



Modellgetriebene homogenisierte Datenhaltung und Synchronisation mit vorhersagbaren Eigenschaften
für plattformübergreifende Anwendungen

Förderkennzeichen: 01 S12012

Installationsanleitung

Autor: T. Heuser (B2M)

Co-Autoren: E. Gailus (B2M), Gerald Hübsch (CAS)

Projektpartner



GEFÖRDERT VOM

Bundesministerium
für Bildung
und Forschung

Änderungshistorie

Version	Änderungen von	Datum	Anmerkung

ToDos

Abschnitt	von	Beschreibung	Priorität	Todo

Inhalt

1	Entwicklungsumgebung	1
1.1	Softwareinstallation.....	1
2.1.1	Basis: Eclipse Modelling Paket	1
2.1.2	Xtend.....	1
2.1.3	Subversion SVN Unterstützung (Subversive)	2
2.1.4	Android SDK.....	3
2.1.5	Android Development Toolkit (ADT)	4
1.2	Konfiguration von Eclipse	5
1.3	Importieren des Projektes aus GitHub	6
2	Anwendung und Nutzung von Annotationen an Modellen	6

Abbildungen

Abbildung 1:	Auswahl von Xtend im Eclipse Marketplace	2
Abbildung 2:	Integration der Annotationen in der Entwicklungsumgebung	7
Abbildung 3:	Annotationen einer Klasse am Beispiel Poi und der Kategorie Database	7
Abbildung 4:	Annotation eines Attributes am Beispiel Poi.name und der Kategorie Database.....	8
Abbildung 5:	Auswahl des Ziels der Generierung.....	8

1 Entwicklungsumgebung

Im Folgenden wird das Aufsetzen einer Entwicklungsumgebung für die MOHITO Generator-Entwicklung beschrieben. Die Anleitung setzt Kenntnisse im generellen Umgang mit Eclipse und der Installation von Plugins im Speziellen voraus. Ist dies nicht der Fall, kann die allgemeine Einführung unter www.eclipse.org zum Wissensaufbau genutzt werden.

1.1 Softwareinstallation

2.1.1 Basis: Eclipse Modelling Paket

Eclipse Kepler Modelling Paket herunterladen und entpacken: <http://www.eclipse.org/downloads/packages/eclipse-modeling-tools/keplerr>

Anpassung des Arbeitsspeichers in der eclipse.ini Datei um den Umgang mit komplexen Modellen auch bei Nutzung der Oracle JVM zu ermöglichen:

```
--launcher.XXMaxPermSize  
512M  
-Xms80m  
-Xmx1024m
```

Eclipse starten und weitere Komponenten über „*Help -> Install New Software*“ hinzufügen

2.1.2 Xtend

Xtend bietet eine Infrastruktur zur Modelltransformation und Code-Generierung. Eine Beschreibung und die Dokumentation ist unter <http://www.eclipse.org/xtend/> zu finden.



Abbildung 1: Auswahl von Xtend im Eclipse Marketplace

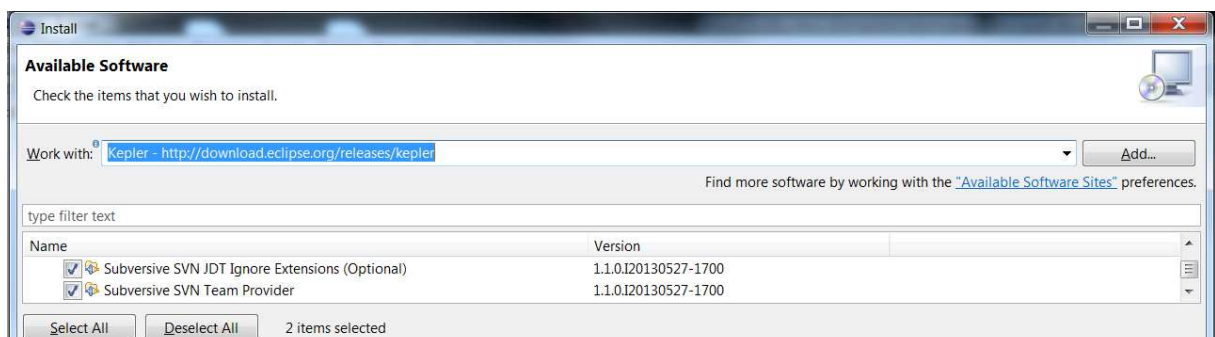
Die Installation über den Eclipse Marketplace wird empfohlen. Die entsprechende Auswahl ist in Abbildung 1 dargestellt.

