Wissensbasiertes und kollaboratives Systems Engineering mechatronischer Produkte

Stephan Husung, Faizan Faheem & Zirui Li

Abstract

Model-Based Systems Engineering (MBSE) gewinnt bei der Entwicklung komplexer mechatronischer Produkte zunehmend an Bedeutung. Zur effizienten Wiederverwendung von vorhandenem Wissen in der Entwicklung können MBSE-Modelle mit Wissensrepräsentationen, wie z.B. Wissensgraphen, gekoppelt werden. Dazu müssen Informationen zwischen den MBSE-Modellen und den Wissensrepräsentationen ausgetauscht werden. Für eine kollaborative Entwicklung müssen die Informationen in den MBSE-Modellen über die eigenen Unternehmensgrenzen hinweg ausgetauscht werden. Im Rahmen des Beitrags wird hierfür die Nutzung von Dataspaces diskutiert. Der Beitrag diskutiert Anforderungen, Konzepte und prototypische Implementierungen sowie anschauliche Anwendungsbeispiele.

DOI: 10.25368/2024.EEE.005

1 Motivation und Zielstellung

Um komplexe mechatronische Produkte zu entwickeln, werden zunehmend modellbasierte Ansätze eingesetzt. Diese reichen von Ansätzen mit semiformalen Modellen bis hin zu Ansätzen mit formalen Modellen für die Detailspezifikation (u.a. CAD- oder PCB-Modelle) sowie die Simulationen (Hick *et al.*, 2019). Insbesondere in den frühen Phasen der Entwicklung, in denen wesentliche Informationen für die Spezifikation noch nicht ausgereift zur Verfügung stehen, mehrere Varianten des Produktes untersucht werden müssen sowie zahlreiche Änderungen und Iterationen (u.a. im Kontext agiler Entwicklungsprozesse) durch den Erkenntnisgewinn in der Entwicklung notwendig werden, spielen Ansätze mit semiformalen Modellen eine entscheidende Rolle (Husung *et al.*, 2022b). Um komplexe mechatronische Produkte zu entwickeln, können für deren Beschreibung systemtheoretische Ansätze verwendet werden (Ropohl, 1975). Systemtheoretische Ansätze gehen davon aus, dass Systeme in mehrere Teilsysteme zerlegt werden können und das System wiederum in übergeordnete Systeme integriert werden kann. Für die Entwicklung ist häufig das sogenannte „System of Interest“ zur Erfüllung der Bedarfe der Stakeholder relevant. Dieses System wird im Folgenden als Gesamtsystem bezeichnet. Bei komplexen mechatronischen Produkten ergeben sich bei der Zerlegung des Systems häufig mehrere Systemebenen mit mechatronischen Subsystemen bis letztendlich auf den unteren Systemebenen domänenspezifische Subsysteme vorliegen (Husung *et al.*, 2022a). Für die semiformale Modellierung des mechatronischen Gesamtsystems und

der mechatronischen Subsysteme hat sich die standardisierte Modellierungssprache Systems Modeling Language (SysML) etabliert (Bajaj *et al.*, 2022; Friedenthal, 2014).

Wichtig für die Entwicklung ist, dass SysML-Modelle mit ihrer semiformalen Modellierung allein nicht ausreichen, um das Produkt umfassend zu spezifizieren und zu analysieren. Für eine konsistente und nachverfolgbare Spezifikation existieren mehrere Ansätze, um die SysML-Modelle mit u.a. Anforderungen (Husung *et al.*, 2022a), Detailspezifikationsmodellen (z.B. CAD-Modellen (Schumacher *et al.*, 2022; Moeser *et al.*, 2015)) oder Simulations-Modellen (Mahboob *et al.*, 2023) zu koppeln (Forte *et al.*, 2021; Hick *et al.*, 2019; Mahboob, 2021). Zur versionsabhängigen Verwaltung der einzelnen Modelle werden Datenmanagementlösungen eingesetzt.

Die Produktentwicklung besteht aus mehreren Phasen der Synthese, Analyse und Bewertung. Für die Synthese müssen zahlreiche Entscheidungen getroffen werden. Informationen über den aktuellen Stand des Produkts, einschließlich der Anforderungen und Spezifikationen, sowie Erfahrungswissen sind für diese Entscheidungen erforderlich (Mandel *et al.*, 2021). Vor allem in den frühen Entwicklungsphasen gibt es viele Unsicherheiten, da das notwendige Wissen über das spezifische Produkt in der geplanten Umgebung oft noch nicht in einem hohen Reifegrad vorliegt. Dazu kommt, dass die heutige Produktentwicklung häufig verteilt durchgeführt wird, so werden Zulieferer für Subsysteme oder Partner eingebunden (prostep ivip Verein, 2023). Daher liegen die Spezifikationen zum Gesamtprodukt und das Erfahrungswissen häufig über viele Personen und Unternehmen verteilt vor. Vor diesem Hintergrund steigt im Kontext der Produktentwicklung unter Nutzung modellbasierter Ansätze der Bedarf:

- Systemmodelle für die Spezifikation des aktuellen Produktes mit Wissensmanagement-Lösungen zu koppeln, um während der Entwicklung kontextabhängig auf bestehendes Wissen zurückgreifen zu können und
- Systemmodelle für die Anforderungen und die Spezifikationen der Subsysteme bis zum Gesamtsystem zwischen unterschiedlichen Entwicklungspartnern auszutauschen.

