

Heterogene Modellierung - Verknüpfung und Integration von Systemmodellen der SysML mit CAD-Modellen

Thomas Schumacher^{1*}, David Inkermann¹

¹ Technische Universität Clausthal, Institut für Maschinenwesen, Lehrstuhl für Rechnerintegrierte Produktentwicklung

* Korrespondierender Autor:

Thomas Schumacher
Fritz-Süchting-Institut für Maschinenwesen der Technischen Universität Clausthal
Robert-Koch-Str. 32
D-38678 Clausthal-Zellerfeld
Telefon: +495323/723504
Mail: schumacher@imw.tu-clausthal.de

Abstract

The development of mechatronic systems requires domain-specific and domain-independent models. Missing approaches to consistently link domain-independent and domain-specific models leads to a lack of acceptance of modelling languages, like SysML. Heterogeneous models present an approach to integrate detailed domain-specific and abstract domain-independent model elements into one model. This paper introduces the heterogeneous modelling approach and presents a technical concept for linking (representation) and integrating (presentation) model elements from SysML-diagrams and CAD-models. In addition, the advantages of heterogeneous models for specific development use cases are explained.

Keywords

Model-based systems engineering, heterogeneous modelling, SysML, CAD, Data exchange

1. Einleitung und Motivation

Jede Entwicklungsdomäne, wie Mechanik, Elektrik/Elektronik und Software, nutzt eigene Entwicklungsmodelle um Anforderungen, Systemverhalten und –struktur zu beschreiben. Mechatronische Systemen erfordern eine domänenübergreifende Entwicklung, die zur Funktionserfüllung und Systemrealisierung Lösungen der verschiedenen Domänen integriert. Bei der rechnerunterstützten Produktentwicklung müssen daher neben domänenspezifischen auch domänenübergreifende Modelle erzeugt, beurteilt und integriert werden. Modellbasiertes Systems Engineering zielt auf die Durchgängigkeit und Konsistenz von Systembeschreibungen der unterschiedlichen Entwicklungsdomänen ab und nutzt semiformale Modelle für die domänenübergreifende Systemmodellierung. Eine Herausforderung die oft nicht ausreichend betrachtet wird, ist die Schnittstelle und Verknüpfung zwischen domänenübergreifender und domänenspezifischer Modellierung. Bisher fehlen Ansätze zur Verknüpfung und Integration domänenübergreifender und –spezifischer Modelle, wodurch Durchgängigkeit und Konsistenz von Informationen deutlich eingeschränkt werden. In diesem Beitrag wird ein technisches Konzept zur Integration und Verknüpfung verschiedener Modellelemente vorgestellt und der Nutzen heterogener Modelle aufgezeigt.

1.1. Herausforderungen in der modellbasierten Systementwicklung

Eine wesentliche Voraussetzung für die erfolgreiche Entwicklung mechatronischer Produkte ist ein domänenübergreifendes Systemverständnis. Vorgehensmodelle des Systems Engineering sehen hierfür ein Top-Down Vorgehen vor, dass über eine schrittweise Definition und Dekomposition des Gesamtsystems Spezifikationen für die domänenspezifische Entwicklung bereitstellt [1]. Ein Ansatz das domänenübergreifende Systemverständnis zu verbessern sind semiformale Sprachen, wie UML/ SysML, die in der modellbasierten Systementwicklung eingesetzt werden. Diese Modellierungsansätze verknüpfen Anforderungs-, Verhaltens- und Strukturbeschreibung in einem Systemmodell [2, 3]. Gleichzeitig werden domänenspezifische Wechselwirkungen und Festlegungen wie z.B. räumliche Anordnungen nicht ausreichend abgebildet. Hierzu sind domänenspezifische Modelle, wie bspw. CAD-Modelle für eine räumliche, geometrische Produktbeschreibung oder zur Analyse und Bewertung potentieller Störeffekte (z.B. Temperatureinflüsse) erforderlich. Für eine durchgängige Lösungsbeschreibung ist eine Verknüpfung domänenübergreifender und domänenspezifischer Modelle notwendig [4, 5]. Fehlende Ansätze domänenübergreifende und -spezifische Modelle konsistent zu verknüpfen, resultiert in geringer Akzeptanz domänenübergreifender Sprachen, auch weil die erforderlichen Abstraktionen bestehender domänenspezifischer Modelle zeitaufwändig sind [6]. Heterogene Modelle sind ein Ansatz, unterschiedliche Modellelemente in eine Präsentation zu integrieren und damit die Konsistenz von Produktinformationen im Entwicklungsprozess zu verbessern. Gleichzeitig muss die Konsistenz sowohl innerhalb der Modellpräsentation als auch -repräsentation sichergestellt werden [4].

