

Ein strukturierter Ansatz zur Ableitung methodenspezifischer UML/SysML-Profile am Beispiel des SPES 2020 Requirements Viewpoints¹

Bastian Tenbergen, Philipp Bohn, Thorsten Weyer

paluno – The Ruhr Institute for Software Technology

Universität Duisburg-Essen

Gerlingstraße 16, 45127 Essen

{bastian.tenbergen | philipp.bohn | thorsten.weyer}@paluno.uni-due.de

Abstract: Eine wesentliche Voraussetzung für die industrielle Akzeptanz von wissenschaftlich entwickelten modellbasierten Entwicklungsmethoden ist, dass sich die Methode in die Werkzeug- und Prozesslandschaft von Industrieunternehmen eingliedert und somit angewendet werden kann, ohne dass methodenspezifische Werkzeuge notwendig sind. Profile erlauben es, UML/SysML für spezielle Entwicklungsmethoden anzupassen, indem Konzepte der Methode in UML/SysML abgebildet werden. Eine Herausforderung bei der Entwicklung von UML/SysML-Profilen besteht u.a. darin, die Profile systematisch und strukturiert abzuleiten, sodass die Konzepte der Methode auf UML/SysML korrekt und vollständig abgebildet werden können. Während in der Literatur Ansätze zur Ableitung domänenspezifischer UML/SysML-Profile existieren, gibt es bisher jedoch keinen Ansatz zur Ableitung methodenspezifischer Profile. Ziel dieses Artikels ist es daher, einen strukturierten Ansatz zur Definition solcher Profile vorzustellen. Zunächst wird dazu die wesentliche Literatur zur strukturierten Definition von domänenspezifischen Profilen untersucht. Anschließend wird ein auf bestehender Literatur basierender Ansatz vorgestellt und anhand des SPES 2020 Requirements Viewpoints illustriert. Durch diesen Ansatz wird eine Grundlage für die Anwendbarkeit einer modellbasierten Entwicklungsmethode insofern geschaffen, als dass die Konzepte der Methode auf UML/SysML abgebildet werden können und die Entwicklungsmethode mit im spezifischen Industriekontext bereits vorhandenen UML-Modellierungswerkzeugen angewendet werden kann.

1 Einführung und Motivation

Eine Möglichkeit, die Herausforderungen bei der Entwicklung heutiger softwareintensiver eingebetteter Systeme zu adressieren, ist die Verwendung von modellbasierten Entwicklungsansätzen, wie z.B. [Br12]. Allerdings haben verschiedene Studien zum Stand der Praxis modellbasierter Entwicklung (z.B. [LPR93]) gezeigt, dass modellbasierte Entwicklungsansätze nur zögerlich in der Industrie eingeführt werden, u.a. auf Grund von fehlenden Nutzensnachweisen im industriellen Kontext [STP12]. Eine wesentliche Voraussetzung für solche Industriestudien ist jedoch, dass die Ansätze im in-

¹ Dieser Beitrag wurde im Rahmen der BMBF-Projekte SPES 2020 (Förderkennzeichen: 01IS08045V) und SPES 2020_XTCore (Förderkennzeichen: 01IS12005C) gefördert.

dustriellen Kontext anwendbar sind, u.a. durch flexible Integration der Entwicklungsansätze in die bestehenden Werkzeugketten [GLT03], Wiederverwendung bestehender Werkzeuge sowie durch den Einsatz gängiger Modellierungssprachen [Da06], wie beispielsweise UML und SysML.

Durch den Einsatz von UML/SysML-Profilen kann diese Voraussetzung geschaffen werden [FV04]. Da Profile konzeptueller Natur und somit werkzeuginspezifisch sind, können sie in der Regel für spezifische Werkzeuge implementiert und in bestehende Werkzeug- und Prozessketten eingegliedert werden. In Abbildung 1 wird dieser Sachverhalt dargestellt: ein konzeptuelles Profil wird für zwei unterschiedliche Werkzeuge spezifisch angepasst. Somit passen sich die werkzeugspezifischen Implementierungen in die bestehende Werkzeugkette ein und entsprechen gleichzeitig dem konzeptuellen Profil. Da Profile konzeptueller Natur sind, entfällt somit die Notwendigkeit Werkzeug- und Prozessketten durch Einbeziehen von neuen methodenspezifischen Modellierungswerkzeugen anzupassen und Entwickler im Umgang mit zusätzlichen Werkzeugen zu schulen.

