

Netzwerkorientierte Speicherung von Metamodellen und Modellen

Seminar Model Driven Software Engineering
Betreuer Martin Gebauer
26.01.2009

Inhalt

- Einführung
 - Model Driven Software Development
 - Eclipse Modeling Framework
 - Neo
- Modellpersistenz in Neo
 - Anwendungsbeispiel
 - Modellorganisation im Nodespace
 - Modellversionierung
 - Subreferences
- Zusammenfassung

Einführung MDSD

Modell

„Ein Modell ist eine abstrakte Repräsentation von Struktur, Funktion oder Verhalten eines Systems.“ [SV05, Seite 20]

- Verschiedene Sichten auf ein System
- Metamodell
„Modell eines Modells“
- Modellierungssprache
- Modelltyp
 - Computation Independent Model (CIM)
 - Platform Independent Model (PIM)
 - Platform Specific Model (PSM)
- Modelltransformation
- Generierung aus Modellen

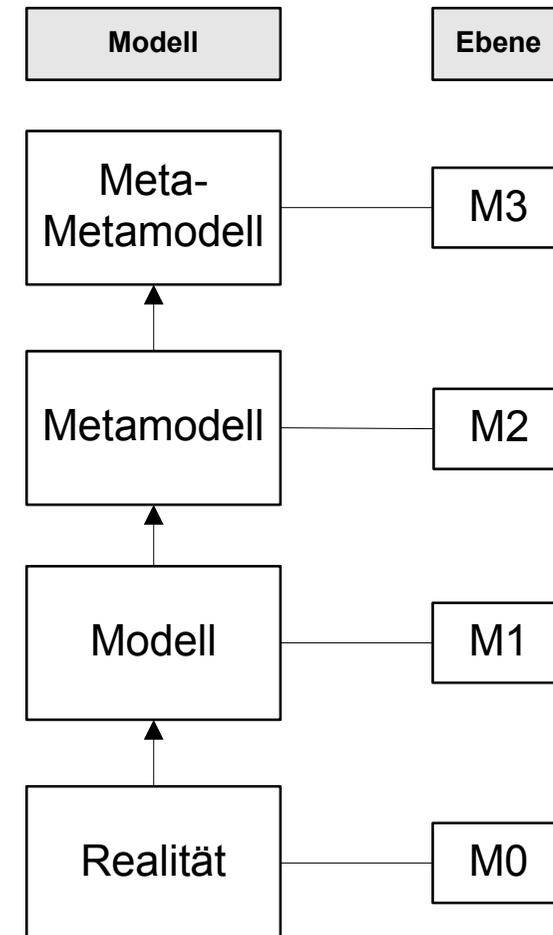


Abbildung: Modellhierarchie
(vgl. auch [PT07, Seite 53])

Eclipse Modeling Framework

- Framework zur Erzeugung von Modellen
- Integration in Eclipse
- Klassenbasierte Modellierung
- Implementierung von Essential MOF 2.0
- Eigenes Metamodell namens ECore
- Modellerzeugung mit Editor oder per Import
- Modelle im XMI-Format gespeichert

