



Software Plattform Embedded Systems 2020

GEFÖRDERT VOM



Bundesministerium
für Bildung
und Forschung

- ZP-AP2 Teilergebnis: UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML-

Version: 1.2

Projektbezeichnung	SPES 2020	
Verantwortlich	Bastian Tenbergen (Universität Duisburg-Essen)	
QS-Verantwortlich	Tayfun Gezgin (OFFIS e.V.)	
Erstellt am	05.09.2010	
Zuletzt geändert	13.10.2010 11:14	
Freigabestatus		Vertraulich für Partner: <Partner1>; <Partner2>; ...
		Projektöffentlich
	X	Öffentlich
Bearbeitungszustand		in Bearbeitung vorgelegt
	X	fertig gestellt

Weitere Produktinformationen

Erzeugung	Bastian Tenbergen (BT), Marian Daun (MD)
Mitwirkend	Ernst Sikora (ES), Heiko Stallbaum (HS), Tayfun Gezgin (TG)

Änderungsverzeichnis

Änderung			Geänderte Kapitel	Beschreibung der Änderung	Autor	Zustand
Nr.	Datum	Version				
1	05.09.10	0.1	Alle	Initiale Produkterstellung	BT	In Bearbeitung
2	07.09.10			Interner Review	ES	In Bearbeitung
3	07.09.10	1.0	Alle	Editorielle Überarbeitung	BT	Vorgelegt
4	30.09.10			Externer Review	TG	Vorgelegt
5	05.10.10	1.1	Alle	Überarbeitung nach externem Review	BT	In Bearbeitung
6	07.10.10			Interner Review	HS	In Bearbeitung
7	13.10.10	1.2	Alle	Überarbeitung	BT	Fertig gestellt

UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML

Kurzfassung

In diesem Dokument wird ein UML/SysML-Profil zur Zielmodellierung auf Basis der Systems Modeling Language (SysML) und Unified Modeling Language (UML) vorgestellt. Dieses Profil wurde im Rahmen der Ausarbeitung zweier Fallstudien in Kooperation mit Bosch und EADS entwickelt und basiert auf den Ausführungen zur Zielmodellierung aus dem initialen, modellbasierten Requirements Engineering Ansatz für Embedded Systems ([LaSiStTe09]). Die Studie zum Stand der Praxis zum modellbasierten Requirements Engineering ([LaGaSiTe10], [DaSiLa10]) hat gezeigt, dass die Zielmodellierung derzeit nur eine geringe Rolle im modellbasierten Requirements Engineering spielt. Ferner wurde gezeigt, dass durch die systematische Verfeinerung von Anforderungen auf mehreren Abstraktionsebenen Vorteile für das Requirements Engineering entstehen können. Zielen liegen Anforderungen zu Grunde. Für die systematische Erhebung und Verfeinerung von Anforderungen ist es daher unerlässlich, zunächst Ziele zu spezifizieren. Im modellbasierten Requirements Engineering werden Ziele in Zieldiagrammen ([Pohl10]) modelliert, wie beispielsweise die KAOS-Methode es vorsieht ([KAOS07], [Lamsweerde09]).

UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML

Inhalt

1	Einordnung und Kurzbeschreibung	5
1.1	Motivation und Einordnung.....	5
1.2	Management Summary	5
1.3	Überblick	5
2	Einführung zur KAOS-Zielmodellierung und UML/SysML	6
2.1	KAOS-Zielmodellierung	6
2.2	UML/SysML.....	6
2.3	Hinweise zur Implementierung des SysML-Profiles mit EA.....	7
3	Beschreibung des MDG-Plug-Ins zur Zielmodellierung nach KAOS in SysML....	8
3.1	Überblick über das UML/SysML-Profil.....	8
3.2	Elemente der Zielmodelle	13
3.3	Beziehungen zwischen den Elementen.....	14
3.4	Die Enumerationen „GoalType“ und „ContributionType“	15
4	Verwendung des MDG-Plug-Ins.....	16
4.1	Installation der MDG-Technologie	16
4.2	Verwendung des Profils.....	16
4.3	Erweiterung des Profils.....	18
5	Zusammenfassung	19
6	Literaturverzeichnis	20

1 Einordnung und Kurzbeschreibung

1.1 Motivation und Einordnung

Das vorliegende Ergebnisdokument beschreibt das Enterprise Architect Plug-In „SysML Goal Modeling“, welches ein UML/SysML-Profil zur Modellierung von Zielen nach der KAOS-Methode beinhaltet. Dieses Dokument beschreibt den Aufbau des Plug-Ins, gibt eine Beschreibung der Architektur des UML/SysML-Profiles und erläutert die Verwendung des Profils in Enterprise Architect (EA).

1.2 Management Summary

In diesem Dokument wird ein UML/SysML-Profil zur Zielmodellierung auf Basis der Systems Modeling Language (SysML) und Unified Modeling Language (UML) vorgestellt. Dieses Profil wurde im Rahmen der Ausarbeitung zweier Fallstudien in Kooperation mit Bosch und EADS entwickelt und basiert auf den Ausführungen zur Zielmodellierung aus dem initialen, modellbasierten Requirements Engineering Ansatz für Embedded Systems ([LaSiStTe09]). Die Studie zum Stand der Praxis zum modellbasierten Requirements Engineering ([LaGaSiTe10], [DaSiLa10]) hat gezeigt, dass die Zielmodellierung derzeit nur eine geringe Rolle im modellbasierten Requirements Engineering spielt. Ferner wurde gezeigt, dass durch die systematische Verfeinerung von Anforderungen auf mehreren Abstraktionsebenen Vorteile für das Requirements Engineering entstehen können. Zielen liegen Anforderungen zu Grunde. Für die systematische Erhebung und Verfeinerung von Anforderungen ist es daher unerlässlich, zunächst Ziele zu spezifizieren. Im modellbasierten Requirements Engineering werden Ziele in Zieldiagrammen ([Pohl10]) modelliert, wie beispielsweise die KAOS-Methode es vorsieht ([KAOS07], [Lamsweerde09]).

1.3 Überblick

Im folgenden Abschnitt 2 wird eine kurze Einführung zur Zielmodellierung nach der KAOS-Methode sowie zu SysML gegeben. In Abschnitt 3 wird das EA-Plug-In „SysML Goal Modeling“ erläutert, indem zunächst der Aufbau der zugehörigen EA-Projektdatei dargestellt wird (Abschnitt 3.1) und anschließend das UML/SysML-Profil anhand seiner Implementierung in Enterprise Architect erläutert wird (Abschnitte 3.2 bis 3.4). In Abschnitt 4 wird die Installation, Verwendung und Erweiterung des EA-Plug-In als MDG-Technologie erläutert. Eine Zusammenfassung dieses Dokumentes kann Abschnitt 5 entnommen werden.

2 Einführung zur KAOS-Zielmodellierung und UML/SysML

2.1 KAOS-Zielmodellierung

Ziele beziehen sich auf funktionale Eigenschaften sowie Qualitätseigenschaften, die ein System externen Akteuren (Personen und anderen Systemen) bietet. Ziele werden auf verschiedenen Abstraktionsstufen definiert und abstrahieren von technischen Einzelheiten, Anforderungen sowie von einer möglichen Systemarchitektur. So kann beispielsweise die beabsichtigte Nutzung des Systems durch einen externen Akteur durch ein Ziel charakterisiert werden.

Ziele verfeinern die für das System definierte Vision (siehe [Pohl10]). Jedes Ziel leistet somit einen Beitrag zur Erfüllung der Systemvision. Gleichzeitig werden Begründungen für detaillierte (lösungsorientierte) Anforderungen an das System durch Ziele gegeben.

KAOS ([KAOS07], [Lamsweerde09]) ist eine Modellierungssprache zur Spezifikation von Zielen und deren Beziehungen. Durch den Einsatz dieser Modellierungssprache ist es im Requirements Engineering möglich, Anforderung anhand von abstrakteren Zielen zu begründen und miteinander in Verbindung zu setzen.

