

Entwicklerdokumentation für das CONSENS SysML 1.3 Profil

Verfasser: Alexander Nyßen, Markus Mühlbrandt (Itemis AG)

Version: 1.0

Stand: 16.04.2013

Inhaltsverzeichnis

1 Übersicht des Projekts.....	3
2 Stereotypen des CONSENS-Profiles.....	4
2.1 Environment Definition und Active Structure Definition.....	4
2.1.1 Knoten.....	4
2.1.2 Beziehungen.....	5
2.2 Environment und Active Structure	6
2.2.1 Knoten.....	6
2.2.2 Beziehungen.....	6
2.3 Functions.....	7
2.3.1 Knoten.....	7
2.3.2 Beziehungen.....	7
2.4 ApplicationScenario.....	9
2.4.1 Knoten.....	9
2.4.2 Beziehungen.....	9
2.5 Requirements	9
2.5.1 Knoten.....	9
2.5.2 Beziehungen.....	9
3 Mapping der CONSENS DSL Elemente auf die SysML Profil Stereotypen	10
4 Diagramme und Toolboxen	11
4.1 Diagram-Profil	11
4.2 Toolbox-Profil.....	12
5 Export von Profilen in eine EA MDG Technology	13

1 Übersicht des Projekts

Die Beauftragung umfasst die Erstellung eines Profils für CONSENS auf Basis der SysML 1.3. Das Profil wurde mit dem Enterprise Architect 10 (im Folgenden als EA abgekürzt) als sogenannte MDG Technology erstellt. Dadurch kann das Profil in den EA importiert und dort verwendet werden. Die Erstellung des Profils wurde anhand des Pedelec-Beispiels durchgeführt, welches in der Benutzerdokumentation für das CONSENS-SysML-Profil näher beschrieben ist [ITE13].



Abbildung 1: Elemente des CONSENS-Profiles

Wird das Pedelec-Beispiel mit dem EA geöffnet sind die Elemente des CONSENS-Profiles im Project Browser ersichtlich (Abbildung 1). Im Ordner „CONSENS Profile“ ist das eigentliche SysML-Profil definiert. Im Paket „ConsensDiagrams“ sind alle mit dem Profil erstellbaren Diagrammtypen definiert. In den weiteren Paketen mit dem Präfix „Toolbox“ sind die in den Diagrammen verwendeten Toolboxes (Werkzeuge) enthalten.

2 Stereotypen des CONSENS-Profiles

Das CONSENS-Profil ist eine Erweiterung der SysML1.3 ([OMG10] Kapitel 18). Es nutzt neben einigen neu definierten Stereotypen auch vorhandene SysML- oder UML4SysML-Elemente. Das Profil wird über ein Profildiagramm abgebildet, das sich im Ordner „CONSENS Profile“ im Paket „<<profile>> CONSENS“ befindet.

In den folgenden Unterkapiteln sind die als Knoten oder Beziehungen verwendbaren CONSENS-Stereotypen sowie die direkt aus der SysML beziehungsweise UML4SysML verwendbaren Stereotypen/Metaklassen für jeden Diagrammtyp des CONSENS-Profiles angeführt. Weitere Informationen zu diesen Stereotypen können aus der SysML-Spezifikation [OMG11] entnommen werden.

Eine tabellarische Darstellung der CONSENS-Modellelemente und deren Abbildungen auf das CONSENS-Profil ist in Kapitel 3 aufgeführt.

2.1 Environment Definition und Active Structure Definition

Im Folgenden sind die Stereotypen, welche in den Environment Definition- und Active Structure Definition-Diagrammen verwendet werden können, aufgeführt, welche die Partialmodelle Umfeld und Wirkstruktur auf Typbene repräsentieren.

2.1.1 Knoten

In Abbildung 2 ist die abstrakte Syntax der Stereotypen *SystemTemplate*, *SystemElementTemplate* (nur Active Structure Definition) und *EnvironmentElementTemplate* (nur Environment Definition) dargestellt. Diese Stereotypen spezialisieren den *SysML1.3::block* Stereotypen [OMG11]. Mit diesen können Vorlagen für das das Gesamtsystem darstellende Systemelement, für alle übrigen Systemelemente des Systems sowie für Systemelemente des Umfelds abgebildet werden.