Der vorliegende Beitrag adressiert diese Bedarfe und diskutiert Konzepte für deren Erfüllung. In diesem Zusammenhang werden folgende Forschungsfragen diskutiert:

- Wie können semiformale Systemmodelle mit Wissensrepräsentationen (hier Wissensgraphen) gekoppelt werden, um bei der Modellierung auf vorhandenes Wissen zurückgreifen zu können?
- Wie können Systemmodelle (oder Teilmodelle) unter Nutzung von Dataspaces zwischen Partnern ausgetauscht werden, um eine gemeinsame Spezifikation des

Produktes zu realisieren und Erkenntnisse für gemeinsame Entscheidungsfindungen zu transferieren?

Der Beitrag diskutiert die prototypische Umsetzung sowie die Anwendung anhand anschaulicher Beispiele.

2 Konzept

2.1 Kopplung Systemmodelle mit Wissensgraphen

Systemmodelle dienen im Rahmen der Produktentwicklung zur strukturierten Erfassung von produkt- und projektspezifischen Informationen. Insbesondere unter Verwendung der standardisierten Modellierungssprache SysML ist eine semiformale Modellierung unter Anwendung des Metamodells und seiner möglichen Erweiterungen über Profile möglich. Da die Informationen meist projektspezifisch sind, ist eine Übertragung von Erkenntnissen zwischen den Projekten heute nur begrenzt möglich (Mandel *et al.*, 2023). Vor diesem Hintergrund rücken ontologie-basierte Repräsentation für semantische Informationen, wie Wissensgraphen, in den Fokus der Diskussionen. Wissensgraphen beschreiben vorliegendes Wissen. Ziel ist die Organisation von Wissen in strukturierter und maschinenlesbarer Form sowie die Ermöglichung einer flexiblen Abfrage von Informationen. Standardisiert sind u.a. Resource Description Framework (RDF) Graphen (Klyne *et al.*, 2014). Die Daten im RDF-Graphen sind durch Tripel repräsentiert. Als Abfragesprache wird u.a. SPARQL (Prud'hommeaux, 2013) genutzt.

Zur Nutzung der Wissensgraphen im Zusammenhang mit den Systemmodellen müssen auf Basis der aktuellen Spezifikation im Systemmodell (vom System of Interest) und ggf. weiteren Informationen die SPARQL-Abfragen erstellt werden. Da SysML-Modelle auf Metamodellen basieren, liegen grundsätzliche Beschreibungen vor, wie die SysML-Modelle durchsucht und im Zusammenhang mit den Typen der Modellelemente und Relationen Abfragen generiert werden können. Die Generierung der Abfragen ist jeweils problemspezifisch, z.B. zur Suche nach bekannten Fehlern. Damit die Erkenntnisse aus dem Wissensgraphen wiederverwendet werden können, ist eine Anreichung der Wissensgraphen notwendig. Dies kann u.a. durch die Analyse von bestehenden Systemmodellen hinsichtlich der relevanten Erkenntnisse erfolgen (siehe Abbildung 1). Die Nutzung von bestehenden Systemmodellen bietet den Vorteil, dass die Kontextinformationen im Systemmodell und mögliche semantische Informationen als Basis für die Anreichung mit verwendet werden können.