KAOS verfügt über eine Reihe von Modellelementen zur Darstellung von Zielmodellen, sowie über eine Menge von definierten Beziehungen zwischen Zielen. Im vorgeschlagenen UML/SysML-Profil werden die Modellelemente „Ziel“ (Goal) und „Verfeinerung“ (Refinement) betrachtet. Die im Profil betrachteten Beziehungen sind Und-Verfeinerungen (And-Refinements) und Oder-Verfeinerungen (Or-Refinements), sowie Beteiligungs- oder Konfliktbeziehungen (Contribution- oder Contradiction-links). Und-Verfeinerungen spezifizieren, dass alle der Verfeinerung angehörigen Unterziele erfüllt sein müssen, damit das Oberziel erfüllt ist. Oder-Verfeinerungen bedeuten, dass nur eines der (beliebig vielen) alternativen Unterziele erfüllt sein muss, damit das Oberziel erfüllt ist. Beteiligungsbeziehungen geben an, dass die Erfüllung eines Zieles die Erfüllung eines anderen Zieles entweder positiv oder negativ beeinflusst, also vereinfacht oder beeinträchtigt. Eine Konfliktbeziehung zwischen zwei Zielen gibt an, dass die Erfüllung eines Zieles verhindert wird, wenn ein konfliktionäres Ziel erfüllt wird. Die Modellierungssprache KAOS spezifiziert außerdem eine Reihe weiterer Modellelemente und Beziehungen, wie beispielsweise Verantwortlichkeiten oder solche Ziele, die zu vermeiden sind. Diese werden in späteren Revisionen des Zielprofils für UML/SysML betrachtet. Zunächst wurden im vorgeschlagenen UML/SysML-Profil lediglich jene Elemente der KAOS-Methode berücksichtigt, die zur strukturierten Modellierung von Zielen zum Zwecke der Verfeinerung durch Szenarien gemäß dem initialen, modellbasierten RE-Ansatz ([LaSiStTe09]) benötigt werden.

2.2 UML/SysML

SysML ist eine universelle, graphische Modellierungssprache zur Darstellung und Spezifikation von Verhaltens-, Zustands- und Strukturaspekten von Systemen ([FrMoSt09], [OMG10]). SysML baut auf der Unified Markup Language Version 2.0 (UML) auf. UML wird durch SysML domänenunabhängig erweitert und erlaubt es, ein System lösungsunabhängig zu spezifizieren. Eine Besonderheit der SysML gegenüber UML2 ist, dass SysML den Diagrammtyp „Requirements Diagram“ bietet, mit dem es möglich ist, Anforderungen und deren Beziehungen graphisch zu modellieren. Allerdings sieht SysML kein Diagramm zur Modellierung von Zielen vor. Das in diesem Ergebnisdokument beschriebene Profil für SysML stellt eine Erweiterung von UML/SysML zur Modellierung von Zielen nach der KAOS-Methode dar.

2.3 Hinweise zur Implementierung des SysML-Profiles mit EA

Obwohl SysML Goal Modeling als Profil für SysML zu verstehen ist, muss die Implementierung in Enterprise Architect (EA) auf Basis des UML2-Metamodells erfolgen. Die Mechanismen, die EA für die Erstellung von Profilen und MDG-Technologien bereit hält machen dieses notwendig. So muss ein Stereotyp beispielsweise EA-internen Metaklassen wie „Class“ oder „Association“ erben, anstelle von SysML-eigenen Metaklassen wie etwa „Block“. Die Erläuterungen in den Abschnitten 3.2 und 3.3 beziehen sich daher auf die Implementierung des SysML Goal Modeling Profils auf Basis des Metamodells von Enterprise Architect. Da das EA-Metamodell auf das SysML-Metamodell übertragbar ist, können diese Erläuterungen allerdings als äquivalent zu einer Erweiterung des SysML-Metamodells aufgefasst werden.

3 Beschreibung des MDG-Plug-Ins zur Zielmodellierung nach KAOS in SysML

In diesem Abschnitt wird das Enterprise Architect Plug-In beschrieben, in dem das UML/SysML-Profil zur Modellierung von Zielen nach KAOS implementiert wurde. Der nächste Unterabschnitt 3.1 gibt einen Überblick über den Inhalt der entsprechenden Enterprise Architect (EA) Projektdatei. Im darauffolgenden Unterabschnitt 3.2 werden die Modellelemente des UML/SysML-Profiles erläutert. In Unterabschnitt 3.3 werden die Beziehungen, die zwischen den Modellelementen aus Abschnitt 3.1 existieren, erläutert.

3.1 Überblick über das UML/SysML-Profil¹

Wie in Abb. 3–1² ersichtlich ist, besteht die EA-Projektdatei aus vier SysML-Paketen. Diese Pakete werden in den folgenden Abschnitten erläutert. Drei der vier enthaltenen Pakete sind mit dem Stereotypen „<<profile>>“ gekennzeichnet. Diese Kennzeichnung gibt an, dass es sich bei dem Paket um eine benutzererstellte Erweiterung des UML/SysML-Metamodells bzw. des Enterprise Architect Programmumfangs handelt. Details und Erläuterungen zur Erweiterung von Enterprise Architect um benutzererstellte Profilen können aus [Sparx09] und [Sparx10] entnommen werden.

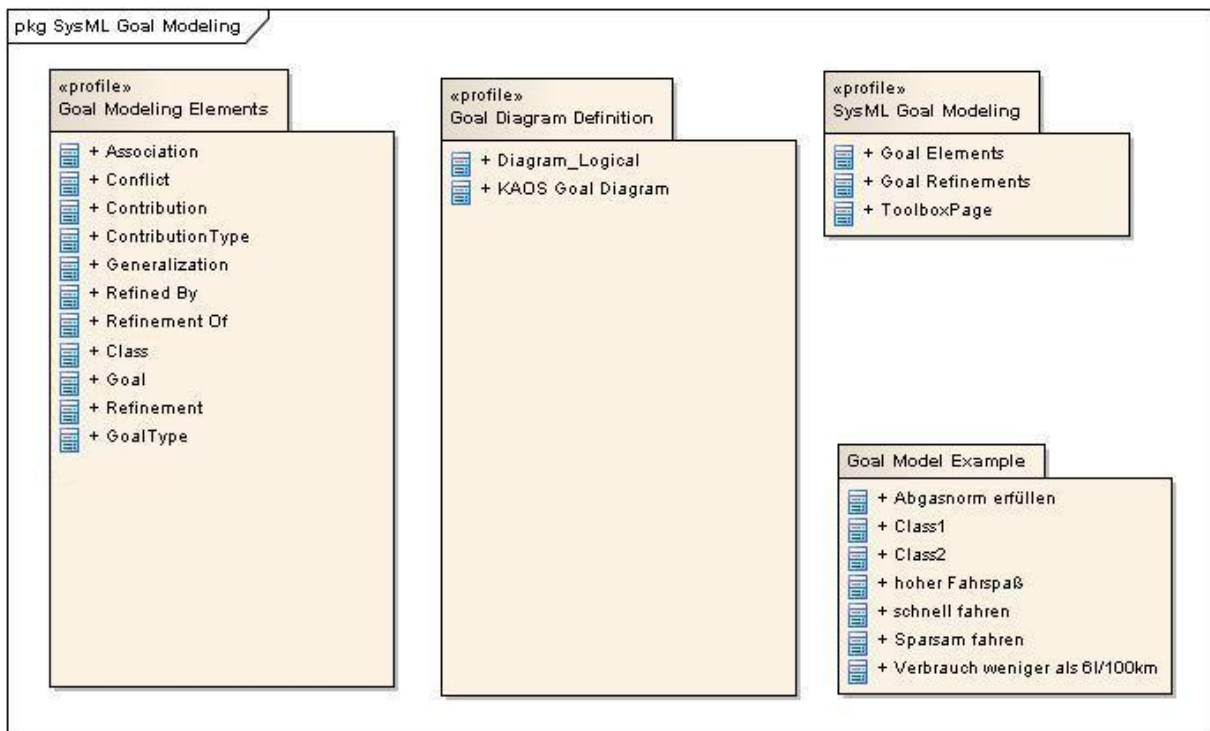


Abb. 3–1 SysML-Paketdiagramm des Inhaltes der EA-Projektdatei

3.1.1 Das Paket „Goal Modeling Elements“

Das Paket „Goal Modeling Elements“ ist vom Stereotyp „<<profile>>“ und implementiert das eigentliche UML/SysML-Profil. In diesem Paket sind alle Modellelemente und Beziehungen der KAOS-Methode als Unterklassen von UML2, bzw. SysML-

¹ Pakete, die der Profildefinition dienen, sind in Englisch beschrieben, während Anwendungsbeispiele auf Deutsch gehalten sind, um so Profildefinition und Anwendung des Profils deutlicher voneinander abzugrenzen.