1.2. Forschungsziel und Vorgehensweise

In diesem Beitrag wird ein Ansatz für die Verknüpfung von Modellelementen aus SysML-Diagrammen und Geometriemodellen der mechanischen Domäne vorgestellt werden. Ziel ist es, die Verknüpfung der einzelnen Modellelemente technisch zu lösen (Repräsentation) und die Zusammenführung der verschiedenen Modellelemente in einer Darstellung (Präsentation) aufzuzeigen. Anhand konkreter Anwendungsfälle werden die Vorteile heterogener Modelle bei der Entwicklung mechatronischer Systeme dargestellt. Hieraus ergibt sich die folgende Forschungsfrage:

-
- Wie können Produktbeschreibungen der SysML mit CAD-Modellen konsistent verknüpft und in einer Darstellung integriert werden und welche Vorteile resultieren daraus?

Dieser Beitrag führt als Grundlage für das Verständnis heterogener Modelle zunächst in Strategien zur Modellstrukturierung ein, hierbei wird u.a. in View und Viewpoint unterschieden. Anschließend werden zentrale Begriffe erläutert und Vorgehensweisen zum Datenaustausch auf Grundlage des integrierten Produktdatenmodells vorgestellt. Weiter wird die Notwendigkeit heterogener Modelle in der modellbasierten Systementwicklung erläutert und zwei Heterogenitätsdimensionen zur Unterscheidung heterogener Modelle eingeführt. Dafür werden die Ergebnisse einer Literaturrecherche hinsichtlich Heterogenität in der domänenübergreifenden Systemmodellierung vorgestellt. Außerdem wird ein Konzept zur Verknüpfung und Integration von SysML-Modellelementen und CAD-Modellen aufgezeigt. Grundlage hierfür sind bestehende technische Lösungen sowie Standards für die Produktdatenrepräsentation und –austausch. Abschließend werden Ansätze zur Sicherstellung der Konsistenz zwischen heterogenen Modellen eingeführt und die zentralen Ergebnisse dieses Beitrags zusammengefasst.

2. Stand der Forschung und Technik

In diesem Abschnitt werden Strategien zur Modellstrukturierung vorgestellt, relevante Begriffe erläutert und Konzepte für den Produktdatenaustausch beschrieben. Zudem wird das Verständnis der heterogenen Modellierung erläutert.

2.1. Strategien zur Modellstrukturierung

Um einen Entwicklungsprozess vollständig modellbasiert zu unterstützen, ist eine willkürliche Erstellung von SysML-Diagrammen nicht zielführend. Da zwischen den verschiedenen Partialmodellen Relationen bestehen, sind Strategien zur Modellstrukturierung erforderlich. Hierbei sind zwei grundsätzliche Ansätze zu unterscheiden: Strukturierung von Modellen anhand des Abstraktionsgrads [7] und Strukturierung anhand der Sichtweise auf das System. Insbesondere für den Architekturprozess, welcher die Zuweisung der Funktionen auf die realisierenden Systemelemente vornimmt, ist die Strukturierung anhand des Abstraktionsgrades relevant. Auch die Strukturierung anhand verschiedener Systemsichtweisen und deren Realisierung durch konkrete Diagramme ist im modellbasierten Systems Engineering etabliert [3]. Anhand der beschriebenen Strukturierungsstrategien wird definiert, welche (Partial-)Modelle zu welchen Zeitpunkten im Systementwurf und der Systemspezifikation erstellt werden sollten. Gleichzeitig bestehen zwischen den (Partial-)Modellen konkrete Abhängigkeiten, wie bspw. die Erfüllung einer Anforderung durch ein Strukturelement, die während der fortschreitenden Entwicklung zu beschreiben sind. Zur Strukturierung dieser Modellvielfalt ist daher innerhalb der Präsentation eine Unterscheidung der Begriffe Viewpoint und View zweckmäßig. Eine View auf ein System ist eine Präsentation aus der Perspektive eines bestimmten Interessenstandpunktes. Dieser Interessenstandpunkt ist u.a. von der Entwicklungsdomäne, der Entwicklungsaufgabe und dem zu betrachtenden Systemumfang abhängig [8]. Die Präsentationen in einer modellbasierten Systementwicklung wird üblicherweise in semiformalen Modellen oder Partialmodellen (Ausschnitt Gesamtsystemmodell) dargestellt, demzufolge entspricht eine View einem Partialmodell bzw. Diagramm. Viewpoints umfassen zwei Bestandteile. Einerseits formuliert ein Viewpoint die Interessen eines jeden relevanten Stakeholders und andererseits definiert ein Viewpoint, die Konstruktions-, Interpretations- und Analysetechniken [8]. Diese Techniken umfassen insbesondere spezifische Notationen und Modellierungstechniken (wie bspw. den Einsatz von Modellierungssprachen, Notationen, Diagrammtypen, Designbestimmungen oder Modellierungsmethoden). Ein Viewpoint kann somit als Sammlung von Vorgaben und Restriktionen zur Erstellung von Views betrachtet werden. In einer modellbasierten

Entwicklung werden üblicherweise vier Viewpoints unterschieden: Anforderungen, Systemverhalten, Systemstruktur und Systemparameter [3], da diese erforderlich sind, um eine umfassende Systemspezifikation und -design zu entwickeln.