Eclipse Modeling Framework

- Objekte
- Referenzen
- Operationen
- Attribute
- Einfache Constraints

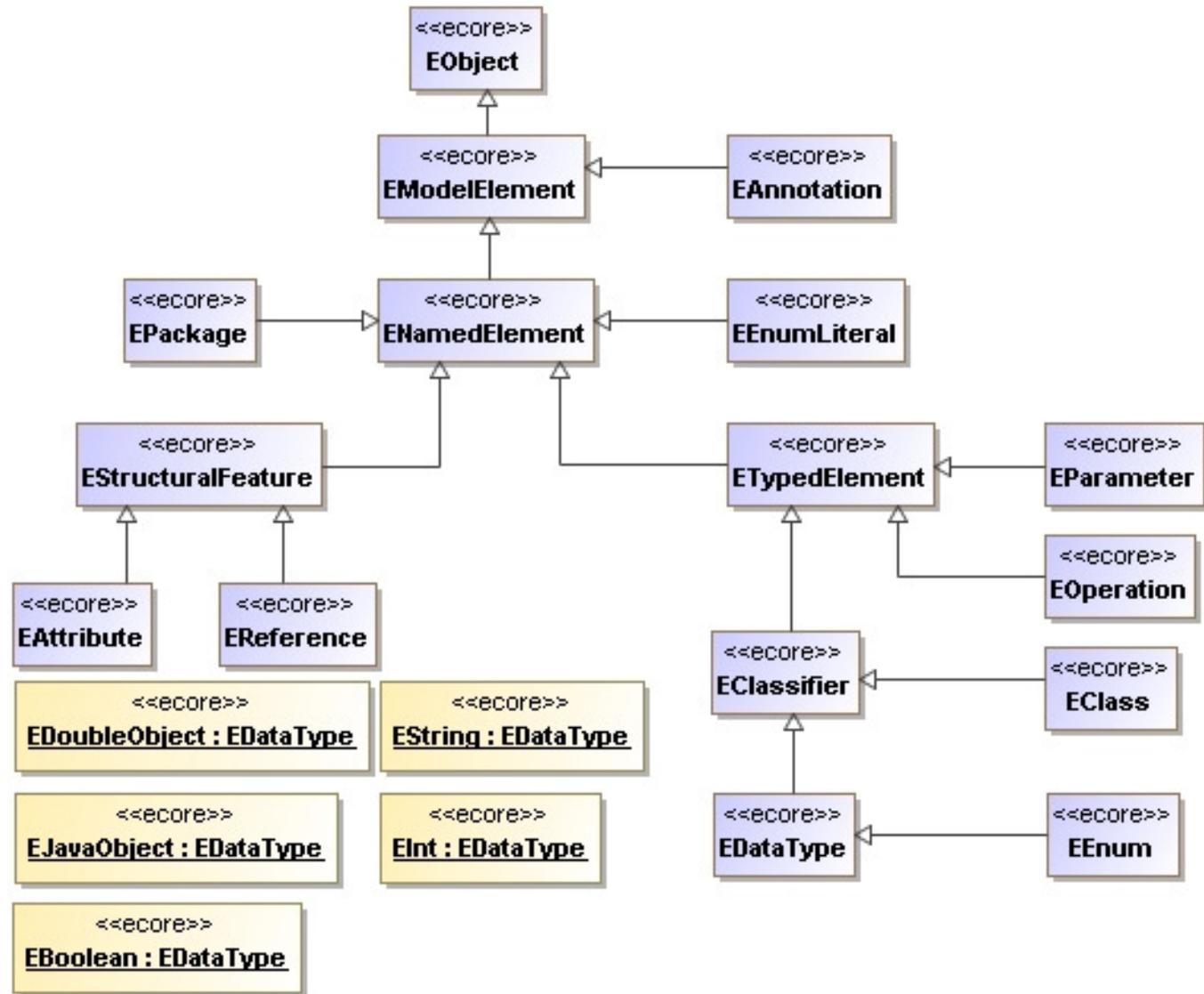


Abbildung: Das
ECore-
Objektmodell

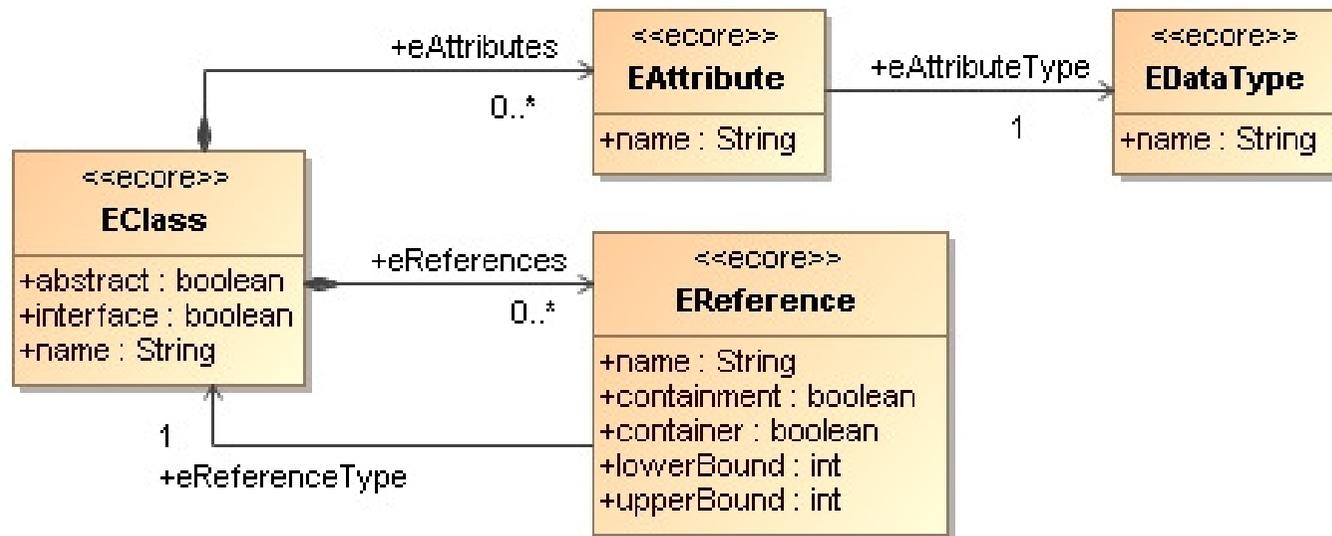


Abbildung: Auszug aus dem ECore-Metamodell [basierend auf [BMS+04, Seite 16]

Neo

- Neo Technology
- Open-Source-Projekt Neo4j
- Embedded Datenbank