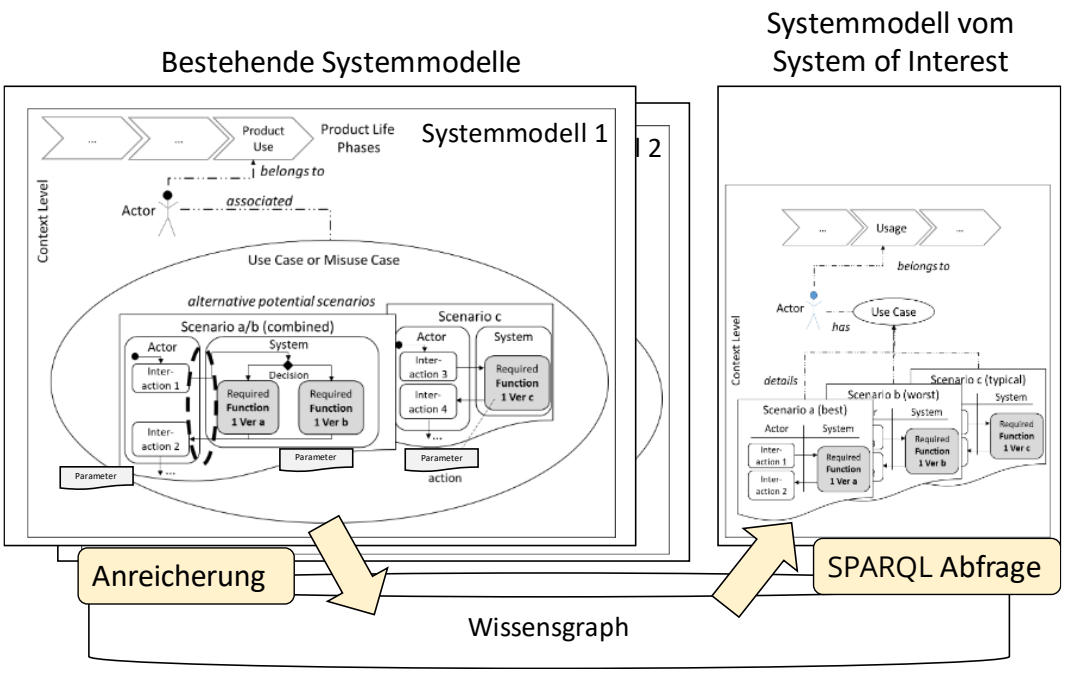

Abbildung 1: Konzept der Kopplung von Wissensgraphen mit Systemmodellen

Im Rahmen der aktuellen Forschungsaktivitäten werden über diesen Ansatz unter anderem Robustheits-Untersuchungen durchgeführt (Faheem *et al.*, 2023). Bei diesen Untersuchungen ist es das Ziel, über die Kopplung der Systemmodelle mit den Wissensgraphen auf Erkenntnisse zu bekannten Fehlern, Störeinflüssen und Maßnahmen zur Beherrschung der Fehler zurückzugreifen. Dafür werden im Wissensgraphen Erkenntnisse zu bekannten Fehlern im spezifischen Kontext abgelegt.

Für die Abfrage im Wissensgraphen müssen die notwendigen SPARQL-Abfragen erstellt werden. Die Abfragegenerierung erfolgt über den Kontext möglicher Fehler im Systemmodell des zu entwickelnden Produktes. Abbildung 2 zeigt für das Anwendungsbeispiel „Mobile Robot“ z.B. die Suche nach möglichen Fehlern einer Funktion „Detect Obstacle“ im Kontext der Funktion inkl. deren Umsetzung sowie den Eigenschaften des Objektes (hier Größe und Geschwindigkeit). Abbildung 3 zeigt vereinfacht einen Ausschnitt aus dem Wissensgraphen.

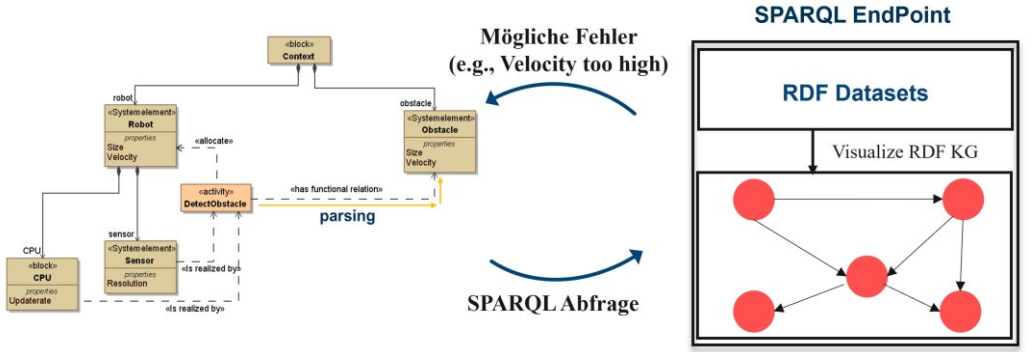

Abbildung 2: Suche im RDF-Wissensgraphen auf Basis der Analyse eines Systemmodells

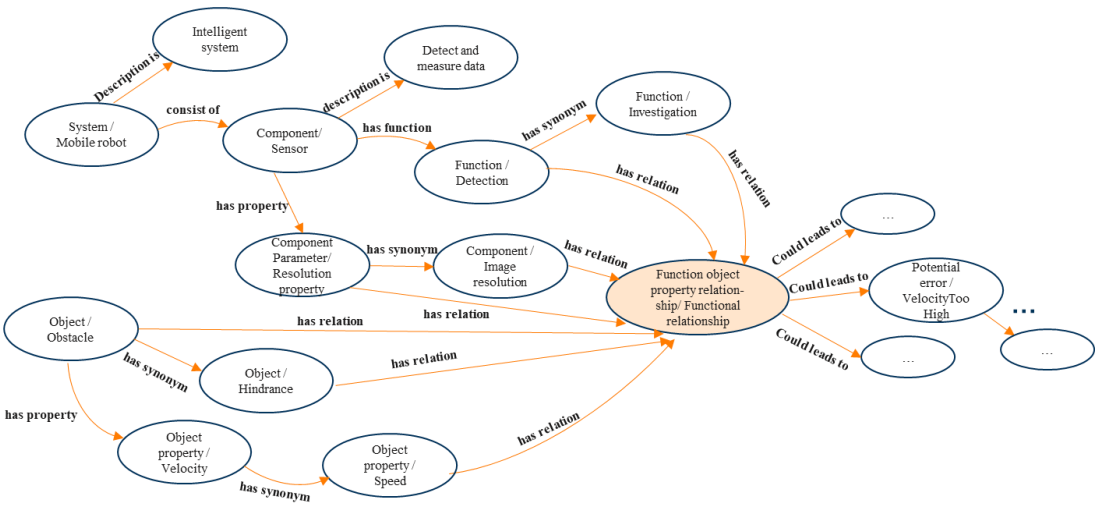

Abbildung 3: Auszug aus dem Wissensgraphen (stark vereinfacht)