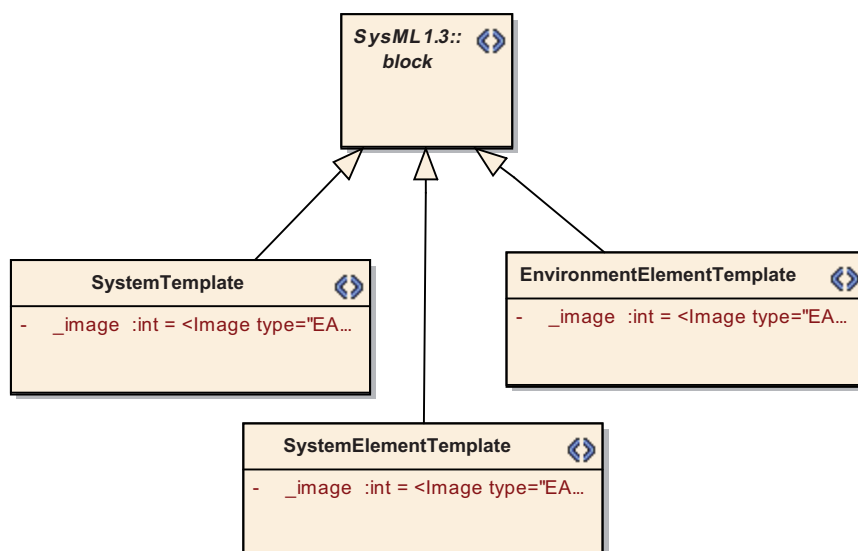


Abbildung 2: Abstrakte Syntax von *SystemTemplate*, *SystemElementTemplate* und *EnvironElementTemplate*

In Abbildung 3 ist die abstrakte Syntax der *FlowSpecification*-Stereotypen dargestellt. Dieser

Stereotyp spezialisiert den *SysML1.3::InterfaceBlock* Stereotypen [OMG11]. Der *FlowSpecification*-Stereotyp ist eine Generalisierung der Stereotypen *EnergyFlowSpecification*, *InformationFlowSpecification* und *MaterialFlowSpecification*. Im Modell des Mechatronic Modellers entsprechen diese Stereotypen einer *FlowSpecification* mit einem entsprechenden *FlowItem*. Zum Beispiel wird der Stereotyp *EnergyFlowSpecification* im Modeller durch eine *FlowSpecification* mit einem *EnergyFlowItem* abgebildet.

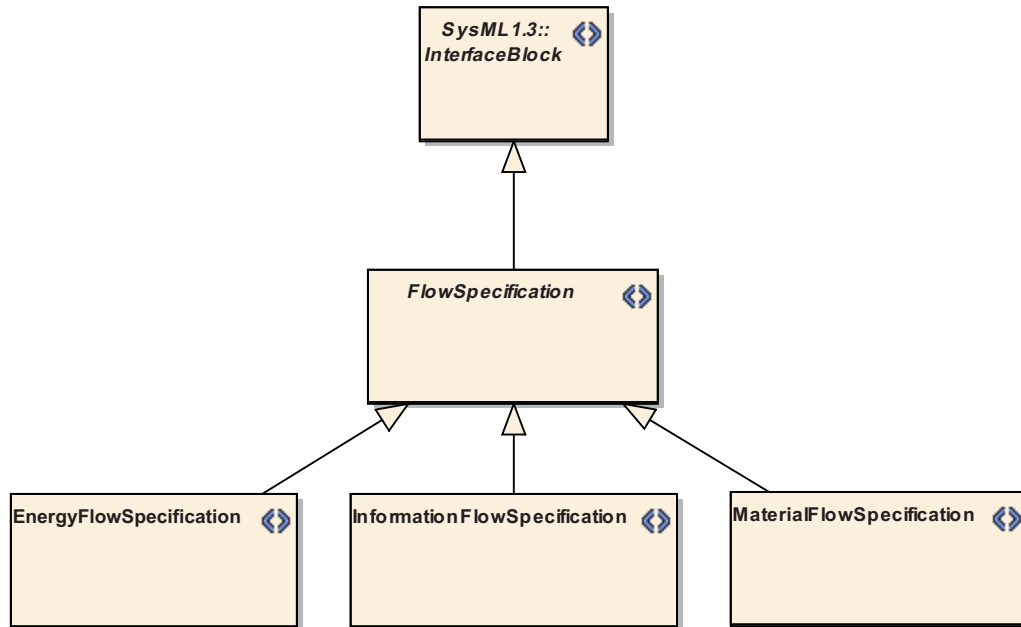


Abbildung 3: Abstrakte Syntax der *FlowSpecifications*

In Abbildung 4 ist die abstrakte Syntax des *MeasurementPoints*-Stereotypen dargestellt. Dieser Stereotyp spezialisiert den *SysML1.3::FullPort*-Stereotypen [OMG11]. Der *FullPort*-Stereotyp wurde in diesem Kontext verwendet, da von einem *FullPort* keine weiteren Beziehungen innerhalb seines Containers weiter verbunden werden können.

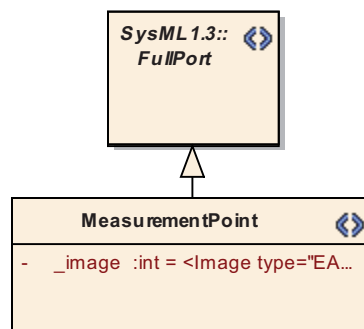


Abbildung 4: Abstrakte Syntax des *MeasurementPoints*

Zusätzlich verwendbare UML4SysML Knoten [OMG10]:

- *UML4SysML::Port* (beide Diagramme)

2.1.2 Beziehungen

Zusätzlich verwendbare UML4SysML Beziehungen [OMG10]:

- *Association*
 - *UML4SysML::Shared Association (Nur ActiveStructureDefinition)*
 - *UML4SysML::Part Association (Nur ActiveStructureDefinition)*

2.2 Environment und Active Structure

Im Folgenden sind die Stereotypen aufgeführt, welche in den Environment und Active Structure Diagrammen verwendet werden können.