- Weitere Teilprojekte z.B. Graph-Algo

- Graphentheorie (Bäume, Netze)
- Geeignet zur Speicherung von semi-strukturierten Daten

- Navigation mittels Objektmethoden oder Traversen

Neo

- Terminologie für Neo
 - Node (Identifikation über Identifier)
 - Relation
 - Property
 - Nodespace

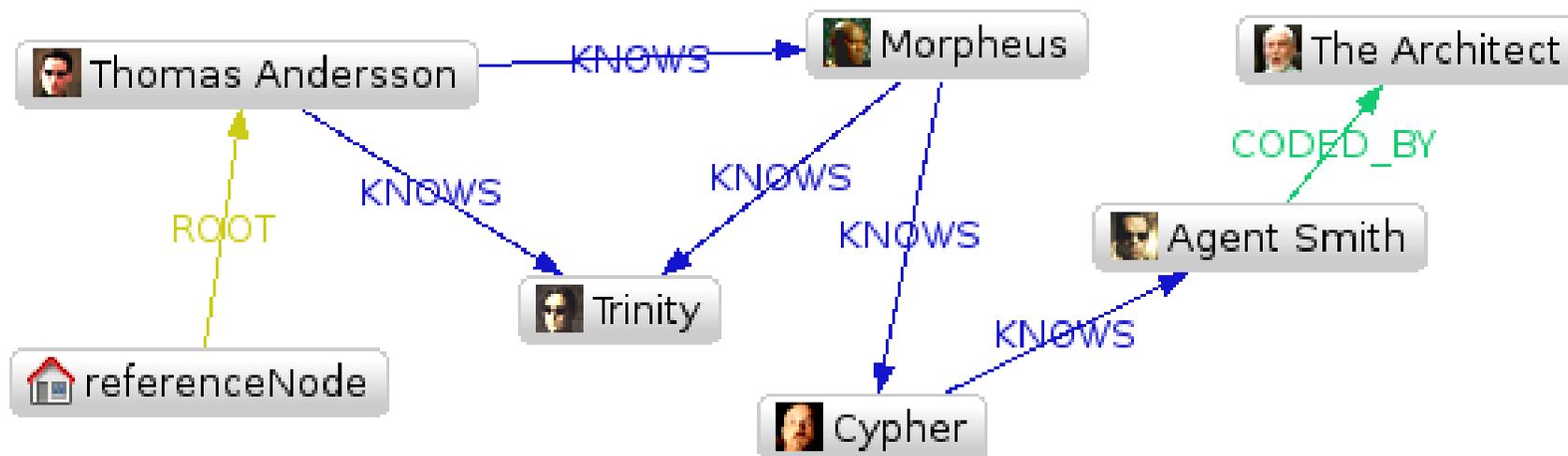


Abbildung: Matrix [Wike]