2.2. Bestehende Konzepte im erweiterten industriellem Umfeld

Im erweiterten industriellem Umfeld existieren Konzepte und Lösungen zur durchgängigen, konsistenten Modellierung technischer Systeme. Beispiele sind der mechatronic concept designer von Siemens, Dassault Systèmes digital MockUp oder Syndeia von Intercax. Der Siemens mechatronic concept designer fördert aufgrund eines funktionalen Entwicklungsansatzes die parallele Entwicklung unterschiedlicher Domänen. Ausgehend von den Anforderungen können funktionale Modelle erzeugt und simuliert werden, um verschiedene Lösungsalternativen zu erarbeiten und eine frühzeitige Validierung des Maschinenverhaltens zu ermöglichen [9]. Der Dassault Systèmes digital MockUp unterstützt insbesondere die Problemanalyse und -lösung während der Integration technischer Systeme durch Verknüpfung virtueller Modelle unterschiedlicher Systemebenen (z.B. Integration von Fahrzeugsystemen). Die Softwareplattform Syndeia ermöglicht eine domänenübergreifende Systementwicklung indem u.a. Modellinformationen zwischen CAD- und SysML-Autorentools ausgetauscht werden können [10]. Im Verbundprojekt FAS4M wurde die systematische Entwicklung und Modellierung mechatronischer Produkte (Funktions-Gestalt-Zusammenhang) aus Sichtweise der mechanischen Domäne betrachtet. Neben einer Modellierungssprache (MechML) wurde eine Schnittstelle entwickelt, um CAD- und SysML-Werkzeuge zu verknüpfen. Dadurch ist bspw. eine Integration von Konstruktionsskizzen in die SysML-Modellierung möglich [11]. Generell lässt sich feststellen, dass die vorgestellten Konzepte in Abhängigkeit zum jeweiligen Verwendungszweck eine durchgängige Verwendung von Modellen unterschiedlicher Konkretisierungsebenen und Sprachen (Repräsentationen) in einer Toolumgebung erlauben. Die Integration von Modellelementen verschiedener Abstraktionen in eine Präsentation wird hierbei jedoch nicht betrachtet.

2.3. Heterogenität und Konsistenz von Modellen

Für die Realisierung heterogener Modelle ist ein grundlegendes Verständnis der Begriffe Heterogenität und Konsistenz im Zusammenhang mit Modellen erforderlich. Allgemein gibt Heterogenität die grundsätzliche Verschieden- oder Ungleichartigkeit sowie eine Uneinheitlichkeit im Aufbau bzw. in der Zusammensetzung an. Bei heterogenen Modellen drückt sich Heterogenität insbesondere durch unterschiedliche Modellierungssprachen, Repräsentationen oder Beschreibungsumfängen (Detaillierungen) der verknüpften Partialmodelle oder einzelner Modellelemente aus. Die Verknüpfung und Integration verschiedener Modellelemente in eine konsistente Präsentation erfordert damit einen besonderen Aufwand, da die Modellelemente aus unterschiedlichen Datenstrukturen (Repräsentation) erfasst werden müssen [12]. Im Abschnitt 2.5 werden auf Grundlage dieser Begriffserläuterung verschiedene Heterogenitätsdimensionen und deren Einfluss auf heterogene Modelle abgeleitet.

Ein zentraler Aspekt bei der Verknüpfung verschiedenartiger Modellelemente zu einem heterogenen Modell ist die Sicherstellung der Modellkonsistenz. Ein konsistentes, heterogenes Modell zeichnet sich durch eine widerspruchsfreie Beschreibung und Darstellung der Modellelemente oder der Partialmodelle aus. Üblicherweise folgen Modelle den gleichen Modellierungsansatz und verwenden kompatible Methoden zur Modellintegration [12] und sind damit per se konsistent. Dies gilt für heterogene Modelle nicht, da verschiedenartige Modellelemente verknüpft werden und die Konsistenz z.B. hinsichtlich Strukturmerkmalen gewährleistet werden muss.

2.4. Produktdefinition, -repräsentation und -präsentation

Da sich Produktdaten in den verschiedenen Produktlebensphasen, insbesondere hinsichtlich Abstraktionsgrad und Verwendungszweck, deutlich unterscheiden, ist ein einheitliches Produktdatenverständnis von besonderer Bedeutung. Einen Ansatz hierzu bietet die Norm ISO 10303 „Product Data Representation and Exchange“. Zentraler Inhalt dieses Standards ist die Unterscheidung von Produktdaten in Daten zur Produktdefinition, Produktrepräsentation und Produktpräsentation.

Tabelle 1: Unterscheidung und Erläuterung von Produktdaten [13]

Produktdaten	Beschreibung / Erläuterung
Produktdefinition	Administrative und organisatorische Daten, bilden Merkmale zur Identifizierung und Klassifizierung von Produkten (z.B. Sachnummer), Einordnung des Produkts in den Produktlebenszyklus (z.B. Freigabestatus) sowie die Produktstruktur (z.B. Variante) ab.
Produktrepräsentation	Produktbeschreibende Daten, verwendet zur rechnerverarbeitbaren Datenabbildung der Produktmerkmale. Durch Analyse- und Simulationsmethoden können Aussagen über das Produktverhalten abgeleitet werden. Die Produktrepräsentation umfasst sowohl Geometriedaten für die Produktgestalt, wie auch technische und organisatorische Produktdaten, die in Form systemspezifischer oder genormter Datenmodelle repräsentiert werden.
Produktpräsentation	Abgeleitet von der Produktrepräsentation, um deren Inhalte graphisch oder textuell darzustellen. Die Produktpräsentation umfasst somit die Produktdokumentation, wie bspw. Architekturmodelle oder Verhaltensdiagramme. Dabei präsentieren die verwendeten Modelle immer benutzerspezifische Sichten auf das Produkt um einen konkreten Zweck abzubilden (siehe Abschnitt 2.1).