Hierfür werden die Systemmodelle im Tool Cameo Systems Modeler beschrieben. Der Wissensgraph wird unter Verwendung von RDF-Datensätzen implementiert und in der Ontotext GraphDB-Plattform (Ontotext, 2024) für die Wissensorganisation und -abfrage verwaltet. Das implementierte Cameo-Plugin ermöglicht den bidirektionalen Informationsaustausch (d.h. die Abfrage von Informationen im Wissensgraphen, z.B. zu potenziellen Fehlern, auf der Basis der spezifizierten SysML-Modelle).

Da das Wissen, insbesondere in der kollaborativen Entwicklung, nicht ausschließlich bei einem Unternehmen vorhanden ist und für ein Gesamtprodukt die Systemmodelle der Subsysteme auf die Partner verteilt werden, stellt sich neben der Frage nach der Kopplung der Systemmodelle mit Wissensgraphen auch die Frage des

Modellaustausches. Die Wissensgraphen selbst können auch unternehmensübergreifend genutzt werden. Dieser Aspekt wird hier nicht diskutiert.

2.2 Modellbasierte Kollaboration mittels Dataspaces

Der Austausch von SysML v1-Modellen war bisher mit zahlreichen Einschränkungen verbunden. Die nun freigegebene Version SysML v2 bietet u.a. mit dem neuen Metamodell und der kombinierten textuellen und grafischen Notation hervorragende Voraussetzungen für den Austausch, der auch auf Paketebene der SysML-Modelle erfolgen kann (Bajaj *et al.*, 2022; Li *et al.*, 2024).

Während der Entwicklung werden verschiedene Versionsstände der Modelle mit unterschiedlichen Modellinhalten (z.B. zu Anforderungen oder zu entwickelnden Subsystemen) und unterschiedlichen Reifegraden der Anforderungen und Spezifikationen zwischen den Partnern ausgetauscht (prostep ivip Verein, 2023). Für den Austausch ist es daher wichtig, dass neue Stände der Modelle (oder Teilmodelle) zielgerichtet in die Systemmodelle des Gesamtsystems oder der Teilsysteme integriert werden können.

Für den Austausch sensibler Systemmodelle ist es wichtig, dass die Partner klar entscheiden und nachvollziehen können, wer welche Daten zu welchem Zweck erhält. Für derartige Zielstellungen sind in den letzten Jahren Dataspaces (Curry *et al.*, 2022) entstanden, die eine Infrastruktur für die verteilte Kollaboration auf Basis heterogener Daten bieten (Farrell *et al.*, 2023).

2.2.1 Einsatz von SysML v2 für die modellbasierte Kollaboration

Im Rahmen einer modellbasierten Kollaboration ist der bidirektionale Austausch von ausgewählten Teilmodellen notwendig. Die Verwendung von Modellen in SysML v2 bietet dabei erhebliche Vorteile (Bajaj *et al.*, 2022). Einer der Vorteile ist die konsistente Übersetzung zwischen grafischer und textueller Notation zur Modellbeschreibung, sodass es möglich ist, grafische Notationen aus textuellen Notationen abzuleiten. Für den Datenaustausch bei der Kollaboration bedeutet dies, dass nur die textuelle Modellbeschreibung übertragen werden kann und die grafische Notation im Ziel-SysML v2-Tool erzeugt wird. Die reine Übertragung der textuellen Notation reduziert auch die Dateigröße. (die Autoren sind sich bewusst, dass die Implementierung von SysML v2 derzeit noch in der Entwicklung ist und dass es zu softwarespezifischen Änderungen kommen kann. Die Aussagen in diesem Papier beziehen sich auf die Möglichkeiten des Standards und die bereits verfügbare Pilotimplementierung von SysML v2 "SysML v2 Pilot Implementation" (Seidewitz, 2024) mit Eclipse Plugin Version 0.37.0). Neben der Modellierungssprache umfasst die SysML v2 auch eine API, die im Rahmen dieses Beitrags nicht adressiert werden kann.

Der Austausch von Teilmodellen setzt voraus, dass die entsprechenden Teilmodelle aus dem Gesamtmodell extrahiert werden können. In der SysML v2-Spezifikation, wie auch in der SysML v1, wird dies durch die Paketstruktur unterstützt. Wichtig ist dabei, dass die einzelnen Teilmodelle jeweils innerhalb eines Paketes beschrieben werden. Innerhalb des Pakets können weitere Pakete importiert werden. Entscheidend ist, dass es keine Referenzen auf SysML-Modellelemente außerhalb des zu extrahierenden Pakets gibt, wenn diese nicht mit ausgetauscht werden. Diese Referenzen würden beim Import beim Partner unauflösbar werden. Anleitungen zur konkreten Modellierung als Basis für die Extraktion von Paketen müssen über Modellierungsrichtlinien erfolgen (Li *et al.*, 2024).

