



Eclipse Platform User Guide V0.1.1

René Baumann

May 23, 2020

Einleitung

Zweck	<p>Diese Dokument beschreibt, wie die fertig installierte und konfigurierte SmartGridready Eclipse Platform zur Modellierung von SmartGridready Komponenten genutzt werden kann.</p>
Adressaten	<p>Das Dokument richtet sich an folgende Zielgruppen:</p> <ul style="list-style-type: none">• Mitglieder der Technischen Kommission Smart-Gridready (SGr)• Hersteller von Geräten, die mit dem SGr - Modell arbeiten wollen• Ev. Prüforganisationen, welche die Einhaltung der SGr - Richtlinien verifizieren wollen
Dokument Status	<p>Das Dokument ist im Status: First Draft</p>
Dokument Historie	<p>Das Dokument hat sich wie folgt entwickelt:</p> <ul style="list-style-type: none">• 200514-RB: V0.1 Created• 200522-RB: V0.1.1 Ergänzt mit XSD Import (cbt)

Fortsetzung...

Einleitung, Fortsetzung

Referenzen

Das Dokument referenziert die folgenden Quellen:

Id	Beschreibung
[1]	Titel: Wiki Eclipse Foundation Autor: Eclipse Foundation [?] Datum: n.a Quelle: Eclipse Wiki home
[2]	Titel: Getting started with CDO Autor: Eclipse Foundation Datum: 17. August 2009 Quelle: Eclipse Wiki CDO
[3]	Titel: CDO The Model Repository V4.9.0 Autor: Eike Stepper Datum: 11. März 2020 Quelle: Eclipse CDO
[4]	Titel: Eclipse Modeling Framework Autor: Wikipedia DE Datum: Quelle: Eclipse EMF
[5]	Titel: SmartGridready Eclipse Modeling Installation Autor: Rene Baumann Datum: 13.05.2020 V0.3 Quelle: t.b.c
[6]	Titel: SmartGridready Eclipse Platform Konfiguration V0.1 Autor: Rene Baumann Datum: 13.05.2020 V0.1 Quelle: t.b.c

Inhaltsverzeichnis

Titel	Seite
Uebersicht	4
Laden von Modellen aus XSD - Files	7
Laden von Modellen aus GitHub	20
Eclipse EMF: CDO - Konfiguration	24
Anhang	27

Uebersicht

SmartGridready Mission

Unser Ziel ist es, die Energiewelt der Zukunft für alle verständlich, sicher und smarter zu machen.

Mit SmartGridready erkennen die Kunden auf einen Blick, dass Ihre Planung und Systemintegrationen mit den smarten Produkten reibungslos in der neuen Energiewelt kommunizieren. Ohne Investitionsrisiko.

Weiter ergibt sich dank SmartGridready eine wesentlich vereinfachte Projektabwicklung, indem alle am Geschäft beteiligten Anbieter mit der gleichen Basisinformation arbeiten.

Damit Sie in der Energiewelt von morgen nachhaltig und erfolgreich sein können. SmartGridready wird ein weit anerkanntes und bewährtes Qualitätslabel für die smart kommunizierende Energieversorgung der Zukunft.

SmartGridready Werkzeuge

Zur Umsetzung der Modellierung setzt SmartGridready ausschliesslich auf OpenSource Tools. Damit soll gewährleistet werden, dass jeder Hersteller, Planer, Operator oder Benutzer einen direkten, kostengünstigen und Herstellerunabhängigen Zugang zu Deklarationen und Tools hat und diese ggf selber - unter Beachtung der SmartGridready Lizenzbedingungen - weiterentwickeln kann.

Zentrales Element dieser Strategie ist die Open Source IDE "Eclipse" mit dem Plug-In "Eclipse Modeling Framework (EMF)". In diesem Dokument wird beschrieben, wie Eclipse EMF - Platform, nach einer Installation gemäss [5], konfiguriert werden muss, damit sie für die SmartGridready Modellierung verwendet werden kann.

Fortsetzung...

Uebersicht, Fortsetzung

Zweck der Plattform

Mit Hilfe der Eclipse IDE will SmarGridready die folgenden Ziele erreichen:

- **Erstellen von (Meta-)Modellen** der Equipments, Devices und Modulen und deren Datenaustausch, in einem SmartGridready konformen Netz von Komponenten, in Programmiersprachen neutraler Form.
- **Dokumentation der Klassen** mit Zweck und korrekter Verwendung im Sinne von SmartGridready. Die Dokumentation soll aus dem erstellen Modell ohne weitere manuelle Arbeit erfolgen können.
- **Generieren eines funktionsfähigen (Teil-)Models** aus dem definierten Matamodells zu Diskussions-, Präsentations-, Testzwecken ohne Programmieraufwand
- **Sicherung und Versionsverwaltung** für die erstellten Modelle. Zile ist die Development history zu sichern um so ggf auf frühere Entwicklungsstände zurückgreifen zu können.
- **Dokumentation** der erstellten Modelle, indem die dort gespeicherten Beschreibungen zu Zweck, Nutzung und Voraussetzungen in eine User - Dokumentation überführt werden kann.
- **Generierung von Modellen** Erzeugen von Editoren, mit welchen die erstellten Datenmodelle mit Daten befüllt und getestet werden können.
- **Persistency und Concurrency** Die Plattform soll es erlauben, konkrete Modelle, die zu Testzwecken erstellt worden sind zu speichern und bei Bedarf wieder laden zu können.

Fortsetzung...

Uebersicht, Fortsetzung

Voraus- setzungen

Für die erfolgreiche Nutzung der Eclipse EMF - Platform müssen folgende Voraussetzungen erfüllt sein:

- **Eclipse EMF - Platform** gemäss [5] installiert
 - **Eclipse EMF - Platform** gemäss [6] konfiguriert
 - **Account** und **Zugriff** auf <https://github.com/SmartGridready> ist vorhanden
-

Laden von Modellen aus XSD - Files

Einleitung

Bei der Erstellung eines EMF - Modells möchte man ev. Teile davon auf bestehende Standards oder Deklarationen abstützen, die aber nur als XSD - Files vorliegen. In diesem Kapitel wird gezeigt:

- Wie aus XSD - Files ein XSD2EMF-Modell wird
- Wie dieses XSD2EMF-Modell analysiert wird
- Wie mein MyEMF-Modell auf das XSD2EMF-Modell zugreift
- Wie man in MyEMF-Modell Deklarationen aus XSD2EMF-Modell nutzt

Voraussetzungen

Wie das Vorgehen zum Erstellen eines XSD2EMF-Modells ist, wird am Beispiel der *eebus Spine Spezifikationen* gezeigt. Dazu benötigt man das Archivfile .

Vorbereitung

Führe die folgenden Arbeitsschritte aus:

1. Kopiere das Archivfile *SpineV1.1.1.zip* in ein temporäres Verzeichnis bspw */home/[user]/temp*
2. Entpacke dort das Archivfile *SpineV1.1.1.zip*
3. Starte nun die Eclipse EMF - Plattform
4. Öffnen in Eclipse die *Console View* durch Menu:
Window => Show View => Other...=> General => Console

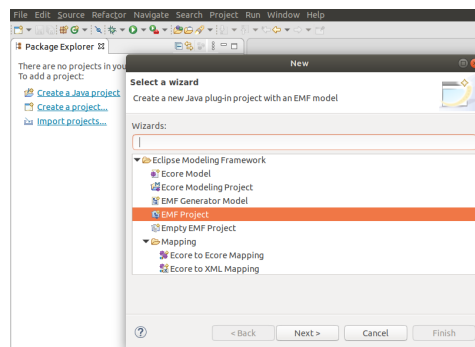
Fortsetzung...

Laden von Modellen aus XSD - Files, Fortsetzung

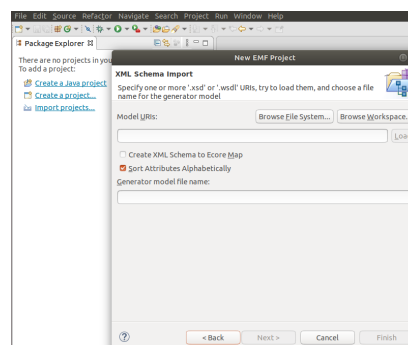
Erstelle
EMF-Projekt

Erstelle ein EMF - Project *eebus_model* in der Eclipse
EMF - Platform durch folgende Aktionen:

1. Klicke Menu: *File => New => Other...*
2. Klicke *Eclipse Modeling Framework => EMF Project => [Next] button*



3. Trage im Feld *Project name*: den Namen *eebus_model* ein
4. Klicke *[Next]* button
5. Wähle *XML Schema* im Window *Select a model Importer*
6. Klicke *[Next]* button



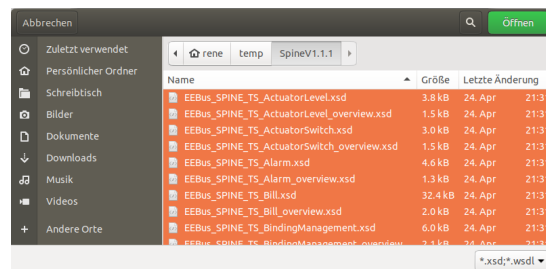
Fortsetzung...