Für den weiteren Inhalt dieses Beitrags sind insbesondere die Produktdaten zur Produktrepräsentation und –präsentation relevant.

2.4.1. Produktdatenaustausch

Innerhalb einer modellbasierten Entwicklung entstehen eine Vielzahl domänenspezifischer und –übergreifender Modelle durch unterschiedliche Entwicklungsbereiche. Um einen effizienten Entwicklungsprozess zu ermöglichen, ist ein Austausch von Modellen bzw. Modellelementen essentiell. Grundsätzlich lassen sich hierbei zwei Varianten unterscheiden: Ein Austausch von Inhalten zwischen einheitlichen bzw. vergleichbaren Modellrepräsentationen und zwischen verschiedenartigen Modellrepräsentationen.

Der Austausch von Produktdaten auf Basis vergleichbarer Modelle ist sowohl in den meisten domänenspezifischen als auch domänenübergreifenden Modellierungstechniken und –werkzeugen etabliert. In der Mechanikdomäne ist ein Austausch von Produktdaten anhand der Normreihe STEP weit verbreitet. Grundlage für den Austausch von Produktinformationen ist die Definition eines übergeordneten Strukturformats und die einheitliche Beschreibung der Produktdaten durch EXPRESS [14]. Dadurch werden alle Produktdaten auf Basis einer international anerkannten, vorgegebenen Struktur beschrieben. Um den Anforderungen ausgewählter Anwendungsbereiche gerecht zu werden, bietet STEP verschiedene Anwendungsprotokolle (AP), die jeweils spezifische Datenmodelle umfassen. Das AP 242 („AP: Modellbasiertes 3D-Engineering“) wird bspw. in der Automobil-, Luftfahrtindustrie und Maschinenbau verwendet um Entwicklungs- und Produktdaten auszutauschen und zu archivieren. Im Kontext dieses Beitrags ist das AP 233 („AP: Systems Engineering – Systementwicklung“) von besonderem Interesse, da es für den Austausch von Systems Engineering Daten entwickelt wurde, siehe Abschnitt 3.1. Auch der Austausch von Modellen bzw. -elementen innerhalb des modellbasierten Systems Engineering ist inzwischen weitgehend standardisiert. Üblicherweise wird bei Anwendung von SysML oder UML das Dateiformat XML Metadata Interchange (XMI) verwendet. Spezifiziert wurde dieser Standard von der Object Management Group und stellt ein XML-basiertes Austauschformat dar [15].

Ein Datenaustausch zwischen verschiedenartigen Modellen kann zumeist nicht über ein standardisiertes, neutrales Datenformat erfolgen. Um dennoch ein Datenaustausch zu realisieren, ist eine Konvertierung der Dateninhalte und -struktur von dem Ausgangsformat in das Zielformat vorzunehmen. Dazu werden s.g. Parser (oder Zerteiler) eingesetzt, dieser identifiziert Sprach- oder Textelemente (Syntaxanalyse) und übersetzt diese in einen lesbaren Inhalt. Anschließend werden die Inhalte durch den Parser in vordefiniertes Datenformat (Zielformat) umgewandelt [16]. Das Bild 1 stellt diesen Vorgang schematisch dar.

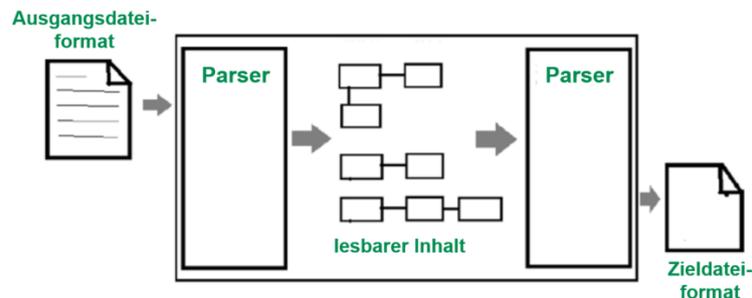


Bild 1: Konvertierung verschiedenartiger Dateiformate [basierend auf 16]

Der dargestellte Konvertierungsvorgang ermöglicht Dateninhalte und -formate zu harmonisieren und somit konsistente Präsentationen, wie bspw. heterogene Modelle abzuleiten.

