



UNIVERSITÄT
DES
SAARLANDES

Einsatz von künstlicher Intelligenz in der Produktentwicklung auf Basis eines Engineering-Graph im Kontext von Model-based-Systems-Engineering

Gregor M. Schweitzer

Einsatz von künstlicher Intelligenz in der Produktentwicklung auf Basis eines Engineering-Graph im Kontext von Model-based-Systems-Engineering

Dissertation
zur Erlangung des Grades
des Doktors der Ingenieurwissenschaften
der Naturwissenschaftlich-Technischen Fakultät
der Universität des Saarlandes

von
Gregor M. Schweitzer

Saarbrücken
2023

Tag des Kolloquiums: 11.04.2024

Prodekan: Prof. Dr.-Ing. Michael Vielhaber

Berichterstatter: Prof. Dr.-Ing. Michael Vielhaber
Prof. Dr.-Ing. Roman Dumitrescu

Vorsitz: Prof. Dr.-Ing. Paul Motzki

Akad. Mitarbeiter: Dr. Niklas König

Kurzfassung

In der modernen Produktentwicklung steigen die Datenmengen exponentiell, sowohl unternehmensintern als auch -extern. Die manuelle Datenanalyse ist angesichts dieser Komplexität nicht mehr durchführbar. Daher werden KI- und Data-Science-Systeme als mögliche Lösungen vorgeschlagen, um Ingenieur_innen bei der Beherrschung der Komplexität zu unterstützen.

Das Hauptziel der Arbeit ist es, zu untersuchen, wie KI-Technologien in der Produktentwicklung eingesetzt werden können. Dies umfasst die Untersuchung der Voraussetzungen für den Einsatz von KI, die Prüfung dieser Voraussetzungen in der Produktentwicklung und die Erarbeitung eines Konzeptes für ein Systemmodell, durch das die Voraussetzungen erfüllt werden können. Darüber hinaus werden die Veränderungen in den Arbeitsweisen der Ingenieur_innen bei der Umsetzung des Systemmodells erforscht.

Der Engineering-Graph als flexibles Systemmodell, basierend auf Graph-Datenbanken, wird vorgeschlagen. Eine prototypische Umsetzung des Engineering-Graph wurde bei Fresenius Medical Care, einem führenden Anbieter von Lösungen für Menschen mit Nierenerkrankungen, durchgeführt. Damit konnte gezeigt werden, dass der Engineering-Graph alle aufgestellten Anforderungen erfüllt.

Abstract

In modern product development, data volumes are increasing exponentially, both internally and externally to the company. Manual data analysis is no longer feasible in the face of this complexity. Therefore, AI and data science systems are proposed as possible solutions to help engineers manage the complexity.

The main objective of this dissertation is to investigate how AI technologies can be used in product development. This includes investigating the prerequisites for the use of AI, examining these prerequisites in product development, and developing a concept for a system model through which the prerequisites can be met. In addition, the changes in the working methods of engineers during the implementation of the system model are explored.

The Engineering-Graph as a flexible system model based on graph databases is proposed. A prototype implementation of the Engineering-Graph was conducted at Fresenius Medical Care, a leading provider of solutions for people with kidney disease. It could be shown that the Engineering-Graph fulfills all requirements.

Vorwort

Die vorliegende Dissertation entstand während meiner Tätigkeit als Doktorand in der Abteilung Engineering System bei Fresenius Medical Care in Bad Homburg. Wissenschaftlich wurde die Arbeit von Herrn Prof. Dr.-Ing. Michael Vielhaber vom Lehrstuhl für Konstruktionstechnik an der Universität des Saarlandes in Saarbrücken begleitet. Diese Arbeit ist das Ergebnis meiner wissenschaftlichen Arbeit bei Fresenius Medical Care.

Meinen Dank möchte ich meinem Doktorvater, Herrn Prof. Dr.-Ing. Michael Vielhaber, für die gute Betreuung, den Ideenaustausch und das Feedback aussprechen. Er hat mich bei allen Schritten sehr gut unterstützt und die entstandenen Veröffentlichungen sowie die Dissertation ermöglicht und signifikant verbessert.

Herrn Prof. Dr.-Ing. Roman Dumitrescu danke ich für die freundliche Übernahme des Zweitgutachtens, das inhaltliche Interesse und die inspirierenden Einladungen ans Fraunhofer IEM in Paderborn.

Einen besonderen Dank möchte ich Herrn Dr.-Ing. Michael Bitzer aussprechen. Er hat mir die Türen in die Welt des Produktlebenszyklusmanagement und der Produktentwicklung geöffnet, mich immer gefördert und mich auf meinem wissenschaftlichen und praktischen Weg mit vielen Ideen und konstruktivem Feedback unterstützt, geprägt und begleitet.

Allen aktuellen und ehemaligen Kollegen bei Fresenius Medical Care sowie der Universität des Saarlandes danke ich für die stets sehr gute Zusammenarbeit und das angenehme Arbeitsklima. Hervorzuheben sind Nico Michels und Oliver Paul von Fresenius Medical Care sowie Simon Mörsdorf von der Universität des Saarlandes. Vielen Dank für die vielen Gespräche auf beruflicher und persönlicher Ebene und eure Begeisterung für die Produktentwicklung.

Des Weiteren danke ich der Organisation Fresenius Medical Care für die Unterstützung und die Bereitstellung von Daten. Ich durfte mich allein der Forschung widmen und konnte Veröffentlichungen schreiben sowie Konferenzen besuchen. Ferner allen Studierenden, die mich durch ihre Studien- und Abschlussarbeiten oder durch ihre studentische Hilfstätigkeit bei meiner Arbeit unterstützt haben.

Größter Dank gilt meinen Freunden, meiner Familie und meiner Partnerin. Ohne euch wäre diese Arbeit nicht möglich gewesen.

Darmstadt, Oktober 2023

Gregor M. Schweitzer

Publikationen

- Schweitzer, G., Bitzer, M., & Vielhaber, M. (2023). Engineering Graph as an approach to support design decisions in product development. *Proceedings of the International Conference on Engineering Design (ICED23)*, 3, 1625–1634. <https://doi.org/10.1017/pds.2023.163>
- Schweitzer, G. M., Bitzer, M., & Vielhaber, M. (2023). Lifecycle Engineering in the Context of a Medical Device Company–Leveraging MBSE, PLM and AI. *Product Lifecycle Management. PLM in Transition Times: The Place of Humans and Transformative Technologies: 19th IFIP WG 5.1 International Conference, PLM 2022, Grenoble, France, July 10–13, 2022, Revised Selected Papers*, 557–566. https://doi.org/10.1007/978-3-031-25182-5_54
- Schweitzer, G. M., Mörsdorf, S., Bitzer, M., & Vielhaber, M. (2022). Detection of cause-effect relationships in Life Cycle Sustainability Assessment based on an Engineering Graph. *DESIGN 2022*. <https://doi.org/10.1017/pds.2022.115>
- Schweitzer, G. M., Bitzer, M., & Vielhaber, M. (2021). Artificial intelligence in engineering: Evolution of virtual product development in the context of medical device industry. *CIRP Design 2021*, 100, 349–354. <https://doi.org/10.1016/j.procir.2021.05.081>
- Schweitzer, G. M., Bitzer, M., & Vielhaber, M. (2020). Produktentwicklung: KI-ready? *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 115(12), 873–876. <https://doi.org/10.3139/104.112464>

Inhaltsverzeichnis

Kurzfassung	iv
Abstract	v
Vorwort.....	vi
Publikationen	vii
1 Einleitung.....	1
1.1 Problematik.....	1
1.2 Zielsetzung.....	3
1.3 Vorgehensweise	4
2 Problemanalyse.....	7
2.1 Begriffsbestimmungen	7
2.1.1 Begriffe der Ingenieurwissenschaften	7
2.1.2 Begriffe der Informatik.....	13
2.2 Herausforderungen.....	20
2.3 Anforderungen an ein KI-fähiges Systemmodell.....	23
2.3.1 Framework zur Ableitung von Anforderungen.....	23
2.3.2 Ableitung der Anforderungen an ein KI-fähiges Systemmodell.....	24
3 Stand der Technik.....	27
3.1 Ingenieurwissenschaften.....	27
3.1.1 Produktlebenszyklusmanagement	27
3.1.2 Model-based-Systems-Engineering	30
3.1.3 Life-Cycle-Engineering.....	35
3.2 Informatik	38
3.2.1 Data-Science.....	38

3.2.2 Künstliche Intelligenz.....	41
3.2.3 Datenbanken	48
3.3 Systemmodelle in der Produktentwicklung.....	50
3.3.1 Modellierung	51
3.3.2 Vernetzung von Daten in der Produktentwicklung	54
3.4 Handlungsbedarf	62
4 Engineering-Graph.....	66
4.1 Engineering-Graph im Überblick	66
4.2 Aufbau und Erweiterung des Engineering-Graph	68
4.2.1 Aufbau des Engineering-Graph.....	68
4.2.2 Erweiterung des Engineering-Graph	71
4.3 Betrieb und Pflege des Engineering-Graph.....	73
4.4 Nutzung des Engineering-Graph.....	78
4.5 Wirtschaftliche Betrachtung	82
5 Anwendung des Engineering-Graph in der Medizintechnik.....	84
5.1 Vorstellung Fresenius Medical Care	84
5.2 Aufbau des Prototyps.....	85
5.3 Engineering-Graph in der Praxis	87
5.3.1 Klassifizierung und Auswahl der Anwendungsfälle	88
5.3.2 Engineering-Analytics	89
5.3.3 Natural-Language-Processing für Anforderungsmanagement.....	92
5.3.4 Life-Cycle-Sustainability-Assessment.....	95
5.3.5 Engineering-Graph zur Unterstützung des Life-Cycle-Engineering	97
5.3.6 Engineering-Graph unterstützt Designentscheidungen	97
5.4 Bewertung der Arbeit an den Anforderungen.....	101
6 Zusammenfassung und Ausblick.....	104

6.1 Zusammenfassung	104
6.2 Limitationen	104
6.3 Fazit anhand der Forschungsfragen	106
6.4 Ausblick auf zukünftige Forschung und Folgearbeiten.....	106
Abkürzungsverzeichnis.....	xii
Abbildungsverzeichnis.....	xiv
Tabellenverzeichnis.....	xvi
Literaturverzeichnis.....	xvii

1 Einleitung

Im ersten Kapitel dieser Dissertation wird die Motivation für die Erstellung der Arbeit aufgezeigt und die Problematik beschrieben, daraus werden Ziele und Forschungsfragen abgeleitet. Abschließend werden das Vorgehen und der Aufbau der Arbeit erklärt.

1.1 Problematik

Moderne Produktentwicklung findet in einer digitalen und komplexen Umgebung statt. Dies zeigt zum Beispiel die Menge an Daten, die von der Menschheit jedes Jahr erzeugt werden. Waren es 2010 noch 2 Zettabytes, sind es 2022 bereits 97 Zettabytes an Daten (statista, 2021). Dieser exponentielle Anstieg bringt mit sich, dass immer mehr Daten und Informationen verfügbar sind. So ist im gleichen Zeitraum die Anzahl der Webseiten um ca. 417 % (statista, 2022b) und der Wikipedia-Artikel um ca. 381 % (statista, 2022c) gestiegen. Diese gestiegene Datenmenge beinhaltet Informationen, die zum Verständnis der Umwelt, in der ein Produkt eingesetzt wird, genutzt werden können.

Neben der gestiegenen verfügbaren Datenmenge von außerhalb eines Unternehmens verändert sich auch das Unternehmen selbst. Es wird mit einer globalen Wertschöpfungskette und mit Entwicklungspartnern, Kunden und Zulieferern aus aller Welt gearbeitet (Eurostat, 2023). Diese Entwicklung ermöglicht es Unternehmen, Standortvorteile in vielen Regionen zu nutzen. Vernetzte Wertschöpfungsketten bringen auch Risiken mit sich. Seit dem Jahr 2022 gibt es einen Krieg in Europa (ZEIT ONLINE, 2023), Spannungen im Handelskonflikt mit China (Bundeszentrale für politische Bildung, 2020) und weitere politische Konflikte, die große Auswirkungen auf die Wertschöpfungsketten haben können. Auch Naturkatastrophen wie Vulkanausbrüche, die den Flugverkehr zum Erliegen bringen (Süddeutsche Zeitung, 2011), oder menschliches Versagen wie bei der Blockade des Suezkanals als wichtige Handelsroute (J. M. Lee & Wong, 2021) können zu Lieferschwierigkeiten und Engpässen führen.

Auch in der Produktentwicklung erhöhen sich die Anzahl und Vernetzung der Daten. Dies zeigt sich zum Beispiel im Anstieg der Anzahl der Internet of Things (IoT)-Verbindungen von 2010 bis 2022 um 2050 % (statista, 2022a). Produktdaten werden mehr und ändern sich häufiger und es werden Felddaten von bereits verkauften Geräten zurück ins Unternehmen gesendet, die ausgewertet und genutzt werden können.

Diese großen Datenmengen von außerhalb und innerhalb eines Unternehmens können zur Verbesserung der Produktentwicklung herangezogen werden. Allerdings ist eine manuelle Auswertung durch Ingenieur_innen nicht mehr möglich. Zur Analyse der Daten müssen große

Datenmengen von außerhalb des Unternehmens ausgewertet und mit den eigenen Produktdaten in Verbindung gesetzt werden, um diese zur Verbesserung der Produktentwicklung nutzen zu können.

Eine maschinelle Unterstützung von Ingenieur_innen bei der Bewältigung dieser Komplexität könnten Systeme auf Basis von Data-Science und Künstlicher Intelligenz (KI) bieten. In der Disziplin der Data-Science werden statistische Methoden genutzt, um große Datenmengen zu untersuchen. KI-Systeme werden durch neuronale Netze umgesetzt, die selbstständig aus großen Datenmengen Informationen erlernen können. Beide Ansätze könnten dazu geeignet sein, Ingenieur_innen bei der Analyse von Daten zu unterstützen. Diese Arbeit leistet einen Beitrag dazu, zu untersuchen, welche Voraussetzungen für den Einsatz von KI vorliegen müssen, diese Voraussetzungen in der Produktentwicklung zu prüfen und ein Konzept zu erarbeiten, durch das die Voraussetzungen erfüllt werden können.

Eine Voraussetzung für den Einsatz von KI ist, dass viele Quellen vorliegen, die als Input genutzt werden können, um ein KI-System zu trainieren, das Produktentwicklung ganzheitlich unterstützen kann. Diese Quellen stammen aus der immer größer werdenden Menge an Daten und Informationen, die im Web verfügbar sind. Zusätzlich können kommerzielle Datenbanken, die Nachhaltigkeitsinformationen enthalten, Normen und Ausschreibungsdokumente sowie unternehmensinterne Quellen genutzt werden.

Zur Verwaltung der Daten werden in Unternehmen verschiedene Konzepte und IT-Systeme eingesetzt. Diese sind Enterprise-Resource-Planning (ERP), Product-Lifecycle-Management (PLM), Systems-Engineering, Internet of Things (IoT) und viele weitere für spezielle Anwendungsfälle optimierte Lösungen. Jedes IT-System verwendet eine eigene Datenbank mit eigener Struktur. Produktmodelle werden im Systems-Engineering erzeugt. Dabei kommt eine Vielzahl von Modellierungssprachen zum Einsatz. Andere Informationen, wie Prozesse und interne Regularien, liegen in Form von vereinzelt Dokumenten vor. Die Daten sind somit teils strukturiert und teils unstrukturiert.

Ein gemeinsames Systemmodell, auf dessen Basis KI trainiert und eingesetzt werden kann, existiert häufig nicht. Daher wird in dieser Arbeit ein Vorschlag präsentiert, wie ein solches Systemmodell erstellt werden könnte. Dazu werden Graph-Datenbanken genutzt, um die unternehmensinternen Quellen aus den Engineering-Systemen mit frei verfügbaren Informationen aus dem Web zu verknüpfen. Alle Relationen zwischen den Datenpunkten werden im Engineering-Graph abgebildet.

Die Entwicklung des Engineering-Graph wird im Kontext Industrie 5.0 durchgeführt, in der der Mensch stärker in den Mittelpunkt der Betrachtung rückt (Alves et al., 2023). Dies bedeutet, dass die Technologie dem Menschen dienen und nicht zu viel vorgeben soll. Daher wird die Entwicklung eines flexiblen Unterstützungssystems untersucht, das Menschen dabei unterstützt, ihre Ziele zu erreichen.

Diese Arbeit entstand am Beispiel eines führenden Medizintechnikhersteller. Um den genannten Herausforderungen zu begegnen, folgt Fresenius Medical Care seiner Digital Engineering Roadmap (Bitzer et al., 2022). Diese wurde gestartet mit ‚Engineering Efficiency‘ und der damit verbundenen Einführung von PLM-Systemen. Der zweite Schritt ist ‚Engineering Excellence‘, in dem System-Lifecycle-Management eingeführt wird. Der dritte Schritt ist ‚Engineering Innovation‘, in dem Engineering-Intelligence angestrebt wird. Engineering-Intelligence ist eine neue Konstruktionsmethodik, die über die "klassische" Konstruktionsmethodik hinausgeht und Methoden und Technologien der Business-Analytik und des wissensbasierten Engineerings kombiniert (Bitzer et al., 2017). Diese Dissertation ist ein Baustein zur Erreichung von Engineering-Intelligence.

1.2 Zielsetzung

In dieser Dissertation soll untersucht werden, wie KI-Technologien in der Produktentwicklung eingesetzt werden können. Diese Technologien sollen Ingenieur_innen in der Produktentwicklung unterstützen, indem sie durch Generierung von Informationen aus großen Datenmengen und die Auswahl relevanter Informationen helfen, die Komplexität zu bewältigen.

Zur Untersuchung wurden in dieser Arbeit Forschungsziele verfolgt, die in Abbildung 1 dargestellt sind. Im Folgenden werden diese vorgestellt.

Forschungsziele

- Verständnis von künstlicher Intelligenz inklusive der Voraussetzungen für den Einsatz in der Produktentwicklung.
- Beschreibung heutiger Modellierung in der Produktentwicklung mit den Elementen Produktdaten, Produktmodelle & unstrukturiertes Expertenwissen.
- Konzeption eines KI-fähigen Systemmodells für die Produktentwicklung.
- Prototypische Umsetzung des entwickelten Konzeptes und Anwendung in der Produktentwicklung.
- Integration von Aufbau und Pflege des Systemmodells in die Arbeitsweise von Ingenieur_innen.

Abbildung 1. Forschungsziele

Zunächst soll ein Verständnis für KI aufgebaut werden. Das Ziel ist die Untersuchung aktueller KI-Technologien und Ableitung deren Voraussetzungen für den Einsatz. Das zweite Ziel ist die Herausarbeitung des aktuellen Standes der Technik der Produktmodellierung. Die Hypothese ist, dass die bestehende Datengrundlage in der Produktentwicklung nicht KI-fähig ist, da Daten in zu vielen Systemen ohne Vernetzung untereinander vorliegen und daher die Voraussetzungen für den Einsatz von KI-Technologie nicht erfüllt sind. Um diese Hypothese zu verifizieren, wird die heutige Modellierung in der Produktentwicklung mit den Elementen Produktdaten, Produktmodellen und unstrukturiertem Expertenwissen beschrieben.

Das dritte Ziel der Arbeit ist die Konzeption eines KI-fähigen Systemmodells für die Produktentwicklung. Eine prototypische Umsetzung des Konzeptes soll realisiert, und in

verschiedenen Anwendungsfällen in der Produktentwicklung eines Medizintechnikherstellers erprobt werden. Das letzte Ziel der Arbeit ist die Untersuchung, wie die Arbeitsweise von Ingenieur_innen sich verändern muss, um die Umsetzung des erarbeiteten Konzeptes im Kontext von Model-based-Systems-Engineering (MBSE) zu verankern.

Um die Forschungsziele zu erreichen, wurden die folgenden Forschungsfragen aufgestellt, die in Abbildung 2 dargestellt sind. Diese werden im Rahmen dieser Dissertation untersucht.

Forschungsfragen

1. Wie kann künstliche Intelligenz (KI) in der Produktentwicklung zur Unterstützung von Ingenieur_innen bei der Bewältigung der Komplexität eingesetzt werden?
 - a. Welche Voraussetzungen werden benötigt, um KI in der Produktentwicklung einsetzen zu können?
 - b. Was sind die Anforderungen der KI-Technologie an ein Systemmodell?
 - c. Wie können bestehende MBSE-Modelle erweitert werden, um die Anforderungen an ein Systemmodell zu erfüllen?
 - d. Welches ist die geeignete Modellierungssprache und -methode für das Systemmodell?
2. Wie kann die Erstellung und Pflege des Systemmodells in die Produktentwicklung integriert werden?

Abbildung 2. Forschungsfragen

Die erste Forschungsfrage ist die Leitfrage dieser Dissertation. Es soll untersucht werden, wie KI in der Produktentwicklung zur Bewältigung der Komplexität eingesetzt werden kann. Um sich dieser Frage zu nähern, wurde sie in vier Unterfragen aufgeteilt. Die erste Unterfrage lautet: Welche Voraussetzungen müssen erfüllt sein, um KI in der Produktentwicklung einsetzen zu können? Sie zielt auf ein Verständnis der KI-Technologie ab und ermittelt die Kernelemente. Die zweite Unterfrage untersucht, welche Anforderungen KI an ein Systemmodell in der Produktentwicklung hat. Die dritte Unterfrage soll untersuchen, ob in der Produktentwicklung bereits ein Modell besteht, das diese Anforderungen erfüllt. Die vierte Unterfrage lautet: Wie könnte ein Systemmodell, das die Anforderungen von KI erfüllt, aussehen?

Die zweite Forschungsfrage ist, wie die Aktivitäten um Aufbau und Pflege des Systemmodells in die Produktentwicklung integriert werden können. Eine Hypothese ist, dass die Rollen im Systems-Engineering angepasst werden können.

1.3 Vorgehensweise

Zur Beantwortung der Forschungsfragen und zur Erreichung der Forschungsziele dient das Vorgehen der Design-Research-Methodology (DRM) als Orientierung (Blessing & Chakrabarti, 2009). In diesem wird zunächst die Phase ‚Research-Clarification‘ durchgeführt, in der die Ist- sowie die gewünschte Zielsituation beschrieben und die Forschungsfragen und -ziele aufgestellt werden. Abbildung 3 zeigt die Vorgehensweise.



Abbildung 3. Vorgehen bei dieser Dissertation

Nach der DRM wird nach der Research-Clarification die erste deskriptive Studie durchgeführt. Diese untersucht den aktuellen Stand der Forschung von Data-Science und KI-Technologie sowie aktueller Modellierungssprachen anhand einer Literaturrecherche. Hier werden Anforderungen der KI-Technologie an ein Systemmodell abgeleitet. Im nächsten Schritt folgt eine präskriptive Studie, in der ein Lösungsvorschlag erarbeitet und aufgezeigt wird, wie ein Systemmodell, das die abgeleiteten Anforderungen der KI-Technologie erfüllt, aussehen kann. Es wird ein Prototyp dieses Engineering-Graph erstellt und beschrieben sowie beispielhaft an einem Anwendungsfall getestet. Abschließend wird eine zweite deskriptive Studie erstellt, in der die Einbettung des Aufbaus und der Pflege des Engineering-Graph in die Arbeitsweise von Ingenieur_innen vorgenommen werden. Außerdem wird gezeigt, wie der Engineering-Graph Designentscheidungen unterstützen kann.

Diese Arbeit gliedert sich, wie in Abbildung 4 dargestellt, in sechs Kapitel. Im ersten Kapitel werden die Problematik sowie die Zielsetzung und der Aufbau der Arbeit beschrieben. In Kapitel 2 wird das Problem analysiert. Dazu werden die wichtigsten Begriffsdefinitionen vorgestellt, die Herausforderungen beschrieben und Anforderungen an die Arbeit abgeleitet. In Kapitel 3 wird der aktuelle Stand der Technik in den für diese Arbeit relevantesten Themen dargelegt. Außerdem wird aus dem Stand der Technik der Handlungsbedarf abgeleitet.

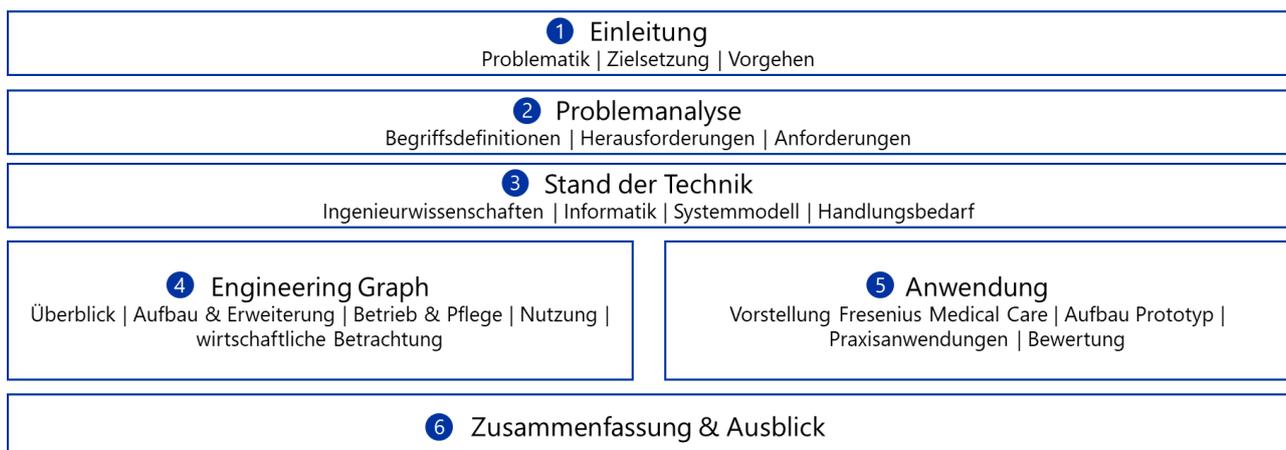


Abbildung 4. Aufbau dieser Dissertation

In Kapitel 4 wird der Engineering-Graph vorgestellt. Es wird beschrieben, wie dieser aufgebaut und genutzt wird und wie er in die Arbeitsweise von Ingenieur_innen eingebettet werden kann. In

Kapitel 5 wird eine beispielhafte Anwendung des Engineering-Graph erläutert. Kapitel 6 enthält eine Zusammenfassung der Ergebnisse der Arbeit, ein Fazit bezüglich der in Kapitel 2 aufgestellten Anforderungen und einen Ausblick auf weitere Tätigkeiten in Wissenschaft und Praxis.

