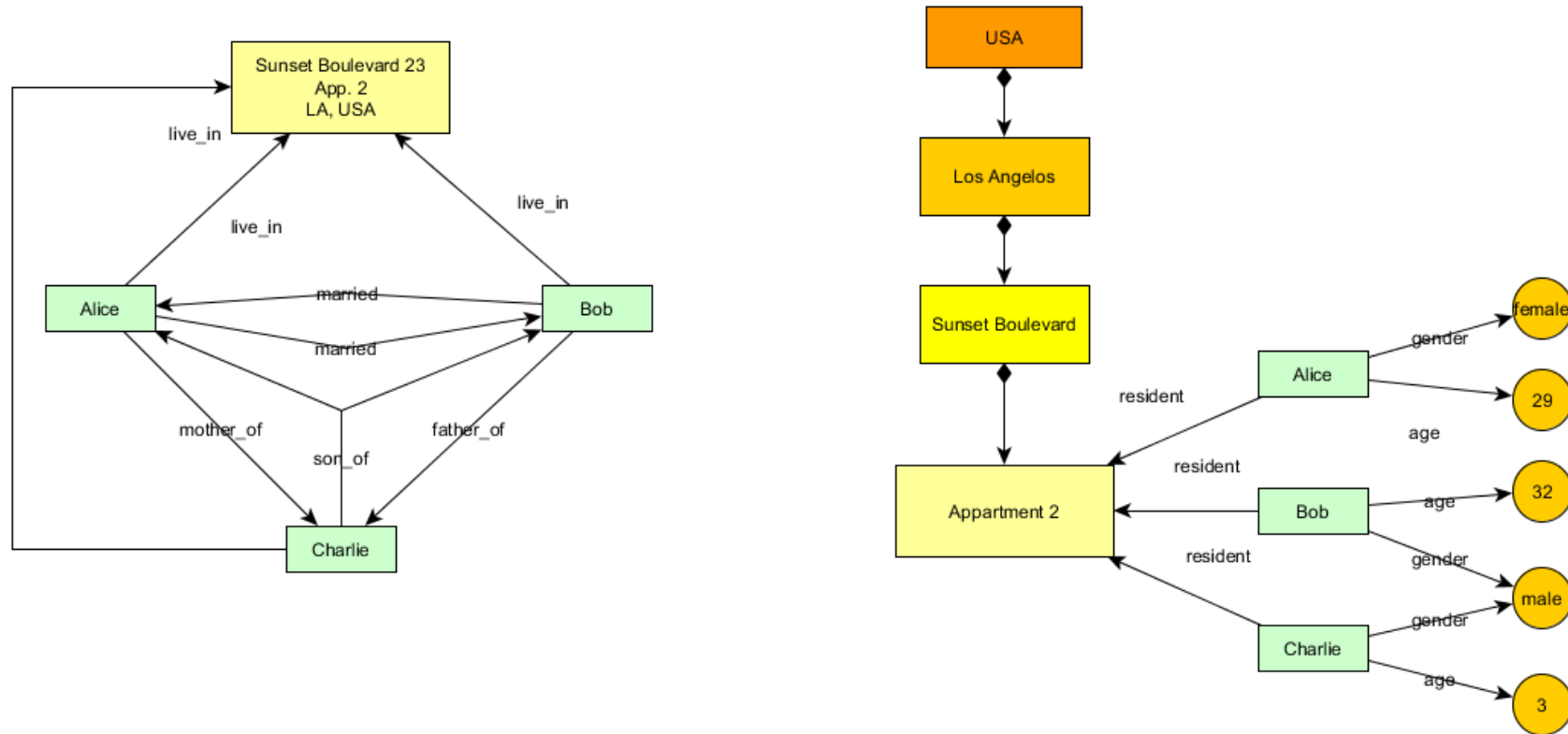
1. Modelle, Metamodelle, überall Modelle
2. Eclipse Modeling Framework (EMF)
3. Hands on EMF
4. Ausblick

1) Modelle, Metamodelle, überall Modelle

- Modell ist ein Ausschnitt der Realität
- Präziser: auf das Wesentliche beschränkte Abstraktion einer Domäne
- Gleicher Sachverhalt kann in unterschiedlichen Domänen verschieden modelliert sein



