

Ganzheitliche (System-)Modellierung mit Hilfe des Artefaktmodells

Markus Brandstätter^{1,2}, Veronica Haber³, Tamara Hofmann⁴, Kim Steinkirchner⁵, Christian Bühler⁶

Abstract: Technische Systeme, wie beispielsweise Fahrzeuge, Mobiltelefone oder Satelliten, bestehen aus einer Vielzahl verschiedener Komponenten. Diese werden i.A. von Ingenieurteams aus unterschiedlichen Domänen geplant, entwickelt und gebaut. Neben den technischen Domänen sind auch nicht-technische Abteilungen (z.B. Unternehmensleitung, Ressourcenplanung, etc.) maßgeblich an der Entwicklung dieser Systeme beteiligt, denn sie koordinieren, steuern und unterstützen die einzelnen (Entwicklungs-)Projekte. Nach dem Systems Engineering (SE) Vorgehensmodell sollten alle an der Entwicklung beteiligten Domänen eine gemeinsame Sicht und Verständnis auf das zu entwickelnde System bekommen und, im Falle des Model-based Systems Engineerings (MBSE) Paradigmas, ein gemeinsames zentrales Systemmodell während der gesamten Systementwicklung nutzen. Aktuell verwenden jedoch meist nur die technischen Domänen eine gemeinsame Modellierungssprache, wie beispielsweise die SysML. Das Geschäftsprozessmanagement hingegen nutzt meist die BPMN zur Modellierung der Geschäftsprozesse. In dieser Veröffentlichung wird ein neuer Modellierungsansatz vorgestellt, welcher eine durchgängige Modellierung, angefangen bei den Geschäftsprozessen, bis hin zu den technischen Disziplinen ermöglicht. Dadurch kann ein gemeinsames Verständnis zwischen den Domänen geschaffen werden.

Keywords: Systems Engineering, Model-based Systems Engineering, Artefaktmodell, Projektmanagement, Interdisziplinäre Modellierung

1 Einleitung

An der Vermarktung, der Entwicklung, dem Vertrieb und der Wartung eines modernen technischen Systems, wie z.B. einem autonom fahrenden Fahrzeug, sind viele unterschiedliche Disziplinen und Fachabteilungen beteiligt. Diese reichen vom Management des Unternehmens, das Geschäftsprozesse erstellt, mit deren Hilfe die unterschiedlichen Fachabteilungen des Unternehmens orchestriert werden, über die Entwicklungsabteilung, die die Vorgaben des Managements und vor allem des Kunden in ein neues System/Produkt überführt, bis hin zur Vertriebs- und der After-Sales-Abteilung. Dabei setzt, im Allgemeinen, jede Fachabteilung und jede Disziplin spezifische Werkzeuge, (Modellierungs-)Sprachen, Methoden und (Entwicklungs-)Prozesse ein. Für die Entwicklung und die anschließende Modellierung und Beschreibung von Geschäftsprozessen ist beispielsweise die „Business Process Model and Notation“ (BPMN) [BPMN1] in Unternehmen weit verbreitet. In den technischen Disziplinen, wie z.B. der Informatik, wird hingegen die „Unified Modeling

¹ FOM Hochschule, Arnulfstraße 30, 80335 München, markus.brandstaetter@fom-net.de

² PROSTEP AG, Systems Engineering, Taunusstraße 42, 80807 München, markus.brandstaetter@prostep.com

³ PROSTEP AG, Systems Engineering, Taunusstraße 42, 80807 München, veronica.haber@prostep.com

⁴ PROSTEP AG, Systems Engineering, Taunusstraße 42, 80807 München, tamara.hofmann@prostep.com

⁵ PROSTEP AG, Systems Engineering, Taunusstraße 42, 80807 München, kim.steinkirchner@prostep.com

⁶ PROSTEP AG, Systems Engineering, Taunusstraße 42, 80807 München, christian.buehler@prostep.com

Language“ (UML) [UML1] zur Beschreibung der Software eingesetzt. Im Systems Engineering dagegen wird die „Systems Modeling Language“ (SysML) [SYSML1] verwendet, um die einzelnen Systembestandteile und deren Verbindungen untereinander zu beschreiben.