2 Problemanalyse

In diesem Kapitel werden zunächst die wichtigsten Begriffe und Konzepte erklärt, im Anschluss werden die Herausforderungen beschrieben sowie Anforderungen an ein Konzept zum Einsatz von KI-Technologie in der Produktentwicklung zur Unterstützung von Ingenieur_innen bei der Bewältigung der Komplexität aufgestellt.

2.1 Begriffsbestimmungen

Die wichtigsten Begriffe und Konzepte, die für das Verständnis der weiteren Arbeit notwendig sind, werden in diesem Unterkapitel vorgestellt. Da die Arbeit sich an der Schnittstelle der Ingenieurwissenschaften und der Informatik bewegt, werden zunächst die Begriffe der Ingenieurwissenschaften und im Anschluss die der Informatik definiert. In beiden Fällen werden zunächst Definitionen vorgestellt, die in der Literatur existieren, bevor die für diese Arbeit gültige Definition begründet ausgewählt wird.

2.1.1 Begriffe der Ingenieurwissenschaften

Hier werden die für diese Arbeit relevanten Begriffe der Ingenieurwissenschaften vorgestellt. Diese sind Model-based-Systems-Engineering, Produktlebenszyklusmanagement und Life-Cycle-Engineering. Zunächst werden die Produkte und Services definiert, die durch Produktentwicklung erstellt werden sollen.

2.1.1.1 *Produkte und Services als Ergebnis der Produktentwicklung*

Das Ziel von Ingenieur_innen in der Praxis ist die Entwicklung von Produkten. Unter einem **Produkt** wird materielles oder immaterielles Gut verstanden, das am Markt verkauft werden kann (Kotler et al., 2007). Das Produkt kann aus mechanischen, elektrischen und Softwarekomponenten bestehen oder ein Service sein.

Als **Produkt-Service-System** werden Produkte bezeichnet, deren Wert für Kund_innen durch eine Kombination aus dem Produkt selbst und weiteren immateriellen Services geschaffen wird (Annarelli et al., 2016). Ein Beispiel hierfür ist ein modernes Smartphone, das als Produkt bereits mechanische, elektrische und Softwarekomponenten besitzt. Der Wert für Nutzer_innen wird allerdings durch weitere Services (von Drittanbietern) wie Streaming von Musik oder Video ergänzt.

Die DIN 19226 definiert ein **System** als „eine in einem betrachteten Zusammenhang gegebene Anordnung von Elementen, die miteinander in Wechselwirkung stehen. Diese Anordnung wird aufgrund bestimmter Vorgaben gegenüber ihrer Umgebung abgegrenzt“ (DIN 19266, 2015).

Ein System kann aus verschiedenen Elementen bestehen, die einzeln betrachtet wiederum als System verstanden werden können. Auf diese Art können Systeme modularisiert, einzeln entwickelt und wieder integriert werden.

Ein **System of Systems** ist ein System, das während der Nutzung mit weiteren Systemen anderer Hersteller in Kontakt treten muss, um eine Marktleistung zu erbringen (Walden et al., 2015). Dabei sollen die Systeme zwei Eigenschaften besitzen: operationale Unabhängigkeit und Verwaltungsunabhängigkeit (Maier, 1998). Operationale Unabhängigkeit bedeutet, dass die einzelnen Systeme, aus denen ein System of Systems besteht, auch einzeln ihren Nutzen erbringen können. Verwaltungsunabhängigkeit bedeutet, dass die einzelnen Systeme auch in der Praxis unabhängig voneinander handeln. Sie werden unabhängig voneinander erworben und integriert, behalten dabei aber ihre Handlungsfähigkeit unabhängig vom System of Systems. Ein Beispiel hierfür ist eine Drohne, die durch ein Smartphone eines beliebigen Herstellers gesteuert werden soll. Die Drohne mit dem Smartphone bildet in diesem Fall ein System of Systems.

Es können vier Typen eines System of Systems unterschieden werden: (1) gerichtet, (2) virtuell, (3) kollaborativ und (4) anerkannt (Holt et al., 2012). Ein gerichtetes System of Systems besteht zu einem übergeordneten Zweck, der den individuellen Zwecken der Systeme, die das System of Systems umfasst, vorsteht. Eine zentrale Autorität setzt diesen übergeordneten Zweck durch. Die einzelnen Systeme können zwar individuell handeln, ihre Partizipation am System of Systems führt aber zur Erreichung des übergeordneten Zwecks. In einem virtuellen System of Systems gibt es keine zentrale Autorität, die einen übergeordneten Zweck festlegt. Ein kollaboratives System of Systems hat zwar einen übergeordneten Zweck, allerdings keine Autorität, die diesen festlegt und durchsetzt. Die Verbindung eines gerichteten und eines kollaborativen System of Systems wird ‚anerkannt‘ genannt. Dieses hat einen übergeordneten Zweck und eine zentrale Autorität, allerdings gibt es auch Systeme, die nicht der zentralen Autorität unterstehen.

Produkte, die am Markt nachgefragt werden, verändern sich. Zunehmende Digitalisierung in allen Bereichen und die Verfügbarkeit großer Datenmengen führen zu einer Entwicklung von Produkten zu sogenannten **Advanced Systems** (Dumitrescu et al., 2021). Diese sind Produkt-Service-Systeme, die durch eine dynamische Vernetzung mit weiteren Produkten, Autonomie in der Handlung und Umsetzung einer Marktleistung sowie interaktive, soziotechnische Integration geprägt sind (Dumitrescu et al., 2021).

2.1.1.2 Model-based-Systems-Engineering

In diesem Kapitel werden die relevantesten Begriffe für MBSE definiert. MBSE ist eine verbreitete Methode, die die Entwicklung von modernen Produkten wie Advanced Systems ermöglicht. Eine Näherung an die Definition erfolgt, indem zunächst Systems-Engineering, dann ein Modell und

zuletzt MBSE definiert wird. Im Anschluss daran werden für MBSE wichtige Begriffe wie System of Interest und verschiedene Möglichkeiten der Modellierung vorgestellt.

Nach dem International Council on Systems Engineering (INCOSE) zielt **Systems-Engineering** darauf ab, sicherzustellen, dass die Elemente zusammenarbeiten, um die Ziele des Ganzen zu erreichen (INCOSE, 2022). Dafür müssen (a) ausgewogene Lösungen entwickelt werden, die den Bedürfnissen der verschiedenen Interessengruppen in Bezug auf Leistungsfähigkeit, Zuverlässigkeit, Nachhaltigkeit, soziale Akzeptanz und Benutzerfreundlichkeit gerecht werden, (b) Anpassungen an sich entwickelnde Technologien und Anforderungen durchgeführt werden und (c) Komplexität und Risiko bewältigt werden (INCOSE, 2022, S. 203). Diese Definition soll für diese Arbeit gelten.

Nach einer weiteren Definition von INCOSE ist Systems-Engineering ein „interdisziplinärer Ansatz und Instrument, um die Realisierung erfolgreicher Systeme zu ermöglichen. Es konzentriert sich auf die Definition der Kundenbedürfnisse und der erforderlichen Funktionalität in einer frühen Phase des Entwicklungszyklus, die Dokumentation der Anforderungen, die Synthese des Entwurfs und die Systemvalidierung unter Berücksichtigung des gesamten Problems“ (Haskins et al., 2006).

Sillitto definiert Systems-Engineering als transdisziplinären und integrativen Ansatz, der die erfolgreiche Realisierung, Nutzung und Ausmusterung von technischen Systemen unter Verwendung von Systemprinzipien und -konzepten sowie wissenschaftlichen, technologischen und Managementmethoden ermöglicht (Sillitto et al., 2019).

Ein **Modell** ist eine vereinfachte Repräsentation eines realen Systems (Department of Defence, 1998). Ein Systemmodell beschreibt umfassend ein Produkt unter Berücksichtigung der Anforderungen sowie der strukturellen und verhaltensbezogenen Aspekte (Hart, 2015). Systemmodelle sollen Produkte mit ihren für den gesamten Lebenszyklus relevanten Informationen digital abbilden. In der Literatur wird auch vom virtuellen Produkt gesprochen (Eigner & Stelzer, 2013; Spur & Krause, 1997).

Model-based-Systems-Engineering ist ein aufstrebendes Paradigma, das herkömmliche Methoden und bewährte Praktiken mit Modellierungstechniken verbindet (Ramos et al., 2011). Die INCOSE definiert MBSE als „die formalisierte Anwendung der Modellierung zur Unterstützung der Systemanforderungen, des Entwurfs, der Analyse, der Verifizierung und der Validierung, beginnend in der Konzeptionsphase und sich über die gesamte Entwicklung und die späteren Lebenszyklusphasen erstreckend“ (Friedenthal et al., 2007).

Zur Erstellung des Produktmodells und damit zur Umsetzung von MBSE wird das **MBSE-Dreieck** benötigt, das aus Modellierungsmethode, -sprache und -werkzeug besteht (Dumitrescu et al., 2015).

Die Weiterentwicklung der Engineering Methoden kann als **Advanced-Engineering** beschrieben werden: „Advanced Engineering berücksichtigt die Prozesse, Methoden und Werkzeuge sowie die

Arbeitsorganisation, um die etablierten Engineering-Ansätze durch Kreativität, Agilität und Digitalisierung zu erweitern.“ (Dumitrescu et al., 2021).

Eine Erweiterung des Systems-Engineering ist das **Advanced Systems-Engineering**. Dieses Leitbild integriert die vielfältigen Aspekte des Systems-Engineering und Advanced-Engineerings und bildet eine fundierte Basis für die Entstehung sowie Erbringung der Advanced Systems als innovative Marktleistung (Dumitrescu et al., 2021).

Das **System of Interest** ist der Teil eines Systems, der im aktuellen Bearbeitungsschritt betrachtet werden soll (ISO 15288, 2015). Dies kann das gesamte System, ein Untersystem oder ein einzelnes Bauteil sein.

2.1.1.3 Produktlebenszyklusmanagement

Der Begriff ‚Produktlebenszyklusmanagement‘ (PLM) lässt sich in die Begriffe ‚Produkt‘, ‚Lebenszyklus‘ und ‚Management‘ aufteilen. Diese Begriffe sollen zunächst einzeln betrachtet werden, um eine Annäherung an die Definition von PLM zu erreichen. Der Begriff ‚Produkt‘ wurde bereits in Kapitel 2.1.1.1 eingeführt, diese Definition gilt weiterhin.

In der Literatur existieren verschiedene Definitionen des Produktlebenszyklus, die sich je nach Anwendungsfall stark voneinander unterscheiden. Dabei wird der Lebenszyklus mal enger und mal breiter gefasst.

In der Produktentwicklung startet die vorherrschende Definition des Lebenszyklus mit der Produkt- und Programmplanung. Die nächsten Phasen sind Entwicklung und Konstruktion, Arbeitsvorbereitung, Produktherstellung (Fertigung, Montage, Beschaffung), Produktvertrieb, Produktnutzung und -wartung sowie Produktrecycling (Anderl, 1998; Eigner & Stelzer, 2013). Diese Definition orientiert sich an den Aktivitäten innerhalb eines Unternehmens, die notwendig sind, um ein Produkt auf den Markt zu bringen, zu betreiben und wieder vom Markt zu nehmen.

Eine weitere Sichtweise auf den Lebenszyklus betrifft die Nachhaltigkeit. Dabei spielt der Materialfluss eine große Rolle. Folglich definiert die DIN EN ISO 14044 den Lebenszyklus als „aufeinanderfolgende und miteinander verknüpfte Phasen eines Produktsystems, von der Rohstoffgewinnung oder -erzeugung aus natürlichen Ressourcen bis zur endgültigen Entsorgung“ (DIN EN ISO 14044, 2006). Diese Definition ist deutlich weitreichender, hier werden auch Abläufe und Phasen miteinbezogen, die nicht innerhalb des Unternehmens passieren oder durch das Unternehmen kontrolliert werden können.

Für diese Arbeit wird eine möglichst umfassende Definition des Lebenszyklus benötigt. Es sollen Informationen aus sehr frühen Phasen miteinbezogen werden, indem sie mit Informationen aus späteren Phasen verknüpft werden. Daher werden die bereits vorgestellten Definitionen verbunden, indem die Phasen der Rohstoffgewinnung zur Produktherstellung und die Phasen der endgültigen

Entsorgung zum Recycling hinzugefügt werden. Des Weiteren sollen der Transport sowohl der Rohstoffe als auch der fertigen Produkte und die Erbringung zusätzlicher Services in der Nutzungsphase berücksichtigt werden.

Daraus ergibt sich für diese Arbeit ein **Lebenszyklus** mit den folgenden Phasen: Produkt- und Programmplanung, Entwicklung und Konstruktion, Arbeitsvorbereitung, Produktherstellung (Rohstoffgewinnung oder -erzeugung, Fertigung, Montage, Beschaffung, Transport), Produktvertrieb, Produktnutzung, -wartung und -service sowie Produktrecycling und endgültige Entsorgung.

Die deutsche Übersetzung von **Management** lautet ‚Verwaltung‘ (Duden, 2022). Darunter fallen die Leitung, Organisation und Planung von Organisationen, Abläufen oder Dingen (Duden, 2022). Für diese Dissertation und dieses Kapitel sollen konkret die produktbezogenen Daten verwaltet werden.

Produktlebenszyklusmanagement ermöglicht es Unternehmen, ihre produktbezogenen Daten während des gesamten Lebenszyklus zu verwalten (Bitzer & Vielhaber, 2016). Terzi et al. (2010) definieren PLM als produktzentrierten, lebenszyklusorientierten Ansatz, der durch IT-Systeme unterstützt wird und in dem Produktdaten zwischen Akteuren, Prozessen und Organisationen in den verschiedenen Phasen des Produktlebenszyklus ausgetauscht werden, um die gewünschten Leistungen und die Nachhaltigkeit des Produkts sowie der damit verbundenen Dienstleistungen zu erzielen (Terzi et al., 2010). Um dies zu erreichen, ist PLM ein Integrator von Werkzeugen und Technologien, der den Informationsfluss durch die verschiedenen Phasen des Produktlebenszyklus rationalisiert. Im Gegensatz zu anderen Technologien handelt es sich bei PLM nicht um eine Punktlösung oder ein serienmäßiges Tool. Vielmehr basiert es auf der Philosophie der Vernetzung von Wissen und zielt darauf ab, die richtigen Informationen zur richtigen Zeit und im richtigen Kontext bereitzustellen (Terzi et al., 2010). Wie PLM dies erreicht, wird in Kapitel 3.1.1 vorgestellt. In dieser Arbeit wird die Definition von Terzi angenommen.

2.1.1.4 Life-Cycle-Engineering

In diesem Unterkapitel werden die Begriffe des Life-Cycle-Engineering und der verschiedenen Nachhaltigkeitsmethoden vorgestellt.

Life-Cycle-Engineering (LCE) ist eine auf Nachhaltigkeit ausgerichtete Engineering-Methode, die die umfassenden technischen, ökologischen und wirtschaftlichen Auswirkungen von Entscheidungen innerhalb des Produktlebenszyklus berücksichtigt (Hauschild et al., 2018). In diesem Kontext wird der Produktlebenszyklus wie in Kapitel 2.1.1.3 definiert angenommen.

Life-Cycle-Engineering konzentriert sich auf die Integration relevanter Nachhaltigkeitsinformationen in die Produktentwicklung und die technischen Aktivitäten (Hauschild et al., 2017). In der Literatur

gibt es verschiedene Ansätze, wie diese Integration erreicht werden kann. Diese werden in Kapitel 3.1.3 vorgestellt.

Die Methoden und Werkzeuge des LCE befassen sich mit den Entwicklungsaktivitäten in verschiedenen Domänen, die in den Lebenszyklus eines Produkts eingebunden sind. LCE-Tools ermöglichen wissenschaftsbasierte Empfehlungen, die sich aus quantitativen Lebenszyklusmodellen und deren Interpretation auf der Grundlage des Life-Cycle-Assessments (LCA) ergeben. Ein Schwerpunkt ist die Identifikation von Interdependenzen und potenziellen Kompromissen entlang des Lebenszyklus von Produktsystemen der nächsten Generation (Dilger et al., 2021). Für diese Arbeit wird die oben vorgestellte Definition von Hauschild angenommen.

Eine umfassende Methode zur Bewertung der Nachhaltigkeit ist das **Life-Cycle-Sustainability-Assessment** (LCSA) (Kloepffer, 2008). Dieses ermöglicht die Analyse und Bewertung eines Produktes in den drei Säulen der Nachhaltigkeit: (1) der ökologischen, (2) der ökonomischen und (3) der sozialen. Zur ökologischen Bewertung kann das LCA verwendet werden (DIN EN ISO 14040, 2009), die ökonomische Bewertung kann durch Life-Cycle-Costing (LCC) erfolgen (Hunkeler & Rebitzer, 2003) und die sozialen Auswirkungen können durch Social-Life-Cycle-Assessment (S-LCA) analysiert werden (Benoît & Mazijn, 2013; Benoît Norris et al., 2020). Life-Cycle-Sustainability-Assessment kann also als $LCSA = LCA + LCC + S-LCA$ definiert werden (Costa et al., 2019; Kloepffer, 2008). Die einzelnen Methoden werden im Folgenden vorgestellt.

Life-Cycle-Assessment ist eine etablierte Systemanalysemethode, die es ermöglicht, die Umweltauswirkungen von Produktsystemen in einer Lebenszyklusperspektive zu quantifizieren (Hauschild et al., 2018). Die in den ISO-Normen definierte Methodik (DIN EN ISO 14040, 2009) besteht aus vier Phasen (d. h. Definition von Ziel und Umfang, Bestandsanalyse, Folgenabschätzung und Interpretation), die die LCA-Expert_in kontinuierlich durchläuft.

Life-Cycle-Costs (LCC) können als alle Kosten, die mit dem System verbunden sind, bezogen auf den definierten Lebenszyklus, definiert werden (Blanchard & Fabrycky, 2006). Sie bilden ein wesentliches Bindeglied für die Verbindung von Umweltbelangen mit zentralen Unternehmensstrategien. Ein Beispiel für die Integration von Wirtschaft und Umwelt durch LCC sind die handelbaren Zertifikate, die Teil des Kyoto-Protokolls sind. Andere Beispiele sind die Umsetzung von umweltgerechtem Design und integrierter Produktpolitik in Unternehmen und entlang von Lieferketten. Hier müssen Synergien zwischen ökologischen und wirtschaftlichen Überlegungen genutzt werden, um eine nachhaltige Entwicklung zu erreichen (Hunkeler & Rebitzer, 2003).

Ein **Social-Life-Cycle-Assessment** ist ein Verfahren zur Bewertung der (potenziellen) sozialen Auswirkungen von Produkten, das darauf abzielt, die sozialen und sozioökonomischen Aspekte von Produkten und ihre potenziellen positiven und negativen Auswirkungen entlang ihres Lebenszyklus

zu bewerten, der die Gewinnung und Verarbeitung von Rohstoffen, die Herstellung, den Vertrieb, die Verwendung, die Wiederverwendung, die Wartung, das Recycling und die endgültige Entsorgung umfasst (Benoît & Mazijn, 2013). Das S-LCA bietet ein systematisches Framework, das sowohl quantitative als auch qualitative Daten kombiniert. Es liefert Informationen zu sozialen und sozioökonomischen Aspekten für die Entscheidungsfindung, mit der Aussicht, die soziale Leistung einer Organisation und das Wohlergehen der Stakeholder zu verbessern (Benoît Norris et al., 2020). Das S-LCA unterscheidet sich von anderen Verfahren zur Bewertung sozialer Auswirkungen durch ihren Gegenstand: Produkte und Dienstleistungen sowie ihren Anwendungsbereich, den gesamten Lebenszyklus. Soziale und sozioökonomische Aspekte, die in der S-LCA bewertet werden, sind diejenigen, die sich während des Lebenszyklus eines Produkts direkt positiv oder negativ auf die Betroffenen auswirken können. Sie können mit dem Verhalten von Unternehmen, mit sozioökonomischen Prozessen oder mit Auswirkungen auf das Sozialkapital verbunden sein.

2.1.2 Begriffe der Informatik

In diesem Unterkapitel werden die Begriffe und Konzepte der Informatik vorgestellt, die zur Analyse von großen Datenmengen geeignet sind oder sich mit der Ablage von Daten beschäftigen. Diese sind Data-Science, Künstliche Intelligenz und verschiedene Datenbanktypen.

2.1.2.1 Data-Science

Die neuen Möglichkeiten der Data-Science-Technologien werden in diesem Unterkapitel vorgestellt. Dazu werden zunächst die Begriffe ‚Daten‘ und ‚Informationen‘ definiert. Im Anschluss daran wird Data-Science definiert.

Daten sind Symbole, die Eigenschaften von Objekten oder Events repräsentieren (Ackoff, 1989). Im Falle der Produktentwicklung könnten dies Engineering-Objekte und deren Metadaten sein. So liegen zum Beispiel für alle technischen Zeichnungen zusätzlich die Daten ‚Erstellungsdatum‘, ‚Ersteller‘, ‚Status‘ und ‚zugehöriges Produkt‘ vor.

Informationen sind verarbeitete Daten, deren Verarbeitung darauf abzielte, den Nutzen zu erhöhen (Ackoff, 1989). Wird zum Beispiel aus dem Erstellungsdatum, dem Status und dem Zeitpunkt des Statuswechsels die Durchlaufzeit errechnet, handelt es sich dabei um eine Information. Informationen können Fragen mit ‚wer‘, ‚was‘, ‚wann‘, ‚wo‘ und ‚wie viele‘ beantworten.

Im Allgemeinen ist **Data-Science** die Anwendung quantitativer und qualitativer Methoden zur Lösung relevanter Probleme und zur Vorhersage von Ergebnissen (Waller & Fawcett, 2013). Dabei steht Data-Science in direktem Zusammenhang mit der Verarbeitung und Kreuzkorrelation von einzelnen Informationsquellen (Rückemann et al., 2016).

Die Personen, die Data-Science-Aktivitäten ausführen, werden Data-Scientists genannt. Sie benötigen neben Methoden- auch Domänenwissen (Waller & Fawcett, 2013).

2.1.2.2 Künstliche Intelligenz

Um sich einer Definition von KI zu nähern, werden zunächst die Begriffe ‚künstlich‘ und ‚Intelligenz‘ betrachtet, bevor eine Definition für KI vorgestellt wird. Anschließend werden die Hauptbestandteile eines KI-Systems beschrieben.

Der Begriff ‚**künstlich**‘ kann auf zwei Arten verstanden werden (Oxford Advanced Learner’s Dictionary, 2020a):

1. nicht natürlich, sondern mit chemischen oder technischen Mitteln nachgebildet, nach einem natürlichen Vorbild angelegt, gefertigt, geschaffen
2. natürliche Vorgänge nachahmend, nicht auf natürliche Weise vor sich gehend

Die erste Definition kann so verstanden werden, dass ein System mit technischen Mitteln nachgebildet wurde, das in bestimmten Anwendungsfällen intelligentes Verhalten zeigt. Die zweite Definition zielt nicht auf das Ergebnis, intelligentes Verhalten, ab, sondern auf den Prozess oder Vorgang, wie dieses Ergebnis zustande kommt. Damit könnten die Prozesse im menschlichen Gehirn gemeint sein, die nachgebildet werden, um intelligentes Verhalten zu erzeugen. Für diese Arbeit steht das Ergebnis im Vordergrund. Es soll ein System entwickelt werden, das in bestimmten Situationen intelligentes Verhalten zeigt. Dabei ist es nicht wichtig, dass der Vorgang im technischen System dem natürlichen Vorgang gleicht. Daher gilt für diese Arbeit die Definition „*nicht natürlich, sondern mit technischen Mitteln nachgebildet*“ (Oxford Advanced Learner’s Dictionary, 2020a).

Intelligenz ist ein abstrakter Begriff, der daher schwierig zu definieren ist. Er bedeutet „die Fähigkeit, zu lernen, zu verstehen und logisch über Dinge nachzudenken; die Fähigkeit, dies gut zu tun“ (Oxford Advanced Learner’s Dictionary, 2020c). Diese Definition besteht aus den Schlüsselwörtern ‚lernen‘, ‚verstehen‘ und ‚logisch nachdenken‘.

Diese Schlüsselwörter decken sich fast mit den Dimensionen der Intelligenz: Wahrnehmen, Lernen, Abstrahieren und Begründen. Es wurde die Dimension der Wahrnehmung hinzugefügt, sodass Intelligenz die Erkennung der Umgebung einschließt. ‚Logisch nachdenken‘ wurde durch ‚begründen‘ erweitert. Es reicht nicht mehr aus, intelligentes Verhalten zu zeigen, sondern dieses Verhalten muss auch begründet werden (Russell et al., 2016).

Eine Expertengruppe der Europäischen Union (EU) hat folgende Definition vorgeschlagen:

„Intelligenz (sowohl bei Maschinen als auch bei Menschen) ist ein vages Konzept, obwohl es von Psychologen, Biologen und Neurowissenschaftlern ausgiebig untersucht wurde. KI-Forscher verwenden meist den Begriff der Rationalität. Darunter versteht man die Fähigkeit, die beste Handlung zu wählen,

um ein bestimmtes Ziel zu erreichen, und zwar unter Berücksichtigung bestimmter zu optimierender Kriterien und der verfügbaren Ressourcen. Natürlich ist die Rationalität nicht der einzige Bestandteil des Konzepts der Intelligenz, aber sie ist ein wichtiger Teil davon.“ (AI HLEG, 2019)

Hier geht es vor allem um die Auswahl einer Handlung, die geeignet ist, unter gegebenen Randbedingungen ein bestimmtes Ziel zu erreichen. Diese Definition ist für die vorliegende Arbeit geeignet. In der Produktentwicklung muss regelmäßig entschieden werden, welche die beste Handlung ist. Dabei gelten ökonomische und ökologische Randbedingungen im Unternehmen, die einzuhalten sind. Daher wird die Definition der EU in dieser Arbeit verwendet.