Laden von Modellen aus XSD - Files, Fortsetzung

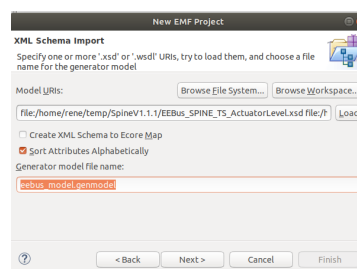
XML - Schema Import

Wenn die vorherigen Aktionen korrekt ausgeführt wurden, ist nun das Window *XML - Schema Import* geöffnet. Fahre wie folgt fort:

1. Klicke den Button *Browse File System*
2. Navigiere im *File Browser* zum Verzeichnis */home/[user]/temp/SpineV1.1.1*
3. Selektiere alle in diesem Verzeichnis vorhandenen Files. Klicke auf das erste File um es zu selektieren, scrolle dann im File Browser ans Ende der File-liste, drücke die *Shift* - Taste. klicke dann mit gedrückter *Shift* - Taste das letzte File in der File-liste.



4. Klicke den Button *Oeffnen*
5. De-selektiere die Option: *CreateXML Schema to Ecore Map*
6. Selektiere die Option: *Sort Attributes Alphabetically*
7. Ändere den Filename im Feld *Generator model file name* auf *eebus_model.genmodel*



8. Klicke *[Next]* button

Laden von Modellen aus XSD - Files, Fortsetzung

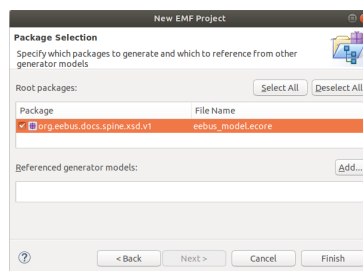
cont..

XML - Schema

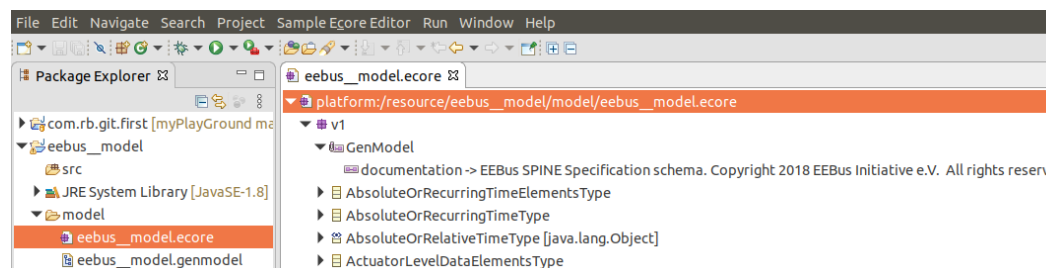
Import

....fast geschafft, nur noch die paar Aktionen..

1. Klicke ins Feld *File Name* im Window *Package Selection* und ändere den Filenamen von *v1.ecore* auf *eebus_model.ecore*



2. Klicke den Button *Finish* um den Import zu starten
3. Prüfe in der Eclipse - *Console* - View ob beim Import Fehler aufgetreten sind.
 - wenn ja, Fehler korrigieren, ev. erzeugtes EMF - Project im *Eclipse Model Explorer* mit *Right-Click* => *Delete* löschen und nochmals mit Import - Prezedere starten
 - wenn nein, sollte im *Eclipse Model Explorer* das EMF - Project *eebus_model* wie unten gezeigt sichtbar sein (ev. expandieren und Doppelklick auf *eebus_model.ecore*



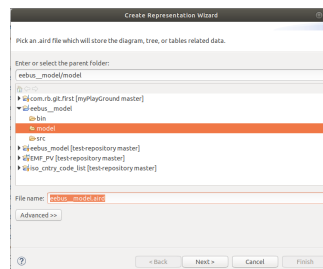
Fortsetzung...

Laden von Modellen aus XSD - Files, Fortsetzung

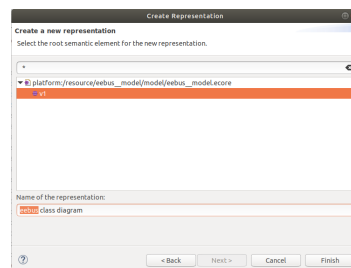
EMF - Project Analyse Vorbereitung

Mit den folgenden Aktionen kann das eben importierte *eebus_model* mit automatisch generierten Informationen ergänzt und damit besser dokumentiert werden. In diesem Block wird das *eebus_model Class Diagram* erstellt - viele weitere Diagramm und Tabellen sind möglich, werden hier aber nicht beschrieben.