2.2.1 Knoten

In Abbildung 5 ist die abstrakte Syntax der Stereotypen *SystemExemplar*, *SystemElementExemplar* (nur Active Structure) und *EnvironmentElementExemplar* (nur Environment) dargestellt. Diese Stereotypen erweitern die *UML4SysML::Property Metaklasse [OMG10]*. Diese Elemente werden im Mechatronic Modeler durch die gleichnamigen Elemente abgebildet.

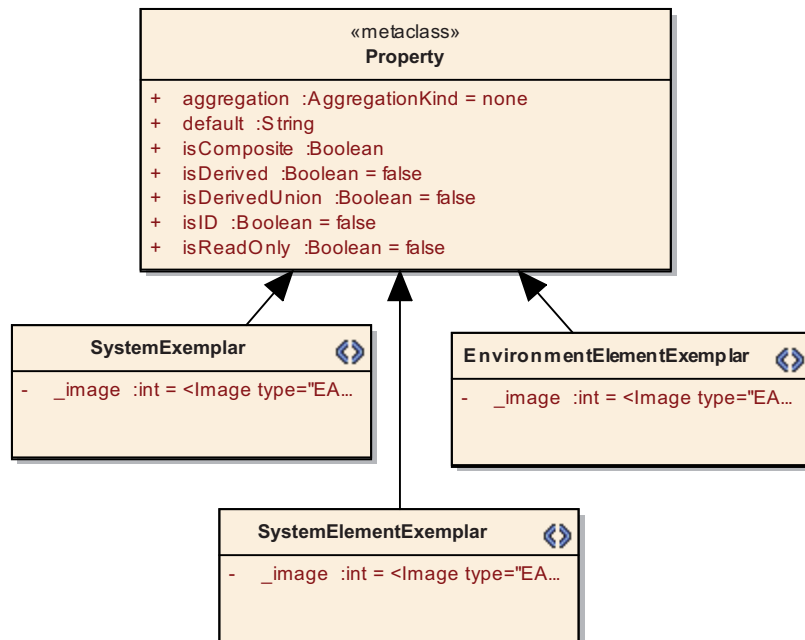


Abbildung 5: Abstrakte Syntax von *SystemExemplar*, *SystemElementExemplar* und *EnvironElementExemplar*

2.2.2 Beziehungen

In Abbildung 6 ist die abstrakte Syntax des Stereotypen *Flow* dargestellt. Dieser Stereotyp spezialisiert den *SysML1.3::itemFlow*-Stereotypen [OMG11].

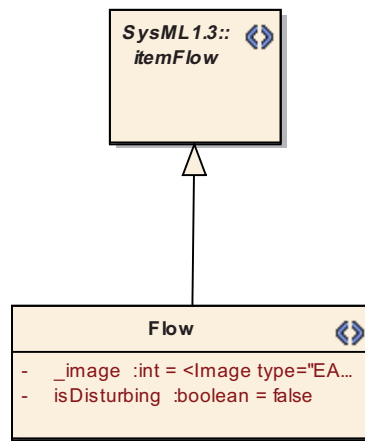


Abbildung 6: Abstrakte Syntax eines Flow

In Abbildung 7 ist die abstrakte Syntax des Stereotypen *LogicalRelation* dargestellt. Dieser Stereotyp erweitert die *UML4SysML::Connector*-Metaklasse [OMG10].

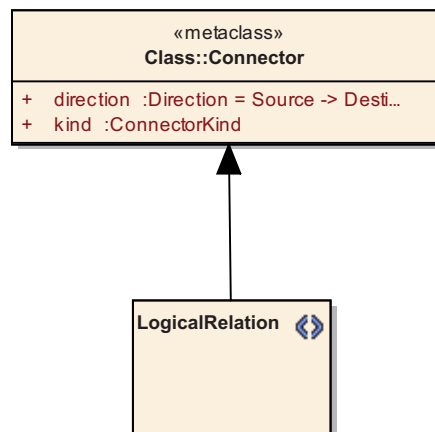


Abbildung 7: Abstrakte Syntax einer LogicalRelation

2.3 Functions

Im Folgenden sind die Stereotypen aufgeführt, welche im Functions Diagramm verwendet werden können.

2.3.1 Knoten

In Abbildung 8 ist die abstrakte Syntax des Stereotypen *Function* dargestellt. Dieser Stereotyp erweitert die *UML4SysML::Class* Metaklasse [OMG10]. Sie dient zu Abbildung von Funktionen.