Künstliche Intelligenz kann als „die Untersuchung und Entwicklung von Computersystemen, die intelligentes menschliches Verhalten nachahmen können“, definiert werden (Oxford Advanced Learner’s Dictionary, 2020b). Weitere Definitionen für KI können in die Dimensionen Denken und Handeln sowie menschlich und rational eingeteilt werden (Russell et al., 2016). Eine Übersicht bietet Tabelle 1.

Tabelle 1. Definitionen Künstlicher Intelligenz (Russell et al., 2016)

Menschlich denken	Rational denken
<p>„die aufregenden neuen Anstrengungen, Computer zum Denken zu bringen ... Maschinen mit Verstand, im vollen und wörtlichen Sinne“ (Haugeland, 1989)</p> <p>„[die Automatisierung von] Tätigkeiten, die wir mit menschlichem Denken in Verbindung bringen, Tätigkeiten wie Entscheidungsfindung, Problemlösung, Lernen...“ (Bellman, 1978)</p>	<p>„die Untersuchung der geistigen Fähigkeiten durch den Einsatz von Computermodellen“ (Charniak & McDermott, 1985)</p> <p>„die Untersuchung der Berechnungen, die Wahrnehmung, Denken und Handeln ermöglichen“ (Winston, 1992)</p>
Menschlich handeln	Rational handeln
<p>„die Kunst, Maschinen zu entwickeln, die Funktionen ausführen, die bei der Ausführung durch Menschen Intelligenz erfordern“ (Kurzweil et al., 1990)</p>	<p>„Computational Intelligence ist die Lehre von der Entwicklung intelligenter Agenten.“ (Poole et al., 1998)</p> <p>„KI befasst sich mit dem intelligenten Verhalten von Artefakten.“ (Nilsson, 1998)</p>

<p>„das Studium der Frage, wie man Computer dazu bringt, Dinge zu tun, in denen Menschen im Moment besser sind“ (Rich & Knight, 1991)</p>	
---	--

Für diese Arbeit soll nicht das menschliche Denken oder Handeln im Vordergrund stehen, sondern die Lösung komplexer Probleme. Um Ingenieur_innen in ihrem Arbeitsalltag bei der Produktentwicklung zu unterstützen, soll KI eingesetzt werden, die konkrete Handlungen vorschlägt oder automatisiert durchführt. Daher gilt für diese Arbeit: KI ist die Realisierung von intelligentem Verhalten und den zugrunde liegenden kognitiven Fähigkeiten auf Computern (Wahlster, 2019).

KI-Systeme können nach ihrer Anwendung kategorisiert werden. Es existieren Systeme, die für einzelne Anwendungsfälle intelligente Lösungen erzeugen können, sogenannte **schwache KI** (I. Lee & Lee, 2015). Systeme, die für generelle Probleme Lösungen erzeugen können, werden **starke KI** genannt. Eine KI, die das menschliche Level der Intelligenz überschreiten kann, wird als Super-KI bezeichnet (Bentley, 2018). Aktuell existieren nur Systeme der schwachen KI. Diese sollen auch in der vorliegenden Arbeit untersucht werden, da spezielle Probleme in der Produktentwicklung adressiert werden.

Aktuelle KI-Systeme fallen alle in den Bereich der schwachen KI. Diese sind auf ein Gebiet spezialisiert und besitzen in diesem ausgeprägte Fähigkeiten, die denen des Menschen gleichkommen oder die menschlichen Fähigkeiten sogar übertreffen. Tabelle 2 gibt einen Überblick über die aktuellen Einsatzgebiete.

Tabelle 2. Einsatzgebiete Künstlicher Intelligenz

Einsatzgebiete	Beschreibung
<p>Maschinelles Lernen</p>	<p>Die künstliche Gewinnung von Wissen aus Erfahrung – das künstliche System – ‚lernt‘ aus Beispielen und kann es am Ende der Lernphase verallgemeinern (Shabestari et al., 2019).</p> <p>Automatische Wissensgenerierung zur Verbesserung des Verhaltens eines Systems (Lunze, 2016).</p> <p>Maschinelles Lernen ist die systematische Untersuchung von Algorithmen und Systemen, die ihr Wissen oder ihre Leistung mit der Erfahrung verbessern (Flach, 2012).</p>

<p>Automatisierte Logik</p>	<p>Beantwortung von Fragen und ziehen von neuen Schlussfolgerungen durch Nutzung der gespeicherten Informationen (Russell et al., 2016). Finden einer Lösung eines Problems durch Durchsuchen des Problemraums (Lunze, 2016).</p>
<p>Natural Language Processing (NLP)</p>	<p>Der Bereich der natürlichen Sprachverarbeitung befasst sich hauptsächlich mit der Interaktion zwischen menschlicher Sprache und Computern. NLP ermöglicht es Computern, die menschliche Sprache auf intelligente und nützliche Weise zu analysieren, zu verstehen und ihre Bedeutung abzuleiten (Lunze, 2016).</p>
<p>Image Processing and Vision</p>	<p>Vision bedeutet die Verarbeitung beliebiger Bild-/Videoquellen, um aussagekräftige Informationen zu extrahieren und auf dieser Grundlage Maßnahmen zu ergreifen (Lunze, 2016).</p>
<p>Expertensysteme</p>	<p>Ein Expertensystem ist ein Programm, das zur Lösung von Problemen entwickelt wurde, die menschliches Fachwissen oder Erfahrung erfordern. Indem es das Denken der menschlichen Experten nachahmt, kann das System die Analyse, den Entwurf oder die Überwachung durchführen, Entscheidungen treffen und vieles mehr. Der Vorteil dieses Systems ist, dass Expertenwissen verfügbar wird (Lunze, 2016).</p>
<p>Qualitative Logik</p>	<p>Qualitative Logik kann zur Darstellung und Analyse physikalischer Systeme anhand qualitativer Beschreibungen eingesetzt werden. Gegenstand der Analyse ist, wie Ingenieur_innen Probleme lösen, welche Methoden und Modelle verwendet werden (Lunze, 2016).</p>
<p>Robotics</p>	<p>Unabhängige Roboter können die Montage planen und ausführen und dabei taktile sowie optische Sensoren zur Wahrnehmung ihrer Umgebung verwenden (Lunze, 2016).</p>

Die Technologie der KI besteht aus fünf Hauptbestandteilen (Paschen et al., 2019). Diese werden in Abbildung 5 vorgestellt und im Folgenden erklärt.

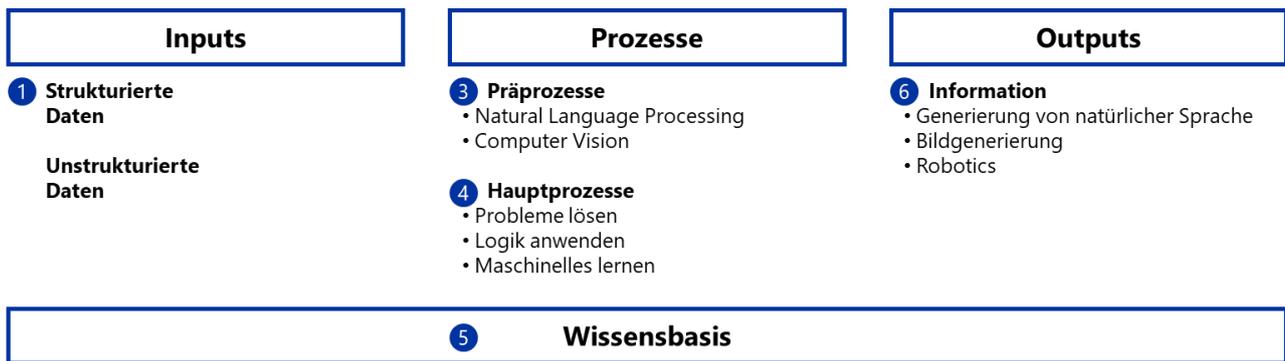


Abbildung 5. Bestandteile der Künstlichen Intelligenz (Paschen et al., 2019)

Der erste Bestandteil eines KI-Systems ist der Input. Dies sind Daten, die typischerweise in großen Mengen vorliegen müssen. Daten können entweder in strukturierter oder in unstrukturierter Form vorliegen. Strukturierte Daten sind dabei standardisiert und nach einem vordefinierten Schema organisiert (Paschen et al., 2019). Bei unstrukturierten Daten fehlen diese Standardisierung und die vordefinierte Organisation. In der Produktentwicklung können sowohl strukturierte als auch unstrukturierte Daten vorliegen. Strukturierte Daten sind beispielsweise in PLM-Systemen, Produktmodellen und Bill-of-Materials zu finden. Unstrukturierte Daten können begleitende Dokumente, Domänenwissen aus Büchern oder Daten aus externen Quellen wie Normen und Regularien sein.

Der zweite Hauptbestandteil sind die Prozesse. Diese können in die Prä- und die Hauptprozesse unterteilt werden. Das Ziel der Präprozesse ist es, unstrukturierte Daten zu strukturieren, sodass die Hauptprozesse diese weiterverarbeiten können. Natural-Language-Processing (NLP) wird eingesetzt, um die Bedeutung der menschlichen (natürlichen) Sprache in geschriebener oder gesprochener Form zu erkennen. Gesprochene Sprache wird dazu zunächst automatisiert in geschriebenen Text transkribiert. Dieser kann dann weiterverarbeitet werden. Computer-Vision wird eingesetzt, um Informationen, die als Bilder vorliegen, verständlich zu machen. Es können Objekte erkannt oder Bildbeschreibungen erzeugt werden.

Die Hauptprozesse sind die Fähigkeiten von KI, eine Logik anzuwenden, Probleme zu lösen und zu lernen. Probleme lösen bedeutet dabei, die richtige Handlung aus einem Set an Handlungsalternativen auszuwählen, um ein Ziel zu erreichen. Logik anwenden bedeutet, dass die KI aus den verfügbaren Daten Schlüsse ziehen kann. Maschinelles Lernen ist die Fähigkeit des KI-Systems, aus Erfahrung zu lernen und ohne ein vordefiniertes Regelwerk die eigenen Fähigkeiten progressiv zu verbessern.

Der dritte Hauptbestandteil eines KI-Systems ist eine Wissensbasis. Diese ist notwendig, damit das System Daten, Informationen und Wissen speichern kann, die für zukünftige Rechenoperationen

benötigt werden. Verschiedene Optionen, wie eine Wissensbasis gestaltet sein kann, werden in Kapitel 3.3 vorgestellt.

Der vierte Hauptbestandteil der KI-Systeme ist der Output. Dieser kann entweder in natürlicher Sprache, Bildform oder als Aktion der Robotik vorliegen.

2.1.2.3 Datenbanken

In diesem Unterkapitel werden verschiedene Datenbanken vorgestellt. Diese werden genutzt, um Daten, Informationen und Wissen zu speichern und für spätere Zeitpunkte verfügbar zu machen. Datenbanken sind in Form der Wissensbasis einer der Hauptbestandteile der KI. Da in der Produktentwicklung viele Daten anfallen und für KI-Systeme nutzbar gemacht werden sollen, ist die Betrachtung von Datenbanken notwendig. Im Folgenden werden verschiedene Konzepte aus der Informatik zur Realisierung von Datenbanken dargestellt.

Eine **Datenbank** ist ein System, das große Datenmengen konsistent, stabil, wiederholbar und schnell verwalten kann (Jatana et al., 2012). Generell können relationale und nichtrelationale Datenbanken unterschieden werden. Relationale Datenbanken sind vorteilhaft, wenn es um Zuverlässigkeit, Flexibilität, Robustheit und Skalierbarkeit geht, aber um den Bedürfnissen moderner Anwendungen gerecht zu werden, bei denen große Datenmengen im Allgemeinen unstrukturiert vorliegen, sind nichtrelationale Datenbanken besonders nützlich (Jatana et al., 2012). Im Folgenden werden zunächst relationale und dann nichtrelationale Datenbanken vorgestellt.

Relationale Datenbanken sind der Standard und im Sprachgebrauch ist eine relationale Datenbank gemeint, wenn von einer Datenbank gesprochen wird. Sie sind für homogene und gut strukturierte Daten geeignet (Henkel et al., 2015). Daten werden in Tabellenform gespeichert, jede Spalte und Zeile ist klar benannt, sodass direkt ersichtlich ist, was ein einzelner Datenpunkt aussagt. Dies bedeutet, dass die Struktur der Datenbank vor der Speicherung der Daten definiert werden muss. Eine spätere Erweiterung um zusätzliche Spalten oder eine Umbenennung von Spalten ist nur mit großem Aufwand möglich, da alle bereits gespeicherten Daten davon betroffen sind. Daten können dabei thematisch geordnet in verschiedenen Tabellen gespeichert werden. Wenn Beziehungen zwischen den Daten bestehen und benötigt werden, müssen diese Tabellen über eine ‚Join‘-Beziehung miteinander verbunden werden.

Graph-Datenbanken sind eine mögliche Ausprägung von nichtrelationalen Datenbanken (Jatana et al., 2012). Sie fokussieren auf die Beziehungen zwischen den Daten (Rawat & Kashyap, 2017). Graph-Datenbanken bestehen aus ‚Nodes‘ und ‚Relationships‘ (Vicknair et al., 2010). Dabei werden die Daten als Nodes gespeichert, die als Kreise visualisiert werden können. Die Beziehungen der Daten werden als Relationships gespeichert, die als Linien zwischen den Kreisen visualisiert werden. Sowohl Nodes als auch Relationships können quantitative und qualitative Eigenschaften haben. Graph-

Datenbanken haben kein fixes Schema, es können also weitere Daten während der Nutzungsphase hinzugefügt werden. Die Untersuchung der Beziehungen zwischen den Daten ist in einer Graph-Datenbank ohne großen Rechenaufwand möglich (Vicknair et al., 2010). Da diese Datenbanken auch horizontal skaliert werden können, bieten sie eine Lösung für die zunehmende Menge an Daten, die heute im Internet gespeichert werden müssen.

2.2 Herausforderungen

In diesem Unterkapitel werden die Herausforderungen beschrieben, die sich aus aktuellen Trends und der in Kapitel 1.1 vorgestellten technologischen Entwicklung ergeben und in der Produktentwicklung zu bestehen sind.

Herausforderung 1: Veränderung der Produkte

Die Veränderung der Produkte hin zu Advanced Products geht mit Herausforderungen in der Produktentwicklung einher. Advanced Products sind hochkomplexe Systeme, die meist ein mechatronisches Produkt mit weiteren Services verbinden, um die Marktleistung für den Kunden zu erbringen (Dumitrescu et al., 2021). Während der Nutzungsphase können Advanced Products in einem System of Systems eingesetzt werden, was die Komplexität weiter steigert. Die wechselseitigen Abhängigkeiten zwischen den einzelnen Subsystemen, die das Advanced Product ergeben, sind auch mit modernen Entwicklungsmethoden wie MBSE nicht durch Menschen allein versteh- und beherrschbar. Daher müssen die Werkzeuge, die in der Produktentwicklung eingesetzt werden, sich entsprechend weiterentwickeln, um die Unterstützung bieten zu können, die für die Entwicklung von Advanced Products benötigt wird.

Herausforderung 2: Steigender Softwareanteil

Da der Softwareanteil in Advanced Products im Vergleich zu herkömmlichen Produkten stark gestiegen ist und die Unterstützung von Ingenieur_innen in der Produktentwicklung durch IT-Systeme an Relevanz gewinnt, verschwimmen die Grenzen zwischen den traditionellen Feldern des Maschinenbaus und der Informatik. Ein gemeinsamer Bereich, bei dem sich die Felder treffen, ist die Modellierung. Systemmodellierung ist eine Kerntätigkeit der Produktentwicklung (ISO 15288, 2015), nutzt aber Modellierungssprachen, die in der Informatik entwickelt wurden (OMG, 2020). Außerdem sind alle IT-Systeme, die in der Produktentwicklung eingesetzt werden, auf die Systemmodelle als Grundlage angewiesen. Auch neue Technologien wie KI benötigen als ein Kernelement eine Wissensbasis. Systemmodelle aus der Produktentwicklung können einen Teil einer Wissensbasis bilden, die benötigt wird, wenn KI in der Produktentwicklung eingesetzt werden soll. Daher rücken die Felder Maschinenbau und Informatik immer stärker zusammen. Diese Entwicklung hat weitreichende Konsequenzen. Ingenieur_innen müssen sich intensiver mit den Themen der

Informatik befassen, da ihre Arbeit zu immer größeren Teilen aus der Bedienung und Weiterentwicklung von IT-Systemen besteht. Da auch die Produkte einen zunehmend größeren Softwareanteil haben, ist die Zusammenarbeit mit Kolleg_innen, die Software erstellen, ebenso ein immer größerer Bestandteil der täglichen Arbeit. Dadurch ändert sich die Rollenbeschreibung von Ingenieur_innen und in der universitären Ausbildung muss ein stärkerer Fokus auf Themen der Informatik wie Modellierung, Programmierung und Verständnis für Datenbanken gelegt werden.

Herausforderung 3: Steigerung der Menge an relevanten Daten

Durch die Digitalisierung der Produktentwicklung steigt die Menge der anfallenden Daten (Bitzer et al., 2017). Da in der Produktentwicklung aber eine Vielzahl an spezialisierten Anwendungen der einzelnen Engineering-Domänen Mechanik, Elektrik/Elektronik und Software eingesetzt wird, liegen diese Daten in verschiedenen Systemen. Jedes dieser Systeme hat sein eigenes Datenmodell, eine eigene Datenbank zur Ablage der Daten sowie seine eigene Logik und Funktionsweise. Übergreifende Systeme wie PLM schaffen eine Möglichkeit der Ablage der Daten in einem gemeinsamen System, allerdings fokussiert die Struktur innerhalb der Systeme noch die Mechanik und Elektrik/Elektronik.

In der aktuellen Entwicklung moderner Methoden der Produktentwicklung wie MBSE, LCE und LCSA liegt der Fokus auf dem gesamten Produktsystem und dem gesamten Lebenszyklus. Dafür werden spezialisierte Modelle als digitale Repräsentation der Realität erstellt, die Ingenieur_innen bei der Erfüllung ihrer Aufgaben unterstützen. So werden durch MBSE mit der Sprache SysML Modelle des Produktsystems erzeugt, die allerdings nicht für die Fragestellungen des LCE oder des LCSA wiederverwendet werden. Es gibt aktuell keine ausreichende Durchgängigkeit der Modellierung und keine ausreichende standardisierte Möglichkeit, Modelle zu verknüpfen und für verschiedene Einsatzgebiete zu nutzen.

Herausforderung 4: Struktur der Daten

Neben den strukturiert vorliegenden Daten in PLM-Systemen oder Modellen aus MBSE und LCE gibt es in der Produktentwicklung viele Informationen, die nur unstrukturiert vorliegen. Diese können allerdings essenzielle Informationen beinhalten, ohne die die Produktentwicklung langsamer oder mit erhöhtem Risiko abläuft. Die verringerte Geschwindigkeit resultiert dabei aus einem erhöhten manuellen Aufwand bei der Suche nach Informationen und die Erhöhung des Risikos ergibt sich durch ein mögliches Übersehen von kritischen Informationen. Diese Informationen können Expertenwissen umfassen, das die erfahrenen Mitarbeiter_innen aufweisen. Ein weiteres Beispiel ist frei verfügbares Wissen aus dem Internet. Es gibt zahlreiche, zum Teil wertvolle Informationen in Artikeln, Blogs und Diskussionen in Gruppen, in denen sich Ingenieur_innen austauschen. Ferner liegen Informationen im Google Knowledge Graph oder Wikimedia Knowledge Graph. Weitere

Informationen werden von Non-Government-Organisationen wie der World Health Organization (WHO) oder der Weltbank bereitgestellt. Diese Datenquellen sind aktuell nicht miteinander verbunden und formalisiert. Die Verknüpfung der Informationen findet in den Gedanken der Menschen statt, die Produktdaten kennen, öffentliche Quellen gelesen haben und daraus Schlüsse ziehen. Eine formalisierte Unterstützung bei der Auswahl der wichtigsten Quellen kann so nicht gewährleistet werden. Daher sind die Ergebnisse nicht reproduzierbar und hängen stark vom Individuum ab.

Eine weitere Herausforderung ist die unterschiedliche Struktur der Daten. Die Daten und Informationen in den verschiedenen Systemen und Modellen sind nicht unmittelbar verknüpf- und kombinierbar, da die Struktur der Daten zu weit voneinander abweicht. Dies erschwert die Erstellung einer gemeinsamen Datenbasis, da unterschiedliche Strukturen zunächst harmonisiert werden müssen und neue Quellen, die im Laufe der Anwendung hinzugefügt werden sollen, die Datenstruktur wieder grundlegend ändern können.

Herausforderung 5: Regulierung der Medizintechnik

Eine weitere Herausforderung, vor allem in der Medizintechnik, ist die hohe Anzahl der Normen und Regularien, die erfüllt werden müssen, um ein Produkt zuzulassen und damit auf den Markt bringen zu können. Die Normen und Regularien liegen in Form von Dokumenten vor. Der Einfluss auf die Produkte muss durch menschliche Fachexpert_innen analysiert werden. Auch bei einer Aktualisierung der Regulierung muss der Einfluss der Änderung manuell untersucht werden. Die Regulierung in verschiedenen Ländern und Regionen kann sich dabei stark unterscheiden.

Herausforderung 6: Ökonomische Betrachtung

Die sechste Herausforderung ist eine ökonomische Betrachtung der Aktivitäten. Die Schaffung eines Systemmodells, auf deren Grundlage KI-Systeme zum Einsatz kommen können, um Effizienzgewinne zu erreichen, erfordert ein erhebliches initiales Investment. Aktuell gibt es noch keine standardisierten und industrialisierten Lösungen am Markt, die den Aufbau des Systemmodells erleichtern oder sogar automatisieren können. Daher ist dieses Investment in der aktuellen Situation vor allem für größere Unternehmen interessant. Eine weitere Option bilden Unternehmen, die als Technologieführer auf die Beherrschung hochkomplexer Systeme spezialisiert sind. Für sie könnten ein solches Systemmodell und der dadurch realisierbare Einsatz von KI-Technologie eine Möglichkeit sein, ihre Marktposition durch innovativere Produkte, die für andere Unternehmen nur schwer bis unmöglich zu replizieren sind, zu stärken. Kleineren Unternehmen oder Unternehmen mit Produkten niedriger bis mittlerer Komplexität ist zu empfehlen, abzuwarten, bis sich standardisierte Lösungen am Markt etabliert haben, die zu niedrigeren Preisen eingekauft werden können.

Herausforderung 7: Voraussetzungen für KI-Einsatz

Das Fehlen einer Datenbasis, die alle Engineering-Objekte und Systemmodelle mit frei verfügbaren Informationen verbindet, führt zum Fehlen eines der Kernelemente der KI-Technologie. Diese kann daher nur eingeschränkt eingesetzt werden. Der Einsatz von KI besitzt das Potenzial, einen erheblichen Beitrag dabei zu leisten, die steigende Komplexität beherrschbar zu machen und Ingenieur_innen bei der Produktentwicklung zu unterstützen. Daher muss diese Datenbasis erforscht und geschaffen werden, bevor ein Einsatz von KI in der Produktentwicklung sinnvoll möglich ist.

2.3 Anforderungen an ein KI-fähiges Systemmodell

Um Ingenieur_innen bei der Bewältigung der vorgestellten Herausforderungen zu unterstützen, sollen KI-Systeme eingesetzt werden. Allerdings zeigt bereits Herausforderung 7, dass für den Einsatz von KI-Technologie verschiedene Voraussetzungen erfüllt sein müssen. Eine davon ist das Vorhandensein einer Wissensbasis. In diesem Kapitel werden Anforderungen an ein Systemmodell aus der Produktentwicklung abgeleitet, das als Wissensbasis dienen kann. Dazu wird zunächst ein Framework zur systematischen Ableitung der Anforderungen vorgestellt, bevor dieses Framework angewandt wird, um die Anforderungen an das Systemmodell dieser Arbeit abzuleiten.

2.3.1 Framework zur Ableitung von Anforderungen

Zur systematischen Ableitung von Anforderungen wurde ein Framework entwickelt (Schweitzer et al., 2021). Dieses wird im Folgenden vorgestellt und am Beispiel einer Dialysemaschine angewandt. Abbildung 6 stellt das Framework dar.