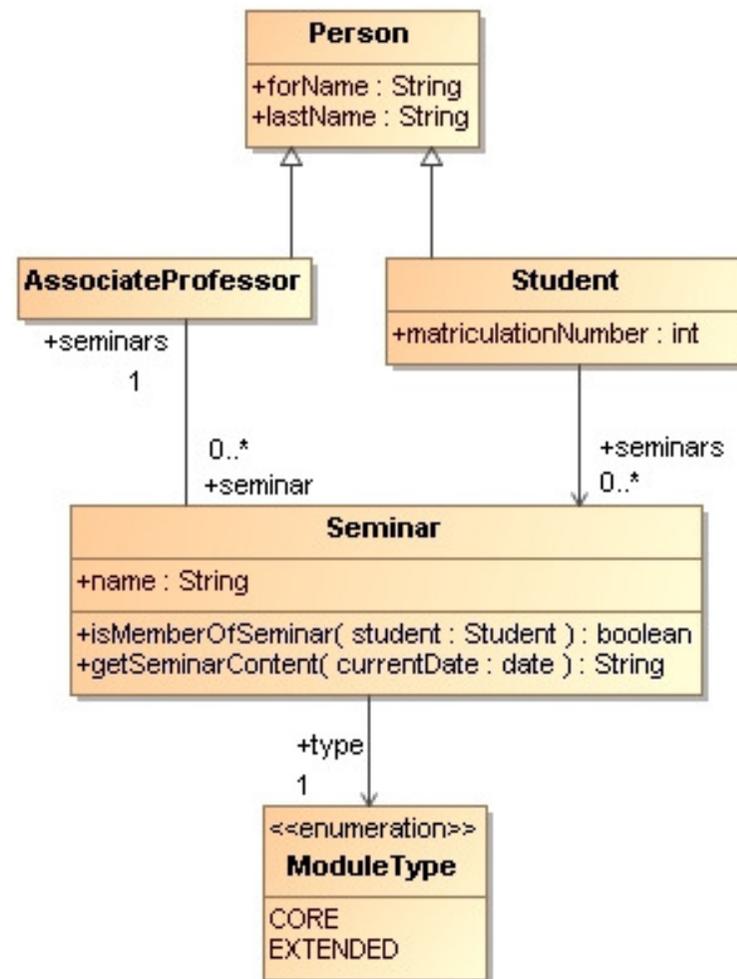


Abbildung: Seminarverwaltung
in UML

DEMO

- Editor in Eclipse
- XMI
- Modellreferenz

Anwendungsbeispiel

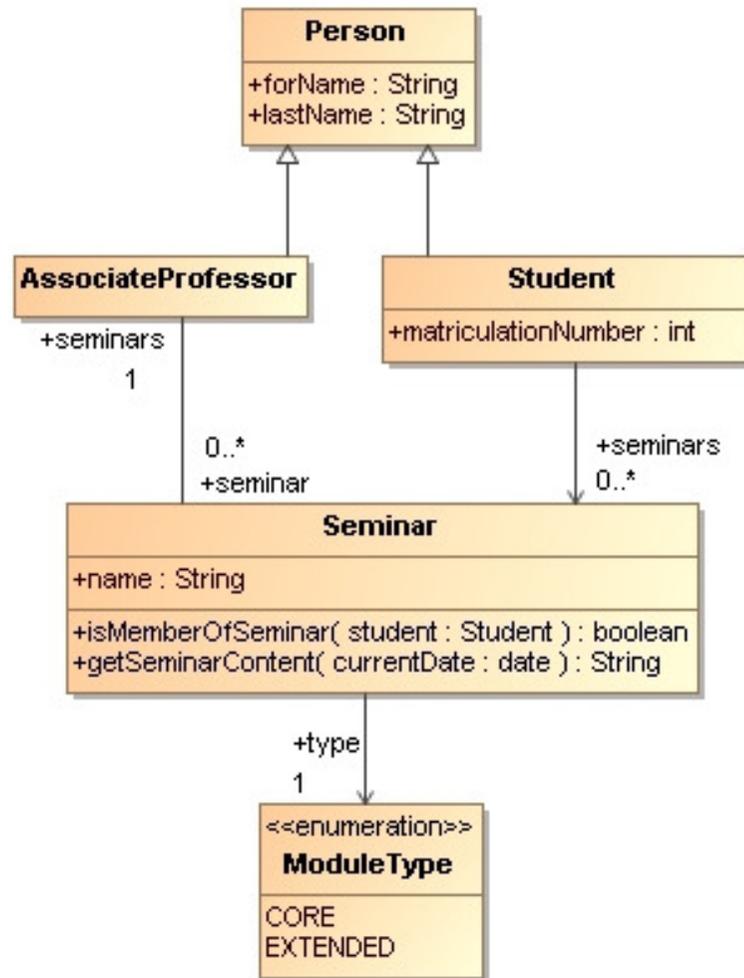


Abbildung: Seminarverwaltung in UML

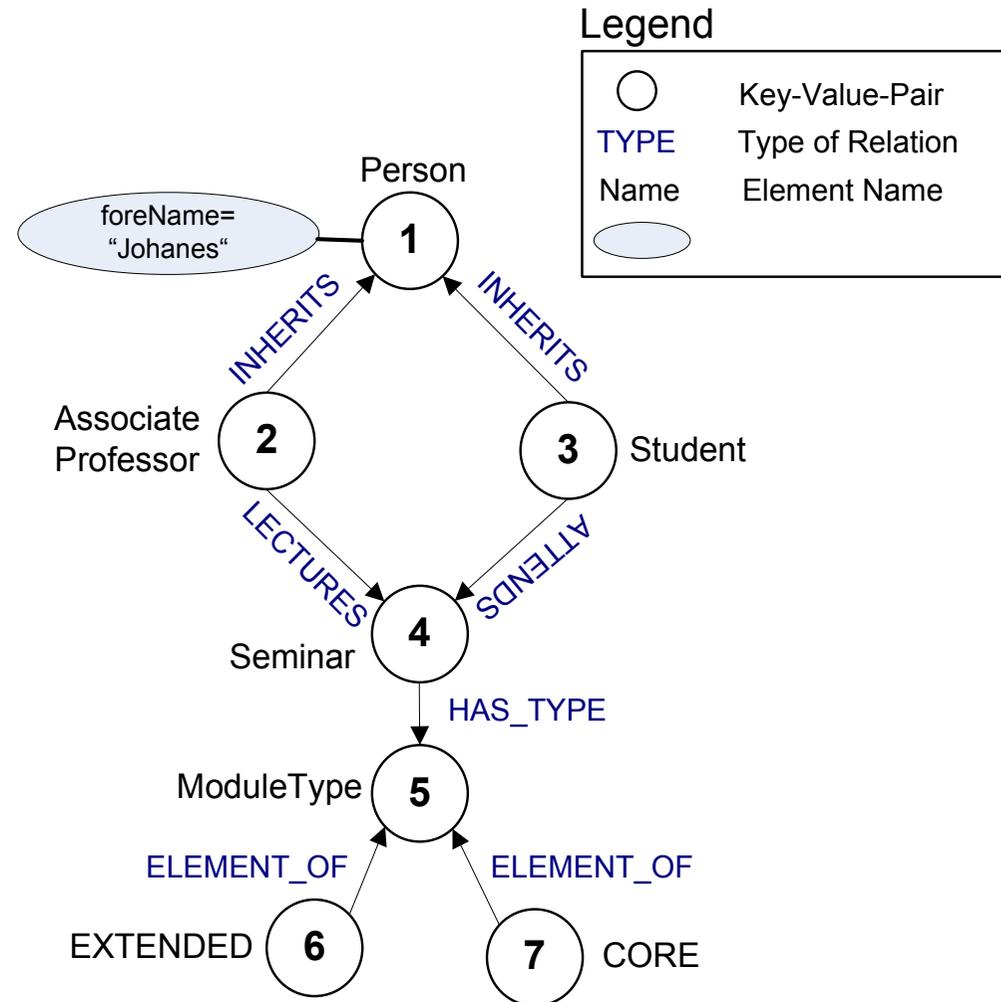


Abbildung: Seminarverwaltung als Graph

Modellgraph

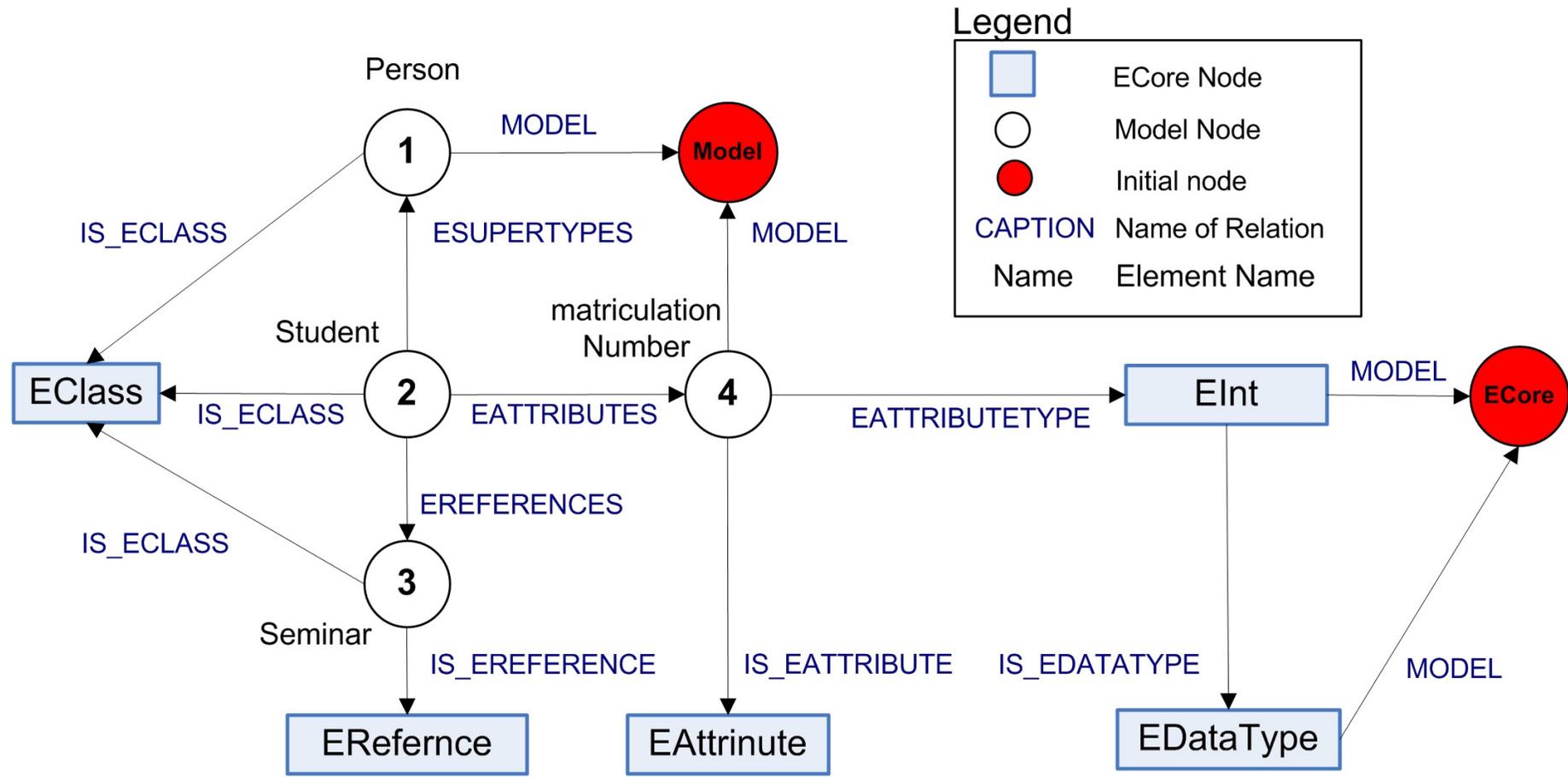


Abbildung: EMF als Graph in Neo

Modellpersistenz im Nodespace

- Referencenode über API erreichbar