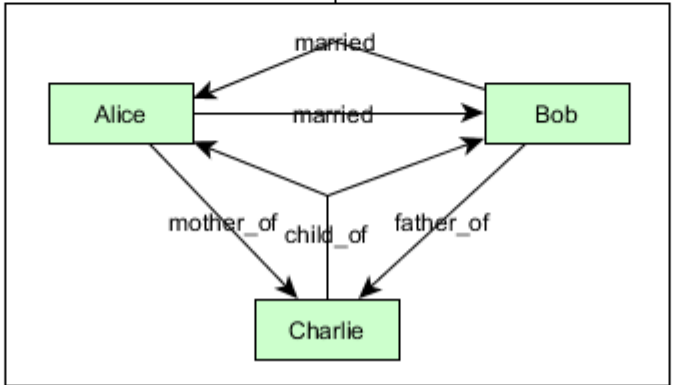
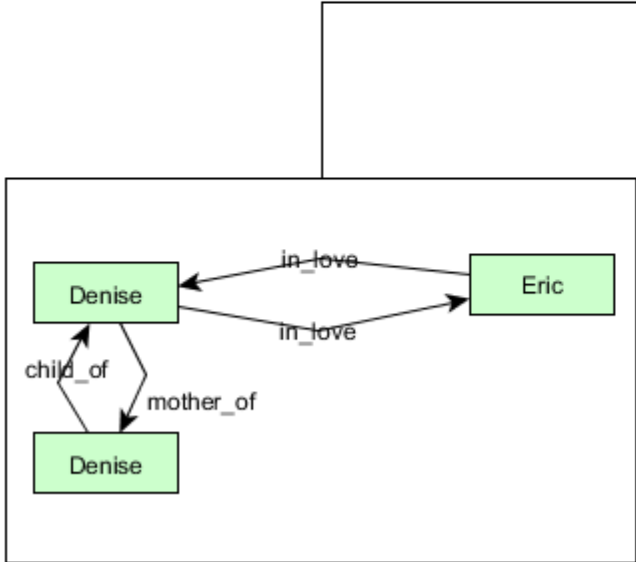
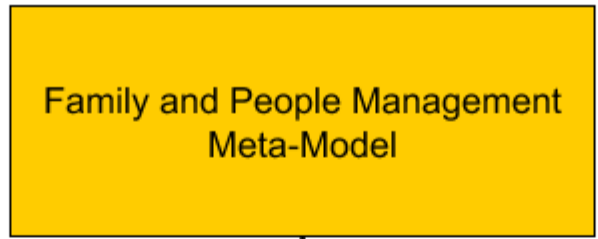
1) Modelle, Metamodelle, überall Modelle



- Auf speziellen Use Case zugeschnitten
 - von nicht relevanten Details abgegrenzt
 - Details können zusammengefasst oder zerlegt werden, je nach Relevanz für den Use Case

1) Modelle, Metamodelle, überall Modelle

- Formale Diskussionsgrundlage
- Interdisziplinäre Kommunikation
- Möglichkeiten der Verifikation und Validierung
- (meist) Verständlich



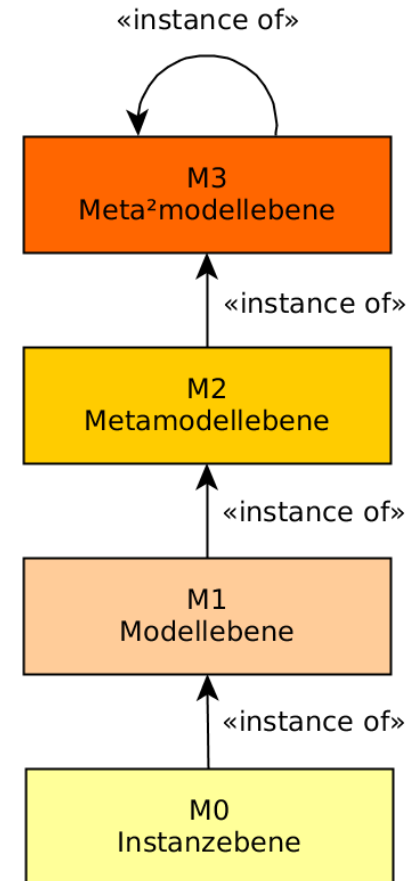
```
person Frank
  age: 42
  gender: male
  best_friend: #person/George

person George
  age: 42
  gender: male
  best_friend: #person/Frank
```