2.2.2 Austausch von Systemmodellen mittels Dataspaces

Bevor Systemmodelle über einen Dataspace zwischen Entwicklungspartnern ausgetauscht werden, müssen die Partner Datenverwendungsrichtlinien definieren. Darüber hinaus sollten Zugangs- und Nutzungsrechte für diese Daten definiert werden (im Kontext dieses Beitrags für die SysML v2-Modelle (Gesamtsystem oder die Teilsysteme)), um die Datenhoheit zu gewährleisten. Um die Zusammenarbeit mit Systemmodellen zu erleichtern, müssen die Modelldaten systematisch gespeichert und mit detaillierten semantischen Informationen, einschließlich Datentypen, Versionsinformationen usw., angereichert werden. Diese semantischen Informationen werden durch entsprechende Metadaten kategorisiert und organisiert, die Titel, Beschreibungen, Lizenzinformationen, Vertragslisten usw. enthalten. Im Dataspace werden Verknüpfungen zwischen Modelldaten und Metadaten hergestellt. (Li *et al.*, 2024)

Im modellbasierten Kollaborations-Prozess kann es erforderlich sein, dass ein Partner die Anforderungen auf der Grundlage der tatsächlichen Eigenschaften der Subsysteme aktualisiert, die dieser vom anderen Partner erhalten hat. Der Dataspace bietet Lieferanten die Möglichkeit, auf die aktuelle Anforderungsdokumentation oder auf frühere Versionen zuzugreifen. Für die von den Lieferanten zur Verfügung gestellten Daten gilt das Gleiche. Dies bedeutet, dass eine iterative Entwicklung des Produktes über den Dataspace während der Zusammenarbeit möglich ist. Darüber hinaus können verschiedene Versionen desselben Modells über den Dataspace aufgerufen und genutzt werden. Der Zugang zu mehreren Versionen der Modelle ist für das Konfigurationsmanagement (ISO/IEC/IEEE, 2018) wichtig.

Im kollaborativen Systems Engineering werden SysML-Modelle oder Teilmodelle für den Austausch definiert. Diese werden als einzelne Elemente in Form von Artefakten im Dataspace bereitgestellt und ausgetauscht (Li *et al.*, 2024). Die SysML-Modelle

werden dafür im Datenmanagement-System abgelegt und der Link über den Dataspace bereitgestellt (siehe Abbildung 4).

Ein weiterer Vorteil des Austauschs auf Basis eines textuellen SysML v2-Modells besteht darin, dass verschiedene ausgetauschte Modelldaten (Teilmodelle in der textuellen SysML-Sprache) durch Import in dieselbe SysML-Datei integriert werden können. Darüber hinaus können diese Modelldaten erweitert werden, um neue Funktionen oder Definitionen zur Anpassung an neue Anforderungen einzubeziehen.

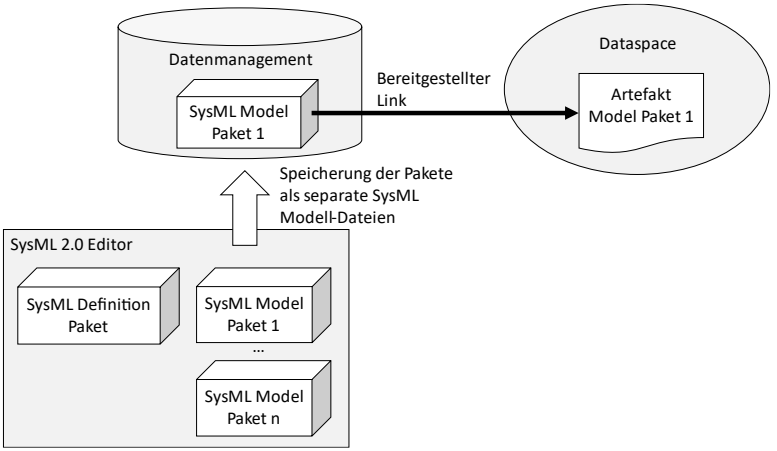

Abbildung 4: Bereitstellung von SysML-Dateien im Dataspace

2.2.3 Umsetzung und Beispiel

Im Rahmen des Beitrags wird die SysML-Dataspace-Integration auf Basis von SysML 2.0 "SysML v2 Pilot Implementation" (Seidewitz, 2024) mit dem Eclipse Plugin Version 0.37.0 und dem OpenSource Dataspace IDS-Test-Bed (Steinbuss, 2022) realisiert. Als Anwendungsbeispiel dient ein präzisionstechnisches Messsystems zur Kraftmessung (siehe Details zum System (Li *et al.*, 2023)). Beim Austausch wird von einer OEM-Zuliefer-Situation ausgegangen. Der OEM entwickelt das mechatronische Gesamtsystem (Kraftmesssystem), der Zulieferer liefert ein Subsystem auf Basis einer Wägezelle. Für den Austausch erzeugt der OEM ein Systemmodell, welches die allgemeinen Definitionen (siehe Abbildung 5) und die Soll-Eigenschaften der Wägezelle (siehe Abbildung 6) spezifiziert.