- Hierarchie der Modelle

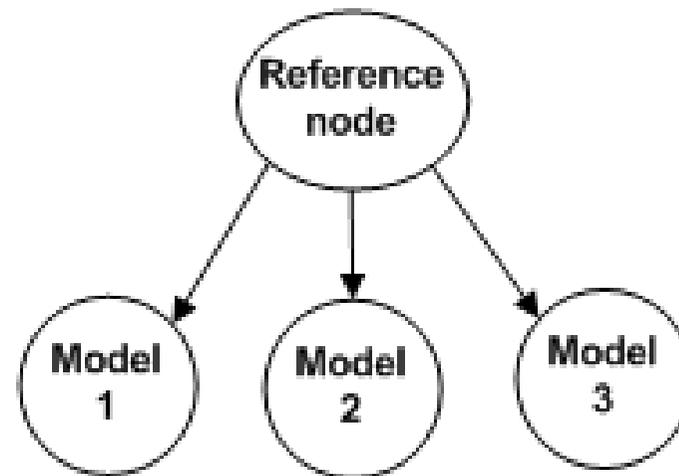


Abbildung: Modellnavigation in Neo

Modellversionierung

- Metainformationen in Properties
- Anwendbar auf Nodes und Relationen

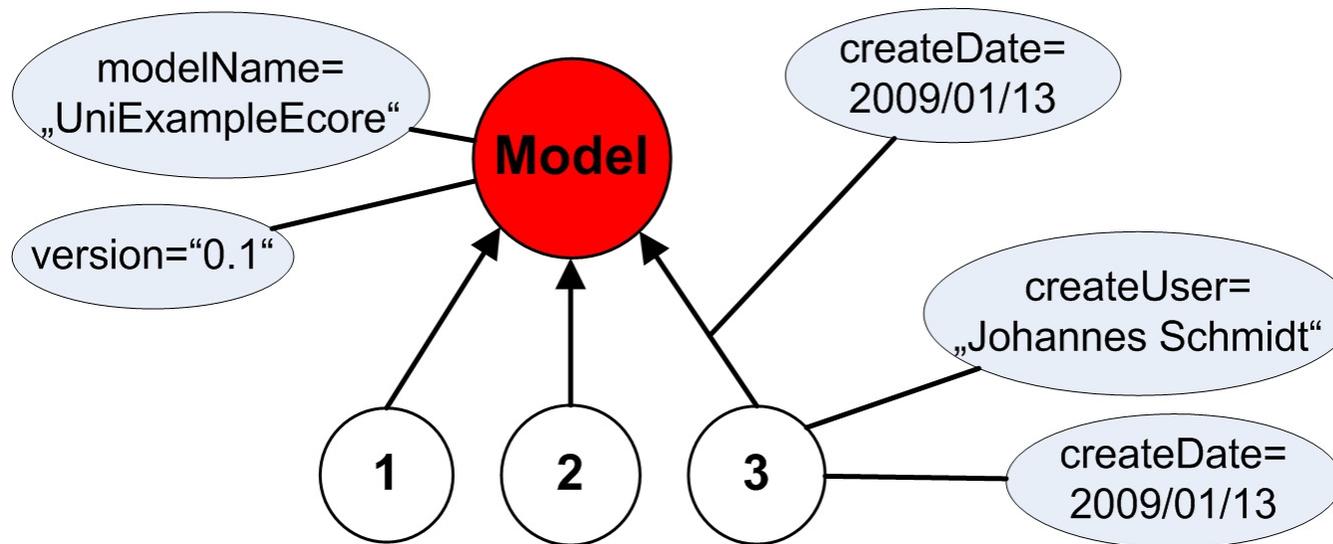


Abbildung: Versionierung von Modellen

Subreferences

- Optimierte Anordnung im Nodespace
- Stellvertreter für einen Relationstyp

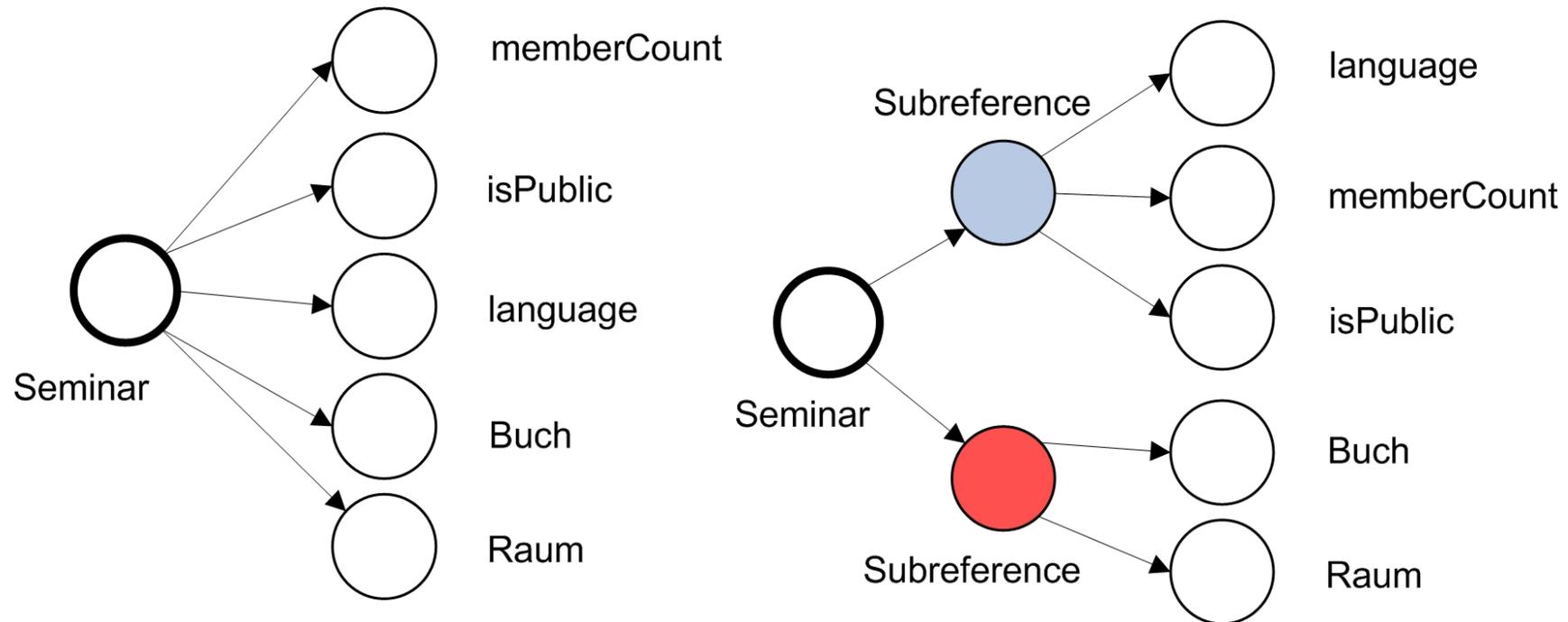


Abbildung: Subreferences