2.5. Heterogene Modelle in der Systementwicklung

Heterogene Modelle bieten die Möglichkeit Partialmodelle oder Modellelemente domänen-spezifischer sowie –übergreifender Modelle zu integrieren [17]. Beispielsweise können Funktionen, Wirkprinzipien, konkrete Systemelemente sowie räumliche Relationen und Anforderungen in einem Modell (Präsentation) zusammengeführt werden. Jansen [18] stellt dies am Beispiel eines mechatronischen Beins dar. In diesem Modell werden sowohl dreidimensionale Körper als auch zweidimensionale Ersatzmodelle einschließlich wesentlicher Relationen abgebildet. Außerdem sind Kontextinformationen, wie Bauraumrestriktion als Designconstraint, enthalten. Durch die Kombination unterschiedlich konkreter Modellelemente entstehen für die jeweilige Fragestellung aussagekräftige Modelle, die Entwicklungsergebnisse und -analysen flexibel erfassen und darstellen [18]. Beispielsweise unterstützen heterogene Modelle bei der Zuordnung von Funktionen zu den realisierenden Domänen (funktionale Partitionierung) oder bei der räumlichen Positionierung von Systemelementen aufgrund funktionaler Abhängigkeiten (räumliche Partitionierung) [18], da sowohl funktionale als auch räumliche Aspekte dargestellt werden. Es muss jedoch beachtet werden, dass die Verknüpfung verschiedenartiger Modellelemente in einer Präsentation, immer auch eine Verknüpfung der Repräsentation (Datenstruktur) erfordert.

Die Heterogenität in Modellen drückt sich durch unterschiedliche Faktoren aus. Auf Basis einer Literaturrecherche konnten folgende Faktoren der Heterogenität in einer modellbasierten Entwicklung identifiziert werden [4, 12, 15, 18, 19, 20]:

- Modellelemente unterschiedlicher Abstraktion bzw. Konkretisierung
- Modellelemente unterschiedlicher Formalisierung
- Modellelemente unterschiedlicher Syntax
- Modellelemente unterschiedlicher Semantik
- Modellelementen aus unterschiedlichen Metamodellen
- Modellelementen mit unterschiedlichen Ursprung

Aus diesen Faktoren lassen sich zwei übergeordnete Heterogenitätsdimensionen ableiten, siehe Tabelle 2. Auf Basis dieser Heterogenitätsdimensionen lässt sich konstatieren, dass Modelle sowohl in der Repräsentation als auch in der Präsentation unterschiedlich sein können. Da die Modellpräsentation üblicherweise von der Modellrepräsentation abgeleitet ist (siehe Abschnitt 2.4), müssen heterogene Modell gleichzeitig eine Verknüpfung der Präsentation als auch eine Integration der Repräsentation sicherstellen. Dieser Beitrag fokussiert insbesondere die Modellrepräsentation als Grundlage für die Modellpräsentation.

Tabelle 2: Ableitung von Heterogenitätsdimensionen

Heterogenitätsdimension	Beschreibung / Erläuterung
Modellrepräsentation	Unterschiede in dem zugrundeliegenden Metamodell, in der angewendeten Modellierungssprache, inkl. Semantik und Syntax.
Modellpräsentation	Unterschiede in dem Abstraktions- bzw. Konkretisierungsgrad sowie der Formalisierung der Modellelemente.

3. Verknüpfung und Integration von Systemmodellen der SysML mit CAD-Modellen

In diesem Abschnitt wird ein technisches Konzept zur Verknüpfung von Modellelemente der SysML und CAD-Modelle (Repräsentation) und die Zusammenführung der unterschiedlichen Modellelemente in einer Visualisierung (Präsentation) vorgestellt. Abschließend wird erläutert warum Inkonsistenzen bei der Entwicklung mechatronischer Systeme auftreten und wie das dargestellte Konzept zur Sicherstellung der Modellkonsistenz beiträgt.

3.1. Technisches Konzept zur Verknüpfung heterogener Modellrepräsentationen

Autorentools der SysML nutzen, wie im Bild 2 dargestellt, üblicherweise das XMI-Format und CAD-Autorentools das STEP-Format für den Datenaustausch. Zentraler Punkt ist die Überführung und Bereitstellung der jeweiligen Dateiinhalte für das andere Autorentool in les- und interpretierbarer Form. Dazu muss zunächst die zugrundeliegenden Dateistrukturen mittels Mapping abgestimmt werden. Dafür gibt die ISO 10303-25 ein erstes, standardisiertes Datenmodell vor, welches auf STEP AP 233 und SysML bzw. XMI aufbaut [21]. Das AP 233 wurde für den Austausch von Systems Engineering Daten, wie Anforderungen, funktionale und strukturelle Daten, physikalische Strukturen, Konfigurations- und Projektmanagementdaten konzipiert [22]. Unter Einbeziehung der in Abschnitt 2.4.1 dargestellten Vorgehensweise zur Konvertierung unterschiedlicher Datenformate lassen sich Modellinformationen zwischen SysML- und CAD-Autorentools austauschen. Konkret ist es auf Basis des standardisierten Datenmodells (ISO 10303-25) und abgestimmten Mappingdefinitionen zwischen SysML/ XMI und STEP AP 233 möglich Blocks, Properties, Activities, Interactions, State Machines, Use Cases, Requirements, Packages und Metadaten zwischen SysML und CAD auszutauschen. Als Beispiel können somit Strukturinformationen (z.B. Systemelemente die zur Funktionsrealisierung erforderlich sind) in Form von Blöcken und Properties (z.B. Geometriemerkmale) an das CAD-System übergeben und fortlaufend detailliert werden. Bild 2 stellt das Konzept zum Austausch von SysML- und CAD-Modellinhalten schematisch dar.