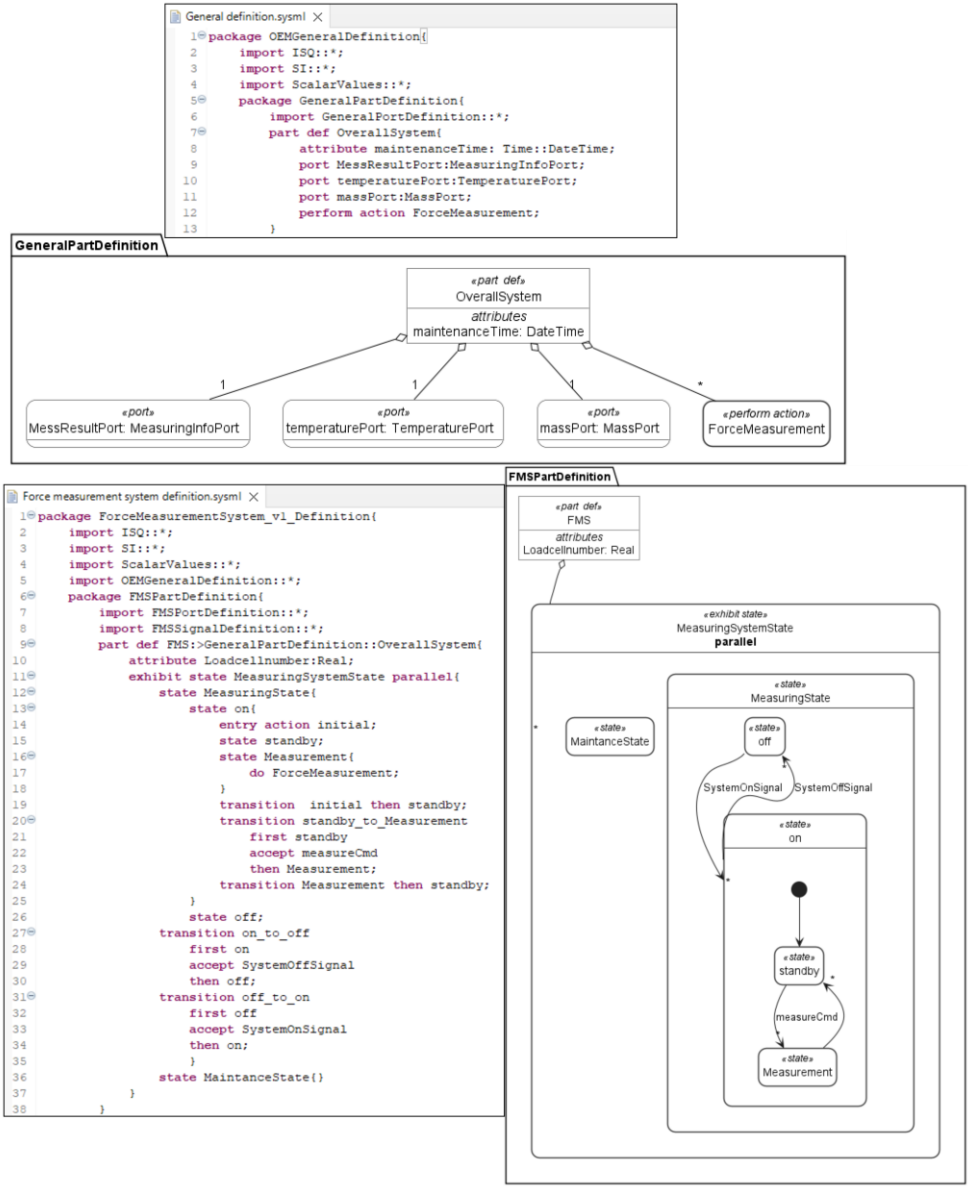

Abbildung 5: Auszug aus dem SysML v2-Modell vom Kraftmesssystem (Li et al., 2024)