Alle Sprachen und Methoden haben gemeinsam, dass sie die spezifischen Artefakte einer Disziplin gut abbilden können – im Falle der BPMN beispielsweise Prozesse und deren prozessualen Zusammenhänge – Artefakte aus einer anderen Disziplin hingegen sehr schlecht oder gar nicht. In BPMN können z.B. keine Klassen modelliert werden, die für die objektorientierte Softwareentwicklung unerlässlich sind. Für eine effektive Zusammenarbeit über Disziplinengrenzen hinweg, ist jedoch sowohl ein gemeinsames Verständnis als auch eine gemeinsame (Modellierungs-)Sprache notwendig, mit deren Hilfe die „Semantische Lücke“ [SEM01] zwischen den Domänen geschlossen werden kann. Genau an dieser Stelle setzt das Artefaktmodell an.

1.1 Artefakte zur Überwindung der „Semantischen Lücke“

Zur Überbrückung der „Semantischen Lücke“ und zur Unterstützung einer effizienten Zusammenarbeit von Ingenieuren aus unterschiedlichen Disziplinen, hat die INCOSE [INCOS1] ein spezielles Vorgehensmodell vorgeschlagen: das „Model-based Systems Engineering“ (MBSE), in dem sämtliche Phasen und Artefakte der Systementwicklung enthalten sind. „*Model-based systems engineering (MBSE) is the formalized application of modeling to support system requirements, design, analysis, verification and validation activities beginning in the conceptual design phase and continuing throughout development and later life cycle phases*“ (INCOSE-TP-2004-004-02, Version 2.03, September 2007).

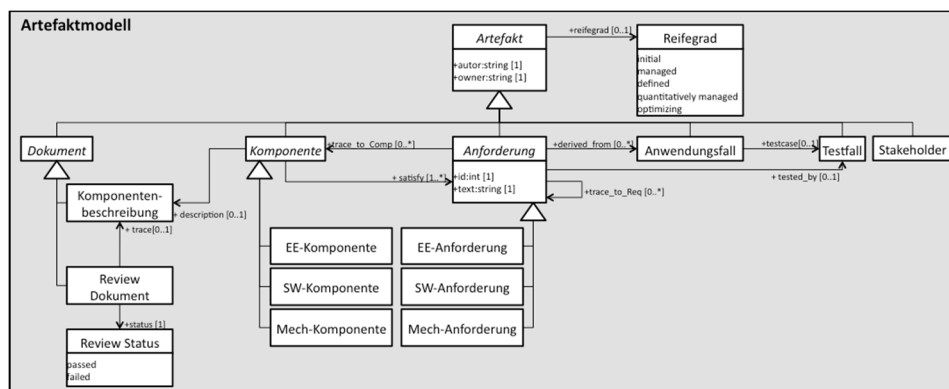


Abb. 1: Auszug aus dem Artefaktmodell aus [ART01]

Nach der obigen Definition werden sämtliche technischen Disziplinen explizit berücksichtigt, die das technische System gemeinsam erstellen, nicht jedoch beispielsweise die Geschäftsebene eines Unternehmens, die jedoch einen maßgeblichen Anteil an der Systementwicklung in einem Unternehmen haben. Dies wird in der INCOSE Vision 2025 [INCOS2] aufgegriffen und weitergeführt: „*Model-based approaches will extend beyond product modeling to enterprise-level modeling and analysis*“ ([INCOS2], S.38). Diese Erweiterung entsteht aus der Motivation heraus, Entscheidungsträgern mit der Formalisierung von

Zusammenhängen eine Hilfe für kosteneffiziente, sichere und nachhaltige Entscheidungen an die Hand zu geben⁷.

Um dem ganzheitlichen Ansatz bei der Systementwicklung Rechnung zu tragen, wurde in [ART01] das Artefaktmodell vorgestellt, mit dessen Hilfe sich die „Semantische Lücke“ zwischen den technischen Disziplinen schließen lässt. Dabei hat sich jedoch gezeigt, dass das Artefaktmodell nicht nur für die Beschreibung der Technischen Artefakte eines Unternehmens geeignet ist, sondern generell eingesetzt werden kann, z.B. zur Modellierung der Artefakte eines Geschäftsprozesses. Wie in Abb. 1 dargestellt, enthält das Artefaktmodell sowohl die wesentlichen Artefakte (Komponente, Anforderung, Anwendungsfall etc.), als auch die Zusammenhänge zwischen den unterschiedlichen Artefakten. So ist beispielsweise der Zusammenhang zwischen einer Komponente und einer Anforderung explizit modelliert („trace_to_Comp“). Dieser Zusammenhang ist sowohl für die technische Modellierung als auch für das Geschäftsprozessmanagement wichtig.