2.1.3 Subversion SVN Unterstützung (Subversive)

Subversive wird über die Standard Kepler Update Site mit angeboten:

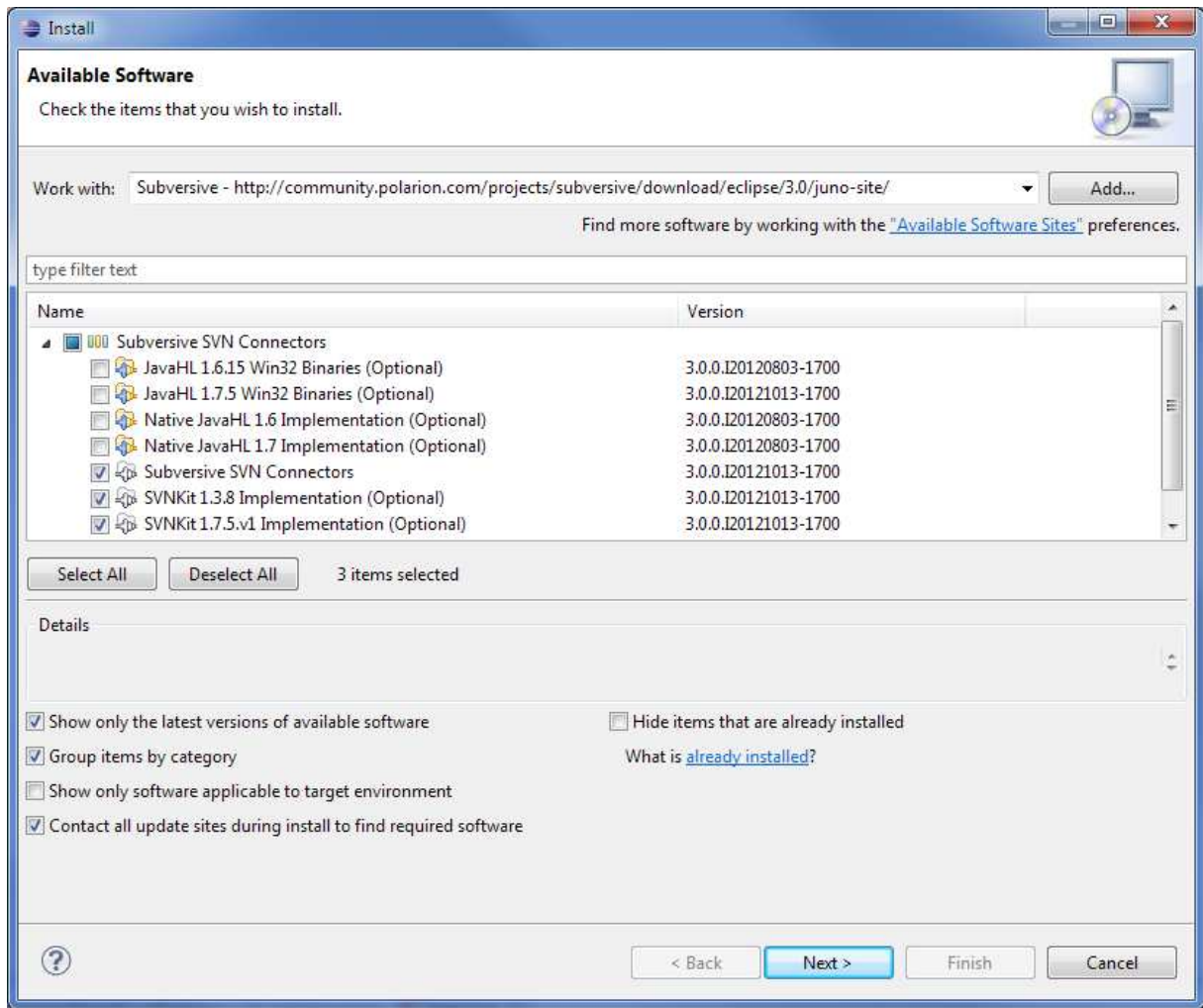
<http://download.eclipse.org/releases/kepler>