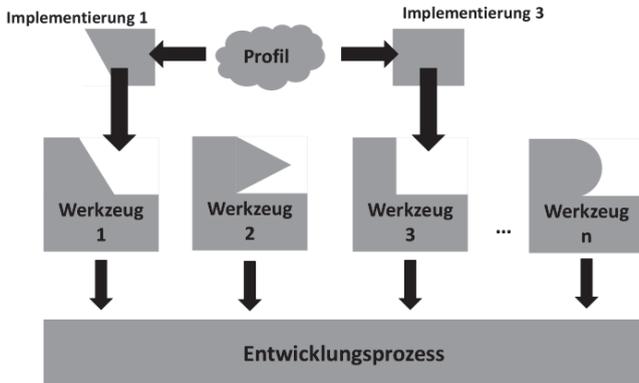


Abbildung 1: Werkzeugunterstützung durch UML/SysML-Profile

Profile werden typischerweise zur Abbildung von Konzepten spezifischer Anwendungsdomänen, wie z.B. Automatisierungstechnik auf das UML-Metamodell verwendet. Durch Profile kann UML/SysML erweitert werden, indem fehlende Konzepte, d.h. Typen oder Klassen mit besonderen semantischen Eigenschaften, die über die der UML eigenen Typen hinausgehen, hinzugefügt werden. Zur Erstellung von UML/SysML-Profilen bedarf es jedoch systematischen Vorgehensweisen, da andernfalls die Gefahr besteht, dass im Hinblick auf den werkzeugtechnisch zu unterstützenden Entwicklungsansatz unvollständige oder inkorrekte Profile gebildet werden und somit u.a. die Integration des Entwicklungsansatzes in bestehende Werkzeuge beeinträchtigt wird.

Während die Literatur einige Ansätze zur Ableitung domänenspezifischer Profile vorgeschlägt, existiert allerdings kein Ansatz zur systematischen Ableitung methodenspezifischer Profile. In diesem Beitrag wird solch ein Ansatz vorgestellt.

In Abschnitt 2 werden zunächst der Stand der Wissenschaft zur Erstellung von UML/SysML-Profilen zusammenfassend diskutiert und die notwendigen Qualitätseigenschaften für methodenspezifische Profile expliziert. In Abschnitt 3 wird anschließend ein Ansatz vorgestellt, der es gestattet, auf systematische Art und Weise Profile für Modellierungsmethoden zu erstellen, die diese Eigenschaften aufweisen. Abschnitt 4 illustriert die Anwendung des Ansatzes anhand eines konkreten Beispiels.

2 UML/SysML-Profile: Grundlagen und Entwicklung

UML/SysML-Profile sind domänenspezifische Modellierungssprachen (DSML), die das UML-Metamodell [MOF06] im Hinblick auf die Bedürfnisse von speziellen Anwendungsdomänen erweitern ([MOF06], [KT08]), indem sie Stereotypen definieren. UML/SysML-Profile können beispielsweise domänenspezifische Konzepte der Automotive-Domäne (z.B. „Drehmoment“ und „Motor“) und die Beziehungen zwischen Konzepten (z.B. „Motor erzeugt Drehmoment“) dem UML-Metamodell hinzufügen, indem bereits existierende Sprachelemente konsistent zu den im UML-Metamodell definierten Regeln wiederverwendet werden [FV04]. In modellbasierten Entwicklungsansätzen können UML/SysML-Profile zudem eingesetzt werden, um spezifische Artefakttypen von Entwicklungsansätzen und Methoden abzubilden, indem UML/SysML-Diagrammarten wiederverwendet und spezifisch um ggf. fehlende Konzepte erweitert werden. Methodenspezifische UML/SysML-Profile erweitern folglich die UML/SysML-Syntax und schaffen darüber hinaus eine Grundlage für die semantische Integration einer Methode, indem sie die spezifischen Konsistenzregeln der Modellierungsmethode berücksichtigen müssen und somit ggf. auch Einschränkungen bei der Verwendung der UML/SysML-Notation nach sich ziehen.