1. Öffne das *eebus_model* und mache Right-Click auf *eebus_model.ecore*
2. Klicke auf den Menu - Eintrag *Initialize Ecore Diagram ...*
3. Ersetzte im Feld *File name:* den Namen *v1.arid* durch: *eebus_model.aird* und klicke *[Next]* button



4. Selektiere im Window *Select type of representation* den Eintrag *Entities in a Class Diagram* und klicke *[Next]* button
5. Ersetzte im Window *Create a new representation* im Feld *Name of the representation* den Eintrag *v1 Class Diagram* durch *eebus Class Diagram* und klicke den *[Finish]* button

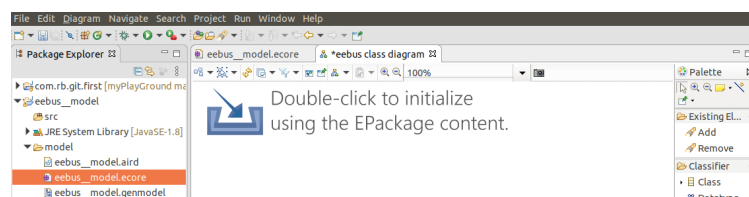


Die Aufbereitung der Daten dauert je nach Leistung des Computers 5-15sec.

Laden von Modellen aus XSD - Files, Fortsetzung

EMF - Project
Analyse
Abschluss

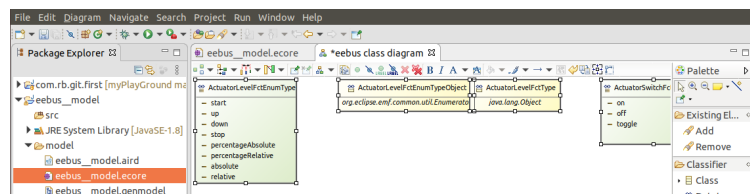
Wenn die vorherige Aufbereitung der Daten abgeschlossen ist, wird der folgende Bildschirminhalt gezeigt.



Zum Abschluss der XSD – Modell Analyse sind nun noch die folgenden Aktionen auszuführen:

1. Führe *Doppelklick* Im Eclipse Window **eebus class diagram* af das Graphiksymbol aus um die Aufbereitung zu starten.

Achtung, diese Aktion kann 60sec oder mehr dauern! Eine erste Bildausgabe erscheint wirr, mit vielen Symbolüberlagerungen - **NICHT eingreifen**, sondern weiter abwarten bis das Bild "entwirrt" ist.



2. Klicke das Diskettensymbol in der Eclipse Window Menu-Leiste um das nun fertige Klassendiagramm zu speichern.
3. Führe ggf eine manuelle Anpassung des Klassendiagrammes mittels "Drag-and-drop" aus um die "suboptimale" oder "nicht-intuitive" Anordnung der Klassen und Listen zu verbessern. Nutze dazu auch die Filterfunktion im Klassendiagramm-editor

Laden von Modellen aus XSD - Files, Fortsetzung

XSD2EMF Model nutzen Uebersicht

Nachdem das XSD - Modell als vollwertiges EMF - Modell vorliegt, kann es mit den Eclipse EMF - Werkzeugen analysiert, dokumentiert und ggf angepasst werden aber auch in anderen EMF - Modellen referenziert, im Folgenden beispielsweise mit *MyEMF_model* und die Deklarationen verwendet und eingebaut werden. In diesem Block wird gezeigt, wie das geht XSD2EMF-Modell in anderen EMF-Modellen genutzt werden kann.

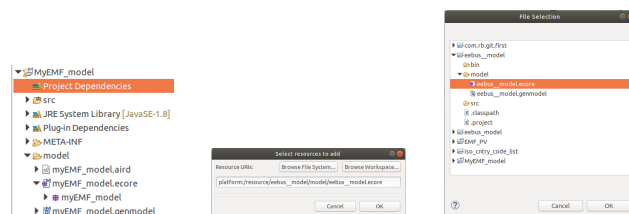
Fortsetzung...

Laden von Modellen aus XSD - Files, Fortsetzung

XSD2EMF
Model nutzen
Verlinkung

Mit den folgenden Aktionen wird das Beispielmmodell *MyEMF_model* mit dem XSD2EMF-Modell *eebus_model* zur weiteren Nutzung verlinkt.

1. Öffne (expandiere) im Eclipse *Model Explorer* das EMF-Modell *MyEMF_model*
2. Rechts-klick auf die Zeile *Project Dependencies* (Bild 1) und klicke dann *+ Add Model*
3. Klicke den Button *Browse Workspace...*
4. Öffne (expandiere) im Window *File Selection* (Bild 2) das *eebus_model*, dann öffne darin das Verzeichnis *model*
5. Selektiere dann das ECORE-file *eebus_model.ecore* (Bild 3) und klicke den Button *Ok*
6. Klicke dann *Ok* im Window *Select resources to add* (Bild 2)



7. Selektiere *Design* im Window *Viewpoints Selection* und klicke *Ok*

Damit ist das Modell *eebus_model* zum EMF-Modell *MyEMF_model* gelinkt (siehe auch *Project Dependencies* im *MyEMF_model*).