Bereits das Artefaktmodell aus [ART01] wurde so ausgelegt, dass sowohl technische als auch nichttechnische Zusammenhänge zwischen Artefakten explizit dargestellt werden konnten. Z.B. kann die oben beschriebene „trace_to_Comp“ Verbindung zwischen einer Komponente und einer Anforderung sowohl technisch als auch für die Beschreibung eines Geschäftsprozesses gleichermaßen genutzt werden. Technisch stellt die Relation explizit dar, dass es eine Verbindung von einer Komponente zu einer Anforderung geben sollte. Ansonsten ist zu hinterfragen, ob die Komponente im System wichtig ist. Aus Business- bzw. Geschäftsprozesssicht hingegen bedeutet die gleiche Verbindung, dass eine (Kunden-)Anforderung von einer Komponente erfüllt werden soll. So kann z.B. der Verkäufer des Systems dem Käufer nachweisen, dass seine Anforderung im System durch folgende Komponente realisiert wurde. Dieser „integrierte Ansatz“ hat jedoch seine Schwächen. Die Verwendung von einem Artefakt in mehr als einer Domäne kann z.B. zu einem Verlust an Übersichtlichkeit führen, da alle Attribute der unterschiedlichen Domänen in dem Artefakt enthalten sind. Bei zunehmender Anzahl an Attributen wird auch die eindeutige Zuordenbarkeit dieser zu den einzelnen Domänen erschwert.

1.2 Erweiterung des Artefaktmodells auf weitere Domänen

Um dem oben beschriebenen Dilemma entgegenzuwirken wurden zwei unterschiedliche Modellierungsansätze verfolgt. Zum einen wurde mit Ansichten („Views“) auf ein Artefakt gearbeitet, bei der jede Domäne ihre spezifischen Attribute des Artefakts explizit dargestellt bekommt, die anderen Attribute jedoch nicht (Abb. 2 unten links). Dieser Ansatz wurde nicht weiterverfolgt, weil aktuell das Sichtenkonzept in den meisten SysML-Werkzeugen noch nicht vollständig implementiert ist und es so zu „unschönen“ Seiteneffekten kommen kann, wie z.B. dem „Überlagern“ von Stereotypen beim Ableiten einer Klasse. Aus diesem Grund wurden zwei Stereotypen eingeführt, mit deren Hilfe die Artefakte unterschieden werden können. Der Stereotype „BusinessArtefakt“ wird zur Kennzeichnung

⁷ „Systems engineering takes its place with other systems-related, integrative disciplines such as economics, human ecology, geography, and economic anthropology to structure more objective cost, benefit and risk assessments of alternative policy executions. The addition of a formal systems approach helps decision-makers to select cost effective, safe, and sustainable policies that are more broadly embraced by the stakeholder community.“ ([INCOS2], S. 26)

der hauptsächlich für das Business notwendigen Artefakte, und der Stereotype „Technisches Artefakt“ zur Beschreibung von hauptsächlich für die Technik relevanten Artefakten benutzt. Die Unterscheidung der beiden Artefakttypen ist wichtig, da sich i.A. die Attribute der Technischen Artefakte stark von den Attributen der Business-Artefakte unterscheiden. Technische Artefakte besitzen z.B. technische Attribute zur Beschreibung der technischen Eigenschaften und Beziehungen zu anderen Technischen Artefakten. Beispielsweise hat die technische Komponente Motor die technischen Attribute Spannung, Strom, Leistung etc., wohingegen die Business Komponente die Business Attribute Preis, Hersteller, Bestellnummer etc. hat. Diese Erweiterung des allgemeinen Artefaktmodells ist in Abb. 2 exemplarisch dargestellt.