² Alle Diagramme in diesem Dokument sind aus Enterprise Architect von SparxSystems exportiert.

UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML

Elementen definiert.. Abb. 3–2 zeigt den Inhalt dieses Paketes im Detail. Einzelheiten zum Inhalt können aus den Abschnitten 3.2 bis 3.4 entnommen werden.

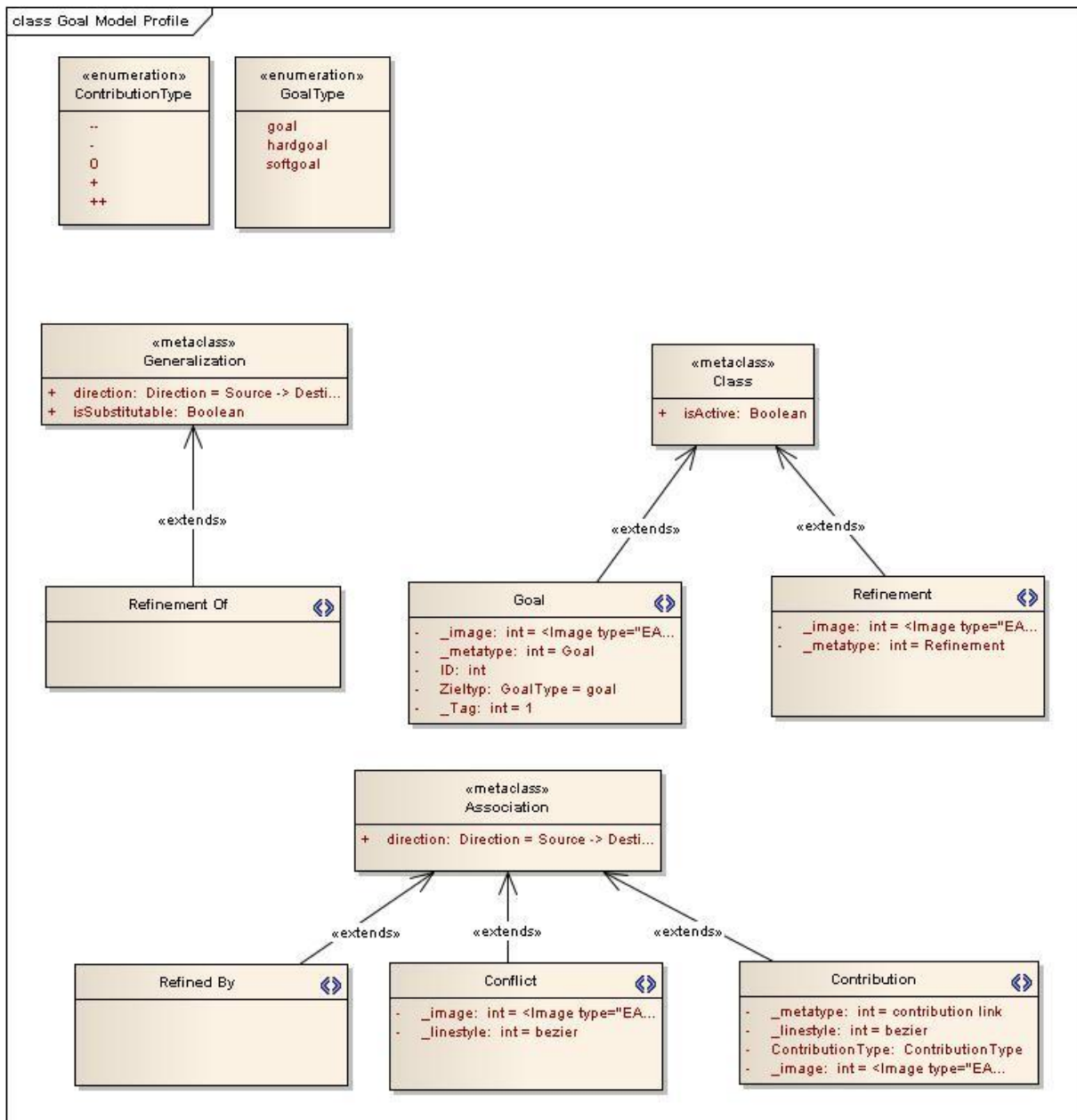


Abb. 3–2 UML-Klassendiagramm des Inhaltes des Paketes „Goal Modeling Elements“

3.1.2 Das Paket „Goal Diagram Definition“

Das Paket „Goal Modeling Elements“ ist ebenfalls vom Stereotyp „<<profile>>“ und implementiert den Diagrammtypen, der zu verwenden ist, wenn in Enterprise Architect Zielmodelle erstellt werden sollen. Dieser Diagrammtyp beruft sich auf die Modellelemente und Beziehungen aus dem Paket „Goal Modeling Elements“ (siehe Abschnitt 3.1.1). Das Paket „Goal Diagram Definition“ ist in Abb. 3–3 dargestellt. Wie in Abb. 3–3 zu sehen ist, besteht das Paket aus zwei Klassen, einer Klasse „Goal Diagram“ und einer Metaklasse „Diagram_Logical“. „Diagram_Logical“ ist eine EA-interne Klasse, von der alle strukturellen Diagrammtypen erben. „Goal Diagram“ erbt von der Metaklasse „Diagram_Logical“ und spezifiziert somit Zieldiagramme als strukturelles Diagramm in Enterprise Architect. In Abb. 3–3 ist zu erkennen, dass „Diagram_Logical“ eine Reihe von Attributen spezifiziert, welche die Eigenschaften

UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML

des neuen Diagrammtypen „Goal Diagram“ festlegen. Eine Erläuterung der Attribute kann aus [Sparx10] entnommen werden. Besonders hervorzuheben ist das Attribut „toolbox“. Dieses Attribut legt die zu verwendende Toolbox im Enterprise Architect Editor fest. Die zu verwendende Toolbox ist im Paket „SysML Goal Modeling“ definiert und wird in Abschnitt 3.1.3 erläutert.