Fortsetzung...

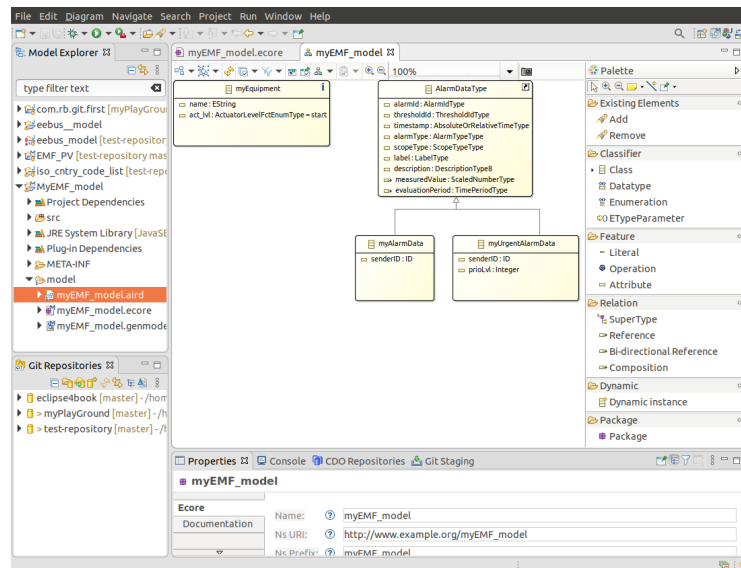
Laden von Modellen aus XSD - Files, Fortsetzung

XSD2EMF Model nutzen Anwendungs- beispiele

Nachdem das Modell *eebus_model* aus dem XSD-Import mit dem Zielmodell *MyEMF_model* verlinkt ist, können im Zielmodell Deklarationen mit Elementen aus dem *eebus_model* gemacht werden. Hier wird gezeigt, was u.a. möglich ist.

Das untenstehende Bild zeigt das EMF-Modell *myEMF_model* mit folgenden Beispielen:

- In der Klasse *myEquipment* wurde ein Attribut *act_lvl* definiert, welches den eebus - Datentyp *ActuatorLevelFctEnumType* mit dem Defaultwert *start* aufweist
- Aus dem *eebus_model* wurde die Klasse *AlarmDataType* importiert und als Super-Klasse für *myAlarmData* und *myUrgentAlarmData* verwendet



Fortsetzung...

Laden von Modellen aus XSD - Files, Fortsetzung

XSD2EMF
Model nutzen
Anwendungs-
Rezept

Die zuvor gezeigten Anwendungsbeispiele können entweder im Klassendiagramm oder im Model-Tree realisiert werden. Es wird je ein Beispiel in einer der beiden Methoden gezeigt, wobei zu bemerken ist, dass beide Methoden gleichwertig sind.

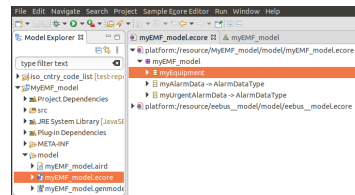
Fortsetzung...

Laden von Modellen aus XSD - Files, Fortsetzung

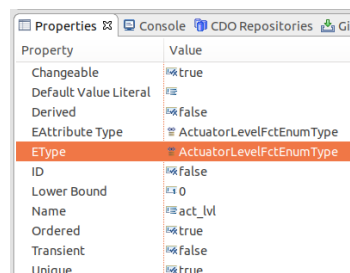
XSD2EMF
Model nutzen
Model-tree

Im Model-tree soll der Klasse *myEquipment* das Attribut *act_lvl* vom eebus - Datentyp *ActuatorLevelFctEnumType* beigefügt werden.

1. Öffnen in Eclipse die *Properties* View durch Menu: *Window => Show View => Other...=> General => Properties*
2. Öffne die *myEMF_model.ecore* Model-tree view durch Doppelklick auf den Eintrag *myEMF_model.ecore* im Eclipse *Model Explorer*
3. Expandiere in der Model-tree view bis die Klasse *myEquipment* sichtbar wird



4. Rechts-klick auf die Zeile *myEquipment*, dann wähle *New Child => EAttribute*
5. Im Reiter *Properties* trage im Feld *Name* den Text *act_lvl* ein und im Feld *EType* wähle aus der Drop-down liste den Datentyp *ActuatorLevelFctEnumType*
6. Prüfe, ob das Attribute sowohl in der Model-Tree View wie auch im Klassendiagramm sichtbar ist.