Dort unter der Kategorie „Collaboration“:



auswählen und installieren. Anschließend Eclipse neu starten und als SVN Connector die SVNKit-Implementierungen von der folgenden Update-Site installieren.

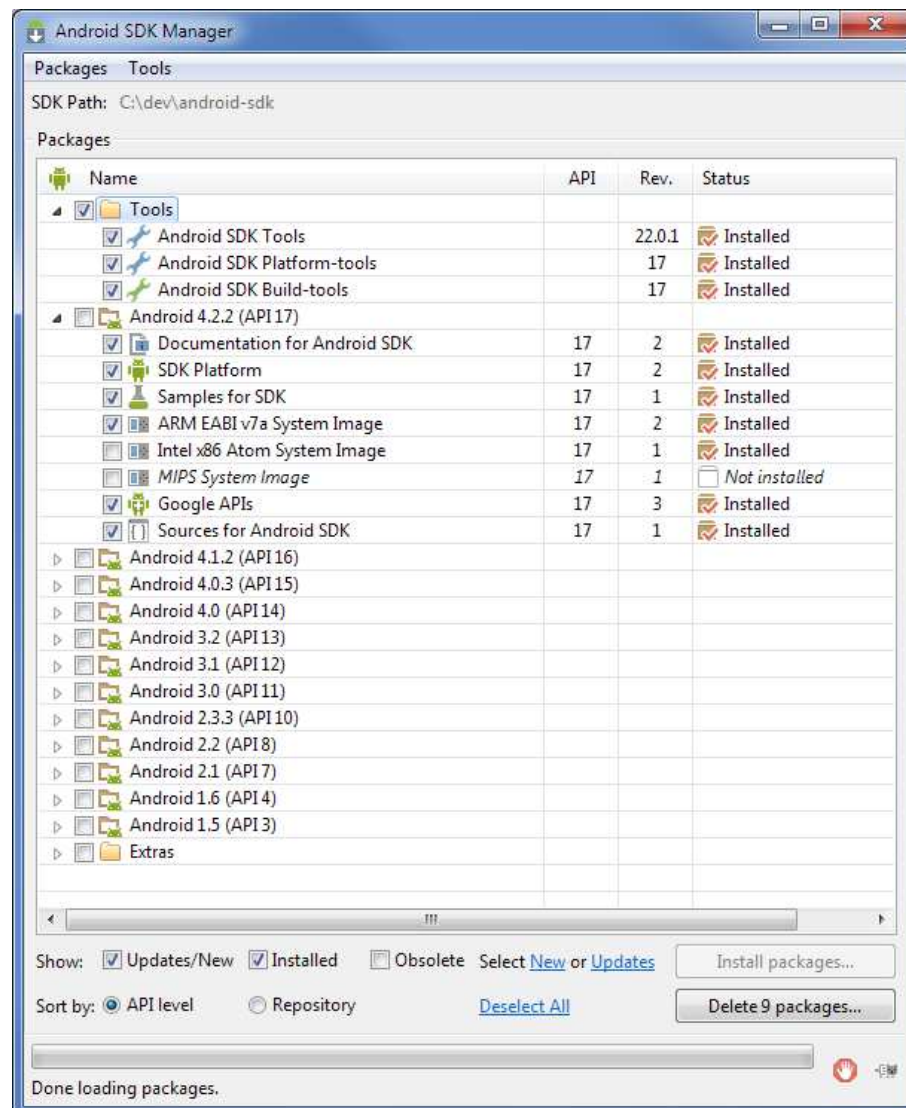
<http://community.polarion.com/projects/subversive/download/eclipse/3.0/juno-site/>