2.3.2 Beziehungen

In Abbildung 9 ist die abstrakte Syntax des Stereotypen *FunctionContainment* dargestellt. Dieser Stereotyp erweitert die Metaklasse *Composition*. Dies Metaklasse ist EA-spezifisch. Im Modell wird dafür eine *UML::Association* [OMG10] verwendet.

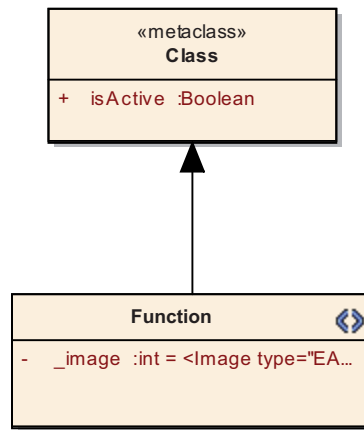


Abbildung 8: Abstrakte Syntax einer Function

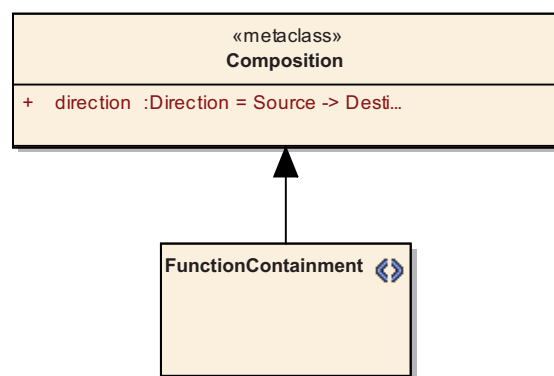


Abbildung 9: Abstrakte Syntax einer FunctionContainment Beziehung

In Abbildung 10 ist die abstrakte Syntax des Stereotypen *Induce* dargestellt. Dieser Stereotyp erweitert die Metaklasse *Abstraction* [OMG10].

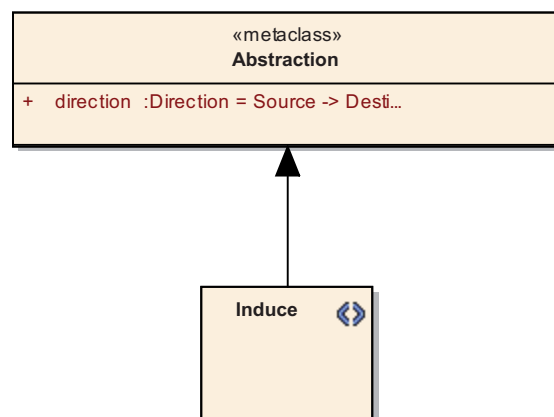


Abbildung 10: Abstrakte Syntax einer Induce Beziehung

Zusätzlich verwendbare SysML Beziehungen [OMG11]:

- *SysML::Satisfy*
Diese Beziehung bildet eine CONSENS *FunctionalRealization* ab.
- *SysML::Refine*
Diese Beziehung bildet ein CONSENS *FunctionalRequirementRefinement* ab.

2.4 ApplicationScenario

Im Folgenden sind die Stereotypen aufgeführt, welche im Application Secenario Diagramm verwendet werden können.

2.4.1 Knoten

In Abbildung 11 ist die abstrakte Syntax des Stereotypen *ApplicationScenario* dargestellt. Dieser Stereotyp erweitert die *UML4SysML::UseCase* Metaklasse [OMG10].



Abbildung 11: Abstrakte Syntax eines ApplicationScenarios

Zusätzlich verwendbare UML4SysML Knoten [OMG10]:

- Actor
- Boundary
 - Dieser Typ ist EA spezifisch und wird nur rein graphisch dargestellt.

2.4.2 Beziehungen

Als Beziehungen stehen zur Verknüpfung von *Actors* und *ApplicationScenarios* sogenannte „Use Case Links“ zur Verfügung. Diese werden als *UML4SysML::Association* [OMG10] abgebildet.

Dazu können noch *Refinement*-Referenzen auf weitere Elemente gesetzt werden. Diese werden durch die *SysML1.3::Refine*-Beziehung [OMG11] abgebildet.

2.5 Requirements

Im Folgenden sind die Stereotypen aufgeführt, welche im Requirements Diagramm verwendet werden können.

2.5.1 Knoten

Als *Requirement*-Knoten wird in diesem Diagramm ein *SysML::Requirement* [OMG11] verwendet, welches standardmäßig Bestandteil der SysML ist.

2.5.2 Beziehungen

Um Requirements hierarchisch strukturieren zu können, wird eine Nesting Beziehung genutzt. Diese wird im Modell durch einen *UML4SysML::NestedClassifier* [OMG10] abgebildet

3 Mapping der CONSENS DSL Elemente auf die SysML Profil Stereotypen

Die folgende Tabelle zeigt wie die Elemente aus CONSENS auf die Stereotypen des CONSENS-SysML-Profiles bzw. auf die Stereotypen/Metaklassen der SysML mappen. Wenn es für einen CONSENS-Element keinen CONSENS-SysML-Stereotypen gibt, wird dieser direkt durch einen SysML-Stereotypen oder eine UML4SysML-Metaklasse abgebildet (z.B. bei einem *Port*).