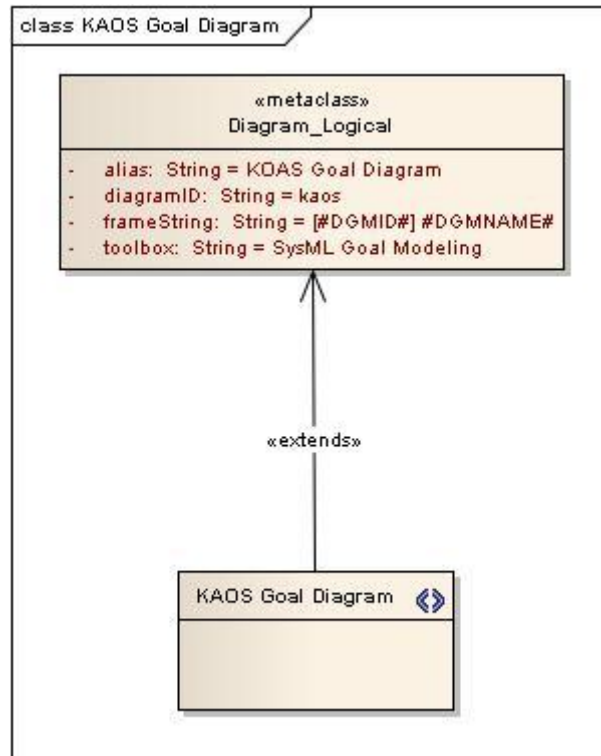


Abb. 3–3 UML-Klassendiagramm des Inhaltes des Paketes „Goal Diagram Definition“

3.1.3 Das Paket „SysML Goal Modeling“

Das Paket „SysML Goal Modeling“ ist ebenfalls vom Stereotyp `«profile»` und implementiert die für den im Paket „Goal Diagram Definition“ definierten Diagrammtypen „Goal Diagram“ (siehe Abschnitt 3.1.2) zu verwendende Toolbox. Die Toolbox ist ein Teil des Enterprise Architect Editors, in dem die für den aktuellen Diagrammtypen verwendbaren Modellelemente und Beziehungen für die Modellierung zur Verfügung gestellt werden. Abb. 3–4 zeigt den Inhalt des Paketes „SysML Goal Modeling“ in Form eines UML-Klassendiagrammes. Die Abbildung zeigt zwei Klassen, die beide von der EA-internen Metaklasse „ToolboxPage“ erben. Diese Klassen implementieren zwei unterschiedliche Sektionen der Toolbox. In einer Sektion „Goal Elements“ werden alle Modellelemente (derzeit Ziele und Verfeinerungen) festgelegt, in der anderen Sektion „Goal Refinements“, alle Beziehungen zwischen den Modellelementen. In den Klassen „Goal Elements“ und „Goal Refinements“ werden Attribute spezifiziert, die die zur Verfügung stehenden User-Interface-Elemente (UI-Elemente) festlegen. Die UI-Elemente repräsentieren die Modellelemente und Beziehungen aus dem Paket „Goal Modeling Elements“ und werden in den Attributen in Paket-Notationsform (siehe dazu [Sparx10] und [OMG10] beschrieben.

UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML

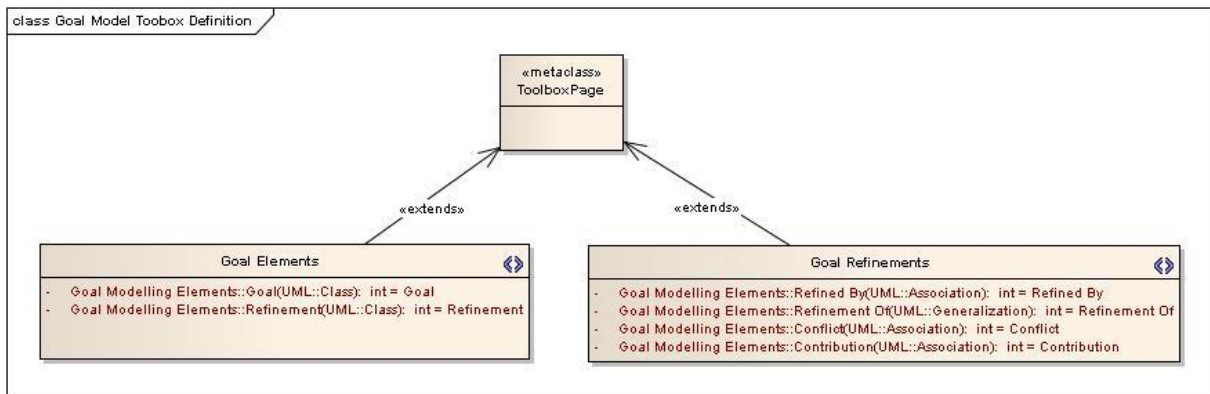


Abb. 3–4 UML-Klassendiagramm des Inhaltes des Paketes „SysML Goal Modeling“

Die folgende Abb. 3–5³ zeigt, wie die Toolbox, die durch das Paket „SysML Goal Modeling“ definiert wird, im User Interface von Enterprise Architect angezeigt wird. Diese Toolbox wird automatisch geöffnet, sobald ein neues Zieldiagramm, wie im Paket „Goal Diagram Definition“ spezifiziert, erstellt wird. Die Toolbox kann ebenfalls in einem beliebigen anderen Diagramm verwendet werden, wenn sie manuell nach einem Aufruf von „More tools...“ (siehe Abb. 3–5) aus dem Menü ausgewählt wird.

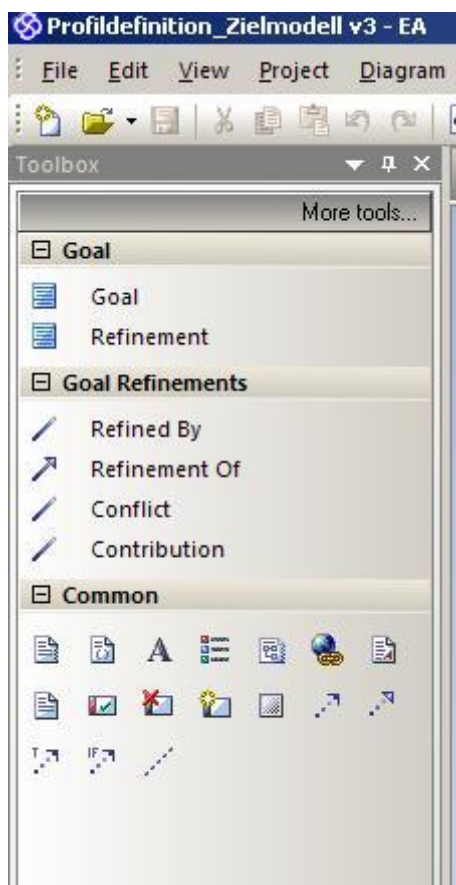


Abb. 3–5 Screenshot der Toolbox, wie sie in Enterprise Architect dargestellt wird

3.1.4 Das Paket „Goal Model Example“

Das Paket „Goal Model Example“ hat keinen Stereotypen und beinhaltet ein Beispiel für die Verwendung der im Paket „Goal Modeling Elements“ spezifizierten Mo-

³ Alle Screenshots in diesem Dokument zeigen Ausschnitte aus dem Programm „Enterprise Architect“ (EA) von SparxSystems.

UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML

dellelemente und Beziehungen. Abb. 3–6 zeigt das Beispiel-Zieldiagramm. Die einzelnen Modellelemente und Beziehungen sind in den Abschnitten 3.2 und 3.3 erläutert. Besonders hervorzuheben ist die angepasste Darstellung des Diagrammtabs in der oberen linken Ecke von Abb. 3–6. Diese besteht aus dem Identifikator „[kaos]“, der anzeigt, dass es sich bei dem angezeigten Diagramm um ein Zielmodell nach KAOS handelt. Der nachfolgende Name „Goal Model Example“ ist der benutzerdefinierbare Name des Diagrammes. Der Text des Diagrammtabs wurde im Paket „Goal Diagram Definition“ in der Metaklasse „Diagram_Logical“ unter dem Attribut „frameString“ festgelegt.