2.1 Ansätze zur systematischen Entwicklung von UML/SysML-Profilen

Bezüglich der Literatur zur Erstellung von domänenspezifischen Sprachen und UML/SysML-Profilen kann u.a. zwischen „leichtgewichtigen“ und „schwergewichtigen“ Ansätzen unterschieden werden [WS07]: leichtgewichtige Ansätze eignen sich für Situationen, in denen bestehende Konzepte des UML-Metamodell weitestgehend zur Abbildung der zu modellierenden Domäne ausreichen [La08]. Schwergewichtige Ansätze erlauben darüber hinaus, die Semantik der Modellierungssprache zum Zwecke spezifischer formaler Analysen anzupassen [AP08, FP10].

Lagarde et al. beschreiben in [La08] einen leichtgewichtigen Ansatz zur systematischen Erstellung von UML/SysML-Profilen. Bei der Anwendung des Ansatzes wird zunächst die zu modellierende Domäne analysiert, die wesentlichen im Profil zu berücksichtigen Konzepte identifiziert und als Stereotypen sowie Relationen zwischen den Stereotypen abgebildet. Anschließend wird ein Grundgerüst des Profils erstellt und mögliche Inkonsistenzen identifiziert und aufgelöst. Der Ansatz kann somit die Vollständigkeit des Profils sicherstellen, d.h. alle für den Modellierungszweck notwendigen Domänenkonzepte im Profil berücksichtigen. Ein weiterer leichtgewichtiger Ansatz wird von Selic in [Se07] vorgestellt. Ähnlich dem leichtgewichtigen Ansatz nach Lagarde et al. analysiert

der Ansatz nach Selic ebenfalls die zu modellierende Domäne. Anschließend werden die identifizierten Konzepte jedoch direkt in Stereotypen übertragen und keine besondere Qualitätssicherung durchgeführt.

Ein schwergewichtiger Ansatz wird von Kelly und Tolvanen in [KT08]² vorgeschlagen. Der Ansatz von Kelly und Tolvanen basiert ebenfalls auf einer Analyse der zu modellierenden Domäne und identifiziert für die Modellierung wesentliche Konzepte. Im Gegensatz zu den leichtgewichtigen Ansätzen wird danach ein domänenspezifisches Metamodell formalisiert und eine Semantik definiert, indem ein Bezug zwischen identifizierten Domänenkonzepten und abgebildeten Metamodellelementen formal hergestellt wird. Anschließend können zusätzliche Stereotypen definiert und gegen das zuvor erstellte Domänenmodell validiert werden.

2.2 Anforderungen an einen Ansatz zur systematischen Entwicklung von methodenspezifischen UML/SysML-Profilen

Bei den vorgestellten Ansätzen zur Entwicklung von UML/SysML-Profilen lässt sich ein grundlegendes Muster erkennen: Zunächst wird die Domäne betrachtet und die wesentlichen Konzepte identifiziert. Diese werden im UML/SysML-Profil durch neue Stereotypen repräsentiert und anschließend validiert. Durch die Validierung wird sichergestellt, dass das resultierende Profil in Bezug auf die zu modellierende Domäne vollständig und korrekt und auch konsistent zu den bereits existierenden Konzepten in UML/SysML ist. Ein Ansatz zur systematischen Entwicklung methodenspezifischer UML/SysML-Profile muss demnach Profile mit den folgenden Qualitätseigenschaften erstellen können:

- *Vollständigkeit:* Der Ansatz muss Profile entwickeln, die in Bezug auf die zu unterstützende Entwicklungsmethode insofern vollständig sind, als dass alle Artefakte der Entwicklungsmethode im Profil repräsentiert werden.
- *Korrektheit:* Der Ansatz muss Profile entwickeln, die in Bezug auf die zu unterstützende Entwicklungsmethode insofern korrekt sind, als dass Konsistenzregeln zwischen Artefakten der Entwicklungsmethode nicht verletzt werden.
- *Konsistenz:* Der Ansatz muss Profile entwickeln, die in Bezug auf die zu unterstützende Entwicklungsmethode insofern konsistent sind, als dass das Profil nicht syntaktisch und semantisch inkompatibel mit dem UML-Metamodell ist und im größtmöglichen Umfang bestehende UML/SysML-Konzepte wiederverwendet.