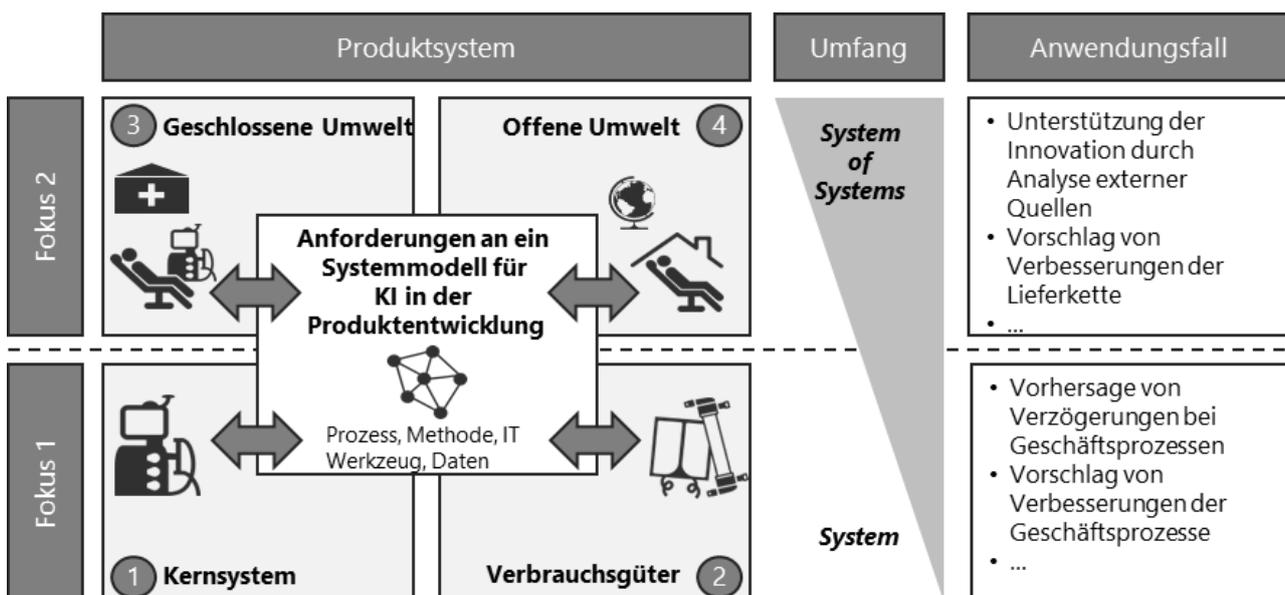


Abbildung 6. Framework zur Ableitung von Anforderungen der Künstlichen Intelligenz an ein Systemmodell (Schweitzer et al., 2021)

Die erste Spalte des Frameworks stellt vier mögliche Ebenen des Produktsystems dar. Dieses ist das System of Interest, das als Ergebnis der Produktentwicklung erstellt werden soll. Es kann ausschließlich auf die Dialysemaschine selbst als mechatronisches System fokussiert werden. Als zweite Stufe können die Verbrauchsgüter mit in Betracht gezogen werden, die zur Durchführung einer Therapie zusätzlich zur Dialysemaschine notwendig sind. Diese beiden Stufen bilden den Fokus 1, bei dem die Produkte betrachtet werden, die von einem Unternehmen produziert und angeboten werden.

Die nächsten Stufen sind in der oberen Zeile, dem Fokus 2, zu sehen. Dieser bezieht die Umwelt, in der die Dialysemaschine und die Verbrauchsgüter eingesetzt werden, mit ein. Bei Stufe 3 ist es noch eine geschlossene Umwelt, die gut kontrolliert werden kann. Dies ist zum Beispiel bei einem Einsatz in der Klinik gegeben. Hier können viele Umwelteinflüsse durch die Betreibenden kontrolliert werden. Stufe 4 stellt dann eine offene Umwelt dar. Dies bedeutet, dass das Produkt frei in der Welt eingesetzt werden kann. Dies könnte eine Umgebung bei den Patienten zu Hause sein, es ist aber auch denkbar, dass ein mobiles Gerät im Urlaub an verschiedenen Orten unter unterschiedlichen Umwelteinflüssen eingesetzt wird (Fresenius Medical Care AG & Co. KGaA, 2023b).

In der Mitte des Produktsystems steht das Systemmodell als Wissensbasis. Diese besteht aus den Kernelementen Prozess, Methode, IT-Werkzeug und Daten. Der benötigte Umfang dieses Systemmodells hängt von der gewählten Stufe ab, in der das Produktsystem eingesetzt werden soll. Die zweite Spalte ist der Umfang, der betrachtet werden soll. Er ergibt sich aus dem gewählten Fokus und der Stufe. Die Stufen 1 und 2 im Fokus 1 betrachten das System an sich, ohne auf externe Umwelteinflüsse sowie weitere Produkte anderer Hersteller, mit denen das Produkt interagieren soll und muss, einzugehen. Die Stufen 3 und 4 im Fokus 2 beziehen diese Umwelt und Produkte anderer Hersteller mit ein. Es kann daher von einem System of Systems gesprochen werden.

Welcher Fokus zu wählen ist, hängt davon ab, welcher Anwendungsfall durch das Systemmodell befähigt werden soll. Sollen ausschließlich Anwendungsfälle mit internem Fokus realisiert werden, reicht ein Systemmodell auf Fokus 1 mit den Stufen 1 oder 2 aus. Diese Anwendungsfälle können die Berechnung von Durchlaufzeiten im Entwicklungsprozess oder die Vorhersage von Verzögerungen sein. Sollen Anwendungsfälle realisiert werden, die zusätzlich mit unternehmensexternen Aspekten zu tun haben, wird ein Systemmodell auf Fokus 2 mit den Stufen 3 oder 4 benötigt. Diese Anwendungsfälle können die Unterstützung von Innovation durch Analyse externer Quellen wie Ausschreibungsdokumente oder der Vorschlag von Verbesserungen der Lieferkette sein.

2.3.2 Ableitung der Anforderungen an ein KI-fähiges Systemmodell

In diesem Unterkapitel werden die Anforderungen an ein KI-fähiges Systemmodell für diese Arbeit abgeleitet. Dazu wird das in Kapitel 2.3.1 vorgestellte Framework angewendet.

Für diese Arbeit gilt dabei, dass der größtmögliche Betrachtungsrahmen eingenommen werden soll. Produktentwicklung soll holistisch unterstützt werden, daher muss die gesamte Komplexität des Problems betrachtet werden. Der Anwendungsfall liegt somit im Fokus 2 und im Bereich der offenen Umwelt. Als Umfang wird das System of Systems gewählt.

Im ersten Schritt sollen die vorhandenen Daten aus der Produktentwicklung, die in verschiedenen Applikationen und Datenbanken vorliegen, miteinander verbunden werden. Daraus ergibt sich Anforderung 1:

A 1: Das Systemmodell soll verschiedene Datenbanken aus dem Bereich Produktentwicklung (zum Beispiel PLM, ERP, ALM, IoT etc.) miteinander verbinden können.

Neben den Datenbanken, in denen die Produktdaten vorliegen, sollen zusätzlich Daten aus den Modellen genutzt werden können. Dazu müssen diese zunächst untereinander vernetzt werden. Daraus ergibt sich Anforderung 2:

A 2: Das Systemmodell soll verschiedene Modelle (zum Beispiel MBSE, Simulation etc.) aus dem Bereich Produktentwicklung miteinander verbinden können.

Im nächsten Schritt sollen die Produktdaten aus den Datenbanken und die Modelle miteinander verbunden werden. Daraus ergibt sich Anforderung 3:

A 3: Das Systemmodell soll Modelle und Datenbanken miteinander verbinden können.

Die ersten drei Anforderungen beziehen sich auf den Fokus 1 und damit auf die Daten, die bereits innerhalb eines Unternehmens vorliegen. Diese zu vernetzen und damit nutzbar zu machen, ist essenziell, um das Systemmodell unternehmensspezifisch zu gestalten und um eine Grundlage zu schaffen, an die weitere Informationen und Daten angehängt werden können.

Anforderung 4 bezieht sich auf den Fokus 2, es werden zusätzlich Daten und Informationen aus der Umwelt hinzugefügt. Diese sollen mit den bestehenden Daten verbunden werden können.

A 4: Das Systemmodell soll unstrukturierte, frei verfügbare Daten (zum Beispiel Normen, Ausschreibungsdokumente, Fachartikel) mit den bestehenden Daten verbinden können.

Neben frei verfügbaren Informationen gibt es bestehende Wissensgraphen, die zum Teil frei verfügbar sind und bereits große Teile des semantischen Webs und damit des Kontextes, in dem Produkte entwickelt sowie eingesetzt werden, abdecken. Diese sollen ebenfalls zum Systemmodell hinzugefügt und mit bestehenden Daten verbunden werden können.

A 5: Das Systemmodell soll frei verfügbares Kontextwissen (zum Beispiel Wikimedia Knowledge Graph) mit den bestehenden Daten verbinden können.

Die Verbindungen zwischen den einzelnen Daten und Informationen sollen als semantische Beziehungen abgebildet werden können. Dies ermöglicht eine semantische Analyse der Beziehungen.

A 6: Das Systemmodell soll semantische Beziehungen zwischen den Daten abbilden können.

Um eine Zukunftssicherheit zu gewährleisten, soll das Systemmodell durch strukturierte und unstrukturierte Daten jedweder Art erweitert werden können. Dies ermöglicht, dass neu veröffentlichte Quellen oder Quellen, deren Relevanz erst später klar wird, jederzeit zum Systemmodell hinzugefügt werden können. Daraus ergibt sich Anforderung 7:

A 7: Das Systemmodell soll um strukturierte und unstrukturierte Daten flexibel und dynamisch erweiterbar sein.

Werden die in diesem Kapitel vorgestellten Anforderungen erfüllt, ergibt sich ein Systemmodell, das sowohl die Produktdaten aus dem Unternehmen als auch relevante frei verfügbare Daten abbildet. Es liegen also sowohl die Daten über das Produkt als auch die Daten über den Kontext, in dem es entwickelt und eingesetzt wird, vor. Dies ermöglicht eine maschinelle Unterstützung von Ingenieur_innen bei der Beherrschung der Komplexität.

3 Stand der Technik

In diesem Kapitel wird der Stand der Technik in Wissenschaft und Praxis für die Konzepte und Methoden vorgestellt, die als Grundlage für Systemmodelle in den Ingenieurwissenschaften dienen oder die Datenverarbeitung in der Informatik beschreiben. Daraus wird der Handlungsbedarf abgeleitet, der in dieser Arbeit adressiert wird.

3.1 Ingenieurwissenschaften

In diesem Unterkapitel wird der Stand der Technik der Ingenieurwissenschaften dargestellt, der für ein Verständnis der Ablage und Verwaltung von Produktdaten und der Erstellung von Systemmodellen wichtig ist. Dies umfasst das PLM, MBSE und LCE. Es wird jeweils der Stand der Wissenschaft und der Stand der Praxis beschrieben.

3.1.1 Produktlebenszyklusmanagement

Eine Definition für den Lebenszyklus und PLM für diese Arbeit wurde in Kapitel 2.1.1.3 präsentiert. In diesem Unterkapitel wird beschrieben, wie PLM funktioniert und welchen Mehrwert es für Unternehmen bietet. Dafür wird auf Theorie und Praxis eingegangen.

3.1.1.1 *Stand der Wissenschaft*

Das Ziel von PLM ist es, alle Informationen zu einem Produkt über dessen gesamten Lebenszyklus, d. h. angefangen von den frühen Phasen der Anforderungsanalyse und Produktkonzeption über die Entwicklung und Produktion bis hin zur Nutzungs- bzw. Betriebsphase konsistent zu verwalten und für alle zukünftigen Aufgaben im Produktentstehungsprozess im Sinne von Wissensmanagement bereitzustellen (Gerhard, 2019). Das PLM verwaltet alle Daten aus Entwicklung, Produktion, Lager und Vertrieb und unterstützt die sogenannte Single Source of Data über den gesamten Lebenszyklus (Bracht et al., 2018). Dies bedeutet, dass jedes Datum nur einmal an einer Stelle vorliegt und an anderen Stellen, an denen die Information benötigt wird, darauf referenziert wird. Die gesamte Produktpalette wird abgedeckt, vom Einzelteil bis zum gesamten Produktportfolio (Terzi et al., 2010). Die fundamentalen Elemente von PLM sind Methodik, Prozesse und IT-Systeme (Terzi et al., 2010). Diese werden in den folgenden Absätzen erklärt.

Mit der Entwicklung von 2D-CAD- zu 3D-CAD-Systemen vergrößerten sich die Möglichkeiten, die Ingenieur_innen zur Verfügung stehen (Gerhard, 2019). Die Menge an Daten stieg rapide an und die Möglichkeit der kollaborativen Zusammenarbeit an einzelnen Teilen oder Baugruppen machte neue Methoden zur Datenverwaltung sowie Regelung der Zugriffsrechte und Zusammenarbeit notwendig.

Um diese Methoden in der Praxis einsetzen zu können, wurden abgestimmte Ablaufschritte definiert, die von Ingenieur_innen eingehalten werden sollen (Gerhard, 2019). Diese Prozesse sind

unternehmensspezifisch und ermöglichen, dass die Produktentwicklung nicht subjektiv von der handelnden Person abhängt. Dadurch können die Kollaboration und der Austausch von Produktdaten realisiert werden.

Auf der IT-Ebene werden die neuen Methoden durch moderne Computer-aided-Design (CAD)-, Computer-aided-Manufacturing (CAM)- und Computer-aided-Engineering (CAE)-Systeme sowie entsprechende Simulations- und Visualisierungstechniken unterstützt. PLM-Lösungen bilden den funktionalen und administrativen Backbone. Sie sind Ende der 1990er Jahre aus einer Erweiterung von PDM-Systemen entstanden (Eigner & Stelzer, 2013).

In diesem Umfeld gewinnt der strategische Ansatz PLM zunehmend an Bedeutung und kann als Erweiterung der langjährigen Aktivitäten im Produktdatenmanagement gelten. Mit einer systemtechnischen Lösung für PLM können Informationen erfasst oder über sogenannte Erzeugersysteme, zum Beispiel CAD-, CAE-, CAM- und Office-Systeme, automatisch übernommen, individuell aufbereitet, abgerufen, administriert, analysiert und weitergeleitet werden. Das System kann durch flexibles Customizing dem Produkt- und Prozessmodell des jeweiligen Unternehmens angepasst werden.

Die verstärkt einsetzende Nutzung von 3D-CAD-Systemen in der Produktentwicklung hatte noch zwei weitere unmittelbare Folgen: erstens ein stark ansteigendes Datenvolumen und zweitens durch die Möglichkeit der Zusammenarbeit an Baugruppen die Notwendigkeit, diese Zusammenarbeit mit entsprechenden Zugriffsrechten auf die Daten zu steuern und zu regeln. Aus dieser Notwendigkeit heraus gab es Ende der 1990er Jahre einen enormen Entwicklungsschub bei den bereits einige Jahre zuvor am Softwaremarkt erschienenen Zeichnungs- oder Dokumentenverwaltungssystemen in Richtung Produktdatenmanagement (PDM-Systeme). Während Erstere als Archive für digitale Zeichnungsdateien dienten, wurde mit den PDM-Systemen eine neue Kategorie technischer Informationssysteme begründet (Gerhard, 2019).

Typische Fähigkeiten und Funktionalitäten eines PLM-Systems sind (1) Stamm und Struktur, (2) Dokumentenmanagement, (3) Gruppentechnik, (4) Projektmanagement, (5) Workflow-Management, (6) Freigabe- und Änderungsmanagement, (7) Viewing, Redlining, Digital Mock-up, (8) Input- und Output-Management, (9) Publishing, (10) Archivierung und Backup, (11) Daten-Replikation und (12) Integrationen (Eigner & Stelzer, 2013). Diese Funktionalitäten zielen darauf ab, die Produktdaten mit den Projektdaten und den beschreibenden Dokumenten in Beziehung zu setzen sowie gemeinsam verwalten und analysieren zu können.

3.1.1.2 Stand der Praxis

PLM-Systeme sind in der industriellen Praxis weit verbreitet: 51 % der produzierenden Unternehmen gaben an, dass sie eine niedrige bis mittlere Implementierung von PLM haben, 8 % haben sogar eine extensive Implementierung (Koudal, 2005).

Produktlebenszyklusmanagement wird von Unternehmen nicht im gesamten Produktlebenszyklus eingesetzt (Abramovici & Schulte, 2004): 100 % der Unternehmen, die ein PLM-System im Einsatz haben, nutzen dieses in der Phase des Produktdesigns. Abbildung 7 zeigt eine Übersicht, in welchen Lebenszyklusphasen die teilnehmenden Unternehmen ihr PLM-System einsetzen. Mehrfachnennungen waren dabei möglich.

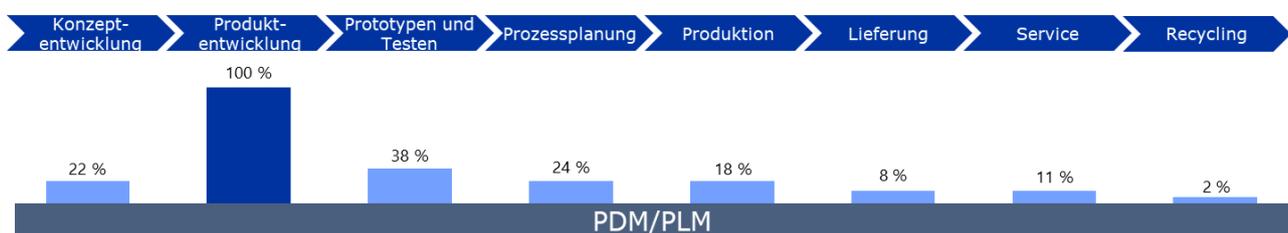


Abbildung 7. Einsatz von Produktlebenszyklusmanagement nach (Abramovici & Schulte, 2004)

Die größten Anbieter für PLM-Software sind Dassault Systemes vor Autodesk und Siemens Digital Industries Software (Pang, 2022). Diese großen Hersteller bieten nicht nur PLM-Software an, sondern auch CAD-Design-, Modellierungs- und Simulationswerkzeuge. Die gesamte Software für die Produktentwicklung ist oft zu einer Toolchain verbunden. Dies bedeutet, dass die einzelnen spezialisierten Systeme herstellerseitig so gestaltet wurden, dass sie gut zusammenarbeiten. Daten können daher ausgetauscht werden, die Datenformate sind aufeinander abgestimmt und die Schnittstellen wurden bereits vorkonfiguriert. Diese Toolchains ermöglichen Unternehmen die Wahl, ob sie alle Entwicklungswerkzeuge von einem Hersteller beziehen möchten, um von den vorgedachten Schnittstellen zu profitieren, oder ob sie einen ‚Best-of-Breed‘-Ansatz wählen, bei dem für jede Aktivität das optimale Werkzeug ausgewählt wird, die Schnittstellen dann aber selbst hergestellt werden müssen.

Konzepte wie Semantic Product-Data-Management (PDM) sollen zukünftige Herausforderungen bewältigen und so Marktanteile gewinnen (Semantic PDM GmbH & Co. KG, 2023). Ein Semantic PDM-System ist ein PLM-System, das nicht auf relationale Datenbanken, sondern auf Graph-Datenbanken beruht. Dadurch besteht die Hoffnung, die Vernetzung der Daten besser darstellen zu können.

Der große Funktionsumfang von PLM-Systemen bringt es mit sich, dass ihre Einführung häufig eine Vielzahl von kleineren Lösungen ablöst. Daher ist die Einführung mit hohem Aufwand und einer großen Änderung der Arbeitsweise verbunden. Nach der Einführung benötigen PLM-Systeme

weiterhin Aufmerksamkeit, da sie gepflegt und aktualisiert werden müssen. Außerdem können neue Projekte neue Herausforderungen und Anforderungen an das PLM-System mit sich bringen, die dann im PLM-System umgesetzt werden müssen. Es ist immer eine Aufwandsanalyse durchzuführen, die ergeben soll, ob sich die Einführung eines PLM-Systems lohnt. Um hier vor allem kleineren und mittelständischen Unternehmen entgegenzukommen, gibt es diverse Open-Source-Optionen (Aziz et al., 2005; Gopsill et al., 2011). Diese bieten die Möglichkeit, mit geringeren Kosten ein PLM-System einführen und betreiben zu können.

3.1.2 Model-based-Systems-Engineering

Model-based-Systems-Engineering ist eine wichtige Methode zur Beherrschung der Komplexität in der Produktentwicklung. Eine Definition wird in Kapitel 2.1.1.2 vorgestellt. Teil der Methode ist die Modellierung, bei der das Produkt systematisch in einer menschen- und maschinenlesbaren Sprache modelliert wird. Das so erstellte Modell kann eine Grundlage für ein KI-fähiges Systemmodell darstellen, daher wird in diesem Unterkapitel der Stand der Wissenschaft und der industriellen Praxis von MBSE vorgestellt.

3.1.2.1 *Stand der Wissenschaft*

Der transdisziplinäre und integrative Ansatz des MBSE ermöglicht die erfolgreiche Realisierung, Nutzung und Stilllegung von technischen Systemen (Sillitto et al., 2019). Er hat sich aus dem dokumentenzentrierten Ansatz der 1960er Jahre herausgebildet, um die Produktdaten aus dem gesamten Lebenszyklus (wieder-)verwenden zu können und die Anforderungen der Stakeholder zu erfüllen (Estefan, 2007).

Model-based-Systems-Engineering deckt alle Prozesse des Systemlebenszyklus ab: Vereinbarungs- und organisatorische Projekteinführungsprozesse, technische Managementprozesse und technische Prozesse selbst (ISO 15288, 2015). Ein Kernelement von MBSE ist die Dekomposition eines System of Interest in Teilsysteme. Diese Teilsysteme können in Systemelementen ausgearbeitet werden. Ein System of Interest kann auch durch sein betriebliches Umfeld und seine unterstützenden Systeme beschrieben werden.

Diese Elemente können ausschließlich einzeln entwickelt und getestet werden, bevor sie wieder zum gewünschten Gesamtsystem zusammengesetzt werden. Diese Vorgehensweise spiegelt sich im V-Modell wider. Abbildung 8 zeigt eine Darstellung des V-Modells.

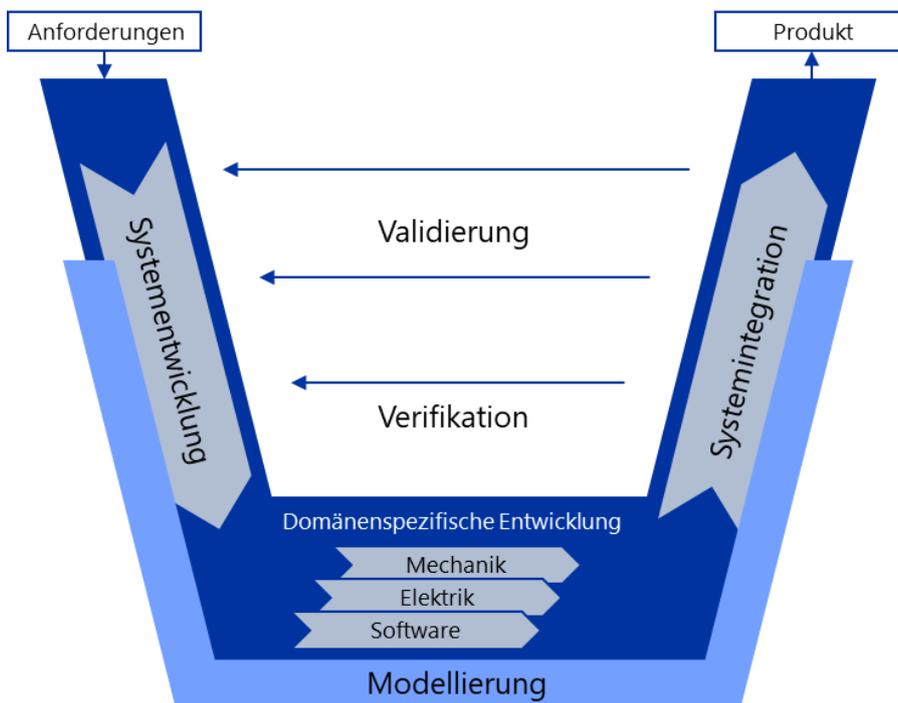


Abbildung 8. V-Modell nach (Gausemeier & Moehring, 2002)

Zur Beschreibung dieser Elemente und Beziehungen wird ein modellbasierter Ansatz verwendet, der in der Regel durch die Modellierungssprache System Modeling Language (SysML) unterstützt wird. Die SysML, die von der Object Management Group (OMG) in Zusammenarbeit mit dem International Council of Systems Engineering (INCOSE) entwickelt wurde, basiert auf den Sprach- und Formalismuskonzepten der Unified Modeling Language (UML) (Eigner, 2016; Korthals et al., 2020). Die UML wird dabei über drei Mechanismen angepasst: Stereotypen, Tagged Values und Constraints (OMG, 2020). SysML-Modelle beschreiben Systeme durch Anforderungen, Struktur und Verhalten auf eine abstrakte Weise. Die Sprache und die Diagramme sind in einer Weise formalisiert, die nicht nur von Maschinen, sondern auch von menschlichen Ingenieur_innen verstanden werden kann (Korthals et al., 2020).

Zur Erzeugung der Systemmodelle wird ein Zusammenspiel aus Modellierungsmethode, -sprache und -werkzeug benötigt. Abbildung 9 stellt dieses MBSE-Dreieck dar.

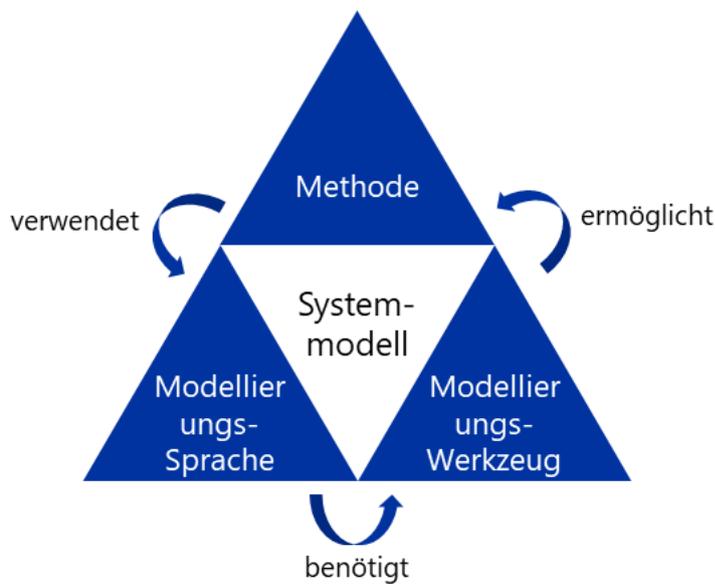


Abbildung 9. MBSE-Dreieck nach (Dumitrescu et al., 2015)

Das Zusammenspiel der Elemente des MBSE-Dreiecks wird in Kapitel 3.3.1 vorgestellt. Dort wird genauer auf die Modellierungsmethode und die Modellierungssprachen eingegangen.