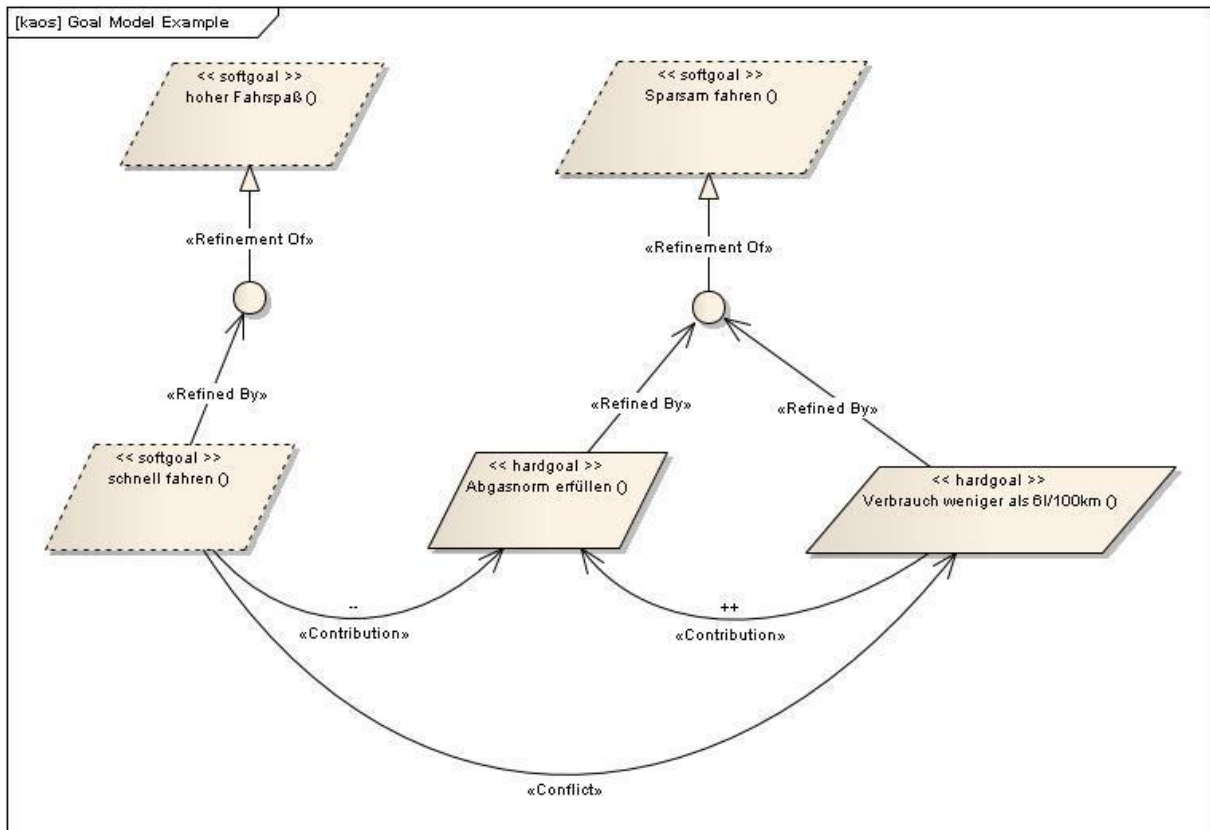


Abb. 3–6 KAOS Zieldiagramm aller derzeit implementierten Modellelemente und Beziehungen

Das Beispiel zeigt zwei Oberziele „hoher Fahrspaß“ und „Sparsam fahren“ vom Typ „softgoal“ (siehe Abschnitt 3.2.1.2). Das Ziel „hoher Fahrspaß“ wird von Unterziel „schnell fahren“, welches ebenfalls vom Typ „softgoal“ ist, verfeinert. Bei dieser Verfeinerung handelt es sich um eine Oder-Verfeinerung, bei der lediglich eine Alternative angegeben ist. Das Unterziel „schnell fahren“ muss also erfüllt sein, damit das Oberziel „hoher Fahrspaß“ erfüllt ist. Das Oberziel „Sparsam fahren“ wird durch die beiden Unterziele vom Typ „hardgoal“ (siehe Abschnitt 3.2.1.1) mit einer Und-Verfeinerung verfeinert. Es müssen folglich beide Unterziele „Abgasnorm erfüllen“ und „Verbrauch weniger als 6l/100km“ erfüllt sein, damit das Oberziel erfüllt ist. Das Unterziel „Verbrauch von weniger als 6l/100km“ steht in einer Beziehung vom Typ „Contribution“ (siehe Abschnitt 3.3.4) zum Ziel „Abgasnorm erfüllen“. Diese Beziehung bedeutet, dass die Erfüllung des Zieles „Verbrauch weniger als 6l/100km“ sich stark positiv auf die Erfüllung des Zieles „Abgasnorm erfüllen“ auswirkt, also die Erfüllung des Zieles „Abgasnorm erfüllen“ vereinfacht wird. Das Ziel „schnell fahren“ steht ebenfalls in einer Beziehung vom Typ „Contribution“ zum Ziel „Abgasnorm erfüllen“. Diese Beziehung besagt, dass die Erfüllung des Zieles „schnell fahren“ die Erfüllung des Zieles „Abgasnorm erfüllen“ schwach beeinträchtigt. Des Weiteren

UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML

steht das Ziel „schnell fahren“ im Konflikt mit dem Ziel „Verbrauch weniger als 6l/100km“, was durch eine Beziehung vom Typ „Contradiction“ (siehe Abschnitt 3.3.3) dargestellt ist. In diesem Fall verhindert die Erfüllung des Zieles „schnell fahren“ die Erfüllung des Zieles „Verbrauch weniger als 6l/100km“.

3.2 Elemente der Zielmodelle

In diesem Abschnitt werden die Modellelemente des KAOS-Zielmodells beschrieben, die im Paket „Goal Modeling Elements“ definiert wurden. Ein Überblick über das Paket kann aus Abschnitt 3.1.1 entnommen werden. Die Erläuterungen in den folgenden Abschnitten stützten sich auf Abb. 3–2. Die Verwendung der Modellelemente ist in Abb. 3–6 illustriert.

3.2.1 Die Klasse „Goal“

Die Klasse „Goal“ definiert Ziele des Zieldiagrammes und stellt somit das zentrale Modellelement im SysML Goal Modeling Profil dar. Ziele erben von der Metaklasse „Class“ und werden somit von Enterprise Architect als eigenes Modellelement aufgefasst. Die Klasse „Goal“ definiert mehrere EA-interne Attribute, dessen Erläuterung aus [Sparx10] entnommen werden können. Besonders hervorzuheben ist dabei das Attribut „_image“ und das Attribut „Zieltyp“, welches ein Ziel entweder genauer als ein „hardgoal“ (siehe Abschnitt 3.2.1.1.) oder „softgoal“ (siehe Abschnitt 3.2.1.2) definiert. Ein generischer Zieltyp „goal“ ist ebenfalls möglich. Dieser generische Typ sagt aus, dass eine genauere Klassifizierung des Zieles noch nicht erfolgt ist und ggf. zu einem späteren Zeitpunkt festgelegt werden kann. Genauere Informationen zum festlegen des Zieltyps können in Abschnitt 3.4 gefunden werden. Das Attribut „_image“ definiert das Aussehen des Modellelementes „Goal“ als rechtslastiges Parallelogramm. Das Attribut „Zieltyp“ definiert die Art des zu erstellenden Zieles. Die Art des Zieles wird in der Enumeration „GoalType“ definiert (vgl. 3.4) und kann entweder „goal“, „hardgoal“ oder „softgoal“ sein. Der Zieltyp wird im Stereotypen des Modellelementes entsprechend dargestellt. Das Attribut „Zieltyp“ wird im EA-Editor im Eigenschaftsfenster des Modellelementes unter dem Registerreiter „Tagged Values“ zusammen mit dem Attribut ID dargestellt. Das Attribut ID gibt dem Benutzer die Möglichkeit, eine eindeutige ID zu vergeben.

3.2.1.1 Zieltyp: „hardgoal“

Der Zieltyp „hardgoal“ wird durch die Enumeration „GoalType“ definiert. Wenn ein Ziel vom Typ „hardgoal“ ist, so wird der Zieltyp im Stereotypen des Ziels dargestellt. Ein Ziel vom Typ „hardgoal“ wird als rechtsseitiges Parallelogramm mit durchgezogener Linie dargestellt. Die semantische Bedeutung eines Zieles vom Typ „hardgoal“ kann aus [KAOS07], [Lamsweerde09] und [Pohl10] entnommen werden.