CONSENS Element	CONSENS SysML Profil Stereotypen	SysML Stereotypen UML Metaklassen
ApplicationScenario	ApplicationScenario	UML::UseCase
EnvironmentElementExemplar	EnvironmentElementExemplar	UML::Property
EnvironmentElementTemplate	EnvironmentElementTemplate	SysML1.3::Block
Flow	Flow	SysML1.3::itemFlow
FlowSpecification with Energy FlowItem	EnergyFlowSpecification	SysML1.3::InterfaceBlock
FlowSpecification with Information FlowItem	InformationFlowSpecification	SysML1.3::InterfaceBlock
FlowSpecification with Material FlowItem	MaterialFlowSpecification	SysML1.3::InterfaceBlock
Function	Function	UML::Class
FunctionalDecompositionl	FunctionContainment	UML::Association (EA::Composition)
FunctionalRealization	-	SysML1.3::Satisfy
FunctionalRequirementRefinement	-	SysML1.3::Refine
FunctionInduction	Induce	UML::Abstraction
LogicalRelation	LogicalRelation	UML::Connector
-	MeasurementPoint	SysML1.3::FullPort
Port	-	UML::Port
Requirement	-	SysML1.3::Requirement
RequirementsInduction	Induce	UML::Abstraction
RequirementsRealization	-	SysML1.3::Satisfy
SystemElementExemplar	SystemElementExemplar	UML::Property
SystemElementTemplate	SystemElementTemplate	SysML1.3::Block
SystemExemplar	SystemExemplar	UML::Property
SystemTemplate	SystemTemplate	SysML1.3::Block

4 Diagramme und Toolboxes

4.1 Diagram-Profil

Um eigene Diagrammtypen im EA zu definieren, muss ein Profil mit einem UML-Klassendiagramm erstellt werden. Dort können die neuen Diagramme als Stereotypen erstellt werden, die dann schon vorhandene EA Diagrammtypen, dargestellt als Metaklassen, erweitern. Abbildung 13 zeigt die Diagrammdefinitionen für das CONSENS-Profil. Zum Beispiel erweitert das neue *ActiveStructureDefinition*-Diagramm das *Diagram_Logical*-Diagramm (Klassendiagramm im EA).