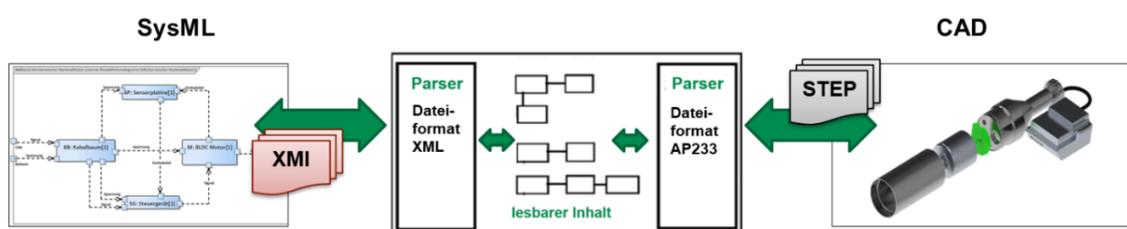


Bild 2: Konzept zum Austausch von Modellinhalten zwischen SysML- und CAD-Autorentools

Für eine durchgängige Modellierung mechatronischer Systeme muss das bestehende Mapping verfeinert und erweitert werden. Beispielsweise unterscheiden sich Blöcke und deren Relationen, welche eine mechanische Baugruppe, wie ein Getriebe, darstellen, grundlegend von Blöcken und deren Relationen die eine Softwaremodul beschreiben. Daher sollten Blöcke inklusive deren Beschreibungen, wie Parametern oder Properties, nach Domänen unterschieden werden [23]. Diese Unterscheidung in verschiedene Blockstereotypen und die Definition zweckmäßiger Verknüpfungen und Schnittstellen sollen in folgenden Arbeiten weiter untersucht werden. Beispielsweise ist es sinnvoll, bereits in SysML festgelegte Strukturmerkmale wie Aggregationen von Bauteilen zu einer Baugruppe oder Schnittstellen zwischen Teilsystemen in ein CAD-Modell zu überführen. Auf diese Weise können funktionale Wechselwirkungen auch in räumlichen CAD-Modellen aufgezeigt werden.

3.2. Integration heterogener Elemente in eine Modellpräsentation

Durch Verknüpfung verschiedener Modellrepräsentationen lässt sich eine integrierte Modellpräsentation ableiten. Wie in Bild 3 an einer Baugruppe eines aktiven Wankstabilisators dargestellt, können domänenübergreifende (SysML) und domänenspezifische (CAD) Modellelemente in eine Präsentation integriert werden. Diese integrierte Modellpräsentation stellt den wesentlichen Mehrwert gegenüber den in Abschnitt 2.2 vorgestellten Konzepten dar.