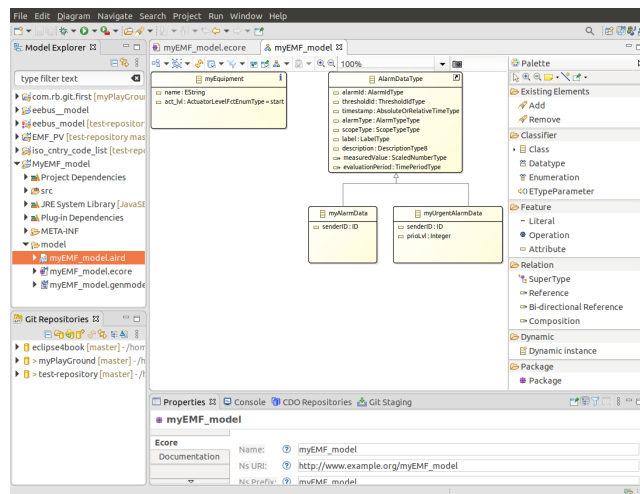


Laden von Modellen aus XSD - Files, Fortsetzung

XSD2EMF
Model nutzen
Klassendiagramm

Im Klassendiagramm soll die eebus - Klasse *Alarm-DataType* Superklasse für die beiden Klassen *myAlarmData* und *myUrgentAlarmData* werden.

1. Öffnen in Eclipse die *Properties* View durch Menu: *Window => Show View => Other...=> General => Properties*
2. Öffne das Klassendiagramm durch Doppelklick auf den Eintrag *myEMF_model.aird => Design => Entities in a Class Diagram => myEMF_model* im *Eclipse Model Explorer*



3. Klicke auf das Tool *add* in der *Palette* für *Existing Elements* und klicke dann eine leere Stelle im Klassendiagramm
4. Wähle im Window *Select element to add in diagram* den Eintrag *AlarmDataType*, dann klicke *Finish*
5. Klicke auf das Tool *Class* in der *Palette* für *Classifier* und klicke dann eine leere Stelle im Klassendiagramm
6. Im Reiter *Properties* trage im Feld *Name*: den Text *myAlarmData* ein
7. Klicke auf das Tool *SuperType* in der *Palette* für *Relations* und klicke dann im Klassendiagramm zuerst auf die eebus-Klasse *AlarmDataType* und dann auf die Klasse *myAlarmData* um die Klassenhierarchie zu erstellen

Laden von Modellen aus XSD - Files, Fortsetzung

cont...

XSD2EMF

Model nutzen

Klassendiagramm

Wähle dann das Tool *Attribute* in der *Palette* für *Features* und klicke in die Klasse *myAlarmData*. Definiere Namen und Typ des Attributes im Reiter *Properties*.

Verfahre gleich um die zweite Subklasse *myUrgentAlarmData* herzustellen.

Laden von Modellen aus GitHub

Warnung

Ab hier ist der Text noch in der BEARBEITUNG....es können Fehler und Inkonsistenzen vorhanden sein!!!!

Voraussetzungen

Der Verein SmartGridready hat auf GitHub ein Model-Repository eingerichtet. Die Eclipse EMF-Platfrom ist bei der Konfiguration [6] bereits mit diesem Repository <https://github.com/SmartgridReady/test-repository> verbunden worden. In diesem Dokument wird dieses Repository in der Folge nur noch mit ***SGr-ModRepo*** bezeichnet.

Aus GitHub - SGr Model-Repository

Die folgenden Schritte zeigen, wie ein Modell aus *SGr-ModRepo* in die eigene Eclipse EMF - Plattform geladen wird, um dieses so dann in eigenen Modellen zu referenzieren, zu analysieren oder ggf. zu ändern. In diesem Beispiel, soll das EMF-Modell *EMF_eebus* vom remote *SGr-ModRepo* Repository in den lokalen Workspace geladen werden.

1. Klicke Folder *Working Tree* im Reiter *Git Repository*
2. Right-Click dann Folder *EMF_eebus*
3. Klicke nun *Import Projects...* im geöffneten Menu - Window
4. Prüfe dass im Feld *Import source:* das lokal geklonte *SGr-ModRepo* Repository steht.
5. Wähle beide Folder: *EMF_eebus* und *EMF_eebus/eebus_model =>Finish*

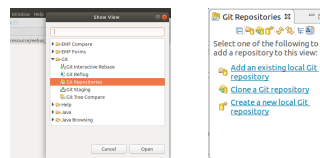
Fortsetzung...

Laden von Modellen aus GitHub, Fortsetzung

GitHub Repository Aktivierung

Die folgenden Aktionen aktivieren die Git - funktionalität auf der Eclipse EMF - platform.