Zusammenfassung und Ausblick

- EMF für Modellierung und Generierung in der Eclipse
- Einfache Anwendungsentwicklung mit Neo
- Referenzierung auf Ebene Modellelemente möglich
- Graphische Metainformationen in Properties
- Für Implementierung eines Modell-Repositories geeignet
- Vorteile gegenüber relationalen Datenbanken (Sparse-Tables)

Vielen Dank für die Aufmerksamkeit

Quellen und Literatur

- [Wike] Neo Wiki. The Matrix.
http://wiki.neo4j.org/content/The_Matrix, letzter Abruf am 25.01.2009
- [BSM+04] Frank Budinsky, David Steinberg, Ed Merks, Raymond Ellersick und Timothy J. Grose. Eclipse Modeling Framework. the eclipse series. Addison-Wesley, 2004.
- [PT07] Georg Pietrek und Jens Trompeter, Hrsg. Modellgetriebene Softwareentwicklung: MDA und MDSD in der Praxis. Entwickler.press, Frankfurt am Main, 2007.
- [MM03] Joaquin Miller und Jishnu Mukerji. MDA Guide Version 1.0.1. Bericht omg/2003-06-01, OMG, Juni 2003. Online verfügbar unter <http://www.omg.org/docs/omg/03-06-01.pdf>.
- [HM08] Wolfgang Hesse und Heinrich C. Mayr. Modellierung in der Softwaretechnik: eine Bestandsaufnahme. Informatik Spektrum, 31(5):377–393, Oktober 2008.
- [Foud] The Eclipse Foundation. EMF.
<http://www.eclipse.org/modeling/emf/>, letzter Abruf am 18.01.2009.
- [Neo] Neo4j. Neo4j Project Homepage. <http://neo4j.org/>, letzter Abruf am 18.01.2009.