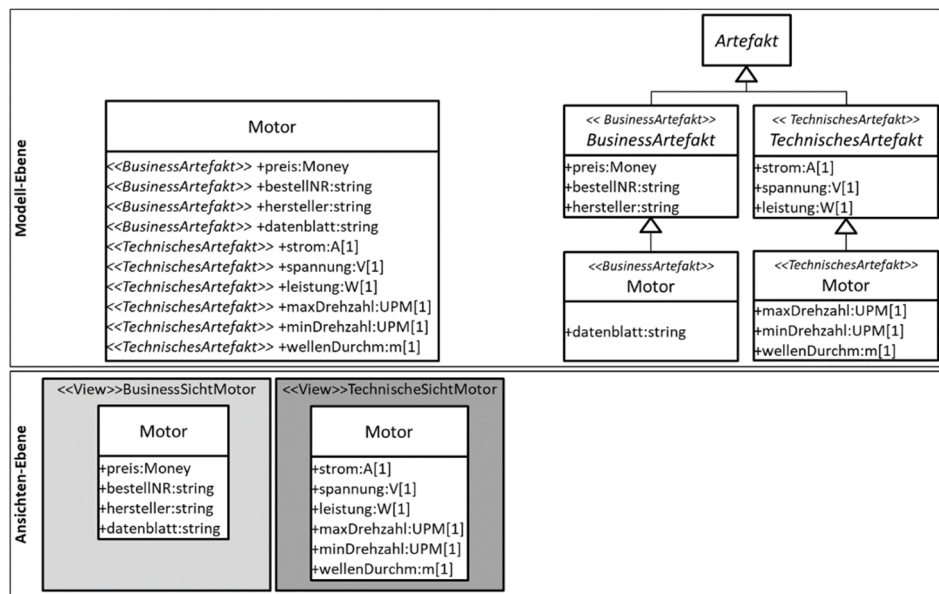


Abb. 2: Erweiterung des Artefaktmodells um Business- und Technische Artefakte

In der Masterarbeit [ART04], sowie in der Veröffentlichung [ART03] wurde das allgemeine Artefaktmodell aus [ART01] aufgegriffen und um die spezifischen Artefakte aus der Geschäftsprozesswelt angereichert (z.B. Markt, Gut, Nachfrage, Kommunikationsprozesse etc.). Außerdem wurde in [ART05] der Zusammenhang zwischen der technischen Modellierung eines Systems und der Business-Ebene anhand eines durchgängigen Beispiels gezeigt. Des Weiteren wurde in [ART02] initial gezeigt, wie der Artefaktgedanke für die durchgängige Beschreibung eines technischen Systems, angefangen bei der Stakeholder Analyse bis hin zu Durchführung der numerischen Simulation (Schwerpunkt der Veröffentlichung), verwendet werden kann und so die „Semantische Lücke“ zwischen den Domänen überwunden werden kann.

1.3 Vorgehen bei der Modellerstellung

Der Erstellungsprozess des Artefaktmodells ist nicht an ein spezielles Vorgehensmodell, wie z.B. das V-Modell [VMO01] oder das Spiralmodell [SPI01], gebunden. Vielmehr

kann es flexibel eingesetzt werden, je nach den spezifischen Belangen des Unternehmens. Von den Autoren wird jedoch ein objektorientiertes Vorgehen bzw. MBSE als Vorgehensmodell zur Erstellung des System- und Artefaktmodells vorgeschlagen. Dabei sollte zunächst mit der Stakeholder Analyse begonnen werden, also der Identifikation aller mit dem System/Projekt/Produkt in Verbindung stehenden Akteuren. Im Anschluss daran werden sämtliche Anwendungsfälle und Anforderungen der Stakeholder an das System/Projekt/Produkt modelliert. Dabei werden im Allgemeinen bereits die ersten Artefakte identifiziert. Diese sollten dann sofort in das Artefaktmodell aufgenommen werden. Nach diesem Schritt folgt entweder die Erstellung des interdisziplinären System- bzw. des Artefaktmodells, oder die Erstellung der Aktivitäten, die zu jedem Anwendungsfall gehören und das vom Stakeholder gewünschte Verhalten aus seiner Sicht beschreiben. Neben der Erstellung der Aktivitäten, sowie während der Erstellung des Systemmodells, wird parallel das Artefaktmodell aufgebaut oder ein vorhandenes generisches Artefaktmodell um die spezifischen Beläge des Systems erweitert. In [ART04] und [ART05] wurde das objektorientierte Vorgehen bei der Modellerstellung anhand einer Firma demonstriert, die Fahrzeuge herstellt, exemplarisch gezeigt. In [MBSE3] wurde ein Vorgehen zur Modellerstellung nach dem MBSE-Vorgehensmodell auf ein Fahrzeugmodell angewendet.

2 Anwendung des Artefaktmodells

Anhand eines stark vereinfachten Beispielsystems – eines Fensterhebers – wird in diesem Kapitel die domänenübergreifende Modellierung mit Hilfe des Artefaktmodells gezeigt. Begonnen wird die Modellierung des Systems mit der Stakeholder Analyse, bei der alle Stakeholder identifiziert werden, die mit dem System Fensterheber in Beziehung stehen. Für diese Stakeholder werden im nächsten Entwicklungsschritt Anwendungsfälle identifiziert und aus diesen wiederum Anforderungen an das System Fensterheber abgeleitet. In Abb. 3 ist das Ergebnis des oben beschriebenen Prozesses dargestellt.