1. Klicke Menu: *Window => Show View => Other...*
2. Klicke Menu: *Git => Git Repositories => Open*
3. Klicke auf *Clone a Git repository* im neuen *Git Repositories* Reiter



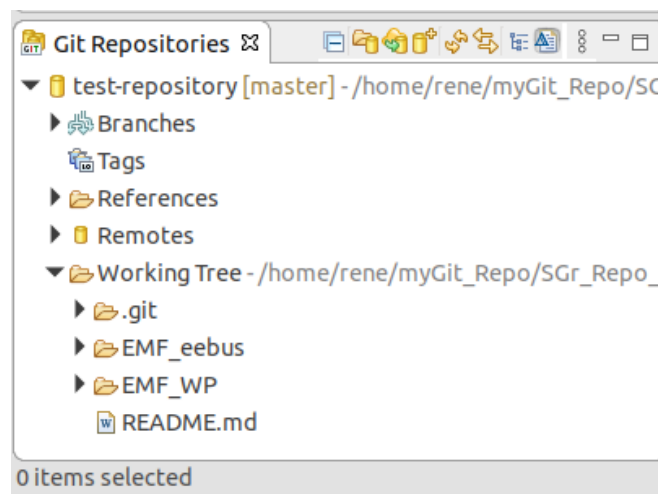
Fortsetzung...

Laden von Modellen aus GitHub, Fortsetzung

GitHub Repository Verbindung

Mit den folgenden Aktionen legt man für die Eclipse EMF - platform einen lokalen Klon des offiziellen Smart-GridReady Git - Repository an, siehe auch Folgeblock: *Clone Git Repository - Screens.*

1. Eintrag bei URI:
https://github.com/SmartGridready/test-repository.git
2. Eintrag bei User:
[eigener user account name auf GitHub]
3. Eintrag bei Password:
[eigenes password auf GitHub]
4. Klicke *Next* Button
5. Prüfe ob im Window *Branch selection* die Zeile *master* angewählt ist, dann klicke *Next* Button
6. Eintrag im Window *Local Destination* bei Directory: *[Pfad zum Verzeichnis auf dem eigenen Rechner, wo die lokale Git-Repo Kopie angelegt werden soll]*, dann klicke den *Finish* Button

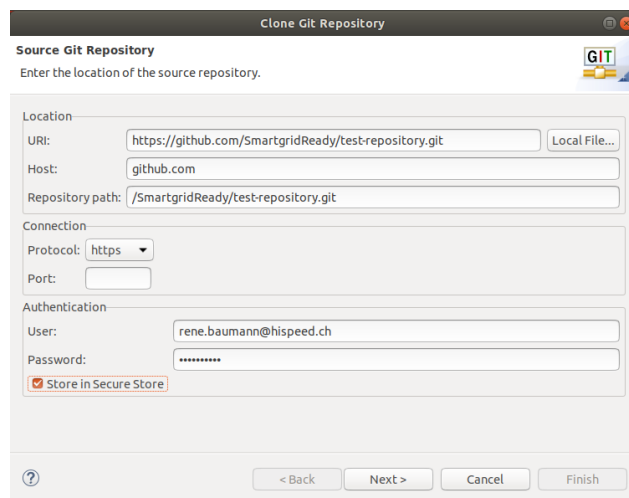


Das SmartGridready Git Repository mit den Modellen ist nun lokal verfügbar um daraus Modelle zum Gebrauch oder Ueberarbeitung in *Eclipse EMF* zu laden.

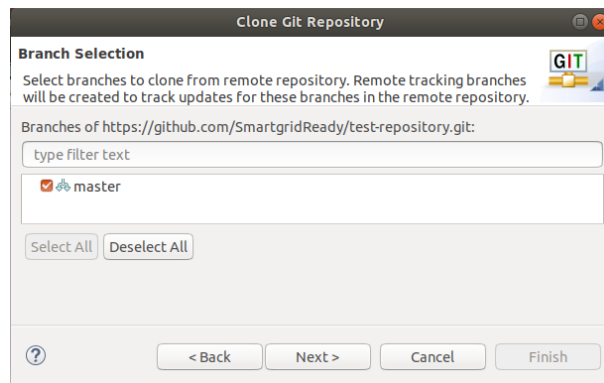
Laden von Modellen aus GitHub, Fortsetzung

Clone Git Repository - Screens

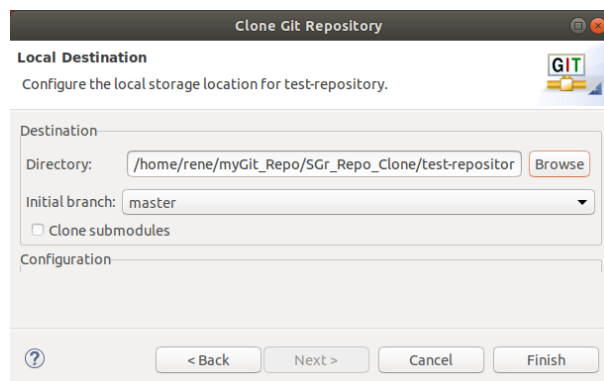
Die folgenden Screen-shoots zeigen den Ablauf des vorherigen Blocks nochmals.