3.2.1.2 Zieltyp: „softgoal“

Der Zieltyp „softgoal“ wird durch die Enumeration „GoalType“ definiert. Wenn ein Ziel vom Typ „softgoal“ ist, so wird der Zieltyp im Stereotypen des Ziels dargestellt. Ein Ziel vom Typ „softgoal“ wird als rechtsseitiges Parallelogramm mit gestrichelter Linie dargestellt. Die semantische Bedeutung eines Zieles vom Typ „hardgoal“ kann aus [KAOS07], [Lamsweerde09] und [Pohl10] entnommen werden.

3.2.2 Die Klasse „Refinement“

Die Klasse „Refinement“ stellt ein Modellelement zur Darstellung von Und- und Oder-Verfeinerungen in einem Zieldiagramm dar. Klassen vom Typ „Refinement“ haben

UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML

typischerweise keinen Namen und werden in EA daher generisch mit „Class1“ oder „Refinement1“ bezeichnet. Verfeinerungen erben von der Metaklasse „Class“ und werden somit von Enterprise Architect als eigenes Modellelement aufgefasst. Verfeinerungen werden im EA-Editor als Kreis dargestellt, was durch das Attribut „_image“ der Klasse „Refinement“ definiert ist. Die KAOS-Methode sieht eine Unterscheidung zwischen And-Refinements (Und-Verfeinerungen) sowie Or-Refinements (Oder-Verfeinerungen) vor. Details hierzu können Abschnitt 2.1 sowie [KAOS07], [Lamsweerde09] und [Pohl10] entnommen werden. Und-Verfeinerungen werden dargestellt, indem mehrere Unterziele derselben Klasse vom Typ „Refinement“ mit jeweils einer Beziehung vom Typ „Refined By“ (siehe Abschnitt 3.3.2) zugeordnet werden. Die für die Unterziele gemeinsame Klasse „Refinement“ wird dann mit einer Beziehung vom Typ „Refinement Of“ (siehe Abschnitt 3.3.1) dem gemeinsamen Oberziel zugeordnet. Ein Beispiel für eine Und-Verfeinerung ist im Beispiel in Abb. 3–6 auf der rechten Seite dargestellt.

Wenn mehrere Klassen vom Typ „Refinement“ demselben Oberziel (jeweils durch eine eigene Beziehung vom Typ „Refinement Of“) zugeordnet werden, stellt dies eine Oder-Verfeinerung dar. Jede Klasse vom Typ „Refinement“ stellt eine Alternative Verfeinerung des gemeinsamen Oberzieles dar. Dabei ist zu beachten, dass jede Alternative ein oder mehrere Unterziele in einer Und-Verfeinerung zusammenschließen kann. In diesem Fall ist die Alternative nur dann vollständig erfüllt, wenn alle Unterziele erfüllt sind.

3.3 Beziehungen zwischen den Elementen

In diesem Abschnitt werden die Beziehungen zwischen den Modellelementen des KAOS-Zielmodells beschrieben, die im Paket „Goal Modeling Elements“ definiert wurden. Ein Überblick über das Paket kann aus Abschnitt 3.1.1 entnommen werden. Die Erläuterungen in den folgenden Abschnitten stützten sich auf Abb. 3–2. Die Verwendung der Beziehungen zwischen den Modellelementen ist in Abb. 3–6 illustriert.

3.3.1 Die Beziehung „Refinement Of“

Die Klasse „Refinement Of“ spezifiziert eine Beziehung von einem Modellelement des Typs „Refinement“ zu einem Modellelement des Typs „Goal“. „Refinement Of“ erbt von der EA-Metaklasse „Generalization“. Die Beziehung „Refinement Of“ bedeutet, dass eine Menge von Unterzielen, die durch eine Menge von „Refined By“-Beziehungen einer Verfeinerung zugeordnet sind (siehe Abschnitt 3.3.2, sowie [KAOS07], [Lamsweerde09], [Pohl10]), alle erfüllt sein müssen, damit das Oberziel erfüllt ist. Nach KAOS ([KAOS07]) kann es lediglich eine „Refinement Of“-Beziehung zwischen einer Verfeinerung und einem Ziel geben.

3.3.2 Die Beziehung „Refined By“

Die Klasse „Refined By“ spezifiziert eine Beziehung von einer Menge von Modellelementen des Typs „Goal“ zu einem Modellelement des Typs „Refinement“. „Refined By“ erbt von der EA-Metaklasse „Association“. Die Beziehung „Refined By“ bedeutet, dass das Ziel ein Unterziel des Oberzieles ist, welchem die Verfeinerung zugeordnet ist (siehe Abschnitt 3.3.1, sowie [KAOS07], [Lamsweerde09], [Pohl10]).

3.3.3 Die Beziehung „Contradiction“

Die Klasse „Contradiction“ spezifiziert eine Beziehung zwischen zwei Modellelementen des Typs „Goal“. Diese Beziehung sagt aus, dass sich zwei Ziele widersprechen, also das Erreichen eines Zieles das Erreichen eines anderen Zieles unmöglich macht. Ebenso wie die Beziehung „Refined By“ erbt die Beziehung „Contradiction“

UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML

von der EA-Metaklasse „Association“. Eine „Contradiction“-Beziehung wird als Bezier-Kurve zwischen beiden beteiligten Zielen dargestellt, wie durch das Attribut „_linestyle“ der Klasse „Contradiction“ festgelegt ist.

3.3.4 Die Beziehung „Contribution“

Die Klasse „Contribution“ spezifiziert eine Beziehung zwischen zwei Modellelementen des Typs „Goal“. Diese Beziehung sagt aus, dass das Erreichen eines Zieles das Erreichen eines anderen Zieles beeinflusst. Die Art der Beeinflussung wird durch die Enumeration „ContributionType“ (siehe Abschnitt 3.4) spezifiziert, wie durch das Attribut „ContributionType“ der Klasse „Contribution“ festgelegt ist. Ebenso wie die Beziehung „Refined By“ und „Contradiction“ erbt die Beziehung „Contribution“ von der EA-Metaklasse „Association“. Eine „Contribution“-Beziehung wird als Bezier-Kurve zwischen beiden beteiligten Zielen dargestellt, wie durch das Attribut „_linestyle“ der Klasse „Contribution“ festgelegt ist.

3.4 Die Enumerationen „GoalType“ und „ContributionType“

Die Enumerationen „ContributionType“ und „GoalType“ definieren Auswahllisten für so genannte „Tagged Values“ der Modellelemente. „Tagged Values“ werden in den Klassen, die Modellelemente repräsentieren als Attribute definiert und können im EA-Editor im Eigenschaftsfenster über den Registerreiter „Tagged Values“ verwendet werden.

Die Enumeration „GoalType“ wird von der Klasse „Goal“ verwendet (siehe Abschnitt 3.2.1) und legt die drei Typen von Zielen fest („goal“, „softgoal“, „hardgoal“). Der Typ „goal“ ist der Standardtyp, der verwendet wird, wenn ein Ziel nicht genauer als „hardgoal“ (siehe Abschnitt 3.2.1.1) bzw. „softgoal“ (siehe Abschnitt 3.2.1.2) festgelegt wird.

Die Enumeration „ContributionType“ wird von der Beziehung „Contribution“ verwendet (siehe Abschnitt 3.3.4) und legt fünf Intensitäten fest, mit der das Erreichen eines Zieles das Erreichen eines anderen Zieles beeinflusst. Diese reichen von sehr schwach („--“) bis sehr stark („++“).

4 Verwendung des MDG-Plug-Ins

Das SysML Goal Modeling Profil wird in Form eines MDG-Plug-Ins (siehe [Sparx10]) für Enterprise Architect zur Verfügung gestellt. In diesem Abschnitt wird die Installation, Verwendung und Erweiterung des Profils kurz erläutert. Detaillierte Informationen zur MDG-Technologie, zur Verwendung und Erweiterung von UML/SysML-Profilen in Enterprise Architect und zur Modellierung mit Enterprise Architect werden in [Sparx09] und [Sparx10] erläutert.