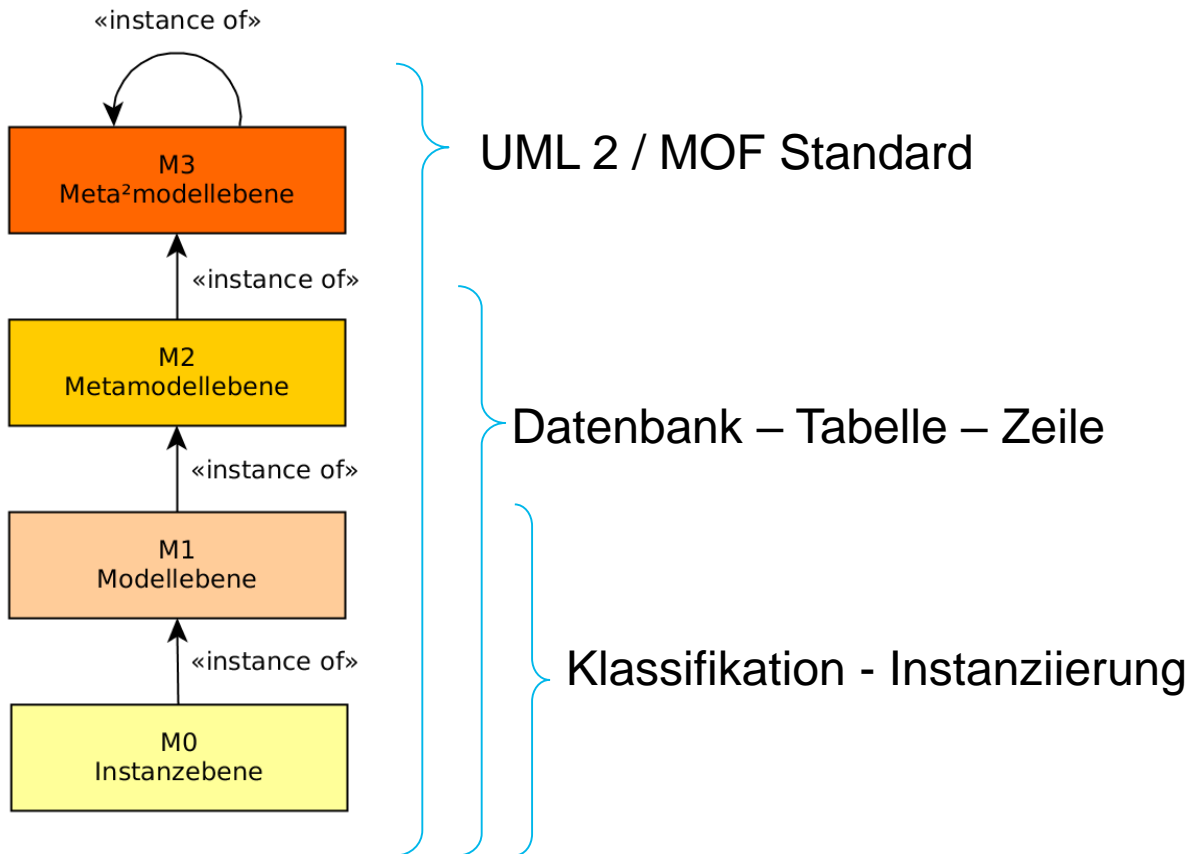
Modell → MetaModell

1) Modelle, Metamodelle, überall Modelle

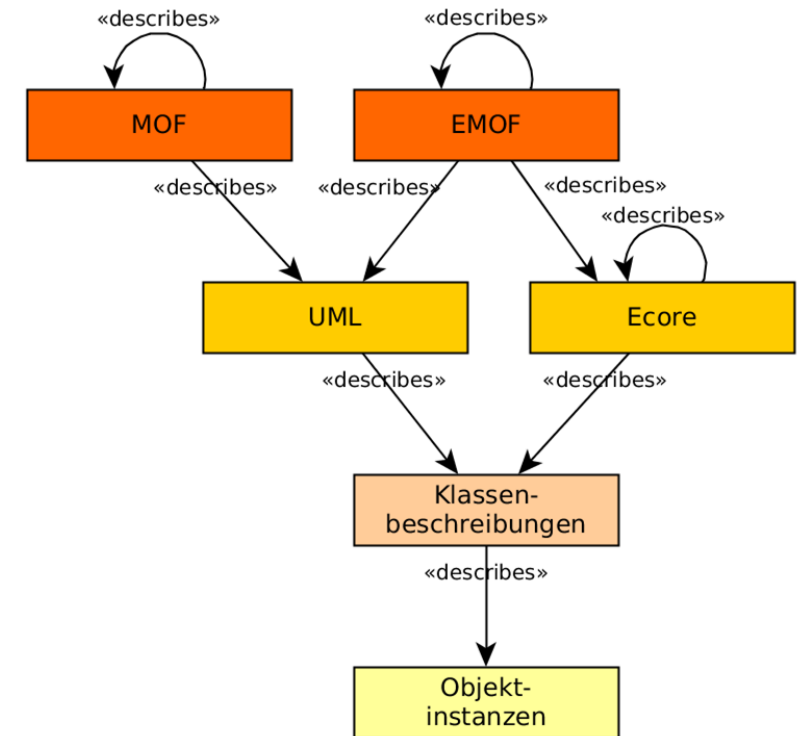
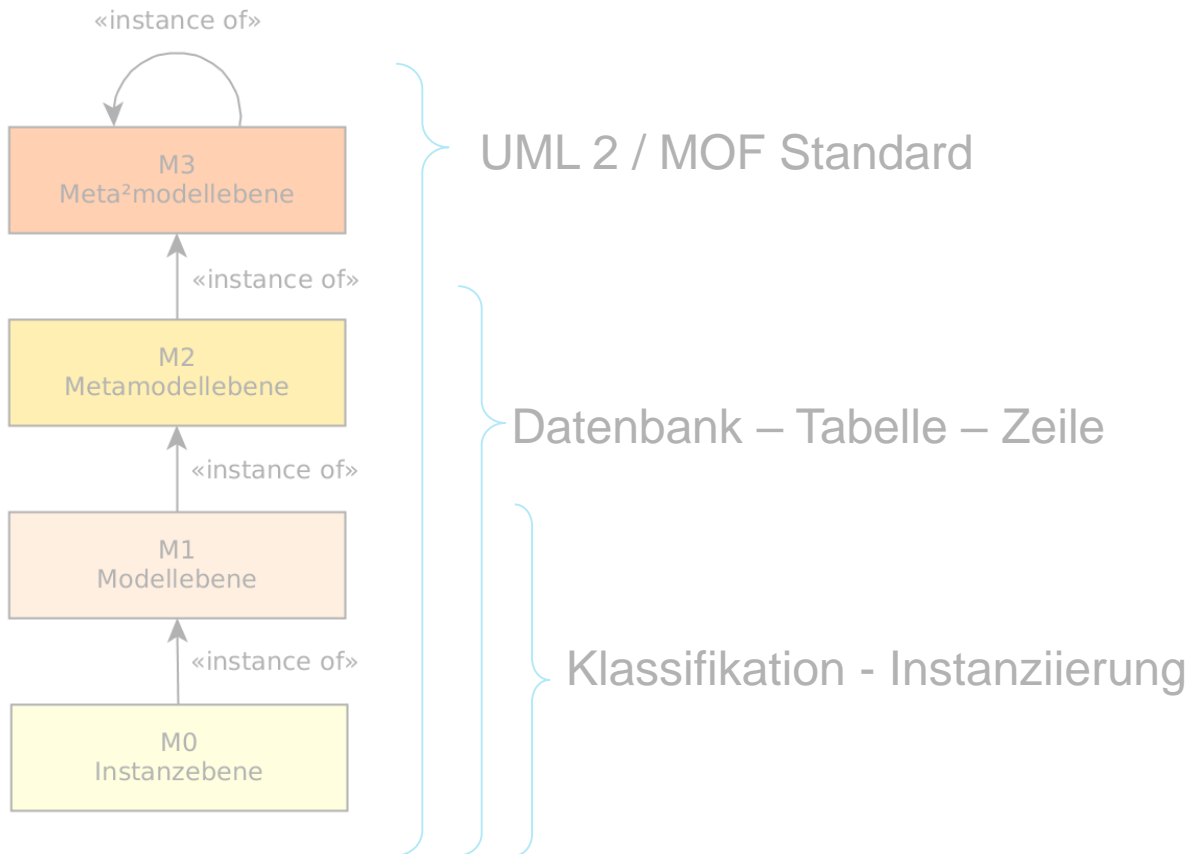
- meta (gr.): über
- Abstraktion aller Modelle, die die gleiche Domäne (mit gleichem Ziel) beschreiben
 - Modell ist eine Instanz des dazugehörigen Metamodells
- Übergeordnete Ebene beschreibt (abstrahiert) darunter liegende Ebene
- OMG empfiehlt vier Modellebenen