In der aktuellen Literatur wird eine Weiterentwicklung des MBSE hin zu Advanced-Systems-Engineering diskutiert (Dumitrescu et al., 2021). Advanced-Systems-Engineering ist die Kombination aus Advanced Systems, Systems-Engineering und Advanced Engineering. Advanced Systems sind die Marktleistung, die in Zukunft erbracht werden soll. Systems-Engineering ist dazu geeignet, die Komplexität der Advanced Systems zu beherrschen, und Advanced Engineering umfasst grundlegende Änderungen im Engineering, die durch den Einsatz aktueller Erkenntnisse in und zwischen den ingenieur- und wirtschaftswissenschaftlichen sowie informationstechnologischen Disziplinen entwickelt wurden.

Die Dekomposition des System of Interest in einzelne handhabbare Elemente kann auch Ebenen enthalten, die weit über das eigentliche Produkt hinausgehen, und den Kontext, in dem das Produkt eingesetzt wird, einbeziehen. Eine Möglichkeit, wie diese Ebenen gestaltet werden können, zeigt Abbildung 10 (Bitzer et al., 2022) am Beispiel eines Therapiesystems.

Die unterste Ebene, das ‚Product-System‘, bilden die einzelnen Komponenten. Dies können ein Produkt und seine Verbrauchsgüter sein. Diese werden auf jener Ebene einzeln gestaltet, die Daten liegen klassischerweise in einem PLM-System. Auf der Ebene des ‚Therapy-Product-System‘ ist das Zusammenspiel der Komponenten modelliert, sodass alle Informationen vorliegen, die zur Bereitstellung der Marktleistung notwendig sind. Diese Informationen werden aktuell in SysML modelliert. Die Ebene des ‚Therapy-Treatment-System‘ bezieht alle Schritte mit ein, die zur Bereitstellung der Marktleistung notwendig sind, aber nicht immer durch Produkte des eigenen

Unternehmens durchgeführt werden. Dies umfasst zum Beispiel den Transport, weitere Medikation, die eine Therapie begleitet, sowie die Abläufe in der Klinik. Nach heutigem Stand werden nur Teile dieser Informationen in SysML modelliert, andere Informationen sind nicht strukturiert verfügbar. Die nächste Ebene ist das ‚Health-System‘. Dieses beinhaltet Informationen über Krankenversicherungen in den Ländern, in denen das Produkt eingesetzt werden soll, allgemeine Informationen, zum Beispiel von der WHO, und rechtliche Rahmenbedingungen. Diese Informationen liegen nur unstrukturiert vor oder sind im Web verfügbar. Die höchste Ebene ist das ‚Society-Eco-System‘. Hier sind demographische und ökonomische Daten über die Welt sowie Umwelteinflüsse enthalten. Auch diese Informationen liegen ausschließlich unstrukturiert vor (Bitzer et al., 2022).

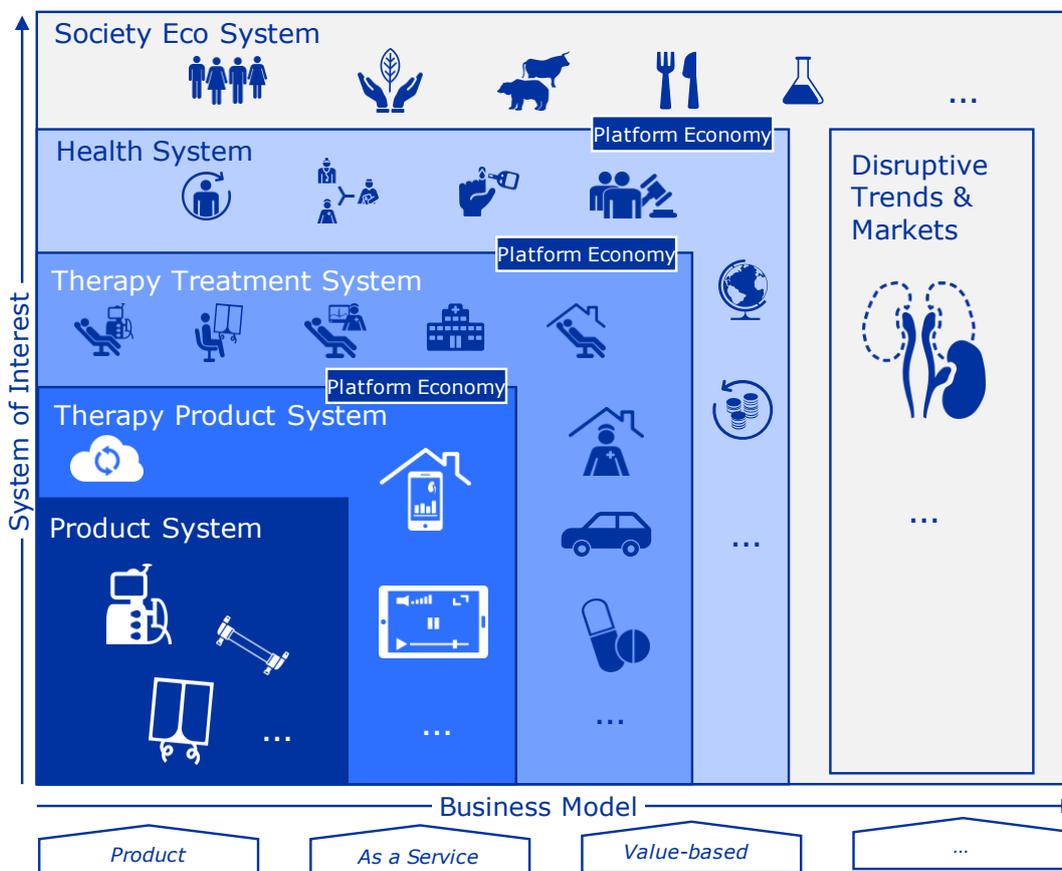


Abbildung 10. Ebenen des System-of-Interest nach (Bitzer et al., 2022)

Es liegen also Informationen auf allen Ebenen des System of Interest vor, allerdings werden diese aktuell nicht systematisch genutzt. Die Daten sind nicht maschinenlesbar vernetzt, eine Nutzung der Daten ist mit großem manuellem Aufwand verbunden.

Die Zusammenfassung und Bündelung von Aktivitäten des MBSE in Rollendefinitionen kann ein wichtiger Hebel bei der Operationalisierung von SE sein (ISO 15288, 2015). Sheard führte zwölf Rollen für SE ein, die häufig als Leitfaden zur Strukturierung der Aktivitäten der ISO 15288 verwendet werden (Sheard, 1996). Tabelle 3 fasst die Rollen zusammen.

Tabelle 3. Rollen im Systems-Engineering nach (Sheard, 1996)

#	Abk.	Name
1	RO	Requirements Owner
2	SD	System Designer
3	SA	System Analyst
4	VV	Validation / Verification Engineer
5	LO	Logistics / Ops Engineer
6	G	Glue Among Subsystems
7	CI	Customer Interface
8	TM	Technical Manager
9	IM	Information Manager
10	PE	Process Engineer
11	CO	Coordinator
12	CA	Classified Ads SE

Zur Vermeidung von Übersetzungsfehlern und Ungenauigkeiten werden hier die englischen Originalbezeichnungen der Rollen weiterverwendet. Die folgenden Absätze gehen genauer auf die Rollen ein, die für die Einbindung des in dieser Arbeit vorgestellten Konzeptes in die Produktentwicklung relevant werden.

Der System Analyst führt Simulationen durch und überprüft während der Entwicklung, ob das System die Anforderungen erfüllt. Daher muss das System modelliert werden, um den System Analyst mit Computersystemen zu unterstützen. Die Rolle ‚Glue Among Subsystems‘ sucht aktiv nach Problemen, die an den Schnittstellen der Subsysteme auftreten könnten. Dabei kann es sich um elektromagnetische Interferenzen oder Grenzwertüberschreitungen handeln. Um diese Rolle gut auszuführen, ist ein tiefes Verständnis des zu entwickelnden Systems erforderlich, das durch ein Systemmodell unterstützt werden kann. Der Information Manager ist für das Konfigurationsmanagement, die Datenverwaltung und die Metriken zuständig.

Die Rolle des System Architect wird als eine Kombination aus der Rolle des System Designer und der Rolle des Customer Interface definiert (Rechtin, 1991). Die Rolle wird hier als außerhalb des Systems-

Engineering verstanden und ist daher in der Lage, sowohl mit dem Ingenieurteam als auch mit dem Kunden zu verhandeln.

3.1.2.2 Stand der Praxis

In einer Studie aus 2018 wurden 661 Unternehmen zu ihren MBSE-Aktivitäten befragt. Die Unternehmen kamen dabei am häufigsten aus den Branchen Defense, Space-Systems, Aircraft, Automobil, IT und Medizintechnik (Cloutier, 2019a).

Abbildung 11 zeigt, in welchen Bereichen die befragten Unternehmen den größten Mehrwert von MBSE erwarten. Es ist der prozentuale Anteil der befragten Unternehmen zu sehen, die einen bestimmten Mehrwert genannt haben. Mehrfachnennungen waren dabei möglich. Die Ergebnisse zeigen, dass MBSE vor allem zur Architektur des Systems eingesetzt wird. Weitere relevante Felder sind die frühe Konzeptphase, die Anforderungsanalyse und die Simulation.

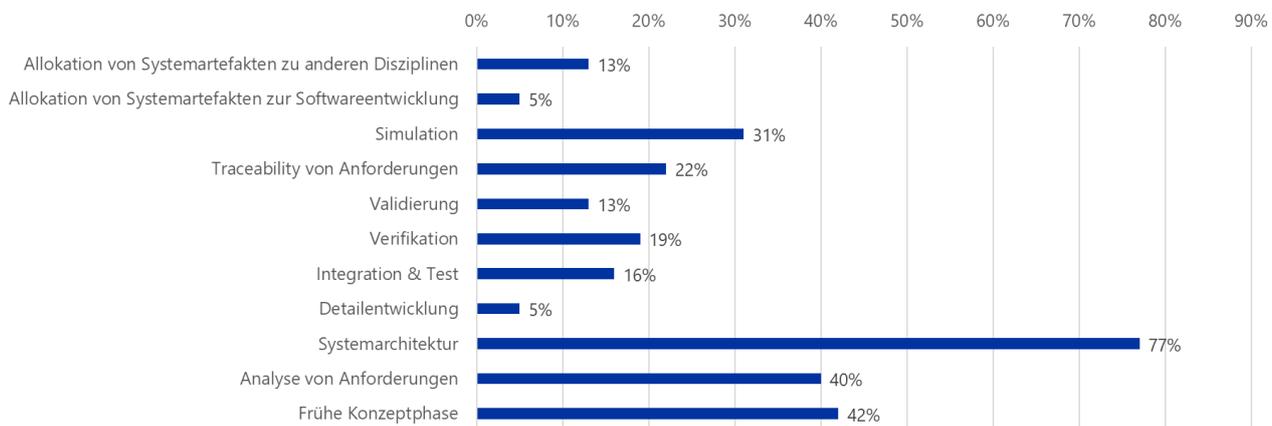


Abbildung 11. Mehrwert von Model-based-Systems-Engineering nach (Cloutier, 2019b)

3.1.3 Life-Cycle-Engineering

Life-Cycle-Engineering (LCE) rückt den gesamten Lebenszyklus eines Produktes in den Fokus der Produktentwicklung. Die Definitionen für LCE wurden in Kapitel 2.1.1.4 vorgestellt. Der Fokus auf den gesamten Lebenszyklus ist für diese Arbeit wichtig, da Produkte in allen Phasen des Lebenszyklus Daten und Informationen erzeugen, die potenziell für ein KI-fähiges Systemmodell relevant sein könnten. Daher wird in diesem Unterkapitel der aktuelle Stand des LCE in Wissenschaft und Praxis vorgestellt.

3.1.3.1 Stand der Wissenschaft

Wie bereits in Kapitel 2.1.1.4 vorgestellt, ist LCE eine Methode, um den gesamten Lebenszyklus bei der Produktentwicklung zu betrachten und Aspekte wie Nachhaltigkeit zu analysieren. Um dies zu gewährleisten, sind in den letzten Jahren verschiedene Konzepte entwickelt worden, die sich unterschiedliche Datenquellen und gestiegene Rechenleistung zu Nutze machen.

Das Integrated-Computational-Life-Cycle-Engineering (IC-LCE) integriert Daten aus dem gesamten Produktlebenszyklus über gekoppelte Modelle (Cerdas et al., 2018). Die Ergebnisse von LCE können visualisiert werden, um sie an Expert_innen und Nichtexpert_innen zu kommunizieren, indem LCE mit Visual Analytics kombiniert wird (Kaluza et al., 2018). Auch das wissensbasierte Engineering kann mit LCE kombiniert werden. Ein manueller Weg, Wissen zu entwickeln und für LCE verfügbar zu machen, wird vorgestellt (von Drachenfels et al., 2020). Darauf aufbauend wird ein Rahmenwerk zur automatischen Datenerfassung während des Lebenszyklus eines Produkts entwickelt (Dilger et al., 2021).

Sakao et al. (2021) identifizieren aktuelle Herausforderungen und Chancen von LCE und entwickeln auf der Grundlage ihrer Erkenntnisse eine Vision für Adaptive and Intelligence LCE (AI-LCE) (Sakao et al., 2021). Hier werden verschiedene Engineering-Fähigkeiten durch Business-Intelligence-Tools unterstützt, die auf einer Datenbank namens ‚Memory‘ und externen Faktoren sowie Anforderungen basieren.

In allen Studien werden Probleme der derzeitigen LCE-Methoden aufgezeigt und daraus wird der Bedarf für ein neues Konzept abgeleitet. Die ermittelten Probleme sind in Tabelle 4 zusammengefasst.

Tabelle 4. Herausforderungen im Life-Cycle-Engineering

#	Herausforderung	Quelle
1	Fehlende Geschwindigkeit	(Cerdas et al., 2018; Sakao et al., 2021)
2	Zu starke Simplifizierung der Modelle	(Cerdas et al., 2018)
3	Unvollständigkeit der Modelle	(Cerdas et al., 2018)
4	Fehlende Transparenz	(Cerdas et al., 2018)
5	Fehlende Integration der Entwicklungsumgebungen der verschiedenen Engineering-Disziplinen	(Kaluza et al., 2018)

Alle Konzepte, die zur Lösung der festgestellten Probleme eingeführt wurden, erfordern eine Art von Datenbank, Repository oder Speicher. Es gibt jedoch kein Konzept, wie diese Datenbank aufgebaut werden kann und wie bestehende Methoden wie PLM und SE in das Datenbankkonzept einbezogen werden können.

3.1.3.2 Stand der Praxis

Zur Standardisierung der Aktivitäten bei der Analyse der Nachhaltigkeit für Produkte wurden die beiden Normen 14040 „Environmental management – Life cycle assessment – Principles and framework“ und 14044 „Environmental management – Life cycle assessment – Requirements and guidelines“ erstellt (DIN EN ISO 14040, 2009; DIN EN ISO 14044, 2006). Diese bieten Unternehmen Rahmenbedingungen und konkrete Anforderungen, wie eine Nachhaltigkeitsanalyse durchzuführen ist. Die Einhaltung dieser Normen ist essenziell, um eine objektive Bewertung durchführen zu können und vergleichbare Ergebnisse zu erhalten.

Bei der Durchführung eines LCSA sollen die drei Säulen Umweltauswirkungen, soziale Auswirkungen und ökonomische Auswirkungen analysiert werden (Benoît & Mazijn, 2013; Benoît Norris et al., 2020). Die aktuellen Ansätze erfüllen dabei nicht alle Anforderungen. Tabelle 5 gibt einen Überblick über die Herausforderungen bei der Durchführung eines LCSA.

Tabelle 5. Herausforderungen bei einem Life-Cycle-Sustainability-Assessment

#	Herausforderung	Quelle
1	Aktuelle Ansätze betrachten nicht alle drei Säulen.	(Valdivia et al., 2021)
2	Fehlende Vernetzung zwischen den Säulen	(Valdivia et al., 2021)
3	Ursache-Wirkungs-Ketten und Mechanismen, die zu einem Endpunkt führen, werden nicht verfolgt.	(Valdivia et al., 2021)
4	Hoher manueller Aufwand und die subjektive Definition von Systemgrenzen und Stakeholdern	(Costa et al., 2019)

Zur Durchführung eines LCSA sind viele standardisierte Informationen wie Materialkennzahlen erforderlich, die von allen Unternehmen gleichermaßen benötigt werden. Daher ist es nicht effizient, wenn Unternehmen diese Informationen selbst erheben und speichern, und es existieren verschiedene Datenbanken mit relevanten Informationen für ein LCSA. Dies gewährleistet außerdem, dass Unternehmen auf eine gleiche Datenbasis zugreifen und somit die Ergebnisse vergleichbar werden. Beispiele für diese Datenbanken sind GaBi (PE-International, 2021) oder Life Cycle Initiative (LifeCycleInitiative.org, 2022).

Die Anbindung dieser Datenbanken und die Auswahl der relevanten Informationen sind aktuell händische Prozesse. Dies führt dazu, dass die Analysen eine subjektive Komponente enthalten und somit nicht optimal vergleichbar sind.

3.2 Informatik

In der Informatik gibt es einige Bereiche, die sich mit der Verwaltung und Analyse von Daten sowie Informationen beschäftigen. Dies ist für diese Arbeit relevant, da die in der Produktentwicklung anfallenden Daten so verwaltet und mit frei verfügbarem Kontextwissen angereichert werden sollen, dass ein KI-System bei der Bewältigung der Komplexität in der Produktentwicklung unterstützen kann. Daher werden in diesem Unterkapitel der Stand der Wissenschaft und Praxis für Data-Science, KI und Datenbanken vorgestellt.

3.2.1 Data-Science

Wie in Kapitel 2.1.2.1 definiert, ist Data-Science die Anwendung quantitativer und qualitativer Methoden zur Lösung relevanter Probleme und zur Vorhersage möglicher Ereignisse. Im Folgenden wird der Stand der Wissenschaft und der Praxis vorgestellt.

3.2.1.1 Stand der Wissenschaft

Das Interesse der Wissenschaft an Data-Science hat in den letzten Jahren stark zugenommen. Dies zeigt der Anstieg der Publikationen, deren Titel oder Abstract den Begriff ‚Data Science‘ enthält, und ist in Abbildung 12 dargestellt.



Abbildung 12. Anzahl der Publikationen mit ‚Data Science‘ in Titel oder Abstract nach (dimensions.ai, 2023)

In der Produktentwicklung werden verschiedene Ansätze diskutiert, wie Data-Science eingesetzt werden kann. So nutzen Bertoni et al. (2017) Data-Science in frühen Phasen der Produktentwicklung, um bestimmte Produkteigenschaften zu simulieren und optimieren zu können (Bertoni et al., 2017). Eine Übersichtsstudie aus dem Jahr 2021 zeigt den Einsatz von Data-Science im Bereich Produktentwicklung (Chiarello et al., 2021). Darin wird vorgestellt, welche Data-Science-Techniken in den verschiedenen Phasen der Produktentwicklung eingesetzt werden. Dafür wurden Artikel zum Engineering auf Inhalt geprüft, der im Data-Science-Lexikon vorkommt. Dieses Lexikon wurde von den Autoren aus Data-Science-Techniken sowie wissenschaftlichen Artikeln erstellt und enthält 223

Einträge. Daraus ergibt sich, dass je nach Phase im Produktlebenszyklus zwischen 5 % und 56 % der Artikel Inhalte zu Data-Science aufweisen.

In den frühen Phasen wird Data-Science eingesetzt, um Kundenbedürfnisse zu verstehen, Anforderungen abzuleiten und zu analysieren (Chiarello et al., 2021). Diese können dann mit bestehenden Anforderungen verglichen werden, um Teile und Wissen aus früheren Produkten zu nutzen. In der Entwicklungsphase werden Data-Science-Methoden eingesetzt, um einzelne Parameter (zum Beispiel Wandstärken von Bauteilen) unter Berücksichtigung der Randbedingungen (zum Beispiel maximale Belastbarkeit) zu optimieren. Begleitend zur Entwicklung kann Data-Science im Projektmanagement eingesetzt werden, um Durchlaufzeiten zu berechnen und vorherzusagen.

Data-Science kann auch auf die Metadaten der Engineering-Objekte angewandt werden. Dadurch können etwa Durchlaufzeiten für Freigaben, der prozentuale Anteil der Objekte, die überarbeitet werden mussten, oder die Durchlaufzeit von Prozessen des Engineering-Change-Managements berechnet und anschaulich dargestellt werden (Lashin et al., 2020). Zusätzlich können die Einflussfaktoren untersucht werden, die dazu führen, dass die Kenngrößen einen bestimmten Wert ausweisen. Dies wurde zum Beispiel im Konzept ‚Product Lifecycle Intelligence‘ bereits in der Praxis umgesetzt (Wolf & Heckman, 2019). Auf Basis dieser Informationen können Vorhersagen getroffen und Vorschläge zur Optimierung der Kenngrößen berechnet werden.

3.2.1.2 Stand der Praxis

Aktuell investieren 80 % der Unternehmen in eine Data-Science-Abteilung oder in das Thema Data-Science (World Data Science Initiative, 2022). Abbildung 13 zeigt die Ergebnisse, die Unternehmen sich dabei erhoffen (Microstrategy, 2020). In der Abbildung ist der prozentuale Anteil der Unternehmen dargestellt, die eine der Antwortmöglichkeiten angegeben haben. Mehrfachnennungen waren möglich.



Abbildung 13. Erhoffte Ergebnisse durch Data-Science nach (World Data Science Initiative, 2022)

Es zeigt sich, dass aus Daten gewonnene Informationen vor allem zur Gewinnung von Effizienzen und zur Steigerung der Produktivität eingesetzt werden sollen. Damit geht eine schnellere und effektivere Entscheidungsfindung einher. Dadurch soll das finanzielle Ergebnis des Unternehmens verbessert werden.

Zur Analyse und Visualisierung der Daten haben sich Business-Intelligence-Werkzeuge wie Tableau von Salesforce (Tableau, 2023) und PowerBI von Microsoft (Microsoft, 2023) durchgesetzt. Diese ermöglichen sowohl die Berechnung der Informationen als auch deren Visualisierung. Ein ausgereiftes Benutzerkonzept und die Möglichkeit, Berechtigungen zu verwalten, erlauben die Bereitstellung der Informationen für die Mitarbeiter_innen, die sie benötigen. Die zugrunde liegenden Daten können sowohl im Werkzeug selbst als auch in einer spezialisierten Datenbank abliegen, auf die das Werkzeug zugreift.

Neben den klassischen Data-Science-Algorithmen gibt es spezialisierte Algorithmen für die Analyse von Graph-Datenbanken (Cook & Holder, 2006). Abbildung 14 zeigt die Analyse von Gemeinschaften, der Zentralität, die Vorhersage von Verbindungen und die Analyse der Ähnlichkeit. Die verschiedenen Analysealgorithmen werden im Folgenden beschrieben.

Graph-Data-Science			
Gemeinschaften	Zentralität	Vorhersage Verbindungen	Analyse Ähnlichkeit
<ul style="list-style-type: none"> • Clusterung einer Gruppe • Bestimmung der Tendenz (Verstärkung oder Bruch) 	<ul style="list-style-type: none"> • Bestimmung der Bedeutung verschiedener Knoten in einem Graph 	<ul style="list-style-type: none"> • Maschinelles Lernen wird auf einem Graph trainiert und angewandt 	<ul style="list-style-type: none"> • Berechnet die Ähnlichkeit zweier Knoten auf Basis der Knoten, zu denen Verbindungen bestehen

Abbildung 14. Übersicht von Graph-Data-Science Algorithmen nach (Neo4J, 2022e)

Die Erkennung von Gemeinschaften bewertet, wie eine Gruppe geclustert oder partitioniert ist, sowie ihre Tendenz, sich zu verstärken oder auseinanderzubrechen (Neo4J, 2022b). Die schwach verbundenen Komponenten können die Struktur der Graphen analysieren und nicht verbundene Teile finden. Darüber hinaus kann die Anzahl der Gemeinschaften innerhalb eines Graphen ermittelt werden, was Aufschluss über die Anzahl der Unterthemen gibt, die ein Graph enthält.

Die Zentralität kann verwendet werden, um die Bedeutung verschiedener Knoten in einem Netz zu bestimmen (Neo4J, 2022a). Einer der am häufigsten angewandten Algorithmen ist Pagerank (Brin & Page, 1998).

Die Vorhersage von Verbindungen erfolgt durch maschinelles Lernen. Ein Modell wird trainiert, um zu lernen, wo Beziehungen zwischen Knoten in einem Graphen bestehen sollten (Neo4J, 2022c). Dieses Modell kann dann verwendet werden, um weitere Beziehungen vorherzusagen.

Ähnlichkeitsalgorithmen berechnen die Ähnlichkeit von Knotenpaaren (Neo4J, 2022d). Die Ähnlichkeit zwischen zwei Knoten wird auf der Grundlage der Knoten berechnet, mit denen sie verbunden sind.

3.2.2 Künstliche Intelligenz

Künstliche Intelligenz hat die Fähigkeit, selbstständig Informationen aus Daten zu gewinnen. Daher könnte ein KI-System dazu geeignet sein, Ingenieur_innen bei der Beherrschung der Komplexität zu unterstützen. In diesem Unterkapitel wird die aktuelle Forschung zum Thema KI vorgestellt. Dazu wird zunächst der Stand der Wissenschaft und anschließend der Stand der Anwendung in der Produktentwicklung beleuchtet.

3.2.2.1 Stand der Wissenschaft

Die KI wird bereits seit den 1950er Jahren erforscht (Wahlster, 2017). Die Definition und das Verständnis von KI haben sich über die Zeit konstant weiterentwickelt. Dabei gibt es das Paradox, dass für Menschen schwierige Probleme für eine KI leicht sind und umgekehrt. So kann eine KI

mühe los einen Schachgroßmeister besiegen, ist aber nicht in der Lage, ein Kind zu trösten (Wahlster, 2019). Abbildung 15 zeigt eine Übersicht, wie sich die KI-Forschung entwickelt hat.