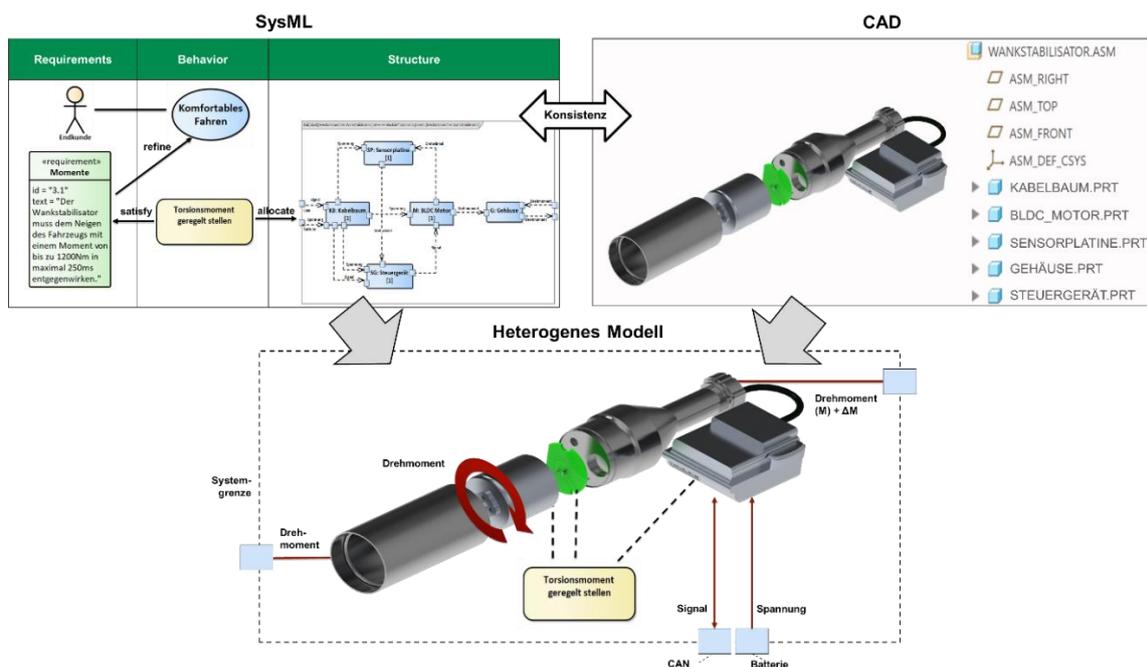


Bild 3: Heterogene Modellpräsentation

In dieser heterogenen Visualisierung werden Modellelemente der SysML zweckgebunden in ein CAD-Modell integriert. Die SysML-Aktivität „Torsionsmoment geregelt stellen“ lässt sich mit den Baugruppen E-Motor, Sensorik und Steuergerät realisieren. Durch die Integration in das CAD-Modell sind die Baugruppen innerhalb der Systemgrenze räumlich positionierbar. In diesem Beispiel würde der E-Motor inklusive Sensorik in das Gehäuse integriert und das Steuergerät über einen Kabelsatz außerhalb des Aktuators positioniert. Außerdem hat dieses heterogene Modell den Zweck, einer umfänglichen Darstellung der Systemschnittstellen. Daher wurden dem CAD-Modell die in SysML modellierten Ports sowie Stoff-, Energie- und Signalflüsse des Systems mit der Umwelt ergänzt. Funktional detailliert dieses heterogene Modell die integrierte Aktivität, indem bei zur Verfügung stehender Spannung und entsprechender

Steuersignale dem vorhandenen Drehmoment durch Aktuierung ein zusätzliches Drehmoment (ΔM) hinzugefügt wird.

3.3. Vorteile heterogener Modelle im modellbasierten Entwicklungsprozess

Verknüpfung und Integration verschiedenartiger Modellelemente in ein heterogenes Modell bietet mehrere Vorteile, diese werden anhand folgender Anwendungsfälle aufgezeigt [10].

1. Eine Visualisierung eines CAD-Modells direkt aus dem SysML-Diagramm heraus verbessert die Vorstellung von Teilsystemen und deren Schnittstellen. Außerdem unterstützt eine räumliche Darstellung, in Form eines CAD-Modells, bei der Positionierung von Teilsystemen, z.B. Sensoren, im Gesamtsystem.
2. Die Überführung eines bestehenden CAD-Modells in eine Strukturbeschreibung der SysML (z.B. Block Definition Diagram) unterstützt bei der Integration bestehender Teilsysteme in ein neues Gesamtsystem und erhöht die Effizienz im Entwicklungsprozess.
3. Die Bereitstellung bereits in SysML-Diagrammen definierten Anforderungen, Strukturmerkmalen (Aggregationen und Schnittstellen) und Designconstraints, wie bspw. Masseigenschaften oder geometrischen Parametern, unterstützen bei der Designdetaillierung. Die vorgenommenen Detaillierungen können anschließend bspw. in Form zusätzlicher Modellelemente (Blöcke, Beschreibungen) zurück in das SysML-Diagramm überführt werden, dies unterstützt die Durchgängigkeit zwischen verschiedenen Modellen.

3.4. Sicherstellung der Konsistenz zwischen verknüpften Modellen

Inkonsistenzen zwischen Modellen sind bei der Entwicklung mechatronischer Systeme auch bei Anwendung geeigneter Prozesse, Methoden und Tools nicht vollständig zu vermeiden. Modellinkonsistenzen können aus den folgenden Gründen auftreten [12]:

1. parallele Entwicklung unterschiedlicher Modelle durch verschiedene Domänen,
2. unzureichendes Verständnis der Abhängigkeiten zwischen Modellen,
3. unklare oder mehrdeutige Anforderungen in einem frühen Entwicklungsstadium,
4. unvollständige Modelle aufgrund unbekannter, aber wesentlicher Informationen.

Das dargestellte technische Konzept (Abschnitt 3.1) trägt durch die Verknüpfung der Modellrepräsentationen zur Verbesserung der Konsistenz heterogener Modellpräsentationen bei. Die Modellkonsistenz soll insbesondere durch die Detaillierung der Mappingdefinitionen (z.B. Blockstereotypen) und der Festlegung zweckmäßiger SysML-CAD-Verknüpfungen (z.B. Systemeigenschaften) zukünftig erhöht werden.

4. Zusammenfassung und Ausblick

Dieser Beitrag beschreibt den Bedarf und die Vorteile einer Verknüpfung domänenübergreifender und –spezifischer Modelle in der Systementwicklung. Hierzu wurden zunächst in die modellbasierte Systementwicklung eingeführt und zentrale Begriffe erläutert. Das Konzept heterogener Modelle wurde erläutert und aufgezeigt, wie SysML-Modellelemente mit CAD-Modellelemente verknüpft werden können, um ein heterogenes Modell zu erzeugen. Abschließend wurden die Vorteile heterogener Modellierung dargestellt. Zukünftig soll das technische Konzept für die Zusammenführung verschiedenartiger Modellelemente in einer gemeinsamen Präsentation als Prototyp realisiert werden. Dabei besteht der Forschungsgegenstand insbesondere in der Definition konkreter Verknüpfungen innerhalb der Modellrepräsentationen um

die räumliche Positionierung von Systemelementen (räumliche Partitionierung) und die Zuordnung von Funktionen zu den umsetzenden Domänen (funktionale Partitionierung) zu realisieren. Abschließend ist kritisch anzumerken, dass das verwendete AP 233 derzeit noch nicht von allen CAD-Autorentools unterstützt wird, welches die technische Anbindung erschwert.