3 Strukturierter Ansatz zur Entwicklung von UML/SysML-Profilen

Für die Entwicklung eines werkzeugunabhängigen, strukturierten Ansatzes für konzeptuelle Profile wurde der leichtgewichtige Ansatz nach Lagarde et al. (siehe Abschnitt 2.1) adaptiert. Die spezifischen Arbeitsschritte des Ansatzes wurden in dem Umfang

² Der Ansatz befasst sich mit der Neuentwicklung domänenspezifischer Sprache, kann aber auf UML/SysML-Profile angewendet werden.

angepasst, wie es für die Erstellung methodenspezifischer Profile notwendig ist. Der Ansatz setzt sich aus den folgenden Schritten zusammen:

- *Schritt 1:* In diesem Schritt werden die wesentlichen Artefakttypen der Entwicklungsmethode identifiziert und in eine initiale Ontologie übertragen. Dies entspricht dem Erstellen des Domänenmodells im Ansatz nach Lagarde et al., unterscheidet sich von Schritt 1 in [La08] jedoch insofern, als dass die von der Methode verwendeten Modelltypen mit den darin enthaltenen Artefakten (z.B. „Zielmodell“, siehe Abschnitt 4.1) dokumentiert werden.
- *Schritt 2:* In diesem Schritt wird die initiale Ontologie validiert und ggf. fehlende Konzepte der Methode ergänzt. Außerdem werden in diesem Schritt die Abhängigkeiten und Konsistenzregeln zwischen den Modell- und Artefakttypen der Methode (z.B. „Ein Szenario erfüllt mindestens ein Ziel“, siehe [Da12], S. 64) ebenfalls in der Ontologie dokumentiert. Dadurch wird sichergestellt, dass das resultierende Profil korrekt und konsistent ist, d.h. nicht der Methode bzw. dem UML-Metamodell widerspricht. Dieser Schritt entspricht dem dritten Schritt in [La08], berücksichtigt aber zusätzlich zu den syntaktischen Vorgaben des UML-Metamodells noch die semantischen Beziehungen der Artefakttypen innerhalb der Modellierungsmethode.
- *Schritt 3:* In diesem Schritt können die gefundenen Konzepte in ein Konzeptmodell der technischen Umsetzung überführt und auf existierende Konzepte des UML-Metamodells abgebildet werden. Den methodenspezifischen Artefakttypen können äquivalente Modelltypen aus UML/SysML in diesem Schritt im Rahmen einer pragmatischen Implementierung (vgl. Abschnitt 1.5.3 in [BB12]) zugeordnet und somit im methodenspezifischen Profil wiederverwendet werden. Dieser Schritt entspricht dem Erstellen einer initialen Profilstruktur aus [La08], sieht jedoch im Besonderen die Wiederverwendung existierender Diagrammarten vor.
- *Schritt 4:* In diesem Schritt wird die initiale Profilstruktur aus Schritt 3 durch die noch fehlenden Stereotypen ergänzt, was Schritt 4 aus [La08] entspricht.

4 Anwendung des Ansatzes

In diesem Abschnitt wird anhand eines konkreten Beispiels die Umsetzung des Ansatzes aus Abschnitt 3 veranschaulicht. Als konkretes Beispiel wurde der SPES 2020 Requirements Viewpoint (ReqVP, [Da12]) gewählt.

4.1 Der SPES 2020 Requirements Viewpoint

Der SPES 2020 Requirements Viewpoint [Da12] zielt darauf ab, die Anforderungen an ein software-intensives eingebettetes System systematisch zu erheben und zu dokumentieren. Der ReqVP unterstützt den Requirements Engineer dabei, die Anforderungen so zu dokumentieren, dass zwischen lösungsneutraler Problembeschreibung und lösungskonzeptbezogenen Anforderungen unterschieden werden kann. Dazu besitzt der ReqVP vier wesentliche Modelltypen:

- *Kontextmodelle*: Dieser Modelltyp dokumentiert die Schnittstellen des geplanten Systems, Entitäten in der operativen Umgebung (z.B. andere Systeme oder Nutzer), die in Interaktion mit dem System stehen, sowie Ein- und Ausgaben, die zwischen System und Umgebung ausgetauscht werden.
- *Zielmodelle*: Dieser Modelltyp dokumentiert Ziele bzw. Intention der Stakeholder hinsichtlich der Systemfunktionalität sowie gewünschte Qualitäten des Systems. Zielmodelle dienen als Begründung für lösungskonzeptbezogene Anforderungen.
- *Szenariomodelle*: Dieser Modelltyp dokumentiert typische Interaktionen zwischen dem System und Entitäten der Systemumgebung und zeigt die beispielhafte Erfüllung konkreter Ziele.
- *Lösungskonzeptbezogene Anforderungsmodelle*: Dieser Modelltyp dokumentiert die funktionalen Anforderungen des geplanten Systems präzise und vollständig. Dabei wird zwischen statisch-strukturellen (z.B. die Informationsstruktur des Systems), operationalen (z.B. Anforderungen an konkrete Nutzerfunktionen) und Verhaltensanforderungen (z.B. von außerhalb des Systems erfahrbare Systemzustände) unterschieden und somit Anforderungen mit Bezug einem konkreten Lösungskonzept dokumentiert.

Anforderungen können im ReqVP auf unterschiedlichen Abstraktionsebenen der Systemdekomposition hinweg spezifiziert werden. Beginnend mit der Betrachtung des Gesamtsystems auf der höchstens Abstraktionsebene, wird mit jeder tieferen Systemebene das geplante System detaillierter beschrieben. Dabei werden auf jeder Abstraktionsebene alle im Requirements Viewpoint definierten Artefakttypen unterstützt.

Details zum Requirements Viewpoint der SPES 2020 Modellierungsrahmenwerkes, insbesondere zu den Konsistenzregeln können [Da12] entnommen werden.

4.2 Erstellung eines methodenspezifischen Profils für den ReqVP

In diesem Abschnitt wird beschrieben, wie der in Abschnitt 3 beschriebene Ansatz auf den SPES 2020 Requirements Viewpoint angewendet wird um ein konkretes SysML-Profil zu erstellen. Die Ergebnisse der einzelnen Arbeitsschritte werden in den Abbildungen dargestellt.

Schritt 1: Identifikation und Dokumentation der Methodenkonzepte

Abbildung 2 zeigt die initiale Ontologie des ReqVP als UML-Klassendiagramm. Die Ontologie spezifiziert die für die in Abschnitt 4.1 beschriebenen Modelltypen notwendigen Artefakttypen (z.B. *Goal*, *Refinement* und *Goal Dependency* für Zielmodelle) und setzt diese zueinander in Beziehung. Neben den für die Modelltypen wesentlichen Konzepten wurden außerdem artefaktübergreifende Konzepte definiert. So wurden beispielsweise die Konzepte *System*, *Interface* und *Stakeholder* definiert, da diese sowohl für Szenariomodelle als auch für Kontextmodelle relevant sind.

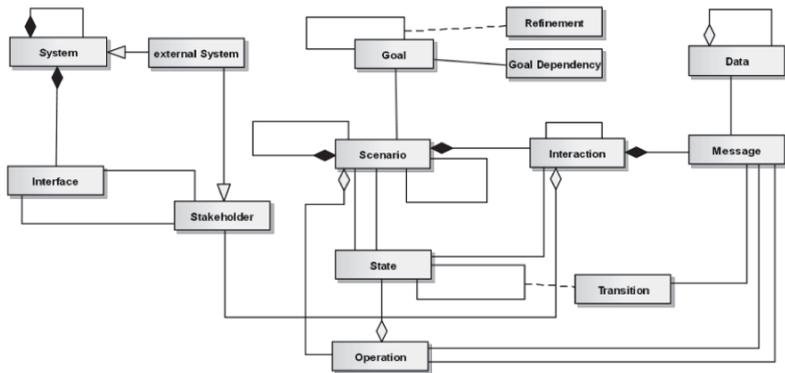


Abbildung 2: Die initiale Ontologie

Schritt 2: Qualitätssicherung und Vervollständigung des Konzeptmodells

Anschließend wurde die Ontologie validiert und vervollständigt. Beispielsweise wurden spezifischen Arten von *Goal*, *Refinement* und *Goal Dependency* definiert, da diese Spezialisierungen wesentliche Auswirkungen auf die Bedeutung des mit dem ReqVP modellierten Sachverhalts haben. Außerdem wurde das Konzept *Stakeholder* verfeinert und gegen einen abstrakten³ *Actor* ausgetauscht und spezialisiert. Anschließend wurde gemäß der Konsistenzregeln des SPES 2020 Requirements Viewpoints (siehe [Da12], S. 64) Assoziationsnamen, Multiplizitäten, Rollen und Leserichtungen der Ontologie hinzugefügt. Das Ergebnis dieses Arbeitsschrittes ist in Abbildung 3 dargestellt.