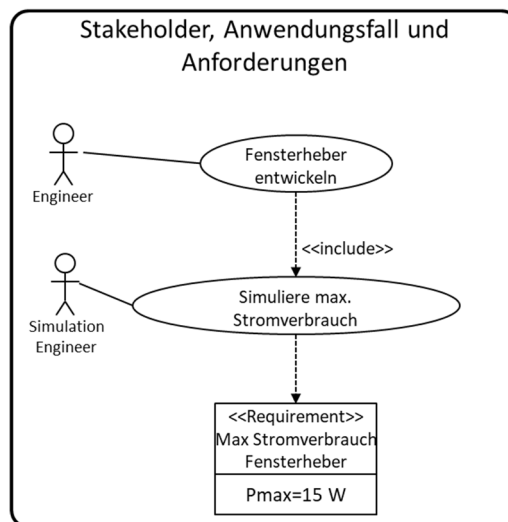


Abb. 3: Stakeholder, Anwendungsfall und Anforderungen an einen Fensterheber

Nachdem der Anwendungsfall und die Anforderungen an das zu entwickelnde System modelliert wurden, erfolgt nun die Beschreibung der Geschäftsprozessartefakte (Abb. 4), sowie der Modellierung des Einkaufsprozesses für ein Artefakt (Abb. 5).

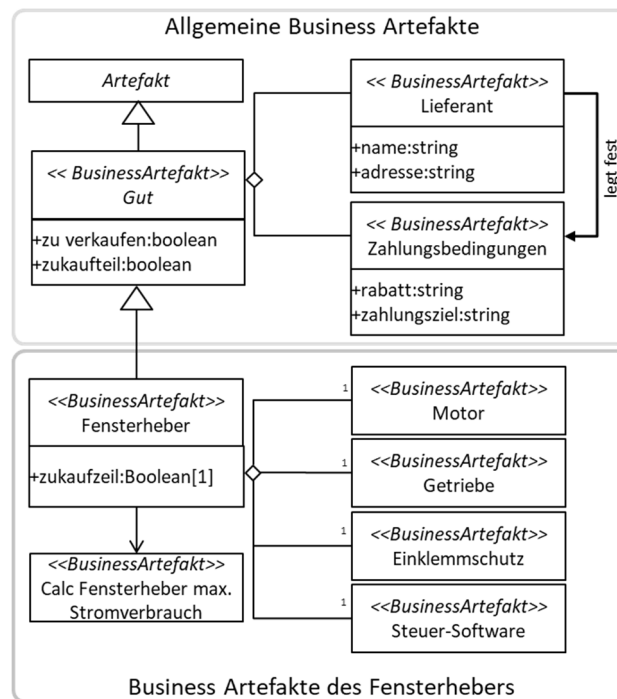


Abb. 4: Geschäftsprozessartefakte

Die Abb. 4 zeigt einen Auszug aus dem Artefaktmodell, in dem hauptsächlich Business-Artefakte modelliert sind, die für die Geschäftsprozessmodellierung notwendig sind, um ein Produkt/System beschreiben zu können (oben). Es zeigt links zunächst ein generalistisches Artefakt „Gut“, welches gerade im Sinne der Volkswirtschaft den „Platzhalter“ für das (ver-)kaufen von Waren einnehmen kann. Daher werden an dieses Artefakt die wichtigen Informationen im Sinne der Unternehmenssicht angehängt. Diese sind in diesem Fall die Zahlungsbedingungen und der Lieferant, der zur Identifikation die Attribute Namen und Adresse trägt. Die Zahlungsbedingungen werden dabei vom Lieferanten beeinflusst, da z.B. Rabatte abhängig vom gekauften Produkt und dem Lieferanten sein können. Dies ist explizit durch die Assoziation „legt fest“ zwischen Zahlungsbedingungen und Lieferant modelliert. Diese Eigenschaften und Beziehungen des generalistischen Business-Artefakts „Gut“ sind nun mit dem Business-Artefakt „Fensterheber“ per Ableitungsbeziehung verbunden, um diesem diese Informationen ebenfalls zur Verfügung stellen zu können. Im unteren Teil der Abbildung ist ein Auszug aus den Business-Artefakten beschrieben, die für die Beschreibung des Fensterhebers notwendig sind. Neben den Bestandteilen des Fensterhebers aus Business-Sicht, ist hier außerdem das Business-Artefakt „Calc Fensterheber max. Stromverbrauch“ dargestellt. Dadurch wird bereits auf der Geschäftsebene festgelegt, dass für die Absicherung des maximal zulässigen Stromverbrauchs des Fensterhebers ein Simulationsmodell erstellt werden muss und dass dieses ebenfalls einen

„Lieferanten“ hat (nicht im Bild dargestellt). So kann die Durchgängigkeit von der Business-Ebene bis hinunter zur Technischen Ebene innerhalb eines ganzheitlichen Modells gewährleistet werden (siehe hierzu auch die Abb. 7).