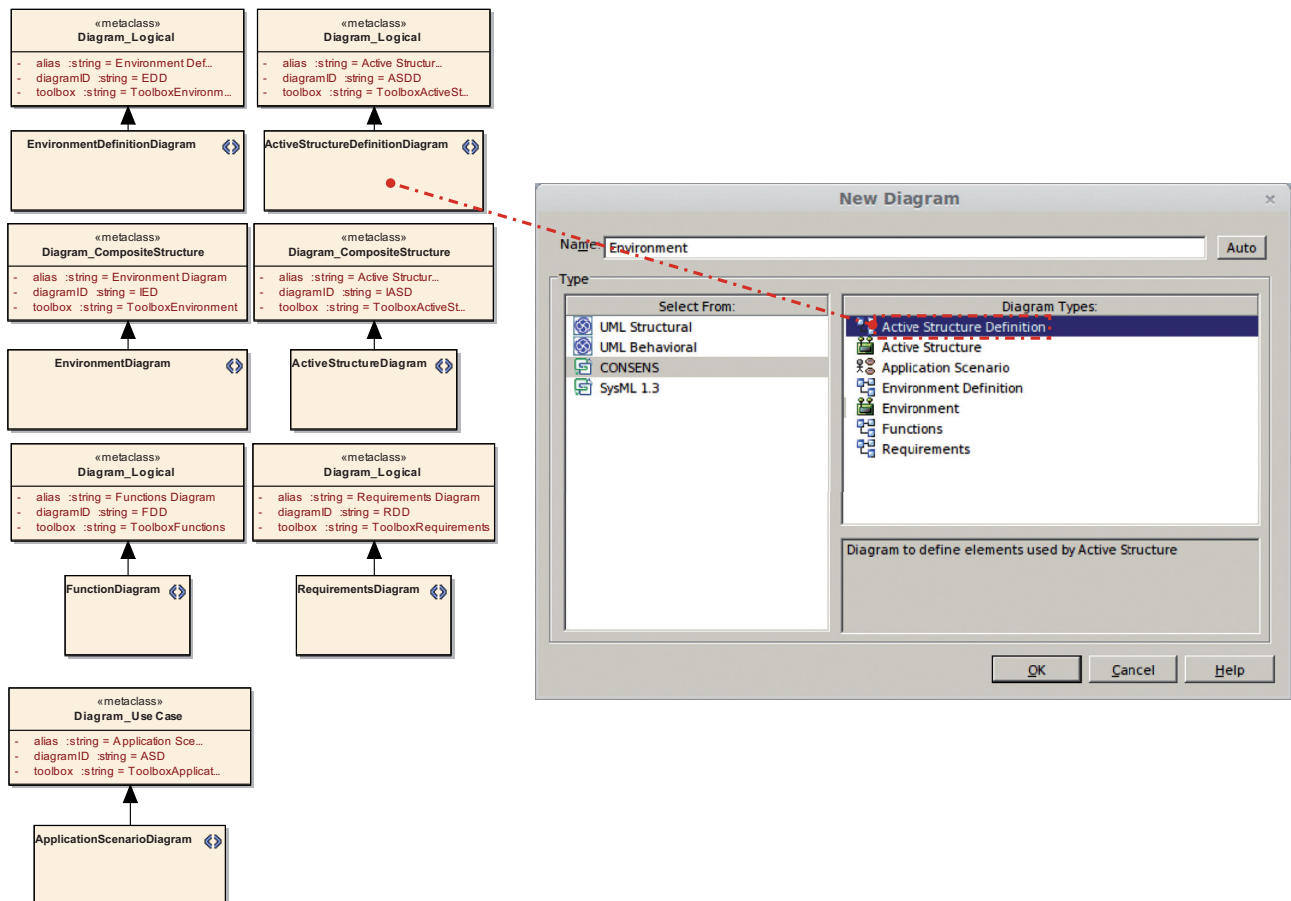


Abbildung 12: Diagram Definitionen im EA

Zu einem Diagramm kann über die Property „toolbox“, welches in der Metaklasse definiert werden muss, noch eine benutzerdefinierte Toolbox an das Diagramm gebunden werden. Diese muss natürlich zuvor im Profil definiert worden sein. Die Toolbox wird dann automatisch beim Öffnen des Diagramms mit geöffnet.

So definierte Diagramme zeigt der EA dann unter dem Profilnamen im „New Diagram“ Wizard an (siehe auch [Spa13]).

4.2 Toolbox-Profil

Selbst erstellte Diagramme können zusätzlich mit eigenen Toolboxes versehen werden. Das Diagramm in Abbildung 13 zeigt die Toolboxdefinitionen für das *ActiveStructure*-Diagramm. Wie bei Diagramm-Profilen müssen auch in diesem Fall in einem UML Klassendiagramm die Definitionen vorgenommen werden. Für jede Kategorie in der Toolbox ist ein eigener Stereotyp angelegt. Der angezeigte Name der Kategorie wird entweder direkt vom Namen oder vom Alias des Stereotypen übernommen.

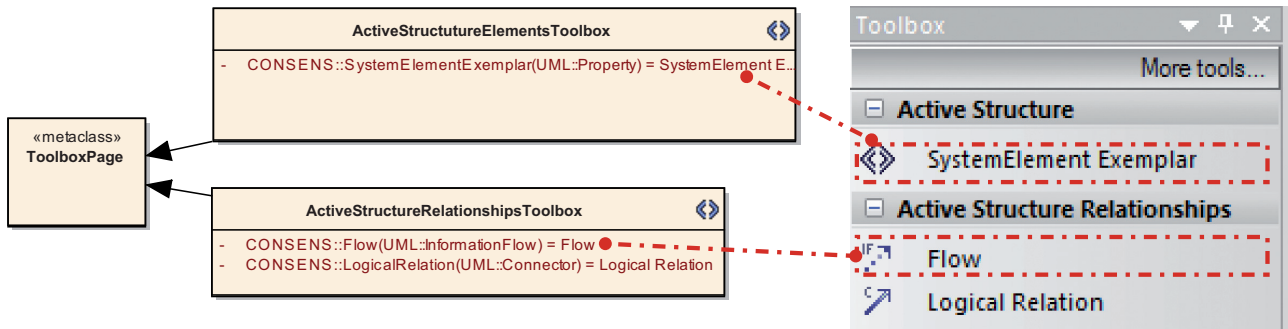


Abbildung 13: Toolbox Definitionen und Darstellung der Toolbox im EA

Für jeden Tooleintrag muss der voll qualifizierte Name des zu erstellenden Profil-Stereotypen sowie der UML Metaklasse, auf der der Stereotyp basiert, über das „Name“-Property gesetzt werden. Den Namen, der für den Tooleintrag in der Toolbox angezeigt werden soll, entnimmt der EA aus dem „Initial“-Property (vgl. Abbildung 14). Siehe auch [Spa13].