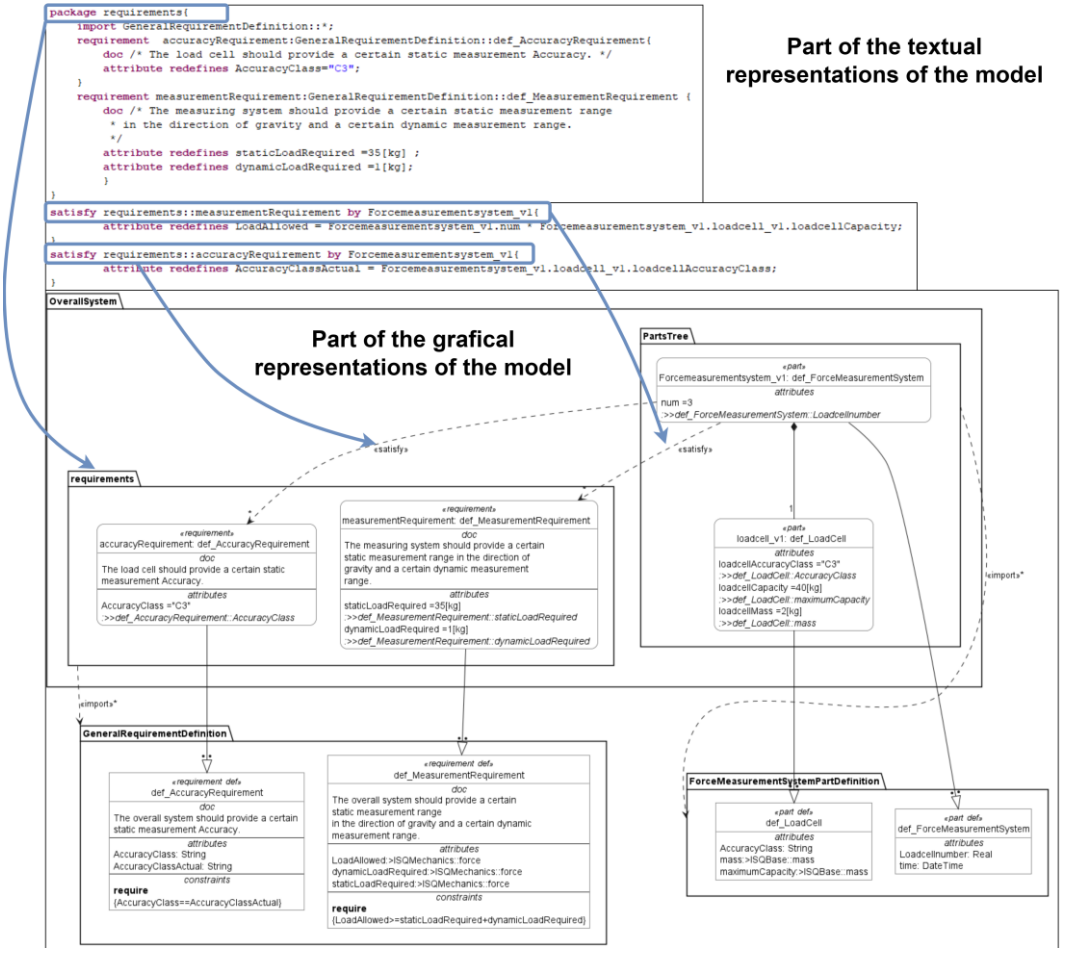

Abbildung 6: Auszug aus den Anforderungen (Soll-Eigenschaften) als SysML v2 Modell

Dieses Systemmodell wird vom OEM über den Datenraum dem Zulieferer zur Verfügung gestellt, damit dieser ein Systemmodell für die Wägezellenlösung entwickeln kann. Das Modell des Subsystems wird im Datenraum bereitgestellt und kann in das Gesamtsystemmodell integriert werden. Um den Reifegrad der Anforderungen und Spezifikationen weiter zu erhöhen, kann dieser Austausch iterativ erfolgen. Das hier verwendete SysML-Modell kann via Github abgerufen werden (GitHub).

3 Zusammenfassung und Ausblick

Modell-Based Systems Engineering wird für die Entwicklung komplexer mechatronischer Produkte immer wichtiger. Dabei müssen die Systemmodelle bzw. Anteile davon (z.B. für die Subsysteme) zwischen Entwicklungspartner ausgetauscht

werden. Damit während der Entwicklung, auch unternehmensübergreifend, auf Erfahrungswissen zurückgegriffen werden kann, können die Systemmodelle mit Wissensrepräsentationen, wie Wissensgraphen, gekoppelt werden. Der vorliegende Beitrag erläutert dafür zwei Konzepte, erstens zur Kopplung von SysML-Modellen mit Wissensgraphen und zweitens zum Datenaustausch von SysML v2-Modellen unter Nutzung von Dataspaces. Auf Basis von prototypischen Implementierungen konnten die Konzepte unter Einsatz von Beispielmodellen evaluiert werden. Die Konzepte sind sehr vielversprechend, es gibt jedoch noch viele offene Fragen, u.a. hinsichtlich Modellierungsrichtlinien, Aufgabenverteilung und Verantwortung bei der Zusammenarbeit, etc.

Weiterführende Forschungsarbeiten beschäftigen sich u.a. mit der Nutzung der API der SysML v2 in Kombination mit den Metadaten, wie den eindeutigen IDs, mit konkreten Modellierungsrichtlinien, mit Hilfsmittel für die Anreicherung der Wissensgraphen sowie mit Unschärfen bei den Abfragen im Wissensgraphen.

Literaturverzeichnis

Bajaj, M., Friedenthal, S. & Seidewitz, E. (2022): Systems Modeling Language (SysML v2) Support for Digital Engineering, INSIGHT, 25. Jg., Nr. 1, S. 19–24. 10.1002/inst.12367.

Curry, E., Scerri, S. & Tuikka, T. (2022): Data Spaces: Design, Deployment, and Future Directions, in Curry, E., Scerri, S. und Tuikka, T. (Hg.), Data Spaces, Springer International Publishing, Cham, S. 1–17. 10.1007/978-3-030-98636-0_1.