4.1 Installation der MDG-Technologie

Dieses MDG-Plug-In befindet sich zusammen mit der Enterprise Architect Projektdatei des Profils im ZIP-Archiv „SysML-Goal-Modeling v4.zip“. Im Folgenden wird beschrieben, wie das SysML Goal Modeling-Profil in Enterprise Architect eingebunden werden kann.

1. **Entpacken der ZIP-Datei.** Entpacken Sie die ZIP-Datei in einen Ordner auf der lokalen Festplatte. Es werden zwei Dateien entpackt: eine EAP-Datei und eine XML-Datei.
2. **Öffnen Sie Enterprise Architect.** Starten Sie die Enterprise Architect Anwendung, um das MDG-Plug-In einzubinden.
3. **Öffnen Sie den MDG-Technologie-Dialog.** Klicken Sie dazu im Enterprise-Architect-Menü auf „Settings“ und danach auf die Menüoption „MDG Technologies“.
4. **Definieren Sie einen neuen MDG-Pfad.** Im MDG-Technologie-Dialog, klicken sie auf „Avance“ und im darauf folgenden Fenster auf „Add“ und dann auf „Add Path...“. Geben Sie den Ordner an, in den Sie die ZIP-Datei in Schritt 1) entpackt haben. Der Zielordner ist der Ordner, in dem die XML-Datei entpackt wurde. Klicken Sie anschließend auf OK. Enterprise Architect wird Sie darauf hinweisen, dass die Änderungen erst nach einem Neustart übernommen werden.
5. **Starten Sie Enterprise Architect neu,** um die Änderungen zu übernehmen.
6. **Überprüfen Sie, ob die MDG-Technologie eingebunden ist,** indem Sie den MDG-Technologie-Dialog wie unter Schritt 4) öffnen. Es sollte nun unter „Technologies“ das SysML Goal Modeling-Plug-In aufgeführt werden.

Weitere Möglichkeiten zur Einbindung von MDG-Plug-Ins in Enterprise Architect sind in [Sparx10] erläutert.

4.2 Verwendung des Profils

Im Folgenden wird beschrieben, wie das SysML Goal Modeling-Profil in Enterprise Architect verwendet werden kann.

1. **Erstellen Sie eine neue EA-Projektdatei.** Klicken Sie dazu in Enterprise Architect auf „File“ und dann „New Project“. Wählen Sie einen geeigneten Pfad und Dateinamen für die Projektdatei und klicken Sie auf „speichern“.
2. **Erstellen Sie ein neues Paket** unter dem Wurzelknoten „Model“ im „Project Browser“. Verwenden Sie dazu entweder die „Add a Package“ oder drücken Sie „Strg+W“. Geben Sie dem Projekt einen geeigneten Nam und verwenden Sie „Class View“ als Sicht auf die Modelle in diesem Paket, wie in Abb. 4–1 dargestellt. Klicken Sie anschließend auf OK.

UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML



Abb. 4–1 Screenshot zur Erstellung eines neuen Paketes für das Zielmodell

3. **Erstellen Sie ein neues Zieldiagramm**, indem Sie mit der rechten Maustaste auf das in Schritt 2) erstellte Paket klicken und unter dem Menüeintrag „Add“ auf „New Diagram...“ klicken. Wählen Sie einen geeigneten Namen für Ihr Zieldiagramm und wählen Sie unter „Type“ die MDG-Technologie „SysML Goal Modeling“ aus. Wählen Sie anschließend unter „Diagram Types“ den Eintrag „Goal Diagram“. Abb. 4–2 zeigt die für diesen Schritt notwendigen Einstellungen.

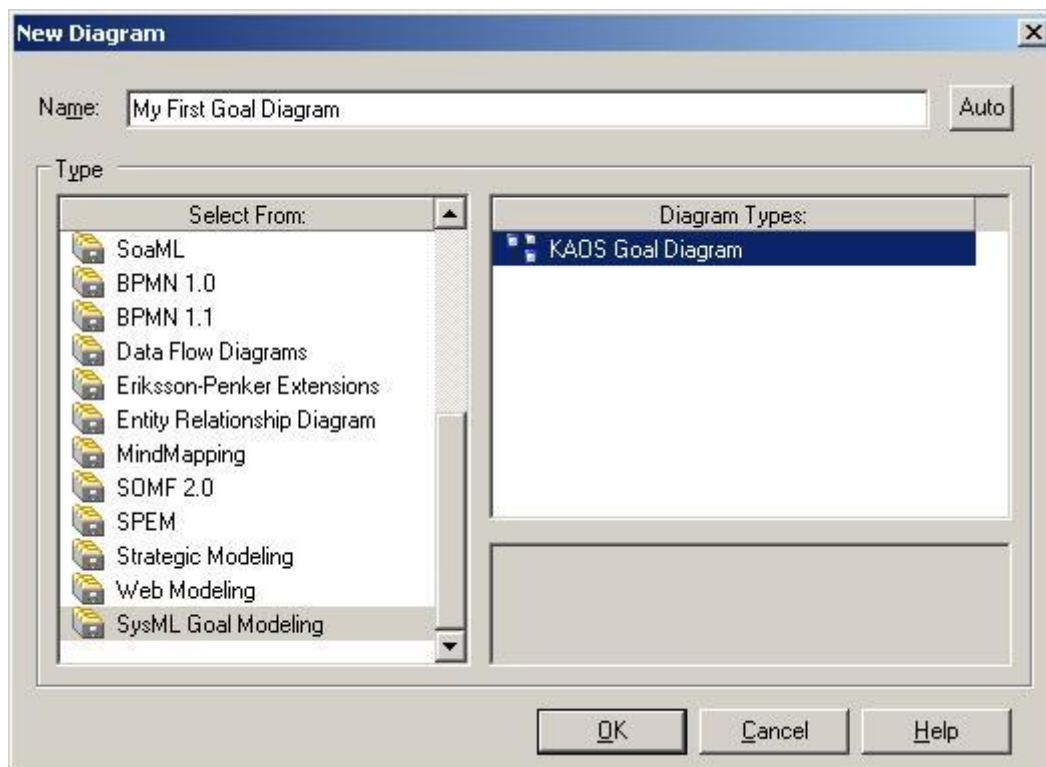


Abb. 4–2 Screenshot zur Erstellung eines neuen Zieldiagrammes

4. **Erstellen Sie ein Zieldiagramm** im Enterprise Architect Editor. Genaue Informationen zur Verwendung von Enterprise Architect können aus [Sparx09] entnommen werden. Informationen zur Zielmodellierung mit KAOS können Sie in

UML/SysML-Profil und MDG-Plug-In für Enterprise Architect zur Modellierung von KAOS-Zieldiagrammen auf Basis von UML/SysML

[KAOS07], [Lamsweerde09] und [Pohl10] finden. Abb. 4–3 zeigt ein Beispiel zur Modellierung mit dem Zieldiagramm.