2.1.4 Android SDK

Die Installation sollte unter Nutzung der Webseite <http://developer.android.com/sdk/> erfolgen. Bei 64-Bit-Systemen ist auf die Auswahl der 64-Bit-Variante zu achten. In dem Paket liegt sowohl das SDK als auch eine Eclipse-Version mit installiertem Development-Toolkit. Diese vernachlässigen und nur das Verzeichnis „sdk“ und den Package Manger kopieren, beispielsweise nach c:\dev\android-sdk.

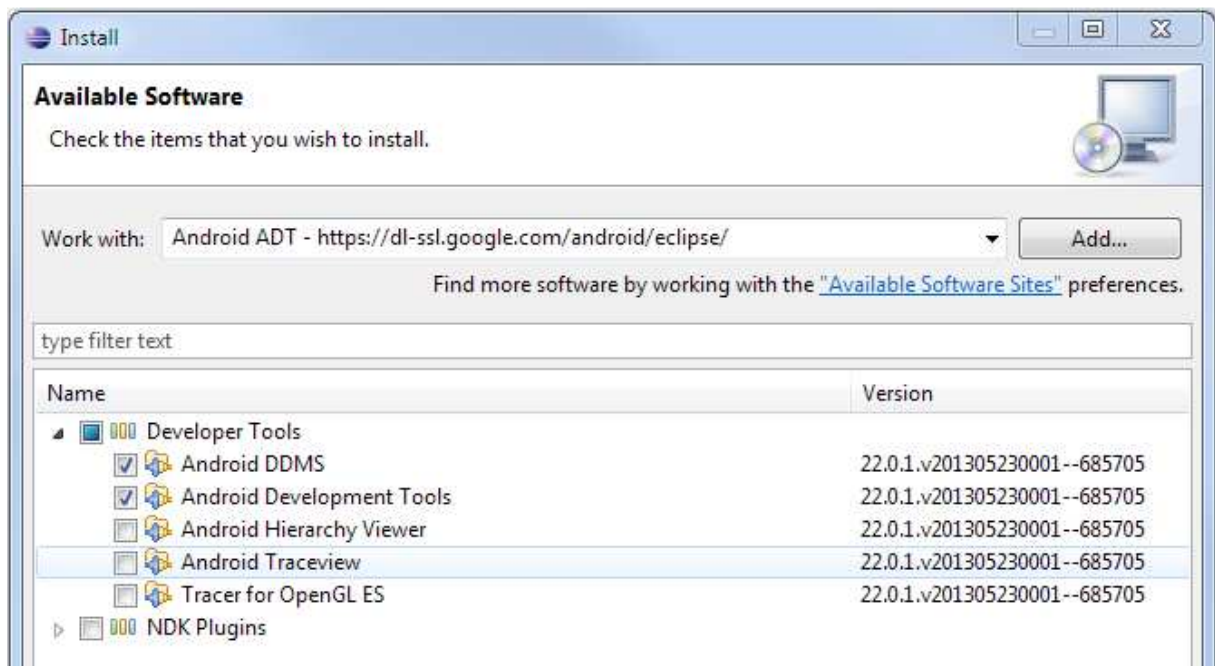
Den SDK-Manager ausführen und die abgebildeten Pakete installieren:



Dabei sicherstellen, dass die Version der SDK-Tools mit der Version des ADT im nächsten Abschnitt übereinstimmt.