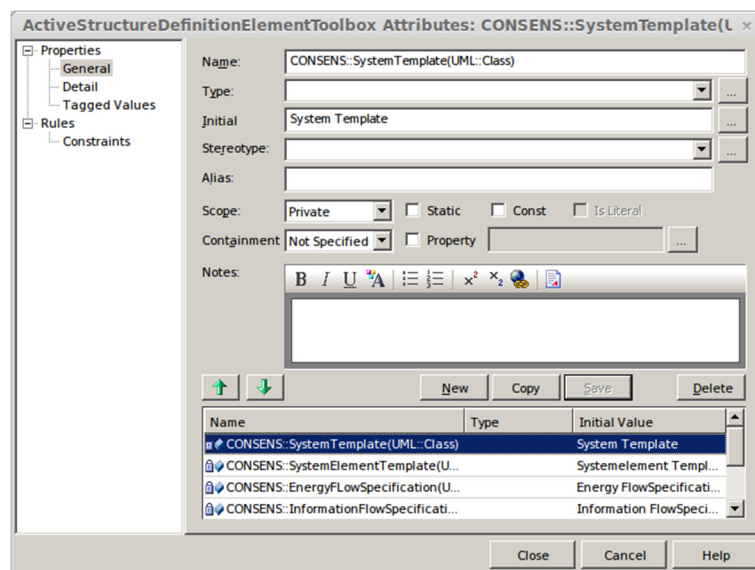


Abbildung 14: Attribute eines Toolboxeintrages

5 Export von Profilen in eine EA MDG Technology

Profile können im EA als sogenannte Model Driven Generation (MDG) Technology exportiert werden, um sie in weiteren EA Installationen insbesondere auf weiteren Computern zu nutzen.

Dazu sind im Fall von CONSENS das eigentliche CONSEN-Profil, das Diagramm-Profil und die Toolbox-Profile notwendig. Diese Profile müssen als erstes als UML-Profile in eine XML Datei exportiert werden. Dazu wählt man im Kontextmenü des jeweiligen „Profil-Packages“ den Punkt „Save Package As UML Profile“ und wählt im angezeigten Dialog einen passenden Namen für die Datei aus.

Sind alle „Packages“ als Profil exportiert worden, sollten im Fall von CONSENS neun Dateien entstanden sein. Jeweils eine für das CONSENS-Profil und die Diagramme und weitere sieben für die Toolboxes.

Mit Hilfe eines Wizards (Tools → Generate MDG Technology File) kann nun die MDG Technology Datei erstellt werden.



Abbildung 15: Wizard zum Erzeugen von MDG Technologies

Auf der zweiten Seite des Wizards kann eine sogenannte MTS Datei erzeugt werden welche die Einstellungen und weitere Modellinformationen (Profilname sowie die verwendeten Profildateien) speichert. Diese kann später wieder geladen werden um diese Angaben nicht wieder neu vornehmen zu müssen (Siehe Abbildung 16).

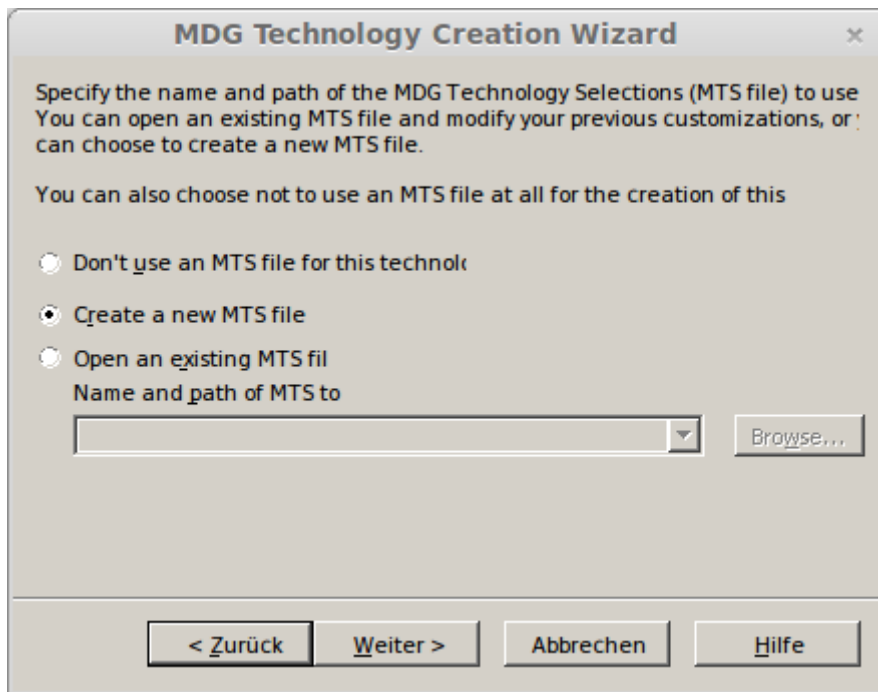


Abbildung 16: Erzeugung einer MTS Datei

Wenn eine MTS Datei verwendet werden soll, muss in der folgenden Wizard Seite noch ein Dateiname angegeben werden.

In der anschließenden Wizard Seite können Informationen zum Profil angegeben werden, wie zum Beispiel der Name, die Version oder eine kurze allgemeine Beschreibung (Abbildung 17). Bei Änderungen muss die Version erhöht werden.