1) Modelle, Metamodelle, überall Modelle



1) Modelle, Metamodelle, überall Modelle



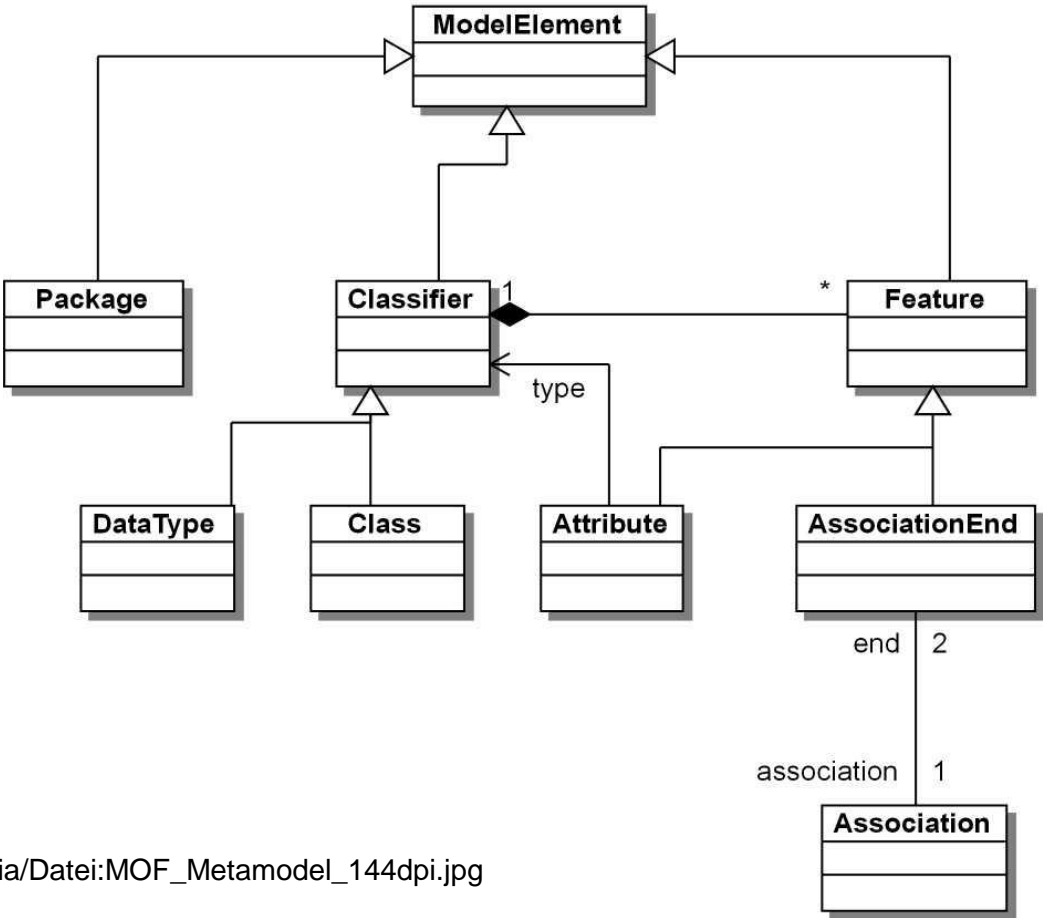
2) EMF - Eclipse Modeling Framework

- <https://www.eclipse.org/modeling/emf/>
- Anfang 2000 in der Eclipse Foundation entwickelt
- Mittlerweile in Eclipse hochintegriert
- Wird als de-facto Implementierung der OMG EMOF angesehen

- EMF Metamodel: Ecore
- Eclipse-basiertes (Daten-) Modellierungsframework
- Basis: adaptiertes UML-Klassenmodell und Java