2.1.5 Android Development Toolkit (ADT)

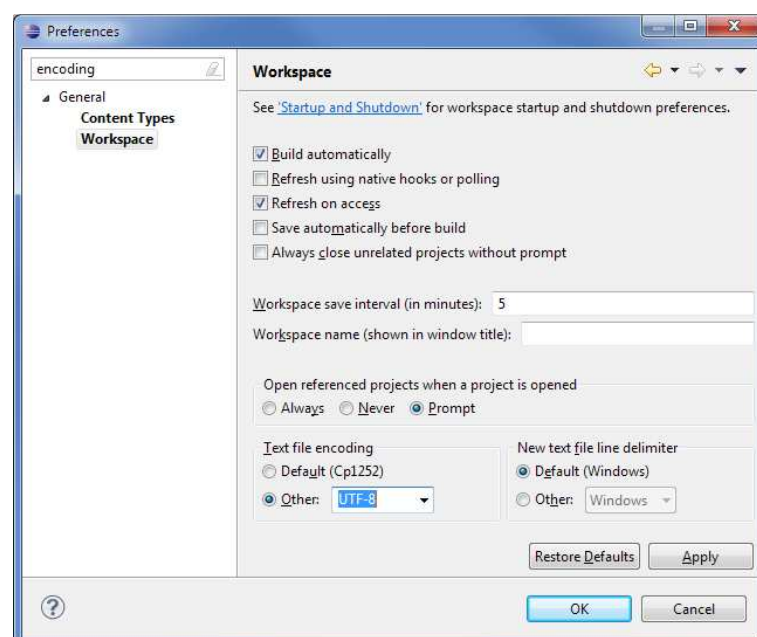
Das ADT über die folgende Update Site in Eclipse installieren:
<https://dl-ssl.google.com/android/eclipse/>



Die Versionsnummer sollte mit der Version der SDK-Tools aus dem Android SDK übereinstimmen (hier: 22.0.1).

1.2 Konfiguration von Eclipse

Für die Kompatibilität der generierten Quellen das Textfile-Encoding auf UTF-8 einstellen.
Window -> Preferences -> General -> Workspace -> Text file encoding: UTF-8



1.3 Importieren des Projektes aus GitHub

Aus dem MOHITO-Repository <https://github.com/MOHITO-PROJECT/MOHITO> alle Projekte auschecken:

Bei Bearbeitung des Frameworks sind die nachfolgenden Projekte von Bedeutung:

- framework/info.multipatform.mohito.framework
- framework/android/info.multipatform.mohito.framework.android

Nach einer Anpassung der Implementierung ist im jeweiligen Plugin ein Ordner mit dem Namen *deployable* auf der Wurzelebene zu erzeugen (falls noch nicht vorhanden). Anschließend ist das Kompilat der Implementierung in Form einer JAR-Datei abzulegen. In der Entwicklungsumgebung ist dies wie folgt umsetzbar: Auswahl eines Plugins im *Package Explorer*, *File* -> *Export* -> *Java/JAR file* -> *JAR file* = „*[Plugin Name]/deployable/[Plugin Name].jar*“ -> *Finish*.

Für die Bearbeitung der Generatoren, oder als Referenz bei Neuentwicklungen, sind die im Nachfolgenden genannten Projekte relevant.

- modagile/multipatform/info.multipatform.mohito.generator.android
- modagile/multipatform /info.multipatform.mohito.generator.java

2 Anwendung und Nutzung von Annotationen an Modellen

Dieser Abschnitt beschreibt die Anwendung der DSL in der MOHITO Entwicklungsumgebung. Die Annotationen erfolgen unter Ausnutzung des *Ecore Diagram*-Editors. Die für Diagramme gegenüber einem reinen Datenmodell notwendige Speicherung von Layout-Informationen erfolgt in *.ecordiag*-Dateien. Ein Doppelklick auf die *.ecorediag*-Datei öffnet den Editor.

Diese können automatisch auf Basis von *.ecore*-Dateien erzeugt werden. Ein Rechtsklick auf die *.ecore*-Datei und Auswahl von *Initialize Ecore Diagram File...* ist hierfür ausreichend.