Faheem, F., Li, Z. & Husung, S. (2023): Analysis of potential errors in technical products by combining knowledge graphs with MBSE approach, in 60th Ilmenau Scientific Colloquium. 10.22032/dbt.58898.

Farrell, E., et al. (2023): European data spaces: Scientific insights into data sharing and utilisation at scale, Publications Office of the European Union, Luxembourg.

Forte, S., Göbel, J. C. & Dickopf, T. (2021): System of Systems Lifecycle Engineering Appraoch Integration Smart Product and Service Ecosystems, 23rd International Conference on Engineering Design (ICED21). 10.1017/pds.2021.552.

Friedenthal, S. (2014): A Practical Guide to SysML: The Systems Modeling Language, 3. Aufl., Elsevier Science.

GitHub: Model Kraftmesssystem, verfügbar unter https://github.com/ziruili-tu-ilmenau/CMBSE.

Hick, H., Bajzek, M. & Faustmann, C. (2019): Definition of a system model for model-based development, SN Applied Sciences, 1. Jg., Nr. 9. 10.1007/s42452-019-1069-0.

Husung, S., Weber, C. & Mahboob, A. (2022a): Integrating Model-Based Design of Mechatronic Systems with Domain-Specific Design Approaches, 17th International Design Conference. 10.1017/pds.2022.192.

Husung, S., Weber, C. & Mahboob, A. (2022b): Model-Based Systems Engineering: A New Way for Function-Driven Product Development, in Design Methodology for Future Products, Springer International Publishing. 10.1007/978-3-030-78368-6_12.

ISO/IEC/IEEE (2018): Part 1:Guidelines for life cycle management No. ISO/IEC/IEEE 24748-1: Systems and software engineering - Life cycle management.

Klyne, G., Carroll, J. J. & McBride, B. (2014): RDF 1.1 Concepts and Abstract Syntax, verfügbar unter https://www.w3.org/TR/rdf11-concepts/.

Li, Z., Faheem, F. & Husung, S. (2023): Systematic use of model-based solution patterns using the example of a load cell, in 60th Ilmenau Scientific Colloquium. 10.22032/dbt.58904.

Li, Z., Faheem, F. & Husung, S. (2024): Collaborative Model-Based Systems Engineering Using Dataspaces and SysML v2, Systems, 12. Jg., Nr. 1. 10.3390/systems12010018.

Mahboob, A. (2021): Modelling and use of SysML behaviour models for achieving dynamic use cases of technical products in different VR-systems, Dissertation, Technische Universität Ilmenau, Ilmenau, 2021.

Mahboob, A., et al. (2023): C.Pulse: an industrial demonstrator for a digital twin powered by MBSE for achieving digital continuity during the complete development process, in 60th Ilmenau Scientific Colloquium. 10.22032/dbt.58906.

Mandel, C., et al. (2021): A Model-Based Systems Engineering Approach to Support Continuous Validation in PGE - Product Generation Engineering, in IEEE ISSE International Symposium on Systems Engineering 2021. 10.1109/ISSE51541.2021.9582475.

Mandel, C., et al. (2023): Improving system of objectives maturity through systematic reuse of knowledge using ontology-based knowledge representations, in 60th Ilmenau Scientific Colloquium. 10.22032/DBT.58908.

Moeser, G., et al. (2015): Fortschrittsbericht zur modellbasierten Unterstützung der Konstrukteurstätigkeit durch FAS4M, in Tag des Systems Engineering, Carl Hanser Verlag, München, S. 69–78. 10.3139/9783446447288.008.

Ontotext (2024): GraphDB, verfügbar unter https://www.ontotext.com/products/graphdb/.

prostep ivip Verein (2023): Recommendation SysML WF/IF.

Prud'hommeaux, E. (2013): SPARQL 1.1 Query Language, verfügbar unter https://www.w3.org/TR/sparql11-query/.

Ropohl, G. (1975): Systemtechnik: Grundlagen und Anwendung, 1. Aufl., Hanser, München.

Schumacher, T., et al. (2022): Fostering Model Consistency in Interdisciplinary Engineering by Linking SysML and CAD-Models, in 2022 IEEE International Symposium on Systems Engineering (ISSE), IEEE. 10.1109/ISSE54508.2022.10005327.

Seidewitz, E. (2024): Systems-Modeling/SysML-v2-Pilot-Implementation: Proof-of-concept pilot implementation of the SysML v2 textual notation and visualization, verfügbar unter https://github.com/Systems-Modeling/SysML-v2-Pilot-Implementation.

Steinbuss, S. (2022): The IDS Reference Testbed. Key in Development and Certification, Zenodo.

Kontakt

Prof. Dr.-Ing. Stephan Husung, M. Sc. Faizan Faheem & M. Sc. Zirui Li
Fachgebiet Produkt- und Systementwicklung
Fakultät für Maschinenbau
Technische Universität Ilmenau
www.tu-ilmenau.de/produkt-und-systementwicklung