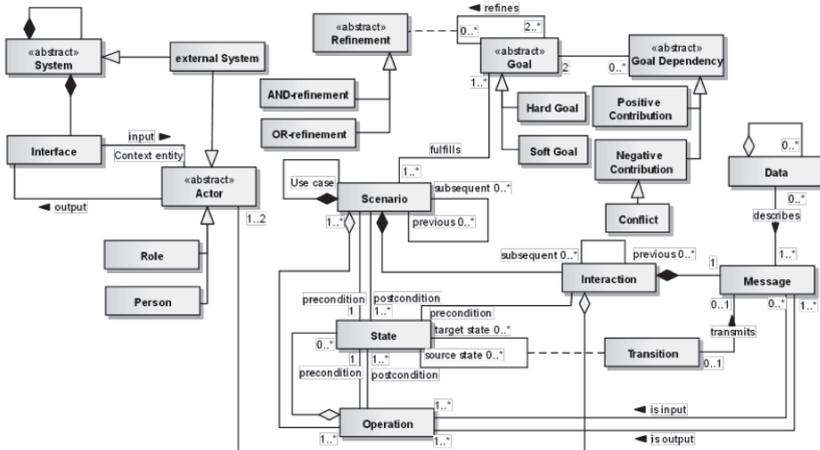


Abbildung 3: Die um Konsistenzmerkmale erweiterte Ontologie

³ „Abstrakt“ bedeutet, dass nicht das ontologische Element selber, sondern eines seiner Spezialisierungen von einem konkreten Modellelement abgebildet wird.

Schritt 3: Erstellen eines Konzeptmodells der technischen Umsetzung

Im nachfolgenden Schritt wurde die vollständige methodenspezifische Ontologie aus Abbildung 3 in ein initiales Konzeptmodell für die technische Umsetzung des Profils überführt und konkreten UML/SysML-Diagrammart zu zugeordnet. Da in diesem Beispiel ein SysML-Profil erstellt wird, wurde die Ontologie mit bereits bestehenden SysML-Konzepten verglichen, um ggf. äquivalente Diagramme aus SysML pragmatisch wiederzuverwenden („Pragmatische Implementierung“, [BB12]). Dabei hat sich beispielsweise gezeigt, dass Kontextmodelle und statisch-strukturelle Anforderungsmodelle durch SysML-Blockdiagramme dargestellt werden können. Es konnten also für diese Modelltypen die entsprechenden SysML-Diagrammart wiederverwendet werden, was in Abbildung 4 durch die <<import>>-Beziehungen dargestellt wurde. Wie in Abbildung 4 entnommen werden kann, gibt es in SysML kein zu Zielmodellen äquivalentes Diagramm, sodass hierfür in Schritt 4 entsprechende Stereotypen abgeleitet werden müssen.