Danksagung

Der vorliegende Beitrag ist Teil der Arbeiten des niedersächsischen Zukunftslabors Mobilität. Das Teilprojekt wird gefördert vom Niedersächsischen Ministerium für Wissenschaft und Kultur (Fördernummer ZN3493) im Niedersächsischen Vorab der VolkswagenStiftung und betreut vom Zentrum für digitale Innovationen Niedersachsen (ZDIN).

Literaturverzeichnis

- [1] Eigner, Martin; Roubanov, Daniil; Zafirov Radoslav: Modellbasierte virtuelle Produktentwicklung. Heidelberg: Springer-Verlag, 2014, S.5.
- [2] Friedenthal, Sanford; Moore, Alan; Steiner, Rick: A Practical Guide to SysML. Amsterdam: MK/OMG Press, 2015, S.17, 18.
- [3] Morkevicius, Aurelijus et al.: MBSE Grid: A Simplified SysML-Based Approach for Modeling Complex Systems. 27th Annual INCOSE International Symposium, Adelaide, 2017.
- [4] Kattner, Niklas et al.: Inconsistency management in heterogeneous models – an approach for the identification of model dependencies and potential inconsistencies. ICED 19, Delft, 2019.
- [5] Zingel, Johannes: Basisdefinition einer gemeinsamen Sprache der Produktentwicklung im Kontext der Modellbildung technischer Systeme und einer Modellierungstechnik für Zielsystem und Objektsystem technischer Systeme in SysML auf Grundlage des ZHO-Prinzips. KIT, Dissertation, 2013, S72.
- [6] Ehrlenspiel, Klaus; Meerkamm, Harald: Integration Versus Spezialisierung: Von der Notwendigkeit einer ganzheitlichen Konstruktionsforschung und -lehre an Universitäten und Hochschulen. Konstruktion - Zeitschrift für Produktentwicklung und Ingenieur-Werkstoffe, 2016, S.8.
- [7] Schumacher, Thomas; Inkeremann, David: Herausforderungen und Anforderungen bei der durchgängigen Architekturmodellierung mechatronischer Systeme. 31. DfX-Symposium, Erlangen, 2020.
- [8] Norm ISO/IEC/IEEE 42010, 2011. Systems and software engineering – Architecture description.
- [9] Siemens PLM Software: Mechatronics Concept Designer - Ein funktionsorientierter Ansatz für den Maschinen- und Anlagenbau. 2010.
- [10] Bajaj, Manas; Cole, Bjorn; Zwemer, Dirk: Architecture to Geometry - Integrating System Models with Mechanical Design. AIAA SPACE, Long Beach, California, 2016.
- [11] Moeser, Georg et al.: Modellbasierter mechanischer Konzeptentwurf: Ergebnisse des FAS4M-Projektes. Herzogenaurach: 2016.
- [12] Avgoustinov, Nikolay: Modelling in Mechanical Engineering and Mechatronics. London: Springer, 2007.
- [13] Anderl, Reiner; Trippner, Dietmar: STEP STandard for the Exchange of Product Model Data. Leipzig: B. G. Teubner Stuttgart, 2000.
- [14] Norm ISO 10303-21, 2015: Industrial automation systems and integration – Product data representation and exchange – Part 21: Implementation methods: Clear text encoding of the exchange structure.
- [15] Object Management Group. OMG Systems Modeling Language (OMG SysML™). V1.6.
- [16] Zoabi, Firas: STEP/XML Based Macro Data Representation for Parametric CAD Model Exchange. Universität Stuttgart, Diplomarbeit, 2014.
- [17] Jansen, Sebastian, Welp, Ewald G.: A heterogeneous modelling approach for domain allocation in mechatronics. ICED 07, Paris, 2007.
- [18] Jansen, Sebastian: Eine Methodik zur modellbasierten Partitionierung mechatronischer Systeme. Ruhr Universität Bochum, Dissertation, 2006, S.82.
- [19] El Hamlaoui, Mahmoud et al.: Heterogeneous design models alignment: from matching to consistency management. SAC '18: Proceedings of the 33rd Annual ACM Symposium on Applied Computing, 2018.
- [20] Stachowiak, Herbert: Allgemeine Modelltheorie. Wien: Springer-Verlag, 1973.
- [21] Norm: ISO/TS 10303-25: Industrielle Automatisierungssysteme und Integration - Produktdatendarstellung und -austausch - Teil 25: Implementierungsmethoden: Datenbindung von EXPRESS zu XMI, 2005.
- [22] Weilkins, Tim: Systems Engineering with SysML/UML. Amsterdam: MK/OMG Press, 2008.
- [23] Kernschmidt, Konstantin: Interdisciplinary structural modeling of mechatronic production systems using SysML4Mechatronics. Technische Universität München, Dissertation, 2018, S.74, 75.