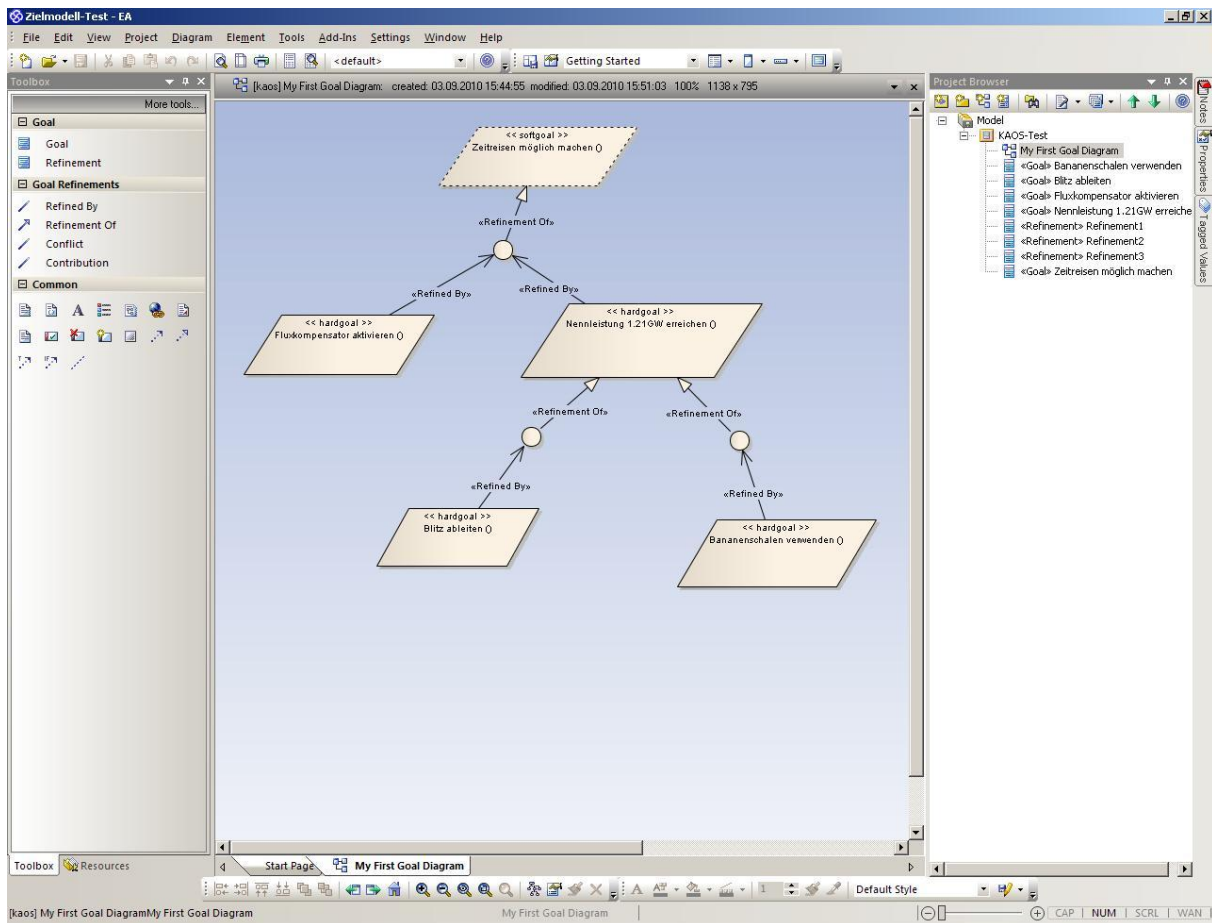


Abb. 4–3 Screenshot zur Modellierung der Ziele in dem neu erstellten Zieldiagramm

4.3 Erweiterung des Profils

Im Folgenden wird beschrieben, wie das SysML Goal Modeling-Profil in Enterprise Architect erweitert werden kann.

1. **Entpacken der ZIP-Datei.** Entpacken Sie die ZIP-Datei in einen Ordner auf der lokalen Festplatte. Es werden zwei Dateien entpackt: eine EAP-Datei und eine XML-Datei.
2. **Öffnen Sie die EAP-Datei in Enterprise Architect.** Starten Sie die Enterprise Architect Anwendung und öffnen Sie die EAP-Datei. Diese Datei enthält die Profildefinition sowie die Definitionen für den Diagrammtypen und die EA-Toolbox.
3. **Modellieren Sie Ihre Erweiterung zu.** Nun können Sie Änderungen am Profil oder den Diagram- und Toolbox-Definitionen vornehmen oder weitere Profile hinzufügen. Hinweise zur Erweiterung von MDG-Technologien und zum Umgang mit Enterprise Architect können aus [Sparx09] und [Sparx10] entnommen werden.

5 Zusammenfassung

In diesem Dokument wurde das SysML Goal Modeling Profil für Enterprise Architect vorgestellt. Bei dem UML/SysML-Profil handelt es sich um eine MDG-Technologie, die als Plug-In für Enterprise Architect verwendet werden kann. Das Profil besteht aus drei Teilen: Der Definition des Zielmodellprofils im Paket „Goal Modeling Elements“, der Definition der Zieldiagrammtypen im Paket „Goal Diagram Definition“ und der Definition für die User-Interface-Komponenten als EA-Toolbox im Paket „SysML Goal Modeling“. Das Profil definiert derzeit lediglich einen Ausschnitt der Modellierungselemente der KAOS-Methode. Im Profil werden zwei verschiedene Zielmodellelemente festgelegt: Ziele („Goals“) und Verfeinerungen („Refinements“). „Goals“ werden zur Modellierung von Zielen verwendet und als „hardgoal“ oder „softgoal“ genauer festgelegt. Verfeinerungen erlauben es, Unterziele zu Oberzielen zu definieren, sodass Oberziele mit Und-Verfeinerungen und Oder-Verfeinerungen durch Unterziele verfeinert werden. Dazu werden die Beziehungen „Refinement Of“ und „Refined By“ verwendet. Die zwei weiteren Beziehungen zwischen den Zielmodellelementen sind Widerspruch („Contradiction“) und Beeinflussung („Contribution“). Mit diesen beiden Beziehungen kann modelliert werden, inwiefern das Erreichen eines Zieles das Erreichen eines anderen Zieles beeinflusst oder verhindert. Verschiedene Typen der Beeinflussung sowie die verschiedenen Typen von Zielen werden durch zwei Enumerationen festgelegt.

Zukünftige Arbeiten zu diesem Thema werden sich mit der Erweiterung dieses Profils und Plug-Ins zur Anwendung auf mehreren Abstraktionsstufen befassen. Des Weiteren soll das Profil um die übrigen Modellelemente der KAOS-Methode erweitert werden.

6 Literaturverzeichnis

- [DaSiLa10] Daun, Marian; Sikora, Ernst; Lauenroth, Kim. Stand der Praxis im modellbasierten Requirements Engineering. SPES 2020 Deliverable D2.1.A, 2010.
- [FrMoSt09] Friedenthal, Sanford; Moore, Alan; Steiner, Rick. A Practical Guide to SysML. Morgan Kaufman & OMG, 2009.
- [KAOS07] Respect-IT. A KAOS Tutorial v1.0. 2007.
- [LaGaSiTe10] Lauenroth, Kim; Gabrisch, Sebastian; Sikora, Ernst; Tenbergen, Bastian. Beschreibung der Durchführung der Studie zum Stand der Praxis im Requirements Engineering für Embedded Systems. SPES 2020 Teilergebnis, Version 1.0 vom 16.07.2010.
- [Lamsweerde09] van Lamsweerde, Axel. Requirements Engineering – From System Goals to UML Models to Software Specifications. Wiley, 2009.
- [LaSiStTe09] Lauenroth, Kim; Sikora, Ernst; Stallbaum, Heiko; Tenbergen, Bastian. Initialer modellbasierter Requirements Engineering Ansatz für Embedded Systems. SPES 2020 Teilergebnis, Version 0.9 vom 18.09.2009.
- [OMG10] Object Management Group. OMG Systems Modeling Language (OMG SysML) Language Specification v1.2. OMG Document Number: formal/2010-06-02. 2010.
- [Pohl10] Pohl, Klaus. Requirements Engineering – Foundations, Principles, Techniques. Springer, 2010.
- [SiStLa10] Sikora, Ernst; Stallbaum, Heiko, Lauenroth, Kim. Anforderungen an den zu entwickelnden Ansatz für modellbasiertes Requirements Engineering. SPES 2020 Teilergebnis, Version 1.0 vom 01.03.2010.
- [Sparx09] SparxSystems. Enterprise Architect User Guide. 2009.
- [Sparx10] SparxSystems. Enterprise Architect Software Developers' Kit. 2010.