Von 1956 bis ca. 1970 wurden in der Forschung vor allem heuristische Systeme untersucht. Dabei wurden durch traditionelle Programmierung große Regelwerke erstellt, die es einem System erlauben sollten, intelligente Handlungen auszuführen. Nach heutigem Verständnis und nach dem Verständnis der Definition für KI für diese Arbeit würden diese Systeme nicht mehr als intelligent bezeichnet werden. Heuristische Systeme können getroffene Entscheidungen sehr gut begründen, da einem starren Regelwerk gefolgt wurde, allerdings kann das System seine Umwelt nicht wahrnehmen, nicht automatisch Neues erlernen und darüber hinaus keine Informationen abstrahieren.

Wissensbasierte Systeme wurden ca. von 1970 bis 1990 untersucht und eingesetzt. Diese kombinieren ein regelbasiertes System mit einer Wissensbasis, die von Hand aufgebaut wurde. Dies ermöglicht es dem System, über die Wissensbasis Umweltinformationen aufzunehmen und in Betracht zu ziehen. Allerdings ist das System statisch und mit einem hohen manuellen Aufwand verbunden, da sowohl das Regelwerk als auch die Wissensbasis durch traditionelle Programmierung händisch erstellt werden muss.

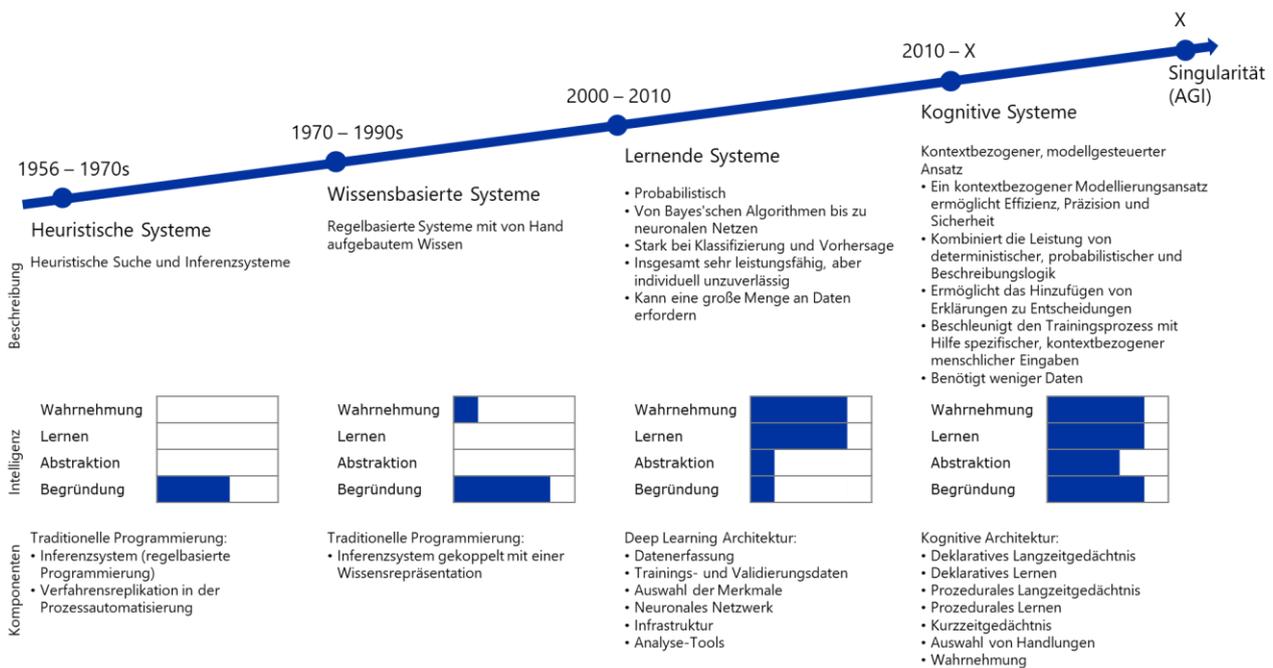


Abbildung 15. Übersicht der Forschung zur Künstlichen Intelligenz nach (Wahlster, 2019)

Lernende Systeme dominierten die Forschung von ca. 2000 bis 2010. Hier wurden die ersten neuronalen Netze eingesetzt. Dies bedeutet, dass das System selbstständig Schlüsse aus Trainingsdaten ziehen kann. Das System kann Informationen wahrnehmen und daraus lernen. Das einmal Gelernte kann allerdings nicht abstrahiert und auf einen neuen Kontext angewandt werden und getroffene Entscheidungen können nur schwer begründet werden.

Kognitive Systeme sind aktuell Gegenstand der Forschung. Hier wird eine komplexere Architektur verwendet, in der ein oder mehrere neuronale Netze mit Kontextinformationen und einem Kurz- sowie einem Langzeitgedächtnis (Speicher) kombiniert werden. Dies ermöglicht sowohl ein hohes Niveau an Wahrnehmung und Lernen als auch eine Abstraktion der Erkenntnisse und die Begründung von Entscheidungen.

Alle aktuellen Systeme fallen unter die in Kapitel 2.1.2.2 vorgestellte Kategorie der schwachen KI. Das bedeutet, dass die Systeme einzelne spezialisierte Anwendungsfälle sehr gut beherrschen, aber keine generelle Intelligenz erreicht haben. Generelle KI (oder artificial general intelligence, AGI) kann in der Zukunft erreicht werden. Eine KI auf diesem Niveau kann sich eigene Ziele setzen und selbstständig neue Fähigkeiten erlernen. Dies bedeutet, dass sie alles erreichen kann, da sie in der Lage ist, sich das benötigte Wissen zu beschaffen und die Ziele zu verfolgen. Das Erreichen von genereller KI wird als Singularität oder Intelligenzexplosion bezeichnet, da ein solches System sich selbst weiterentwickeln und somit die Intelligenz der Menschen weit überschreiten kann. Wie weit diese Singularität in der Zukunft liegt, wird aktuell stark diskutiert (Tegmark, 2017). Im Folgenden werden vor allem lernende und kognitive Systeme fokussiert, da diese aktuell in der Praxis eingesetzt oder entwickelt und untersucht werden.

Wie bereits in Kapitel 2.1.2.2 vorgestellt, wird KI für diese Arbeit als die Realisierung von intelligentem Verhalten und den zugrunde liegenden kognitiven Fähigkeiten auf Computern definiert. Eine Hauptfunktion von KI ist maschinelles Lernen, das es erlaubt, aus großen Datensätzen selbstständig zu lernen, ohne explizit programmiert zu werden (Samuel, 1959). Weitere Hauptfunktionen sind NLP, also das Verständnis von natürlicher Sprache (Goldberg, 2017), und Computer-Vision (Voulodimos et al., 2018), das heißt die Verarbeitung von Bild- oder Videoinformationen. Abbildung 16 zeigt, aus welchen Hauptfunktionen KI besteht.

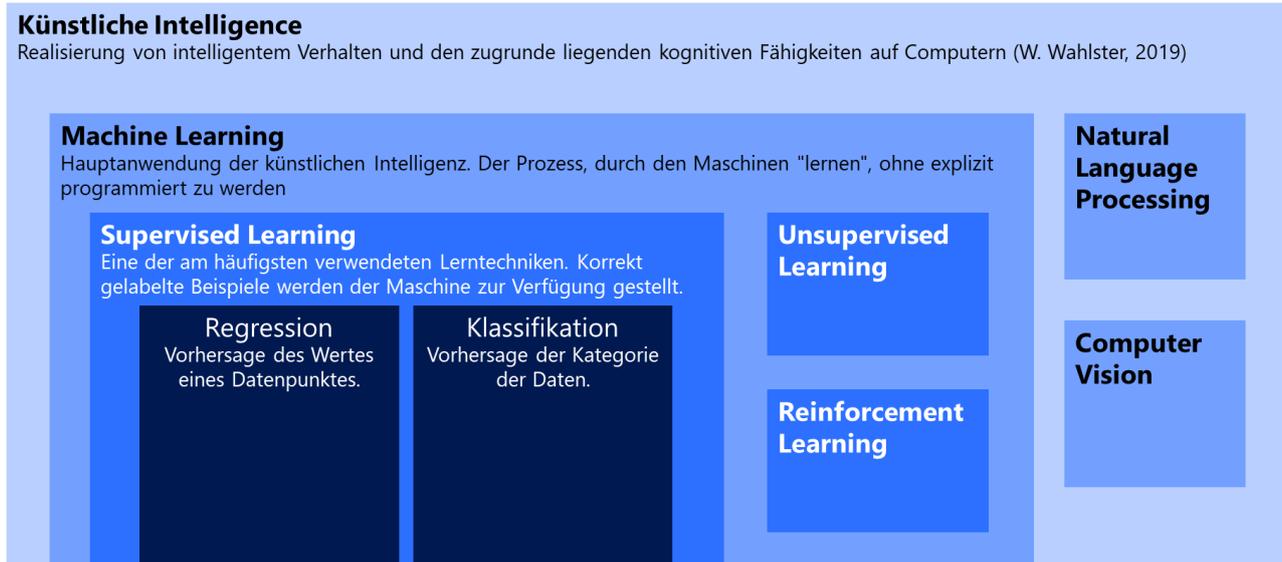


Abbildung 16. Künstliche Intelligenz und ihre Hauptfunktionen

Das maschinelle Lernen kann auf drei Arten erreicht werden: (a) Supervised Learning, (b) Unsupervised Learning und (c) Reinforcement-Learning (Mahesh, 2020). Beim Supervised Learning zeigt man dem neuronalen Netzwerk einen Datensatz mit gelabelten Daten. Diese beinhalten neben dem Eintrag selbst (zum Beispiel einem Bild von einem Hund) auch eine Beschreibung des Eintrags (zum Beispiel die Beschriftung ‚Hund‘). Diesen Trainingsdatensatz verwendet das System, um die Gewichtungen der Knoten im neuronalen Netz so lange anzupassen, bis die gewünschte Trefferquote oder eine bestimmte Anzahl an Iterationen erreicht ist. Danach wird das System an einem Testdatensatz, den es bisher nicht gesehen hat, getestet. Dies verhindert, dass das System ausschließlich die Trainingsdaten optimal beherrscht, aber nicht auf andere Daten anwendbar ist. Diese Lerntechnik heißt Supervised Learning, da ein Datensatz vorliegen muss, der von Hand gelabelt wurde. Dies kann in der Praxis hohe manuelle Aufwände bedeuten.

Unsupervised Learning bedeutet, dass die Trainingsdaten nicht gelabelt vorliegen. Das System kann darin verschiedene Klassen erkennen und eine Klassifizierung in die erkannten Klassen vornehmen. Beim Reinforcement-Learning werden im entsprechenden Anwendungsfall gewünschte Aktionen verstärkt, sodass sie häufiger ausgeführt werden. Dies kann eingesetzt werden, um den Weg durch ein Labyrinth zu finden, indem das System eine positive Rückmeldung erhält, wenn es das Labyrinth abgeschlossen hat. Die Größe der positiven Rückmeldung hängt dann von der Anzahl der benötigten Schritte ab.

Generell können KI-Systeme vor allem zwei Arten von Problemen lösen, Regression und Klassifikation. Eine Regression bedeutet, dass ein zukünftiger Datenpunkt auf Basis der Vergangenheit vorhergesagt werden kann. Klassifikation ist die Erkennung und Zuordnung von Daten zu Klassen (zum Beispiel Erkennung von Hund oder Katze auf einem Bild).

In den letzten Jahren wurden mehrere Anwendungen von maschinellem Lernen für Herausforderungen in der Produktentwicklung wie die Identifikation neuer Produktideen (Christensen et al., 2017), Anforderungserhebung (Wang & Zhang, 2017), Kreativität (Hein & Condat, 2018), Konfigurationsmanagement (Haifeng Liu et al., 2007) und Entscheidungsunterstützung in frühen Entwurfsphasen (Bertoni et al., 2017) erforscht.

Die Verarbeitung natürlicher Sprache ist der Teil der KI, der sich auf das Verstehen, Übersetzen und Erzeugen natürlicher Sprache konzentriert (Bates & Weischedel, 2006; Goldberg, 2017). Sie wird in der Regel durch eine Kombination aus einem regelbasierten System und maschinellem Lernen realisiert. Die Schritte, die ein NLP-System durchführen muss, um die gewünschte Ausgabe zu erzeugen, sind Vektorisierung, Part-of-Speech-Tagging und Dependency-Parsing (Chen, 2020). Vektorisierung ist die Umwandlung von Textinformationen in numerische Vektoren. Dies ist notwendig, da das System nur Berechnungen mit numerischen Daten durchführen kann. Daher werden die Informationen zunächst aus ihrer Quelle als Text extrahiert und dann vektorisiert. Der nächste Schritt ist ein Part-of-Speech-Tagging, bei dem die Wörter als Substantive, Verben, Adjektive usw. getaggt werden (Devlin et al., 2019). Der letzte Schritt ist das Parsen von Abhängigkeiten. Hier werden die Abhängigkeiten zwischen den Wörtern identifiziert und hervorgehoben. Dadurch wird zum Beispiel deutlich, welches Adjektiv oder Verb zu welchem Substantiv gehört, um die Semantik des analysierten Satzes zu verstehen (Goldberg, 2017). Es gibt Komponenten, die von anderen Forschern für die englische Sprache trainiert worden sind (Devlin et al., 2019). Diese können die oben beschriebenen Schritte durchführen, ohne dass ein Modell von Grund auf neu trainiert werden muss. Die Verarbeitung natürlicher Sprache findet allmählich ihre Anwendung in der Produktentwicklung. Forscher und Praktiker wenden sie in den Phasen an, in denen große Mengen natürlicher Sprache generiert oder verarbeitet werden müssen, zum Beispiel beim Anforderungsmanagement (Iqbal et al., 2018; Piquié et al., 2016).

Als Computer-Vision werden KI-Systeme bezeichnet, die Bild- oder Videodaten verarbeiten. Die Hauptfunktion ist dabei die Objekterkennung, bei der Objekte in Bild- oder Videodaten korrekt identifiziert werden sollen. So kann etwa auf einem Bild ein Bauteil erkannt werden, sodass die dazugehörigen Produktdaten direkt angezeigt werden können. Die Gesichtserkennung wird zum Beispiel von modernen Smartphones zur Authentifizierung eingesetzt (Voulodimos et al., 2018).

Eine Limitation der Technologie ist, dass ausschließlich erlernt werden kann, was Teil der Trainingsdaten war. Wenn ein System darauf trainiert wurde, Hunde von Katzen zu unterscheiden, wird es bei einem Bild einer Kuh melden, dass es wahrscheinlich ein Hund ist, da das System nicht weiß, dass mehr als die erlernten Klassen existieren können. Dies bedeutet, dass die Systeme immer für einen spezifischen Anwendungsfall geeignet sind. Ändert sich dieser, muss das System neu erstellt

und trainiert werden. Eine weitere Limitation ist, dass eine große Zahl an vergleichbaren Trainingsdaten vorliegen muss. Unterliegen diese Trainingsdaten einem Bias, wird dieser erlernt und kann sich somit verfestigen.

KI-Systeme sind häufig Black Boxes. Dies bedeutet, dass nicht ersichtlich ist, auf welchem Weg das System einen Input zu einem Output verarbeitet. Es kann also nicht nachvollzogen werden, welcher Lösungsweg gewählt wurde, um zu einem Ergebnis zu kommen. Dadurch kann das Vertrauen in ein KI-System leiden (Chowdhury & Sadek, 2012).

Aktuell gibt es keine klaren Vorgaben für die Feature-Auswahl und das Finetuning bei der Erstellung von KI-Systemen (Chowdhury & Sadek, 2012). Die Auswahl der Features sind die Inputfaktoren, auf deren Basis ein KI-System den Output berechnet. Hier muss entschieden werden, welche Faktoren relevant genug sind, um berücksichtigt zu werden. Im Anschluss daran muss bestimmt werden, wie viele Ebenen das neuronale Netz haben soll und mit welcher Lernrate die Anpassung geschehen soll. Beide Entscheidungen basieren stark auf der Erfahrung der Ingenieur_innen und nicht auf objektiven Vorgaben.

Zuletzt wird die Limitation genannt, dass die Haftung noch ungeklärt ist (Chowdhury & Sadek, 2012). Wird zum Beispiel ein Fahrzeug von einer KI gefahren, ist aktuell unklar, ob einer der Insassen, Fahrzeugbetreibende, das Herstellerunternehmen oder die KI selbst die Haftung für Fehler übernehmen müssen. Daher können aktuelle Systeme nur Vorschläge unterbreiten, aber nicht vollständig selbstständig handeln. Die finale Entscheidung und damit die Übernahme der Haftung finden immer durch einen Menschen statt.

Abbildung 17 zeigt die Analyse der aktuellen Fähigkeiten der Technologie KI. Hier wurde eine SWOT-Analyse (Strengths, Weaknesses, Opportunities, Threats; deutsch: Stärken, Schwächen, Chancen und Gefahren) durchgeführt (Helms & Nixon, 2010; Learned et al., 1969).

<ul style="list-style-type: none"> • Selbstständige Erkennung von Mustern in großen Datensätzen • Vorhersage der Zukunft auf Basis von Daten der Vergangenheit • Erkennung von Klassen in großen Datensätzen • Klassifikation von Objekten <p style="text-align: right;">Stärken</p>	<ul style="list-style-type: none"> • Transfer Learning: Transfer von gelerntem auf ein neues Gebiet • Kontextwissen: Was nicht in den Trainingsdaten war existiert nicht • Lernen mit kleinen Datenmengen <p style="text-align: right;">Schwächen</p>
<ul style="list-style-type: none"> • Unterstützung bei der Beherrschung von Komplexität • Analyse großer Datenmengen, die von Hand nicht analysierbar wären • Automatisierung von repetitiven Tätigkeiten • Vorschläge und Simulation <p style="text-align: right;">Chancen</p>	<ul style="list-style-type: none"> • Treffen von Entscheidungen aufgrund falscher Zusammenhänge • Bias in den Daten und damit im Modell <p style="text-align: right;">Gefahren</p>

Abbildung 17. SWOT-Analyse der aktuellen Fähigkeiten von Künstlicher Intelligenz

Die größten Stärken der KI sind die selbstständige Erkennung von Mustern in großen Datensätzen und die Vorhersage zukünftiger Datenpunkte auf Basis von Daten der Vergangenheit. Beides kann in der Produktentwicklung eingesetzt werden. Dabei können spezialisierte Anwendungsfälle wie die Berechnung und Vorhersage der Festigkeit von Bauteilen oder generellere Anwendungsfälle wie die Vorhersage der Durchlaufzeit im Änderungsmanagement unterstützt werden. Weitere Stärken sind die Erkennung von Klassen in großen Datensätzen und die Klassifikation von Objekten. In der Produktentwicklung kann dies zum Einsatz kommen, indem zum Beispiel mögliche Gleichteile gefunden oder Anforderungen automatisiert klassifiziert werden können. Aus diesen Stärken ergeben sich Chancen. So können durch den Einsatz von KI größere Datenmengen analysiert und in Betracht gezogen werden, was die Qualität der Designentscheidungen verbessern kann. Repetitive Tätigkeiten können automatisiert werden, sodass die Aufgabe nicht mehr händisch durchgeführt, sondern nur noch das Ergebnis auf Plausibilität geprüft werden muss.

Schwächen von KI sind, dass trainierte Systeme nur für den speziellen Anwendungsfall eingesetzt werden können, für den sie erstellt wurden. Einmal Gelerntes kann nicht auf weitere Anwendungsfälle übertragen werden. Ferner kennt das System nur die Aspekte, die in den Trainingsdaten vorkamen. Kommt später eine weitere Variable hinzu, kann diese nicht durch das System verarbeitet werden. Außerdem wird für das Trainieren ein großer Datensatz benötigt. Aus diesen Schwächen ergeben sich die Gefahren, dass Entscheidungen aufgrund falscher Zusammenhänge getroffen werden können, wenn das System nicht ausreichend trainiert und getestet wurde. Kamen in den Trainingsdaten nicht alle späteren Fälle vor oder unterlagen die Trainingsdaten einem Bias, wird auch das System verzerrte Ergebnisse liefern.

Der aktuelle Einsatz der KI in der Forschung der Produktentwicklung wurde in einem Übersichtsartikel von Shabestari et al. (2019) analysiert (Shabestari et al., 2019). Dabei haben sich die Autoren auf die frühen Designphasen konzentriert, also die Anforderungserhebung und Konzeptionierung.

Bei der Anforderungserhebung wird KI eingesetzt, um Anforderungen zu identifizieren, zu kategorisieren, zu analysieren und zu spezifizieren. Zur Identifikation von Anforderungen werden Texte untersucht, um Nutzeranforderungen zu gewinnen. Diese Texte können von Nutzern zum Beispiel in den sozialen Medien geschrieben worden sein (Christensen et al., 2017; Xu et al., 2010). Bei der Kategorisierung von Anforderungen werden diese nach der Identifikation verschiedenen Klassen zugeteilt. Diese können entweder von Expert_innen definiert oder selbstständig erkannt worden sein (Aguwa et al., 2017). Zur Analyse können Anforderungen nach der Qualität sortiert werden (Pinquié et al., 2018). Das Ziel der Anforderungsspezifikation ist die Zuordnung von Anforderungen zu Produktmerkmalen (Wang & Zhang, 2017).

Die Konzeptionierung wird zum einen bei der Konzepterstellung unterstützt, indem Ingenieur_innen Vorschläge von KI-Systemen unterbreitet werden (Hein & Condat, 2018). Die erstellten Konzepte können dann kategorisiert und analysiert werden (Zhang et al., 2017).

Der Einsatz einer Technologie, die potenziell selbstständig Entscheidungen treffen kann, bringt auch ethische Herausforderungen mit sich. Daher wurde in den letzten Jahren eine Vielzahl von ethischen Grundsätzen für KI-Systeme entwickelt. Es hat sich herausgestellt, dass KI zum Wohle der Allgemeinheit eingesetzt werden sollte, dass sie Menschen nicht schaden oder ihre Rechte untergraben sollte und dass sie Werte wie Fairness, Privatsphäre und Autonomie respektieren sollte (Whittlestone et al., 2019). Diese Grundsätze werden als zu vage kritisiert, um in der Entwicklung von KI-Systemen eine konkrete Hilfestellung zu sein.

3.2.2.2 *Stand der Praxis*

Wie im vorherigen Unterkapitel vorgestellt, wird KI aktuell in vielen Bereichen eingesetzt, beispielsweise in der Forschung zur Produktentwicklung. In der Praxis gibt es auch bereits kommerzielle Lösungen, die von Unternehmen genutzt werden.

In Deutschland wird aktuell an einem Marktplatz gearbeitet, über den Unternehmen bereits trainierte Modelle und Datensätze kaufen und verkaufen können. Außerdem können sich hier Anbieter von KI-Lösungen für die Produktentwicklung und Unternehmen treffen (its owl, 2023).

Im Bereich Anforderungsmanagement gibt es nicht nur eine große Anzahl von Forschungsaktivitäten, sondern auch erste kommerzielle Lösungen, die erworben und eingesetzt werden können. Dazu zählen QVscribe (QRA Corp, 2023) und reQlab (IT Designers, 2023). Beide Systeme können Anforderungen nach bestimmten Gütekriterien analysieren und eine Klassifizierung vornehmen. QVscribe hat zusätzlich die Fähigkeit, Anforderungen aus einem Fließtext zu extrahieren.

Die in Kapitel 2.1.1.3 vorgestellten großen Anbieter von PLM-Software implementieren Features, die auf KI-Technologie basieren. So können zum Beispiel durch generatives Design Vorschläge für geometrische Formen von Bauteilen auf Basis eingegebener Randbedingungen erstellt werden. Außerdem kann die Simulation unterstützt werden (Autodesk, 2023; Dassault, 2023; Siemens, 2023).

3.2.3 Datenbanken

Zur Erstellung eines Systemmodells müssen Daten aus verschiedenen Quellen in Beziehung gebracht werden. Diese Daten liegen in Unternehmen in unterschiedlichen Datenbanken. Daher werden in diesem Unterkapitel Möglichkeiten beschrieben, Daten zu speichern und zu verwalten. Dabei wird vor allem der Unterschied zwischen relationalen Datenbanken und Graph-Datenbanken erklärt. Für beide wird zunächst der Stand der Wissenschaft und anschließend der Stand der Praxis in der Produktentwicklung vorgestellt.

3.2.3.1 Stand der Wissenschaft

Wie in Kapitel 2.1.2.3 beschrieben, gibt es relationale und nichtrelationale Datenbanken. Relationale Datenbanken sind tabellarisch aufgebaut und Graph-Datenbanken, ein Beispiel für nichtrelationale Datenbanken, können als Nodes und Relationships visualisiert werden. Die beiden Konzepte haben verschiedene Vor- und Nachteile, die im Folgenden erläutert werden. Abbildung 18 visualisiert die beiden Konzepte.