Neben dem Halten von businessrelevanten Attributen und für die Geschäftsebene notwendigen Informationen und Zusammenhängen, bietet diese Art der Businessartefaktmodellierung auch noch weitere Vorteile. So können auch die Geschäftsprozesse beschrieben werden, welche die modellierten Artefakte durchlaufen.

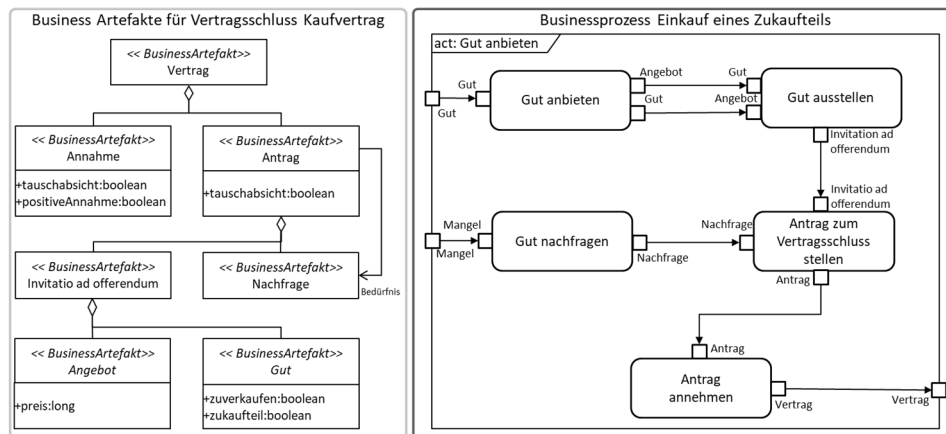


Abb. 5: Geschäftsprozess mit Artefakten zum Einkauf einer Komponente

Abb. 5 zeigt einen abstrakten und vereinfacht dargestellten Einkaufsprozess mit den zugehörigen Business-Artefakten, die im Prozess eingesetzt werden. Rechts im Bild ist der Prozess zu sehen, der mit den Artefakten auf der linken Bildseite korreliert. Bei Betrachtung des Prozesses „Einkauf eines Zukaufteils“ ist z.B. zu sehen, dass zunächst das „Gut“ in den Prozess einfließt und als Input für die Aktivität „Gut anbieten“ fungiert, aus welcher anschließend ein Angebot generiert wird, welches als Output aus der Aktivität herausgegeben wird. Welche einzelnen Schritte dabei in der Aktivität durchlaufen werden ist in der Beschreibung der Aktivität einsehbar (siehe dazu auch [ART04]). Neben der Erstellung eines Angebots wird auch das Gut in der Aktivität bearbeitet und als Output wieder ausgegeben. Dabei wird das Attribut „zu verkaufen“ in einem Schritt der Aktivität auf den Wert „true“ gesetzt und somit das Gut als verkäuflich gekennzeichnet. Um eine höhere Genauigkeit in der Prozessbeschreibung zu erreichen, müsste vor der Weiterverarbeitung des Artefakts „Gut“ in der Aktivität „Gut ausstellen“ der zugehörige Input Port der Aktivität mit einer Zugangsbeschränkung belegt werden. Dies lässt sich z.B. über die Verwendung von OCL-Constraints realisieren [OCL01], welche jedoch nicht Teil der Betrachtung in diesem Paper sind. Innerhalb der nachfolgenden Aktivitäten werden die Artefakte weiterhin modifiziert beziehungsweise miteinander verkettet. Diese Verkettungen führen dabei zu der auf der linken Seite dargestellten Zusammensetzung der Artefakte. Am Ende des Einkaufs-Prozesses ist somit ein (Kauf-)Vertrag entstanden, welcher aus dem Prozess ausgegeben wird und in einen anschließenden Geschäftsprozess einfließen kann.