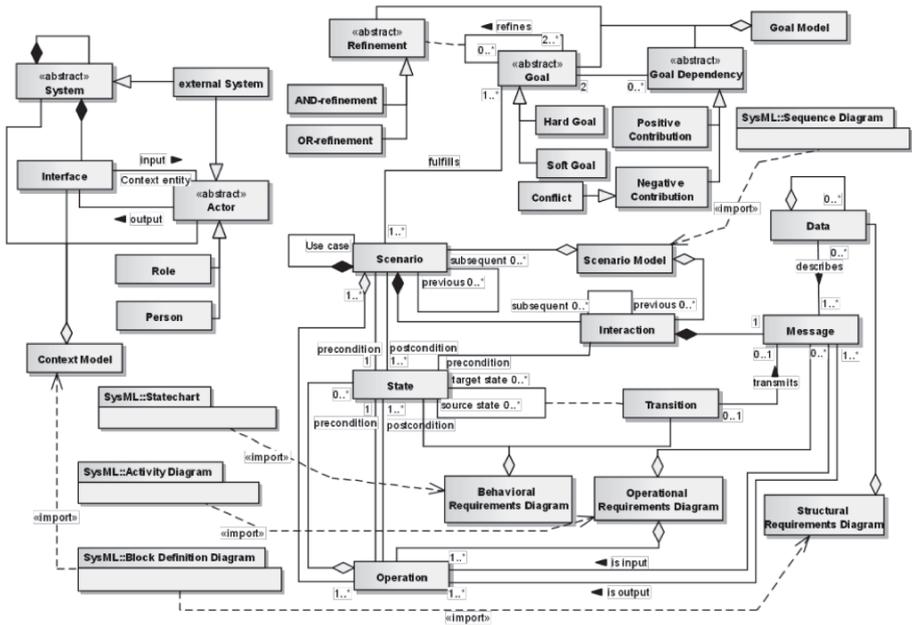


Abbildung 4: Konzeptmodell der technischen Umsetzung des Profils

Schritt 4: Definition fehlender Stereotypen

In diesem Schritt wurde für die zur Zielmodellierung notwendigen Artefakte Stereotypen definiert. Dazu wurde ein Auszug der im ReqVP verwendeten Konzepte des KAOS Rahmenwerks [Da12, La09] modelliert (vgl. Abbildung 5). Die Stereotypen *Goal* und *Refinement* erweitern die SysML-Metaklasse *Block*. *Refinement* realisiert die Konzepte *AND-Refinement* und *OR-Refinement* aus Abbildung 3. Da beide Konzepte durch das gleiche Notationselement modelliert werden und eine Unterscheidung nur auf Modellebene getroffen werden kann (vgl. [La09]), ist ein Stereotyp für beide Arten von *Refinement* ausreichend.

Die Konzepte *Refined By*, *Refinement Of* und *Contribution* stellen syntaktische und hinsichtlich ihrer Notation verschiedene Arten von Assoziationen zwischen Zielen dar, weshalb diese die Metaklasse *Association* bzw. *Generalization* erweitern. Spezielle Typen von *Goal* bzw. *Contribution* werden durch die Enumerationen *GoalType* und *ContributionType* festgelegt.

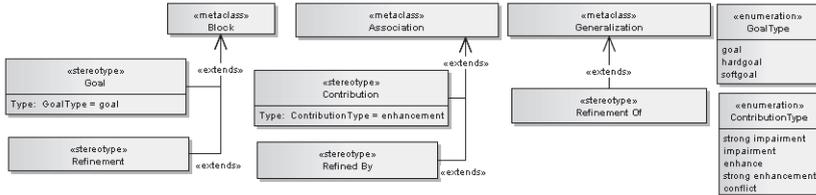


Abbildung 5: Definition von Stereotypen zur Zielmodellierung

5 Zusammenfassung

In diesem Artikel wurde ein strukturierter Ansatz zur Definition methodenspezifischer UML/SysML-Profile vorgestellt. Dazu wurde zunächst die wesentliche Literatur zur strukturierten Definition von UML/SysML-Profilen untersucht und ein auf der bestehenden Literatur basierender Ansatz vorgestellt. Die Anwendung des Ansatzes wurde am Beispiel des SPES 2020 Requirements Viewpoints gezeigt. Der Ansatz versetzt Entwickler in die Lage, methodenspezifische Profile entwickeln zu können, die hinsichtlich der in der Modellierungsmethode verwendeten Artefakte vollständig und hinsichtlich der Modellierungsmethode zu Grunde liegenden Konsistenzregeln korrekt sind. Ferner können Profile entwickelt werden, die syntaktisch zum UML-Metamodell sowie der zu unterstützenden Modellierungsmethode konsistent sind. Dadurch ermöglicht der in diesem Beitrag vorgestellte Ansatz eine grundlegende Werkzeugunterstützung für Anwender einer Modellierungsmethode, da dem entstandenen Profil ein Konzeptmodell der Modellierungsmethode zu Grunde liegt, welches werkzeugspezifisch unter Verwendung der für spezifische UML-Werkzeuge gängigen Verfahren implementiert werden kann. Somit leistet der in diesem Artikel beschriebene Ansatz einen wesentlichen Beitrag für die Industrieakzeptanz und industrielle Anwendbarkeit von Modellierungsmethoden, da neu entwickelte Modellierungsmethoden mit den in industriellen Werkzeugketten vorhandenen UML/SysML-Werkzeugen verwendet werden können, was bedeutet, dass keine neuen Werkzeuge, zusammen mit einer neuen Modellierungsmethode, eingeführt werden müssen.