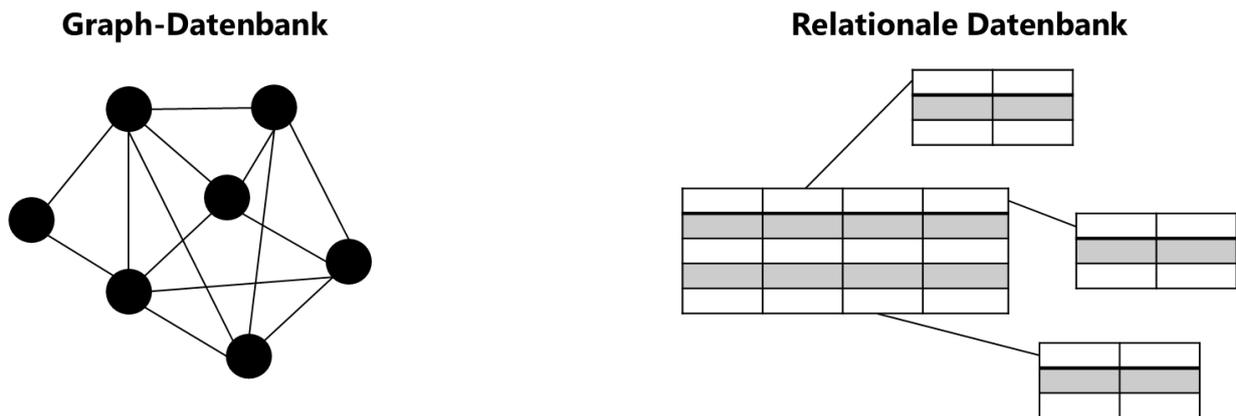


Abbildung 18. Visualisierung einer Graph-Datenbank und einer relationalen Datenbank nach (BI Connector Team, 2021)

Relationale Datenbanken sind geeignet, um Daten strukturiert abzulegen und zu verwalten. Lese- und Schreibvorgänge sowie Zugriffsrechte können verwaltet und kontrolliert werden. Daher sind diese Datenbanken im industriellen Kontext zum Standard geworden. Umgangssprachlich ist mit einer Datenbank fast immer eine relationale Datenbank gemeint. Allerdings kommen relationale Datenbanken an ihre Grenzen, wenn Daten unterschiedlicher Quellen, die dann in unterschiedlichen Tabellen vorliegen, miteinander in Beziehung gesetzt werden sollen. Dafür müssen verschiedene Tabellen nach einer bestimmten Spalte miteinander verbunden werden. Bei stark vernetzten Informationen, wie sie in der Produktentwicklung vorliegen, kann die Verbindung einer Vielzahl an Tabellen notwendig werden. Dies benötigt eine hohe Rechenleistung und kann somit auch viel Zeit in Anspruch nehmen.

Graph-Datenbanken hingegen fokussieren auf die Verbindungen zwischen einzelnen Datenpunkten. Sie sind nicht tabellarisch aufgebaut, sondern bestehen aus Nodes und Relationships. Daher können Objekte und ihre Relationen auf natürliche und klare Art repräsentiert werden (Angles & Gutierrez, 2005). Es muss bei der Erstellung kein Schema definiert werden, sodass verschiedene Schemata miteinander verbunden werden können (Angles & Gutierrez, 2008). Daher können Daten aus unterschiedlichen Quellen integriert werden, ohne dass die Schemata zunächst abgeglichen werden müssen, was die Erweiterung des Graphen um neue und nicht vorgesehene Quellen ermöglicht. Dies ist vor allem in einer komplexen Umgebung wie der Produktentwicklung nützlich.

Ein Vergleich der beiden Konzepte für Datenbanken zeigt, dass Graph-Datenbanken bei Datensätzen, die viele Beziehungen enthalten, Effizienzvorteile haben (Vicknair et al., 2010). Sie können in Betracht gezogen werden, wenn „1. Tabellen mit vielen Spalten vorhanden sind, von denen jede nur von wenigen Zeilen verwendet wird, 2. Tabellen mit Attributen existieren, 3. viele Many-to-many-Beziehungen existieren, 4. baumähnliche Merkmale vorliegen, 5. häufige Schemaänderungen erforderlich sind“ (Vicknair et al., 2010).

Durch die Konzentration auf die Verbindungen zwischen Datenpunkten kann die Abfragesprache Pfade, Nachbarschaften und Muster in den Daten finden und analysieren (Angles & Gutierrez, 2008). Dies kann neue Einblicke in stark vernetzte Datensätze ermöglichen.

3.2.3.2 *Stand der Praxis*

Die meisten großen Anbieter verwenden relationale Datenbanken. Die in Kapitel 2.1.1.3 vorgestellten PLM-Werkzeuge basieren alle auf relationalen Datenbanken. Allerdings gibt es einzelne Anbieter, die bereits Graph-Datenbanken im Bereich Produktentwicklung einsetzen. Im Folgenden werden einige davon beispielhaft vorgestellt.

Conweaver ist eine Firma, die sich auf die Erstellung von Wissensgraphen für Unternehmen spezialisiert hat (Conweaver, 2023). Dabei setzt sie einen Fokus auf den Bereich Produktentwicklung. Eine heterogene Systemlandschaft kann mit dem Linksphere-Wissensgraphen domänenübergreifend vernetzt werden. Conweaver verspricht dadurch eine Zeitersparnis von 20 bis 30 % bei der Suche nach Informationen, eine dreimal höhere Wiederverwendbarkeit von Bauteilen und 100 % Rückverfolgbarkeit von Anforderungen.

Die Robert Bosch Manufacturing Solutions GmbH bietet den Bosch Semantic Stack an (Robert Bosch Manufacturing Solutions GmbH, 2023). Dabei werden Knowledge-Graphen genutzt, um die Produktdaten über die Datensilos der Produktentwicklung hinweg miteinander zu vernetzen. Im Fokus stehen die Vernetzung von Daten der Produktentwicklung und die Rückführung von Daten während des Betriebs.

Ein Beispiel für einen Anbieter von PLM-Software, die nicht auf relationalen Datenbanken beruht, ist semanticPDM (semanticPDM, 2023). Hier wurde ein komplettes PLM-System auf Basis einer Graph-Datenbank, in diesem Fall Neo4J, erstellt. Dies soll allen domänenspezifischen Applikationen eine gemeinsame Datenbasis bieten, über die die Vernetzung der Daten gewährleistet wird.

3.3 Systemmodelle in der Produktentwicklung

In diesem Unterkapitel werden aktuelle Ansätze vorgestellt, die den Anspruch haben, ein durchgängiges Systemmodell für die Produktentwicklung schaffen. Zunächst wird die Modellierung genauer betrachtet, die bereits in Kapitel 3.1.2 als Bestandteil des MBSE erwähnt wurde. Im Anschluss

werden bestehende Konzepte dargestellt, bei denen Graph-Datenbanken in der Produktentwicklung eingesetzt werden.

3.3.1 Modellierung

Für die Modellierung werden die drei Elemente des MBSE-Dreiecks benötigt, die miteinander harmonisieren und zusammenspielen müssen. Diese sind die Modellierungsmethodik, die Modellierungssprache und das Modellierungswerkzeug. An dieser Stelle wird das Modellierungswerkzeug vernachlässigt, da die Vendoranalyse und die Auswahl eines geeigneten Werkzeuges stark von der IT-Landschaft im Unternehmen sowie vom angestrebten Anwendungsfall geprägt sind. Beides steht nicht im Fokus dieser Arbeit. Daher wird im Folgenden zunächst auf die Modellierungsmethodik und danach auf die Modellierungssprache eingegangen.

Eine Modellierungsmethode leitet den Benutzer bei der Anwendung von MBSE an (Schmidt & Stark, 2020), das heißt bei der Erstellung und Verwendung eines Systemmodells, indem sie beschreibt, wie jede Entwicklungsaktivität durchgeführt (Estefan, 2007) und die Semantik des Modells bestimmt wird (Dumitrescu et al., 2015).

In der Literatur werden verschiedene Modellierungsmethoden des MBSE beschrieben. Estefan (2007) stellt die folgenden Methoden vor: Harmony-SE von IBM, Object-Oriented Systems Engineering Method (OOSEM) von INCOSE, Rational Unified Process for Systems Engineering (RUP SE) von IBM, MBSE Methodology von Vitech, State Analysis (SA) von JPL und Object-Process Methodology (OPM) von Dori (Estefan, 2007). Zusätzlich führt Roques (2016) die Methode Architecture Analysis and Design Integrated Approach (ARCADIA) ein (Roques, 2016). Weitere Methoden sind SYSMOD (Weilkiens, 2016) und CONceptual design Specification technique for the ENgineering of complex Systems (CONSENS) (Gausemeier et al., 2012). Keine der Methoden hat sich bis heute als Standard durchgesetzt (Schmidt & Stark, 2020).

In einer vom Autor betreuten unveröffentlichten Masterthesis wurden verschiedene Modellierungssprachen untersucht (Kutaganahalli, 2022). Dazu wurden zunächst Anforderungen aus der Literatur herausgearbeitet, die für die Modellierung eines Systemmodells erfüllt sein sollen. Diese Anforderungen werden im Folgenden beschrieben:

- Formalität: Die Modellierungssprache sollte eine formale Definition eines Modells mithilfe seiner Syntax und Semantik liefern (van der Maas et al., 2001).
- Ausführbarkeit: Die Modellierungssprache sollte ausführbar sein (van der Maas et al., 2001).
- Erweiterbarkeit: Die Modellierungssprache sollte erweiterbar sein, um erwarteten und unerwarteten Anwendungen gerecht zu werden (Frank, 2013).
- Rückverfolgbarkeit: Die Modellierungssprache sollte eine Rückverfolgbarkeit ermöglichen (Nielsen et al., 2015).

- Anpassungsfähigkeit: Die Modellierungssprache sollte anpassungsfähig sein (Chiprianov et al., 2014).
- Interoperabilität: Die Modellierungssprache sollte mit anderen Modellierungssprachen interoperabel sein (Chiprianov et al., 2014).

Eine Modellierungssprache soll nicht nur die funktionalen Anforderungen erfüllen, sondern auch in der Praxis eingesetzt werden können. Um dies in der Bewertung zu berücksichtigen, wurden die folgenden nichtfunktionalen Anforderungen abgeleitet:

- Weiterentwicklung: Die Modellierungssprache sollte aktualisiert und verbessert werden können (Pereira & Silva, 2016).
- Nachvollziehbarkeit: Die Modellierungssprache sollte leicht zu verwenden, zu lernen und zu verstehen sein (Engels et al., 1992). Dies kann in die Unterpunkte Lesbarkeit, Nutzbarkeit und Lernbarkeit unterteilt werden.
- Flexibilität: Die Modellierungssprache sollte flexibel sein (Pereira & Silva, 2016).
- Toolunterstützung: Die Modellierungssprache sollte die Möglichkeit bieten, durch IT-Werkzeuge effizient umgesetzt zu werden (Fettke, 2009).
- Nutzerfreundlichkeit: Die Modellierungssprache sollte nutzerfreundlich sein (Entringer et al., 2021).

Im Anschluss daran wurden Modellierungssprachen durch eine Literaturrecherche identifiziert. Es wurden 56 Modellierungssprachen gefunden, die potenziell für die Modellierung von Produkten, Prozessen, medizinischen Verfahren, Lebenszyklen oder Zusammenhängen verwendet werden könnten. Diese Liste wurde nach ihrem Einsatz und ihrer Relevanz in der Produktentwicklung sowie nach der Relevanz zur Erstellung eines Systemmodells, das die Anforderungen an diese Arbeit erfüllen kann, reduziert. Die ausgewählten Modellierungssprachen werden im Folgenden beschrieben:

- Business Process Modelling Notation (BPMN): Modellierung von Business Prozessen (Shuman, 2010)
- Resource Description Framework (RDF): Es wird verwendet, um Daten im Web miteinander zu verknüpfen, und ermöglicht einen standardisierten Datenaustausch auf der Grundlage von Beziehungen (Beckett et al., 2014).
- Label Property Graphs (LPG): gerichteter, etikettierter Mehrfachgraph mit der besonderen Eigenschaft, dass jede Node und jede Relationship eine (möglicherweise leere) Menge von Eigenschafts-Werte-Paaren enthalten kann (Angles, 2018)
- Universal Modeling Language (UML): Allzweck-Modellierungssprache, die eine Standardmethode zur Visualisierung des Systems bieten soll (Booch et al., 2005)

- Lifecycle Modeling Language (LML): Erweiterung der Prinzipien von MBSE über die Phasen der Produktentwicklung und Produktion hinaus in die Phasen der Konzeptentwicklung, Nutzung, Support und Recycling (Lifecycle Modeling Organization, 2022)
- System Modeling Language (SysML): Diese wurde in Kapitel 3.1.2.1 vorgestellt.
- Petri Nets: mathematisches Modell, das in verschiedenen Szenarien eingesetzt werden kann, zum Beispiel bei der Analyse medizinischer Daten, der Simulation von Behandlungsverfahren und der Modellierung von Prozessen im Gesundheitswesen (Al-Fedaghi & Shbeeb, 2019)
- Modelica: Sprache für die Modellierung von cyberphysischen Systemen, die eine kausale Verbindung von Komponenten unterstützt, die durch mathematische Gleichungen gesteuert werden (Modelica Association, 2023)
- COMPASS (Comprehensive Modeling for Advanced SoS): Erweiterung existierender Sprachen zur Modellierung von Systems of Systems (Woodcock et al., 2012)
- SysML + CML: Ansatz, der die Modellierungsrichtlinien, das Requirements-Engineering und die Architekturmodellierung von SysML umfasst und in eine formale Modellierungsnotation CML übergeht, die ausschließlich für die Modellierung von SoS entwickelt wurde (Ingram et al., 2015)

Alle ausgewählten Modellierungssprachen wurden nach dem Erfüllungsgrad der Anforderungen bewertet. Tabelle 6 bietet einen Überblick über die Anforderungen, die Modellierungssprachen und die Bewertung.

Tabelle 6. Bewertung ausgewählter Modellierungssprachen nach (Kutag-anahalli, 2022)

		Modellierungssprachen										
		BPMN	RDF	LPG	UML	LML	SysML	PetriNets	Modelica	COMPASS	SysML + CML	
Funktionale Anforderungen	1	Formalität	✓	✓	✓	●	X	●	✓	●	✓	✓
	2	Ausführbarkeit	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
	3	Erweiterbarkeit	✓	✓	✓	✓	✓	✓	✓	✓	?	?
	4	Rückverfolgbarkeit	X	✓	✓	●	?	✓	●	●	?	✓
	5	Anpassungsfähigkeit	●	?	✓	●	✓	●	●	?	●	●
	6	Interoperabilität	X	✓	✓	●	X	✓	X	●	✓	✓
Nicht-funktionale Anforderungen	1	Weiterentwicklung	✓	●	✓	✓	✓	✓	●	?	✓	?
	2	Nachvollziehbarkeit	✓	✓	✓	✓	✓	✓	X	?	●	●
		2.1 Lesbarkeit	✓	✓	✓	✓	✓	✓	X	?	●	●
		2.2 Nutzbarkeit	✓	✓	✓	✓	✓	✓	✓	?	●	●
		2.3 Lernbarkeit	✓	✓	✓	✓	✓	✓	●	?	●	●
	3	Flexibilität	✓	✓	✓	✓	✓	✓	?	X	✓	✓
	4	Toolunterstützung	✓	●	✓	✓	?	✓	✓	✓	✓	✓
	5	Nutzerfreundlichkeit	✓	✓	✓	✓	✓	✓	✓	●	X	?
			✓	: Erfüllt.		●	: teilweise erfüllt.					
			X	: Nicht erfüllt.		?	: keine Information.					

Da bereits eine Vielzahl an Modellierungssprachen existiert, ist die Auswahl einer geeigneten Sprache für den aktuellen Anwendungsfall eine große Herausforderung. Für diese Arbeit ist die Erkenntnis relevant, dass es verschiedene Modellierungssprachen gibt, die in unterschiedlichen Anwendungsfällen geeignet sind. Es ist also davon auszugehen, dass in der Entwicklung eines komplexen mechatronischen Produktes in verschiedenen Entwicklungsphasen und von unterschiedlichen Entwicklungsdomänen verschiedene Modelle in unterschiedlichen Sprachen erstellt werden, die den aktuellen Anwendungsfall unterstützen.

3.3.2 Vernetzung von Daten in der Produktentwicklung

Es gibt aktuell Ansätze, Produktdaten aus verschiedenen Systemen miteinander in Beziehung zu setzen. Dabei werden meist Daten aus PLM-, ALM- und ERP-Systemen in eine semantische Beziehung zueinander gebracht. Im Folgenden werden diese Ansätze vorgestellt und verglichen.

Eigner und Sindermann (2014) präsentieren einen Ansatz, in dem neutrale Dateninterfaces verwendet werden. Diese erlauben die Nutzung von Best-of-Breed-IT-Systemen für verschiedene Anwendungen. Es müssen also nicht alle Funktionen in ein einzelnes PLM- oder ERP-System implementiert werden, sondern es können spezialisierte Systeme für die einzelnen Anwendungsfälle genutzt werden, die über das Dateninterface miteinander verbunden werden können. Ein Beispiel für ein solches Interface ist STEP (Eigner & Sindermann, 2014). In der Abbildung 19 ist das Konzept visualisiert. Prototypen wurden für die PLM-Systeme Windchill und den ARAS Innovator erstellt.

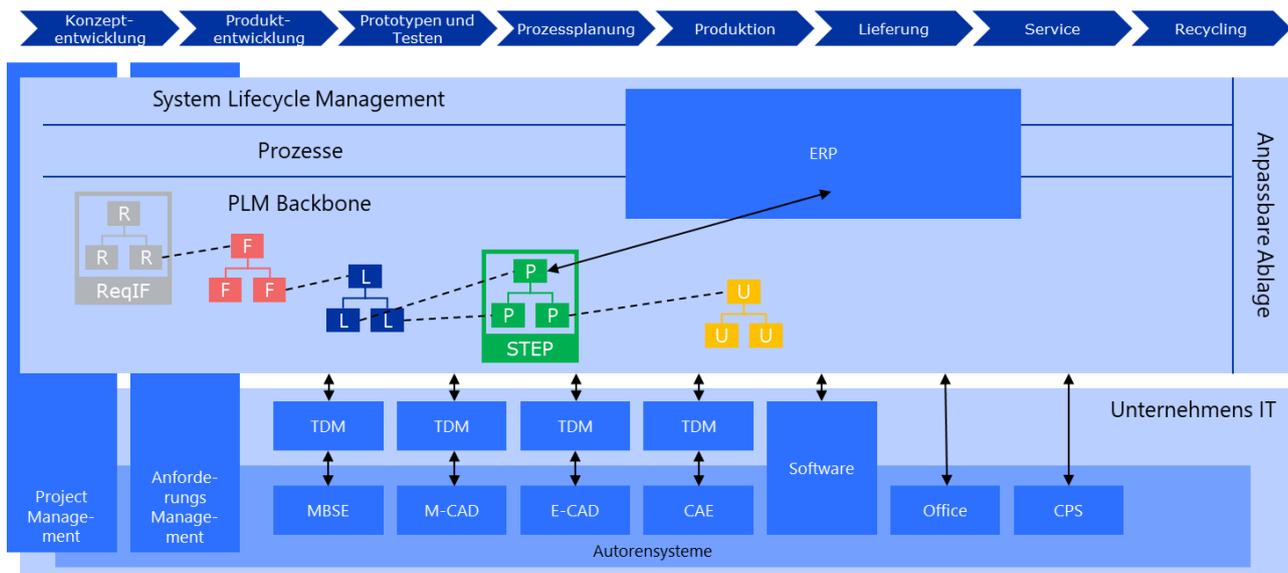


Abbildung 19. Neutrale Dateninterfaces (Eigner & Sindermann, 2014)

In Tabelle 7 wird der Erfüllungsgrad der in Kapitel 2.3.2 aufgestellten Anforderungen für das Konzept der neutralen Dateninterfaces dargestellt. Über die neutralen Dateninterfaces können verschiedene Datenbanken miteinander verbunden werden, weshalb A1 vollständig erfüllt ist. MBSE-Modelle können eingebunden werden und mit Informationen aus den Datenbanken verbunden werden. Allerdings können keine beliebigen Modelle angebunden werden und die Verbindung der Modelle zu den Datenbanken ist nicht hinreichend beschrieben. Daher sind die Anforderungen A2 und A3 nicht vollständig erfüllt. Unstrukturierte Dokumente können in Form von Office Dokumenten angebunden werden. Daher sind die Anforderungen A4 und A7 teilweise erfüllt. Frei verfügbares Kontextwissen und semantische Beziehungen können nicht repräsentiert werden. Daher sind Anforderungen A5 und A6 nicht erfüllt.

Tabelle 7. Erfüllungsgrad der Anforderungen neutrale Dateninterfaces

#	Anforderung	Erfüllungsgrad neutrale Dateninterfaces
A1	Das Systemmodell soll verschiedene Datenbanken aus dem Bereich Produktentwicklung (zum Beispiel PLM, ERP, ALM, IoT etc.) miteinander verbinden können.	●
A2	Das Systemmodell soll verschiedene Modelle (zum Beispiel MBSE, Simulation etc.) aus dem Bereich Produktentwicklung miteinander verbinden können.	◐
A3	Das Systemmodell soll Modelle und Datenbanken miteinander verbinden können.	◐
A4	Das Systemmodell soll unstrukturierte, frei verfügbare Daten (zum Beispiel Normen, Ausschreibungsdokumente, Fachartikel) mit den bestehenden Daten verbinden können.	◑
A5	Das Systemmodell soll frei verfügbares Kontextwissen (wie Wikimedia Knowledge Graph) mit den bestehenden Daten verbinden können.	○
A6	Das Systemmodell soll semantische Beziehungen zwischen den Daten abbilden können.	○
A7	Das Systemmodell soll um strukturierte und unstrukturierte Daten flexibel und dynamisch erweiterbar sein.	◑

Weitere Forschung zu dem Thema findet im Fachgebiet virtuelle Produktentwicklung (VPE) der Universität Kaiserslautern statt (VPE, 2023). Dort ist das Ziel der Arbeiten im Forschungsgebiet ‚Vernetzte IT-Toolketten und Standards‘, eine effektivere Kommunikation und Interpretation von im Unternehmen und in der Lieferkette verteilten komplexen Produktinformationen zu gewährleisten. Dazu werden flexible Lösungen zur zentralen Verwaltung von Produktinformationen erarbeitet, die eine Anknüpfung und Austauschbarkeit diverser Legacy-Systeme erlauben. Ein weiteres Forschungsgebiet, ‚Semantic Product Lifecycle Management‘, beschäftigt sich mit der Bereitstellung von Produktdaten entlang des gesamten Lebenszyklus. Dazu sollen Produktdaten, die in mehreren

heterogenen Systemen verteilt vorliegen, auf Basis semantischer Datenmodellierung integriert werden.

Das Forschungsprojekt ‚Semantic Product/Process Information and Digitized Engineering Repository‘ (SP²IDER) befasst sich mit der Erstellung einer weiteren Ebene des Informationsmanagements zur Anzeige von Daten aus verschiedenen IT-Systemen (Eickhoff et al., 2020). In der Studie werden drei Möglichkeiten aufgezeigt, Engineering-Daten miteinander zu verbinden. Diese sind Ontologien, Standards und Webservices. Alle drei Möglichkeiten können Daten miteinander verbinden, allerdings sind sie nicht geeignet, um alle Engineering-Objekte zu inkludieren. Daher wurde das SP²IDER-Konzept entwickelt und vorgestellt.

Abbildung 20 zeigt, wie der Graph im SP²IDER-Konzept aufgebaut wird. Er basiert auf einer Ontologie, die aus dem PLM-System abgeleitet wurde. Daraus kann dann der initiale Graph erstellt werden, der dann um zusätzliche Quellen aus dem Engineering erweitert wird (Eickhoff et al., 2020). Dabei bleibt der Fokus auf unternehmensinternen Quellen, die Systeme der Zulieferer oder externe Quellen werden nicht berücksichtigt.

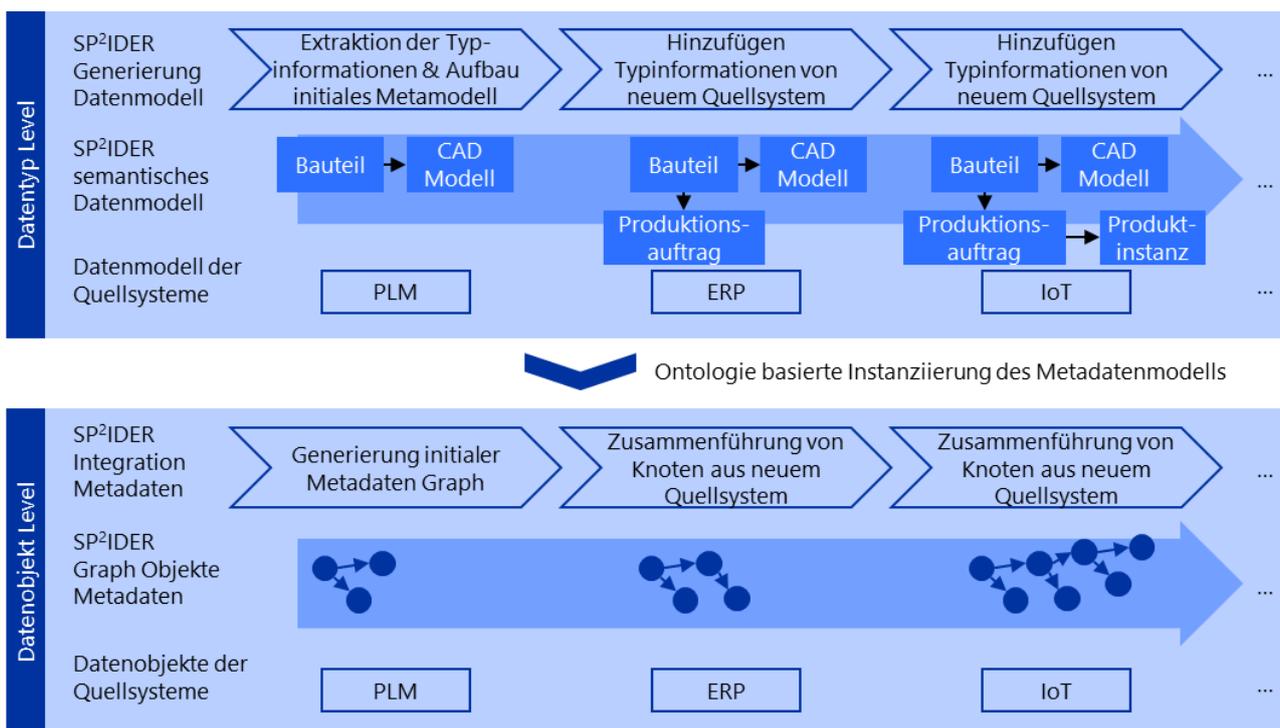


Abbildung 20. Aufbau des SP²IDER-Graph (Eickhoff et al., 2020)

Das in SP²IDER entwickelte Konzept wird auf den Erfüllungsgrad der in Kapitel 2.3.2 aufgestellten Anforderungen untersucht. Tabelle 8 fasst die Ergebnisse zusammen. Verschiedene Datenbanken wie PLM- und ERP-Systeme können miteinander verbunden werden, weshalb A1 erfüllt ist. Modelle können nur rudimentär eingebunden und mit Daten vernetzt werden. A2 und A3 sind daher nur niedrig erfüllt. Auch unstrukturierte Daten können nicht systematisch eingebunden werden, weshalb

auch die Erweiterbarkeit beschränkt ist. A4 und A7 sind daher nur niedrig erfüllt. Frei verfügbares Kontextwissen kann nicht eingebunden werden, A5 ist daher nicht erfüllt. Die Stärke des Konzeptes liegt in der semantischen Verbindung der Daten. A6 ist daher als größtenteils erfüllt anzusehen. Es fehlt allerdings noch eine systematische Vergabe der semantischen Bezeichnungen.