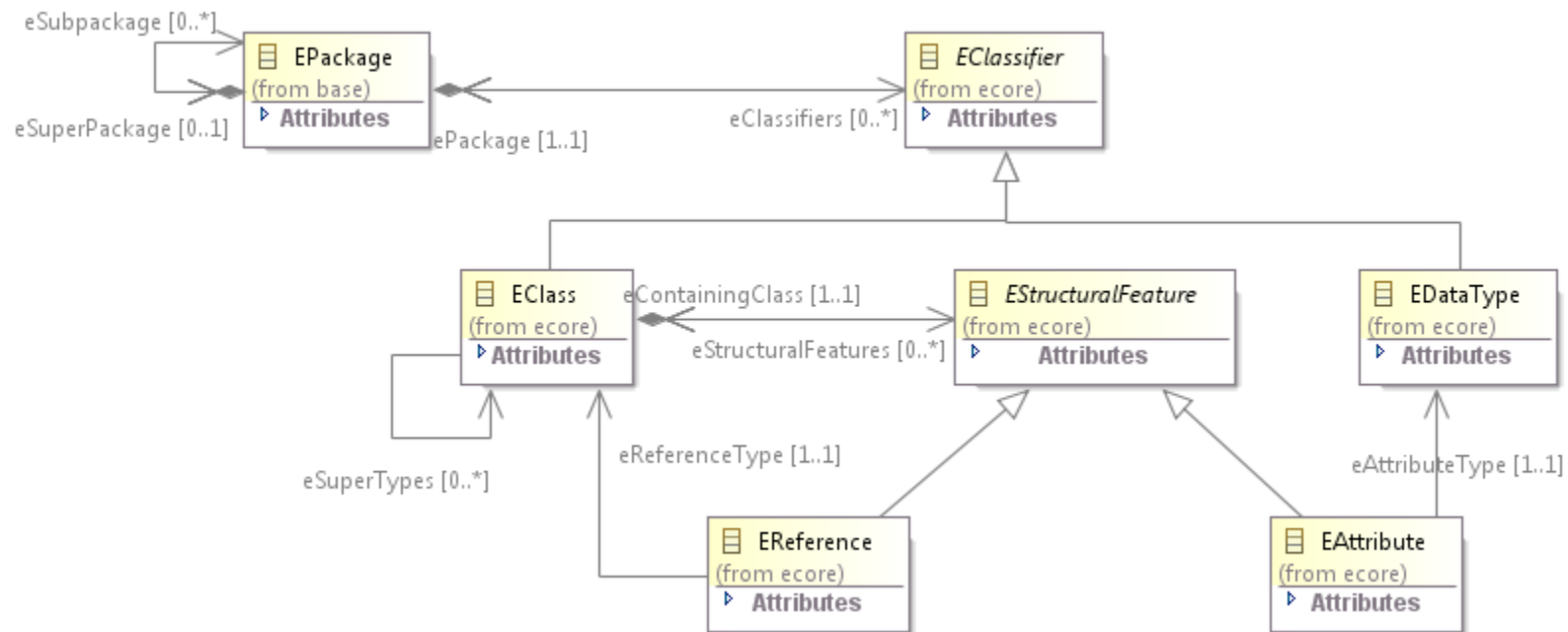
2) EMF – MOF Modell

MOF Metamodel (vereinfacht)