Literaturverzeichnis

- [AP08] Alanen, M.; Porres, I.: A Metamodeling Language Supporting Subset and Union Properties. *Softw Syst Model* 7(1), 2008, S. 103-124.
- [BB12] Beetz, K.; Böhm, W.: Challenges in Engineering of Software-Intensive Embedded Systems. In (Pohl, K.; Hönniger, H.; Achatz, R.; Broy, M. Hrsg.): *Model-Based Engineering of Embedded Systems – The SPES 2020 Methodology*, Springer, Heidelberg, 2012.
- [Br12] Broy, M.; Damm, W.; Henkler, S.; Pohl, K.; Vogelsang, A.; Weyer, T.: Introduction to the SPES Modeling Framework. In (Pohl, K.; Hönniger, H.; Achatz, R.; Broy, M. Hrsg.): *Model-Based Engineering of Embedded Systems – The SPES 2020 Methodology*, Springer, Heidelberg, 2012.
- [Da06] Davis, I.; Green, P.; Rosemann, M; Idulska, M.; Gallo, S.: How do practitioners use conceptual modeling in practice? *Data & Knowledge Engineering* 58, 2006, S. 358-380.
- [Da12] Daun, M.; Tenbergen, B.; Weyer, T.: Requirements Viewpoint. In (Pohl, K.; Hönniger, H.; Achatz, R.; Broy, M. Hrsg.): *Model-Based Engineering of Embedded Systems – The SPES 2020 Methodology*, Springer, Heidelberg, 2012.
- [FP10] Fowler, M.; Parsons, R.: *Domain Specific Languages*. Addison-Wesley Longman, Amsterdam, 2010.
- [FV04] Fuentes-Fernández, L.; Vallecillo-Moreno, A.: An Introduction to UML Profiles. *Upgrade* 5(2), 2004, S. 6-13.
- [GLT03] Graaf, B.; Lormans, M.; Toetenel, H.: Embedded Software Engineering: The State of the Practice. *IEEE Softw* 20(6), 2003, S. 61-69.
- [KT08] Kelly, St.; Tolvanen, J.-P.: *Domain-specific Modeling - Enabling Full Code Generation*. John Wiley & Sohns, New Jersey, 2008.
- [La08] Lagarde, F.; Espinoza, H.; Terrier, F.; André, Ch.; Gérard, S.: Leveraging Patterns on Domain Models to Improve UML Profile Definition. In (Luiz, J.; Inverardi, P. Hrsg.): *Proc. Fundamental Approaches to Software Engineering*. Springer, Heidelberg, 2008.
- [La09] van Lamsweerde, A.: *Requirements Engineering: From System Goals to UML Models to Software Specifications*. John Wiley & Sohns, New Jersey, 2009.
- [LPR93] Lubars, M.; Potts, C.; Richter, C.: A review of the state of the practice in requirements modeling. In: *Proc. IEEE Int. Symp. Requirements Engineering*, 1993.
- [MOF06] Object Management Group: *Meta Object Facility Version 2.0*, OMG Document Number formal/2006-01-01, 2006.
- [Se07] Selic, B.: A Systematic Approach to Domain-Specific Language Design Using UML. In: *Proc. 10th IEEE Int. Symp. Object and Component-Oriented Real-Time Distributed Computing*, 2007.
- [STP12] Sikora, E.; Tenbergen, B.; Pohl, K.: Industry needs and research directions in requirements engineering for embedded systems. In: *Requirements Engineering* 17(1), 2012, S. 57-78.
- [WS07] Weisemöller, I.; Schürr, A.: A Comparison of Standard Compliant Ways to Define Domain Specific Languages. In: *Proc. Models in Software Engineering*, Springer, 2007.