Nachdem die Business-Ebene beschrieben wurde, müssen die Technischen Artefakte ebenfalls modelliert werden, sodass anschließend eine durchgehende Verbindung zwischen den Business- und den Technischen Artefakten des Systems „Fensterheber“ hergestellt werden kann. In Abb. 6 sind auf der linken Seite die zuvor identifizierten Business-Artefakte des Fensterhebers und auf der rechten Seite die Technischen Artefakte dargestellt, die bei der Modellierung des Fensterhebers erstellt wurden. Bei der Modellierung der Technischen Artefakte wird darauf geachtet, dass neben den Artefakten auch deren technische Beziehungen zueinander explizit modelliert werden. Dies geschieht auf einer abstrakten Ebene. So ist in Abb. 6 explizit dargestellt, dass das Artefakt „Fensterheber“ aus den Artefakten „Motor“ und „Getriebe“ besteht. Diese beiden Komponenten sind notwendig, um die gewünschte Struktur und das gewünschte Verhalten des Fensterhebers zu beschreiben und umzusetzen. Zur Umsetzung der Funktion des Fensterhebers sind außerdem eine „Stromversorgung“ und eine „ECU“ notwendig, auf der die „Steuer-Software“ ausgeführt wird. Außerdem ist im Technischen Artefaktmodell beschrieben, dass die Funktion „Einklemmschutz“ durch die „Steuer-Software“ realisiert wird.

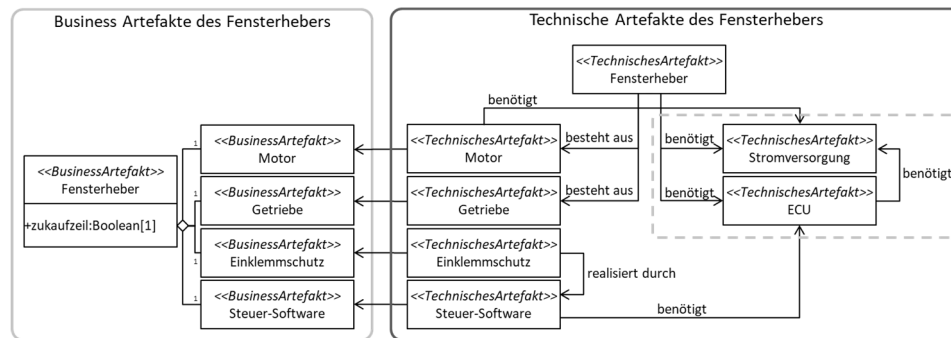


Abb. 6: Business- und Technische Artefakte des Fensterhebers

Bei der technischen Modellierung des Fensterhebers sind zwei neue (Technische) Artefakte hinzugekommen (gestrichelter Bereich: Stromversorgung und ECU). Diese beiden Artefakte waren auf der Business-Ebene für die Beschreibung des Fensterhebers aus Business-Sicht nicht notwendig. Im nächsten Schritt müssen die beiden neuen Technischen Artefakte noch im Business-Artefaktmodell nachgetragen werden (nicht dargestellt).

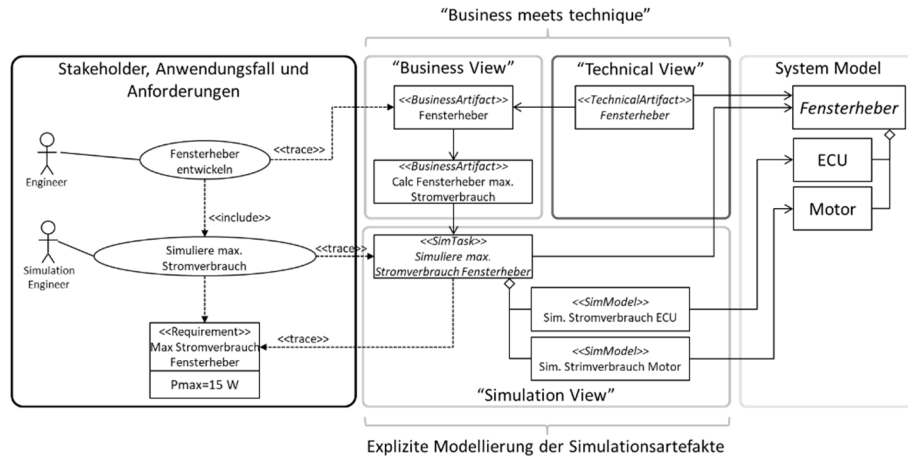


Abb. 7: Systemmodell eines Fensterhebers inkl. der Anbindung an die Simulation (nach [ART02])

Nachdem nun alle Teilaspekte des Systems „Fensterheber“ modelliert sind, zeigt die Abb. 7 das vereinfachte Systemmodell des Fensterhebers – angefangen bei den Anwendungsfälle (links), über die Business-, Technischen und Simulations-Artefakte bis hin zum Systemmodell (rechts). Dabei wurde das Artefaktmodell zur Modellierung der Komponenten, sowie für die Beschreibung der (numerischen) Simulation gleichermaßen verwendet.