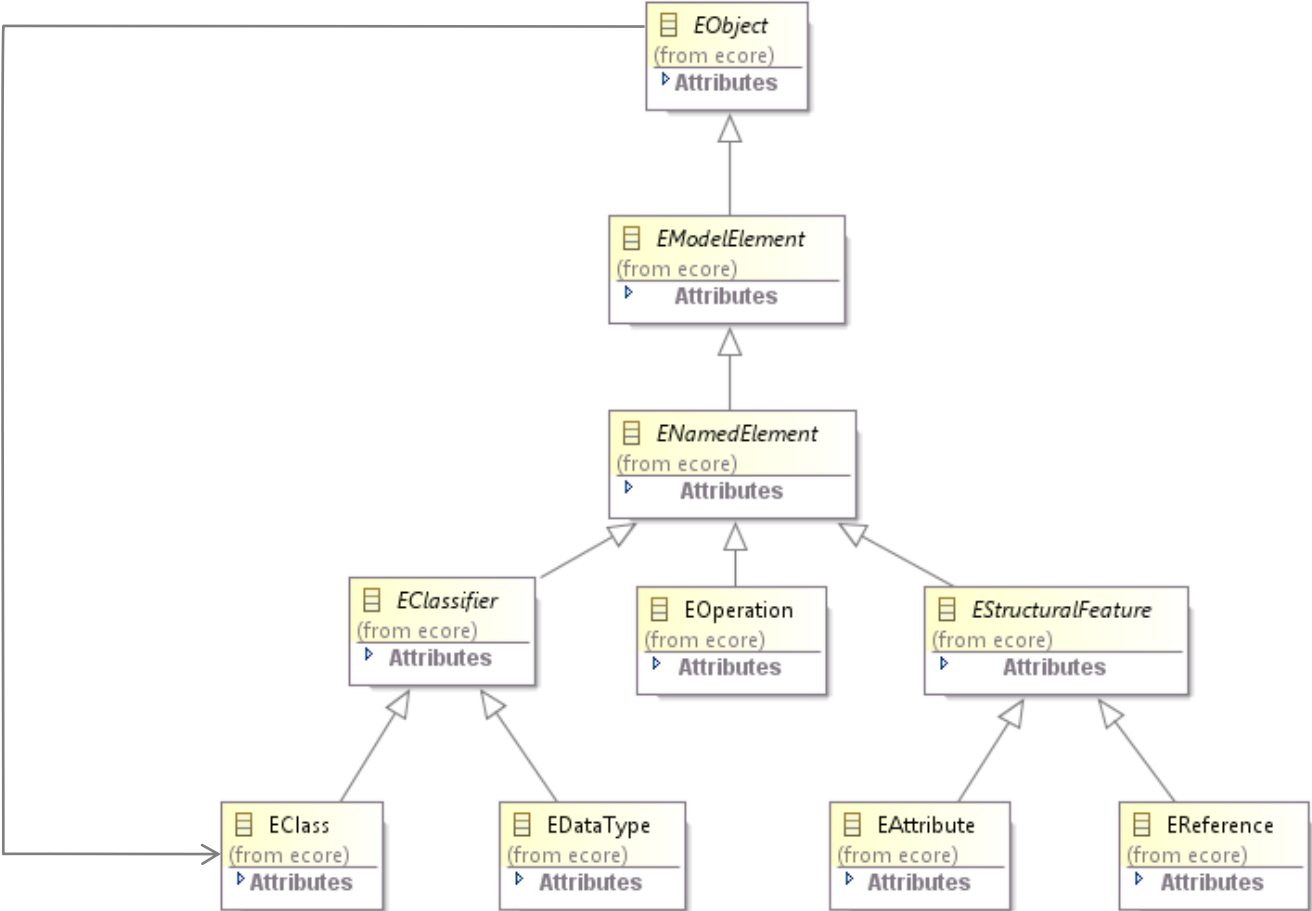


https://de.wikipedia.org/wiki/Meta_Object_Facility#/media/Datei:MOF_Metamodel_144dpi.jpg

2) EMF - Metamodell



2) EMF - Metamodell





One model to rule them all

2) Eclipse Modeling Framework

Mehr Vorteile?

- Validierungsframework (per EOperation)
 - Einfaches, aber mächtiges Persistierungsframework
 - Notification- & Command Framework
 - Generierung von Edit und Editor Code (MVC)
 - Umfangreiche Integration in das Eclipse Ökosystem
 - Persistenz: CDO, EMFStore, Teneo
 - Graphisches (Editoren, UI): GMF, Sirius; Parsley, EMF Forms
 - Model Transformationen (M2M, M2T): QVTo; Xtext, Acceleo
 - Laufzeitumgebung: Eclipse Client Platform (ECP), Remote Application Platform (RAP)
- } built-in

3) Hands on EMF

- Eclipse Modeling Tools
 - <https://www.eclipse.org/downloads/packages/>
- Beispielmodell:
 - Verzeichnis mit Personen
 - Personen können Eltern-Kind-Beziehungen haben

Ausblick

- Modellgetriebene Entwicklung ist sehr mächtig
- Nicht nur wird ein Framework an die Hand gegeben, mit dem sich allerlei Aufgaben leichter erfüllen lassen
- Modelle können auch transformiert werden und letztendlich zu Code oder anderen Artefakten transformiert werden
 - Z.B. OpenAPI-YML Spec parsen (benötigt yml Model) und diese durch geeignete Model Transformationen zu einem REST-API-Modell (noch Platform agnostig) und letztlich per Code Generierung zu z.B. Spring-REST-Controllern oder ASP.NET REST-API

Danke!

Fragen?