Abbildung 17: Zusatzinformationen zum Profil

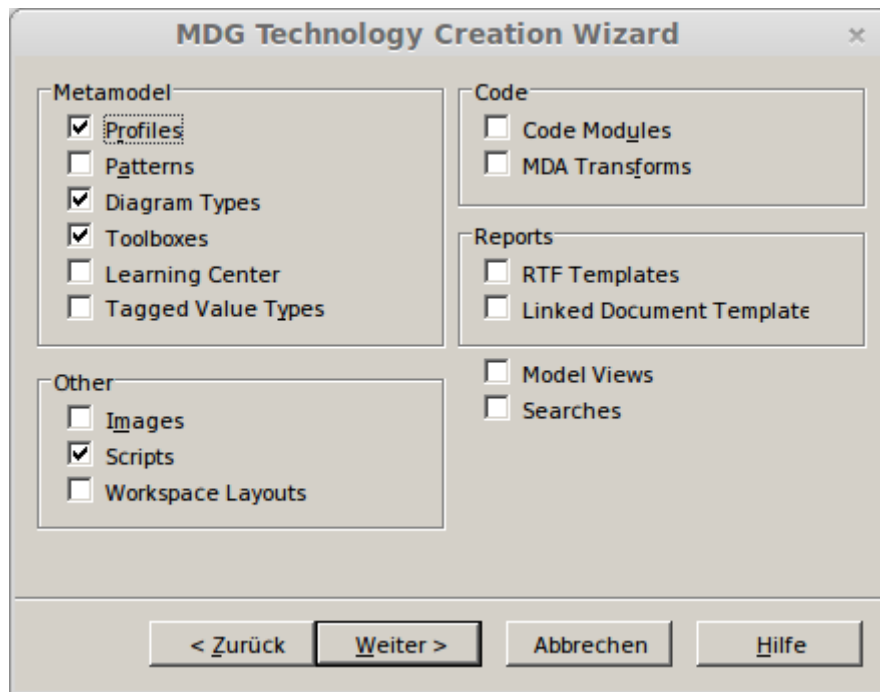


Abbildung 18: Angabe der im Profil genutzten Elemente

In der nächsten Seite müssen die vom Profil genutzten Elemente angegeben werden. Im Fall von CONSENS sind das Profiles, Diagram Types, Toolboxes und Scripts (Abbildung 18). Die Skripte sind direkt bei den Stereotypen des CONSENS-Profiles hinterlegt.

In den folgenden drei Seiten müssen nun die Profil-Dateien angegeben werden, die der Wizard verwenden soll. Diese wurden zuvor bereits als UML Profil exportiert. Auf der ersten Seite werden die zu nutzenden UML Profile angegeben, auf der Zweiten die Diagramm-Profile, und auf der Dritten die Toolbox-Profile.

Wenn der Wizard bis zum Ende durch gearbeitet wurde, wird mit einem Klick auf „Fertig“ das Profil als MDG Technology XML Datei erstellt (Siehe auch [Spa13]).

Abbildungsverzeichnis

Abbildung 1: Elemente des CONSENS-Profiles.....	3
Abbildung 2: Abstrakte Syntax von SystemTemplate, SystemElementTemplate und EnvironElementTemplate	4
Abbildung 3: Abstrakte Syntax der FlowSpecifications.....	5
Abbildung 4: Abstrakte Syntax des MeasurementPoints.....	5
Abbildung 5: Abstrakte Syntax von SystemExemplar, SystemElementExemplar und EnvironElementExemplar.....	6
Abbildung 6: Abstrakte Syntax eines Flow	7
Abbildung 7: Abstrakte Syntax einer LogicalRelation.....	7
Abbildung 8: Abstrakte Syntax einer Function.....	8
Abbildung 9: Abstrakte Syntax einer FunctionContainment Beziehung.....	8
Abbildung 10: Abstrakte Syntax einer Induce Beziehung.....	8
Abbildung 11: Abstrakte Syntax eines ApplicationScenarios	9
Abbildung 12: Diagram Definitionen im EA.....	11
Abbildung 13: Toolbox Definitionen und Darstellung der Toolbox im EA.....	12
Abbildung 14: Attribute eines Toolboxeintrages.....	12
Abbildung 15: Wizard zum Erzeugen von MDG Technologies.....	13
Abbildung 16: Erzeugung einer MTS Datei	14
Abbildung 17: Zusatzinformationen zum Profil	14
Abbildung 18: Angabe der im Profil genutzten Elemente.....	15

Literaturverzeichnis

[ite13]	ITEMIS AG: Benutzerdokumentation für das CONSENS SysML 1.3 Profil, 2013
[Spa13]	SPARX SYSTEMS: Enterprise Architect User Guide for Enterprise Architect V10, 2013
[OMG11]	OBJECT MANAGEMENT GROUP (OMG):, Systems Modeling Language (OMG SysML™), Version 1.3, June 2011
[OMG10]	OBJECT MANAGEMENT GROUP (OMG): Unified Modeling Language, Superstructure, 2010