Anwendung und Nutzung von Annotationen an Modellen

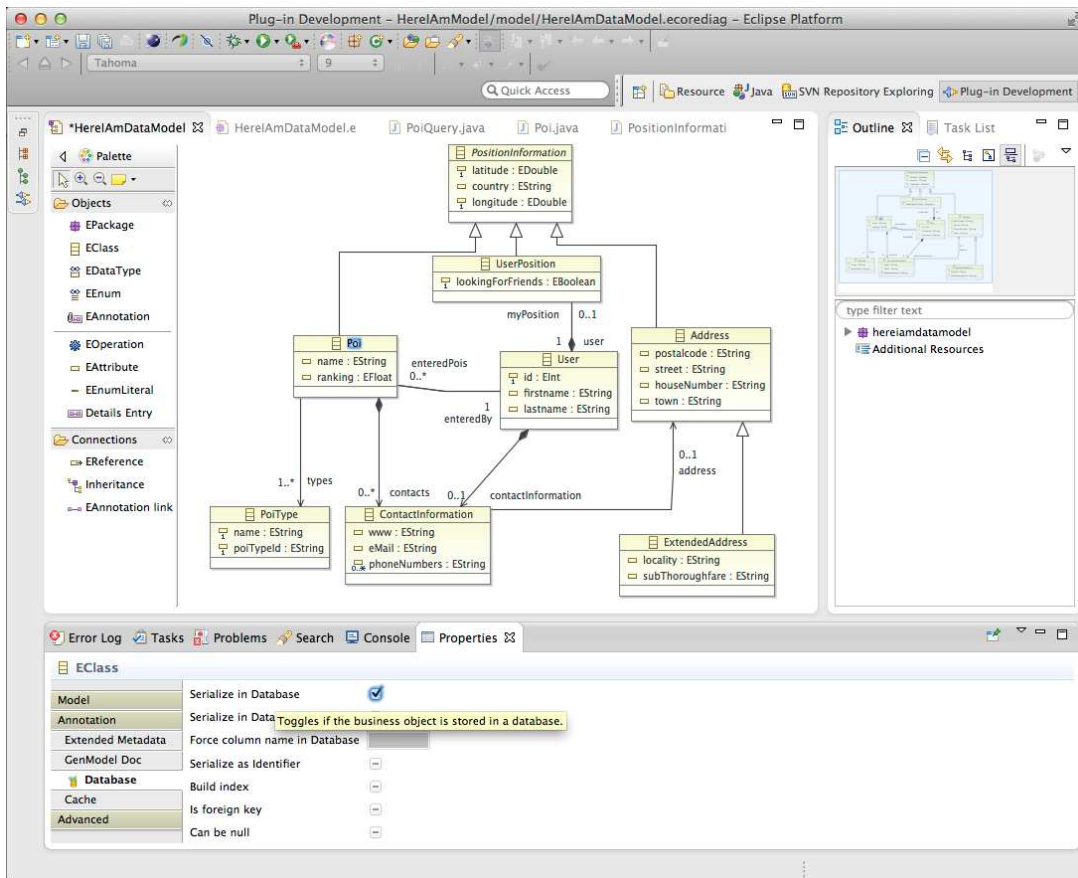


Abbildung 2: Integration der Annotationen in der Entwicklungsumgebung

Die definierten Annotationskategorien und Annotationen werden in der *Properties* Ansicht im unteren Teil der Entwicklungsumgebung dargestellt. Am Beispiel der Klasse *Poi* und der Kategorie *Database* ist dies in Abbildung 2 dargestellt. Verweilt der Cursor auf einem Eintrag wird der Beschreibungstext der Annotation als Tooltip angezeigt.