Eintragen von SGr-Repo URI und GitHub User Login, dann *Next*



Klicke *Next*



Definiere in *Directory*, wo der lokale Clone angelegt wird
Klicke *Finish*

Eclipse EMF: CDO - Konfiguration

Voraussetzungen

Für die erfolgreiche Konfiguration der Eclipse EMF - Plattform müssen folgende Voraussetzungen erfüllt sein:

- **Eclipse EMF - Platform gemäss [5] installiert**

CDO Repository Einführung

Das CDO (Connected Data Objects) Model Repository dient den EMF - Modellen als Basis für die Speicherung der Modelleigenschaften oder bei deren Instanziierung als Speicher für die Objektdaten.

CDO kann auch die Zusammenarbeit der Entwickler unterstützen, indem bei einer Änderung gemeinsam verwendeter Modelle alle Nutzer sofort informiert werden, falls ihr importiertes und verwendetes Modell nicht länger identisch mit dem zentralen Modell ist. Diese Funktionalität bedingt jedoch einen zentralen CDO - Datenserver, was aktuell SmartGridredy nicht unterstützt.

Mehr Informationen zur Funktion und den Möglichkeiten findet man bei: [CDO Introduction](#)

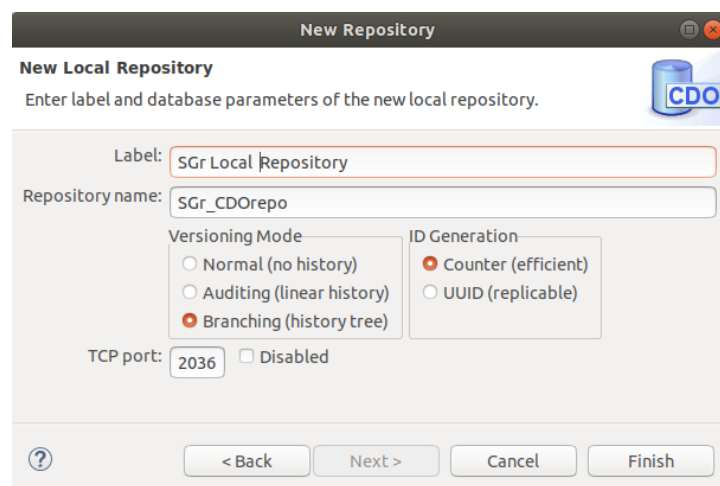
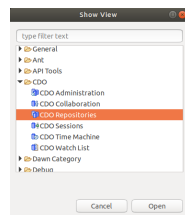
Fortsetzung...

Eclipse EMF: CDO - Konfiguration, Fortsetzung

CDO Repository Erstellung

Die folgenden Aktionen erstellen ein **lokales** CDO Repository mit Versionierung und Mehrfachzugriff über port 2036.

1. Klicke Menu: *Window => Show View => Other...*
2. Klicke Menu: *CDO => CDO Repositories => Open*
3. Klicke auf *[+]* Symbol in der Titelzeile des neuen Reiters *CDO Repositories* um ein Repository hinzuzufügen
4. Wähle im Window *Create Repository* die Option *Create a new local repository* klicke dann auf *Next*
5. Trage im Feld Label: *SGr Local Repository* ein
6. Trage in Repository name: *SGr_CDOr Repo* ein
7. Wähle Optionen: *Branching* und *Couter*
8. Enable Option: *TCP port: 2036* und klicke *Finish*



Eclipse EMF: CDO - Konfiguration, Fortsetzung

CDO Repository Aktivierung

Die Aktionen aktivieren das vorgängig kreierte CDO Repository *SGr Local Repository*.

1. Rechts-klick auf das CDO - Repository *SGr Local Repository* im Reiter *CDO Repositories*
2. Klicke im Popup - Menu *Connect*
3. Eintrag bei Password:
[eigenes password auf GitHub]

Das Speichersymbol vor dem Repository - Namen sollte von zuvor grau, auf jetzt hellblau gewechselt haben. Damit ist das CDO Repository für die Verwendung bereit.

Anhang

Hinweis	Nachdem die Eclipse Platform nun konfiguriert ist, können und EMF Modellen geladen und erstellt werden. Wie dies gemacht wird, ist im Handbuch <i>SmartGridready Eclipse Platform User Guide</i> nachzulesen.
---------	---

Eclipse	t.b.c
---------	-------