Durch den Einsatz des Artefaktmodells und der disziplinenübergreifenden Systemmodellierungssprache SysML zur ganzheitlichen Modellierung des Systems „Fensterheber“, ist es möglich, die Grenzen zwischen den Disziplinen und somit die „Semantische Lücke“ zwischen ihnen zu überwinden. Außerdem wird durch den Einsatz des Artefaktmodells eine Nachverfolgbarkeit über die Disziplinengrenzen hinweg ermöglicht, sodass jederzeit, von allen an der Systemmodellierung beteiligten Ingenieuren, das System als Ganzes verstanden und Verbindungen nachvollzogen werden können.

3 Zusammenfassung und Ausblick

In der vorliegenden Veröffentlichung wurde exemplarisch anhand eines Fensterhebers gezeigt, wie mit Hilfe des Artefaktmodells und der Nutzung der Systemmodellierungssprache SysML eine gemeinsame Basis für die interdisziplinäre Modellierung eines Systems geschaffen werden kann. Angefangen beim Business bis hin zur technischen Modellierung und Absicherung/Simulation des Systems.

Aktuell laufen wissenschaftliche Arbeiten zum Thema „Anwendung und Nutzen des Artefaktmodells“ in der Medizintechnik, der Bekleidungsindustrie und in der Raumfahrt. Diese werden im Laufe des Jahres ausgewertet und die dort identifizierten Artefakte dem allgemeinen Artefaktmodell hinzugefügt.

4 Literaturverzeichnis

- [ART01] Dr. Brandstätter, M.; Roder, K.: Ein Artefaktmodell zur Verbesserung der Prozessmodellierung. Tag des Systems Engineering 2017, Carl Hanser Verlag GmbH & Co. KG, 2017.
- [ART02] Dr. Brandstätter, M.; Bühler, C.: Einsatz des Artefaktmodells zur Modellierung von CAE-Artefakten im MBSE-Prozess. Better Products with Model Based Systems Engineering (MBSE) and CAE, 2018.
- [ART03] Hofmann, T.; Haber, V.: The collaboration of business and technology: Seeing MBSE with fresh eyes. EMEASEC/TdSE, 2018.
- [ART04] Haber, V.: Die Möglichkeiten und Grenzen der SysML - Eine Betrachtung der Prozessmodellierung und Businessicht. Masterarbeit, Hochschule für angewandte Wissenschaften München, 2019.
- [ART05] Hofmann, T.; Die Grenzen und Möglichkeiten der SysML-Modellierung. Eine Betrachtung der Systemmodellierung und Techniksicht. Masterarbeit, Hochschule für angewandte Wissenschaften München, 2019.
- [BPMN1] „Documents Associated With Business Process Model And Notation™ (BPMN™) Version 2.0”, <http://www.omg.org/spec/BPMN/2.0/>, 2011.
- [INCOS1] „International Council on Systems Engineering – INCOSE”, <http://www.incose.org/>.
- [INCOS2] INCOSE (2014): Vision 2025 - Systems Engineering. A World in Motion. Hg. v. INCOSE. Online verfügbar unter <https://www.incose.org/docs/default-source/aboutse/se-vision-2025.pdf>, zuletzt geprüft am 01.08.2018. (S. 26)
- [MBSE1] „OMG MBSE wiki“, <http://www.omgwiki.org/MBSE/doku.php>.
- [MBSE2] „INCOSE 2007 – Systems Engineering Vision 2020”, INCOSE-TP-2004-004-02 September, 2007.
- [MBSE3] Brandstätter, Dr. M.; Eckl, C.: Einsatz von Model-based Systems Engineering in der Automobil Industrie. TdSE, 2015.
- [OCL01] „Object Constraint Language”, <https://www.omg.org/spec/OCL/About-OCL/>.
- [UML1] „Documents Associated With Unified Modeling Language™ (UML®) Version 2.5”, <http://www.omg.org/spec/UML/2.5/>, 2016.
- [VMO01] Dröschel, Wolfgang; Wiemers, Manuela: Das V-Modell 97: der Standard für die Entwicklung von IT-Systemen mit Anleitung für den Praxiseinsatz, Walter de Gruyter GmbH & Co KG, 2015.
- [SEM01] Chitra Dorai; Svetha Venkatesh: „Bridging the Semantic Gap with Computational Media Aesthetics”; IEEE MultiMedia. Bd. 10, Nr. 2, 2003, S. 15–17, doi:10.1109/MMUL.2003.1195157
- [SPI01] Boehm, Barry W.: A spiral model of software development and enhancement. IEEE Computer, Nr. 5, 1988, S. 61-72.
- [SYSML1] „Documents Associated With OMG Systems Modeling Language™ (SysML®), Version 1.5”, <http://www.omg.org/spec/SysML/1.5/>, 2019