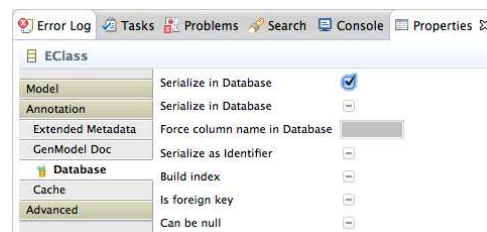


Abbildung 3: Annotationen einer Klasse am Beispiel Poi und der Kategorie Database

Abbildung 3 zeigt die Annotationen für das Modellelement *Poi* und die Kategorie *Database*. Die Kategorie wird sichtbar, sobald in einem Editor für ecore-Diagramme ein annotierbares Modellelement ausgewählt wird. Je nach Typ des Elements sind nur bestimmte Annotationen

Anwendung und Nutzung von Annotationen an Modellen der Kategorie auswählbar. Im Beispiel ist dies der erste Eintrag. Dieser ist zusätzlich vom Anwender auf wahr gesetzt. Die weiteren Annotationen werden zur Übersicht mit aufgeführt und ausgegraut dargestellt.

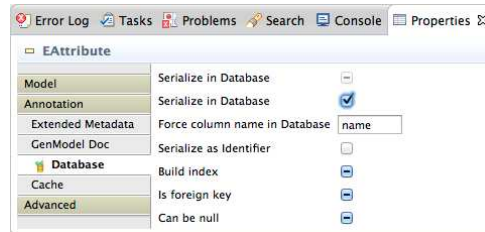


Abbildung 4: Annotation eines Attributes am Beispiel *Poi.name* und der Kategorie Database

Abbildung 4 zeigt die Annotationen für das Attribut *name* des Modellelements *Poi*. Für dieses sind die Kategorien in den Zeilen 2 bis 7 auswählbar. Die Annotation *Serialize in Database* ist auf wahr gesetzt. Die Annotation *Force column name in Database* ist auf *name* gesetzt. Die Annotation *Serialize as Identifier* ist auf falsch gesetzt. Die anderen Annotationen behalten ihren Standardwert und werden nicht vom Benutzer überschrieben.

Die Generierung von Code für die unterschiedlichen Plattformen kann direkt für einzelne Modelle angestoßen werden.

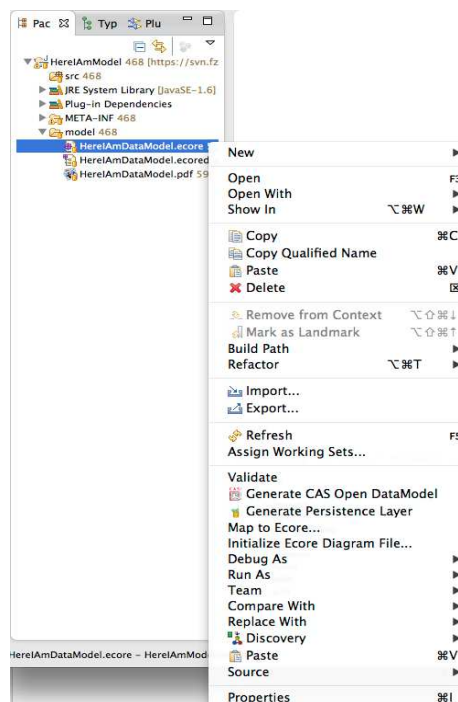


Abbildung 5: Auswahl des Ziels der Generierung

Die Alternativen zur Generierung können mittels Rechtsklick auf eine *.ecore*-Datei angezeigt und ausgewählt werden. Abbildung 5 stellt die Auswahl des Ziels einer Generierung dar. Eine

Anwendung und Nutzung von Annotationen an Modellen

mögliche Auswahl ist *Generate CAS Open DataModel* oder *Generate Persistence Layer*. Basierend auf der Auswahl werden entsprechende Projekte für die Zielumgebung erzeugt. Die Einstellungen des Generators können mittels der Annotationen der Kategorie *Generator* am Paket, welches alle Klassen des Domänenmodells enthält, übergeben werden. Die generierten Klassen sind im aktuellen Eclipse-Workspace sicht- und bearbeitbar.