Tabelle 8. Erfüllungsgrad der Anforderungen SP²IDER Konzept

#	Anforderung	Erfüllungsgrad SP ² IDER Konzept
A1	Das Systemmodell soll verschiedene Datenbanken aus dem Bereich Produktentwicklung (zum Beispiel PLM, ERP, ALM, IoT etc.) miteinander verbinden können.	
A2	Das Systemmodell soll verschiedene Modelle (zum Beispiel MBSE, Simulation etc.) aus dem Bereich Produktentwicklung miteinander verbinden können.	
A3	Das Systemmodell soll Modelle und Datenbanken miteinander verbinden können.	
A4	Das Systemmodell soll unstrukturierte, frei verfügbare Daten (zum Beispiel Normen, Ausschreibungsdokumente, Fachartikel) mit den bestehenden Daten verbinden können.	
A5	Das Systemmodell soll frei verfügbares Kontextwissen (wie Wikimedia Knowledge Graph) mit den bestehenden Daten verbinden können.	
A6	Das Systemmodell soll semantische Beziehungen zwischen den Daten abbilden können.	
A7	Das Systemmodell soll um strukturierte und unstrukturierte Daten flexibel und dynamisch erweiterbar sein.	

Huet et al. (2020) empfehlen einen Wissensgraphen mit Designregeln, der einen Context Aware Cognitive Design Assistant (CACDA) speist (Huet et al., 2020, 2021). Dazu wurden Designregeln in einem semantischen Netzwerk in Neo4J modelliert, das dann Ingenieur_innen bei der Entwicklung unterstützt, indem es bei der Auswahl der relevanten Regeln hilft. Eine Erweiterung von Regeln und die Verlinkung findet durch Expert_innen manuell statt. In einem Test wurde gezeigt, dass mit dem

CACDA mehr Regeln korrekt angewendet wurden und die wahrgenommene Einfachheit der Benutzung signifikant gestiegen ist (Huet et al., 2021). Abbildung 21 zeigt, wie CACDA Designregeln findet und vorschlägt.

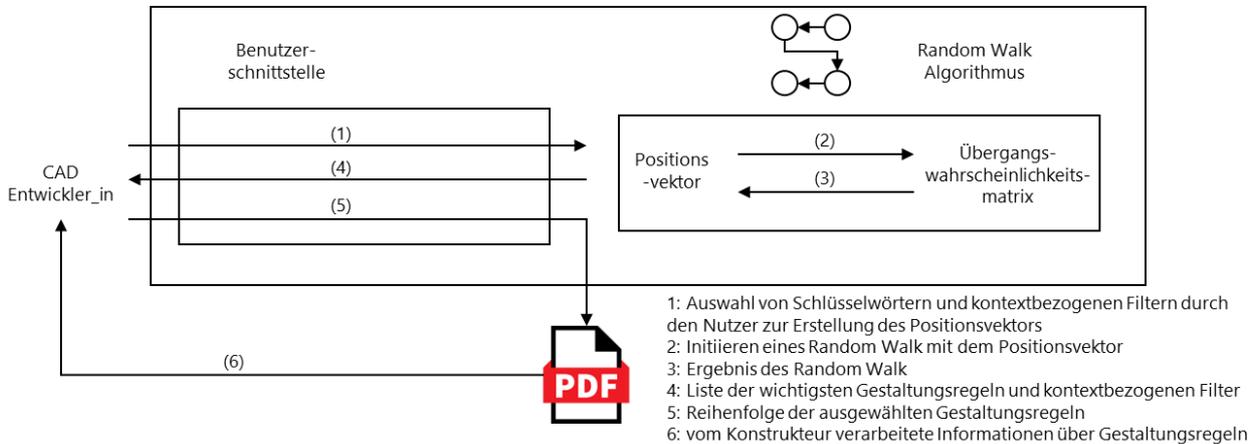


Abbildung 21. CACDA-Vorschlagsystem für Designregeln (Huet et al., 2021)

In Tabelle 9 wird der Erfüllungsgrad der in Kapitel 2.3.2 aufgestellten Anforderungen des CACDA Konzeptes zusammengefasst. CACDA wurde als Graph Datenbank in Neo4J umgesetzt. Damit besteht die Möglichkeit, semantische Beziehungen abzubilden und das Konzept erfüllt Anforderung A6. Alle anderen Anforderungen werden nicht erfüllt.

Tabelle 9. Erfüllungsgrad der Anforderungen CACDA

#	Anforderung	Erfüllungsgrad CACDA
A1	Das Systemmodell soll verschiedene Datenbanken aus dem Bereich Produktentwicklung (zum Beispiel PLM, ERP, ALM, IoT etc.) miteinander verbinden können.	○
A2	Das Systemmodell soll verschiedene Modelle (zum Beispiel MBSE, Simulation etc.) aus dem Bereich Produktentwicklung miteinander verbinden können.	○
A3	Das Systemmodell soll Modelle und Datenbanken miteinander verbinden können.	○
A4	Das Systemmodell soll unstrukturierte, frei verfügbare Daten (zum Beispiel Normen, Ausschreibungsdokumente, Fachartikel) mit den bestehenden Daten verbinden können.	○

A5	Das Systemmodell soll frei verfügbares Kontextwissen (wie Wikimedia Knowledge Graph) mit den bestehenden Daten verbinden können.	○
A6	Das Systemmodell soll semantische Beziehungen zwischen den Daten abbilden können.	●
A7	Das Systemmodell soll um strukturierte und unstrukturierte Daten flexibel und dynamisch erweiterbar sein.	○

In ihrer Studie zeigen Abramovici et al. (2016) das semantische Datenmanagement für die Entwicklung von intelligenten Produkten (Abramovici et al., 2016). Dabei werden Daten aus allen für die Entwicklung relevanten Systemen aus dem gesamten Unternehmen zusammengezogen. Eine Besonderheit ist, dass dabei auch Daten von Zulieferern einbezogen werden können. Abbildung 22 enthält eine Darstellung der empfohlenen Architektur.

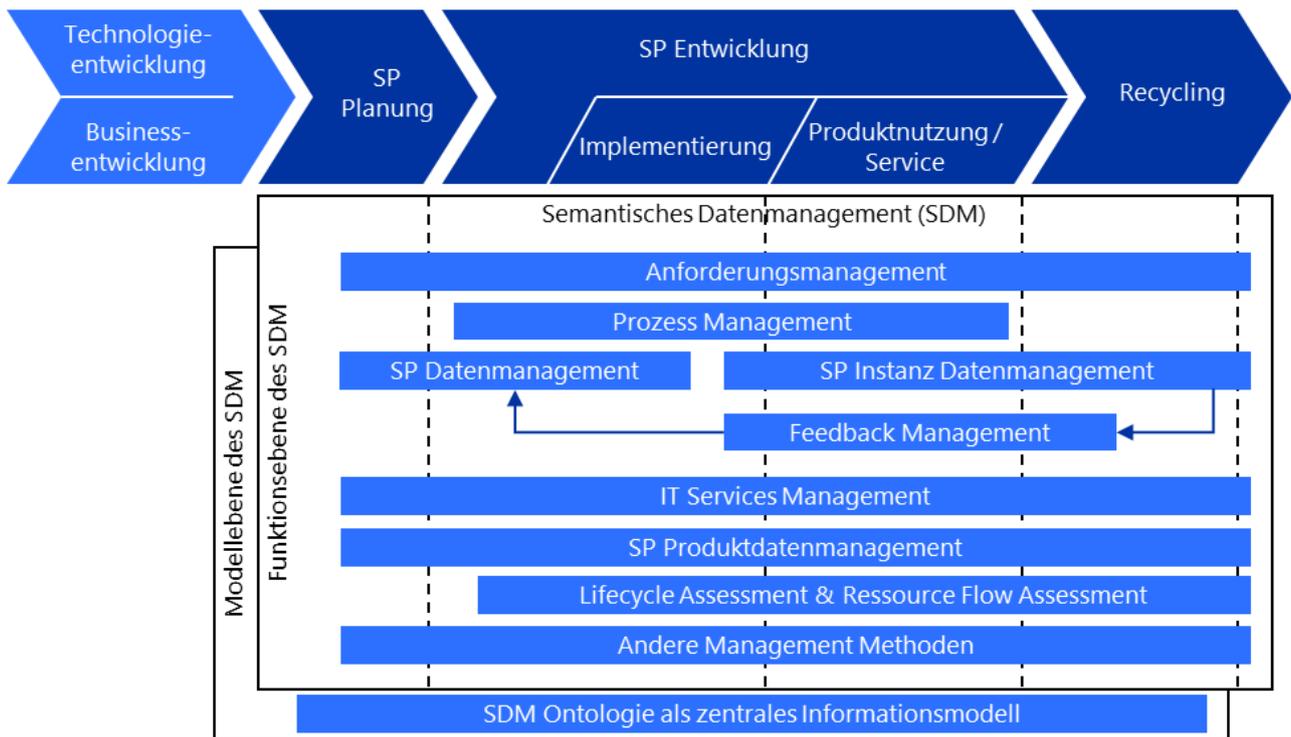


Abbildung 22. Semantisches Datenmanagement (Abramovici et al., 2016)

In Tabelle 10 wird der Erfüllungsgrad der in Kapitel 2.3.2 aufgestellten Anforderungen des semantischen Datenmanagements dargestellt. In dem Konzept werden Datenbanken und Modelle miteinander über semantische Beziehungen verbunden. Daher sind die Anforderungen A1, A2, A3 und A6 erfüllt. Unstrukturierte Daten und frei verfügbares Kontextwissen können nicht aufgenommen werden. Daher sind die Anforderungen A4, A5 und A7 nicht erfüllt.

Tabelle 10. Erfüllungsgrad der Anforderungen semantisches Datenmanagement

#	Anforderung	Erfüllungsgrad semantisches Datenmgmt
A1	Das Systemmodell soll verschiedene Datenbanken aus dem Bereich Produktentwicklung (zum Beispiel PLM, ERP, ALM, IoT etc.) miteinander verbinden können.	●
A2	Das Systemmodell soll verschiedene Modelle (zum Beispiel MBSE, Simulation etc.) aus dem Bereich Produktentwicklung miteinander verbinden können.	●
A3	Das Systemmodell soll Modelle und Datenbanken miteinander verbinden können.	●
A4	Das Systemmodell soll unstrukturierte, frei verfügbare Daten (zum Beispiel Normen, Ausschreibungsdokumente, Fachartikel) mit den bestehenden Daten verbinden können.	○
A5	Das Systemmodell soll frei verfügbares Kontextwissen (wie Wikimedia Knowledge Graph) mit den bestehenden Daten verbinden können.	○
A6	Das Systemmodell soll semantische Beziehungen zwischen den Daten abbilden können.	●
A7	Das Systemmodell soll um strukturierte und unstrukturierte Daten flexibel und dynamisch erweiterbar sein.	○

Neben den vorgestellten Ansätzen aus der Wissenschaft gibt es auch Ansätze, die bereits in der Praxis umgesetzt und gelebt werden. Diese sind eine Kooperation von PTC und Microsoft und die Datenbank der Firma DataHub.

PTC und Microsoft haben in einer Kollaboration eine Lösung gestaltet, bei der On-Premise-IT-Systeme, wie sie in der Entwicklung häufig eingesetzt werden, in der Cloud gehostet werden können (Möring & Feldmann, 2022). Dies bedeutet, dass die Systeme nicht mehr auf den eigenen Servern der Unternehmen laufen und von ihnen verwaltet werden, sondern auf von Microsoft bereitgestellten Servern. Somit sind diese Systeme meistens sicherer und können über bereitgestellte APIs

angebunden werden. Dadurch erhöhen sich die Möglichkeiten zur Zusammenarbeit und zum Datenaustausch mit Zulieferern. Abbildung 23 zeigt eine Darstellung der Lösung.

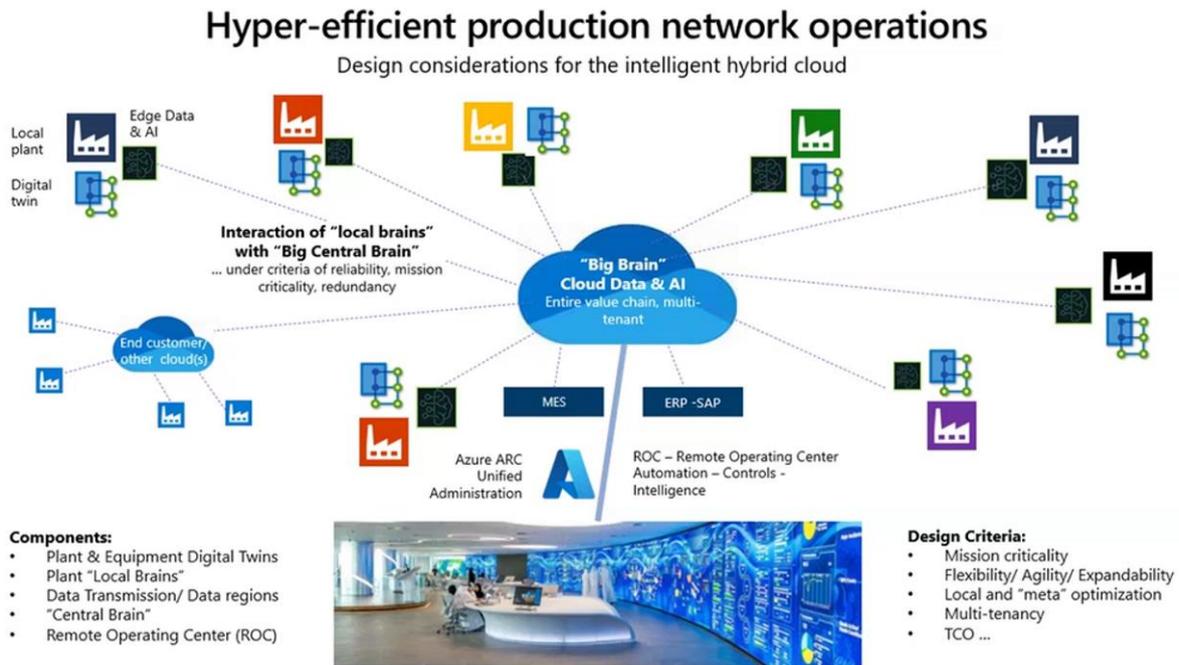


Abbildung 23. Architektur der Kooperation von PTC und Microsoft (Möring & Feldmann, 2022)

Die Firma DataHub bietet eine Plattform zur Verwaltung von Metadaten für verschiedene Systeme (DataHub, 2023). Hier können Daten miteinander semantisch verbunden werden, was die Datenermittlung, Datenbeobachtung und föderierte Kontrolle ermöglicht, um die Komplexität des Datenökosystems zu bewältigen.

3.4 Handlungsbedarf

In diesem Unterkapitel wird der aktuelle Erfüllungsgrad der in Kapitel 2.3.2 vorgestellten Anforderungen analysiert. Daraus wird dann der weitere Handlungsbedarf abgeleitet.

In Kapitel 1.1 wurde die Motivation für die Arbeit beschrieben. Die steigende Datenmenge und damit die zunehmende Komplexität sind Herausforderungen, denen mit dem Einsatz geeigneter Technologien wie Data-Science und KI begegnet werden soll. Aktuelle Ansätze beschränken den Einsatz der KI-Technologie auf einzelne, spezialisierte Anwendungsfälle. Dies bietet bereits einen Mehrwert und kann zu großen Effizienz- und Effektivitätsgewinnen führen. Allerdings birgt die Anwendung auf der Ebene der gesamten Produktentwicklung auf allen Ebenen des System of Interest, wie sie in Kapitel 3.1.2 vorgestellt wurden, ein hohes Potenzial.

Ein KI-System, das zur holistischen Verbesserung der Produktentwicklung eingesetzt werden soll, benötigt Kontextwissen über die höheren Ebenen des System of Interest. Die Analyse des aktuellen

Forschungsstandes in Kapitel 3.3 hat gezeigt, dass es dafür noch keine ausreichenden Konzepte gibt. Die Ebene des ‚Health-System‘ und die des ‚Society-Eco-System‘ werden in keinem der gefundenen und vorgestellten Konzepte berücksichtigt.

Die Integration von Daten und Informationen aus den höheren Ebenen des System of Interest bedeutet, dass unternehmensexterne Daten integriert werden müssen. Dabei wird auf frei verfügbare Daten aus dem semantischen Web zurückgegriffen. Diese können zum Beispiel der Google Knowledge Graph (Google, 2021) oder der Wikimedia Knowledge Graph (Wikimedia, 2021) sein. Aktuelle Konzepte berücksichtigen als unternehmensexterne Daten in manchen Fällen Daten, die mit Entwicklungspartnern entlang der Wertschöpfungskette ausgetauscht werden. Dies ist ein wichtiger Schritt, um Kontextdaten einzubeziehen, aber er ist für die Erfüllung der Vision dieser Arbeit nicht ausreichend.

Nach der Analyse des aktuellen Forschungsstandes und des Standes der Praxis kann der Erfüllungsgrad der in Kapitel 2.3.2 aufgestellten Anforderungen bewertet werden. Tabelle 11 fasst die Bewertungen der in Kapitel 3.3.2 vorgestellten Konzepte zusammen und zeigt eine Gesamtbewertung.

Tabelle 11. Erfüllungsgrad der Anforderungen an diese Arbeit

#	Anforderung	Erfüllungsgrad neutrale Dateninterfaces	Erfüllungsgrad SP ² IDER Konzept	Erfüllungsgrad CACDA	Erfüllungsgrad semantisches Datenmgmt	Erfüllungsgrad Gesamtbewertung
A1	Das Systemmodell soll verschiedene Datenbanken aus dem Bereich Produktentwicklung (zum Beispiel PLM, ERP, ALM, IoT etc.) miteinander verbinden können.	●	●	○	●	●
A2	Das Systemmodell soll verschiedene Modelle (zum Beispiel MBSE, Simulation etc.) aus dem Bereich Produktentwicklung miteinander verbinden können.	◐	◐	○	●	◐
A3	Das Systemmodell soll Modelle und Datenbanken miteinander verbinden können.	◐	◐	○	●	◐

A4	Das Systemmodell soll unstrukturierte, frei verfügbare Daten (zum Beispiel Normen, Ausschreibungsdokumente, Fachartikel) mit den bestehenden Daten verbinden können.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A5	Das Systemmodell soll frei verfügbares Kontextwissen (wie Wikimedia Knowledge Graph) mit den bestehenden Daten verbinden können.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>
A6	Das Systemmodell soll semantische Beziehungen zwischen den Daten abbilden können.	<input type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>	<input checked="" type="radio"/>
A7	Das Systemmodell soll um strukturierte und unstrukturierte Daten flexibel und dynamisch erweiterbar sein.	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>	<input type="radio"/>

Es existiert aktuell kein Konzept, das die Anforderungen in seiner Gesamtheit erfüllt. Einzelne Anforderungen, wie beispielsweise A1 oder A6, werden von vielen Konzepten bereits erfüllt.

Insbesondere Anforderung A4, A5 und A7 werden durch die Konzepte der Literatur und der Praxis noch nicht oder nicht ausreichend erfüllt. Frei verfügbares Kontextwissen wird von keinem Konzept integriert und nutzbar gemacht. Außerdem werden unstrukturierte Daten (z. B. Normen, Ausschreibungsdokumente oder Fachartikel) nicht berücksichtigt. Eine flexible und dynamische Erweiterbarkeit des Modells durch strukturierte oder unstrukturierte Daten ist nicht gegeben.

Zusammenfassend kann gesagt werden, dass ein Handlungsbedarf zur Konzeptionierung und prototypischen Umsetzung eines Systemmodells für die Produktentwicklung besteht, die die aufgestellten Anforderungen erfüllt. Des Weiteren sollen auch die höheren Ebenen des System of Interest in das Systemmodell integriert werden können. Dies ist nach aktuellem Stand nicht gegeben.

Abschließend soll erwähnt werden, dass andere Forscher mehr Forschung zu Data-Science in der Produktentwicklung fordern (Chiarello et al., 2021). Daher wird in dieser Arbeit im Folgenden ein Konzept für ein Systemmodell vorgestellt, das die aufgestellten Anforderungen erfüllt und Daten

sowie Informationen über alle Ebenen des System of Interest integriert. Auf Basis dieses Systemmodells können Technologien wie Data-Science und KI eingesetzt werden.

Das Konzept der vorliegenden Arbeit wird alle Anforderungen erfüllen. Dazu wird der Ansatz des „Engineering-Graph“ vorgestellt, der auf Basis von Graph Datenbanken in der Lage ist, Datenbanken der Produktentwicklung zu verbinden, Modelle der Produktentwicklung zu integrieren und externe unstrukturierte Daten sowie frei verfügbares Kontextwissen zu integrieren. Dazu ist der Engineering-Graph flexibel und dynamisch erweiterbar. In Kapitel 4 wird das Konzept allgemein vorgestellt, in Kapitel 5 wird eine beispielhafte Anwendung des Engineering-Graph beschrieben.

4 Engineering-Graph

In diesem Kapitel wird das Konzept des Engineering-Graph vorgestellt. Dieses Konzept wurde im Rahmen der Dissertation erarbeitet, um den in Kapitel 3.4 vorgestellten Handlungsbedarf zu adressieren. Es ermöglicht die Verbindung vieler Datenquellen zu einem Systemmodell und schafft dadurch eine KI-fähige Wissensbasis. Im Folgenden wird zunächst ein Überblick über den Engineering-Graph gegeben, bevor auf den Aufbau und die Erweiterung, die Nutzung sowie den Betrieb und die Pflege eingegangen wird.

4.1 Engineering-Graph im Überblick

In diesem Unterkapitel wird der Engineering-Graph im Überblick dargestellt. Ein Engineering-Graph ist die Anwendung von Graph-Datenbanken in einer Engineering-Umgebung. Er schafft damit ein Systemmodell, das den Einsatz von KI-Technologien zur Unterstützung von Ingenieur_innen bei der Bewältigung der Komplexität ermöglicht.

Der Engineering-Graph verbindet die Daten aus SysML-Modellen, traditionellen technischen Anwendungen wie domänenspezifischen Systemen, PLM- und MES-Systemen, Unternehmensdaten aus ERP-Systemen und Felddaten aus IoT-Systemen. Diese Systeme haben alle ein eigenes Schema und eine eigene Logik, wodurch die Interoperabilität und die Fähigkeit der Systeme, miteinander zu arbeiten, eingeschränkt wird. Sollen Daten aus allen Ebenen des System-of-Interest im Kontext miteinander betrachtet und analysiert werden, müssen diese in einem Modell zusammengeführt werden. Der Engineering-Graph ist ein Konzept für ein solches Modell. Die Fähigkeit von Graph-Datenbanken, Daten mit unterschiedlichen Schemata einzubeziehen, ist entscheidend und ermöglicht die Erstellung eines Engineering-Graph. Mit dieser Fähigkeit können die Informationen aus der Ebene ‚Product-System‘, die im PLM-System vorliegen, mit Informationen der Ebene ‚Product Environment‘, die als SysML-Modell vorliegen, und mit Informationen der Ebene ‚Society-Eco-System‘, die zum Beispiel als Normen oder als Veröffentlichung der WHO im semantischen Web vorliegen, verbunden werden. Abbildung 24 veranschaulicht dies beispielhaft.

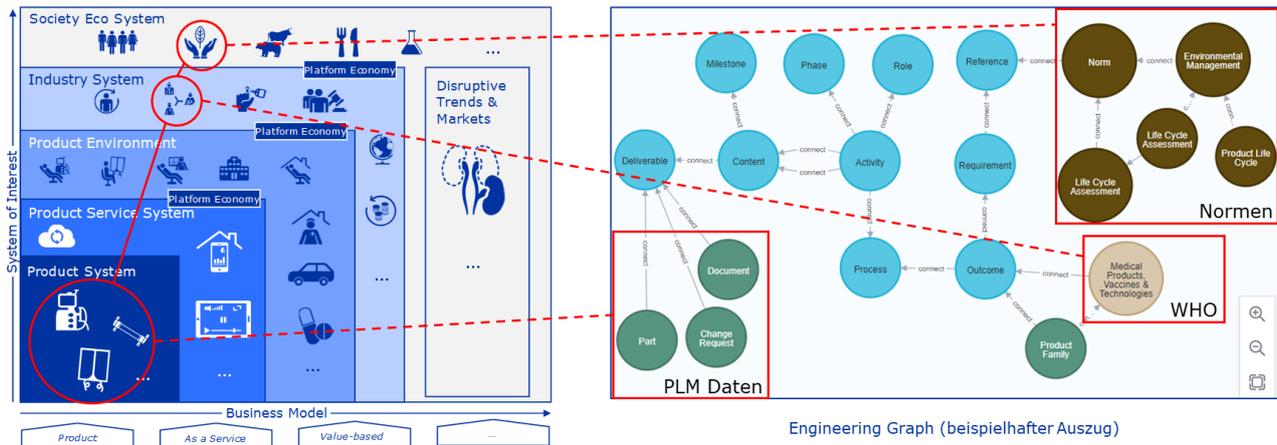


Abbildung 24. Engineering-Graph verbindet Informationen über alle Ebenen des System-of-Interest nach (Schweitzer et al., 2023b)

Im Engineering-Graph liegt der Fokus auf den Verbindungen der Daten, nicht auf den Daten selbst. Es werden keine Daten aus den spezialisierten Systemen dupliziert. Nur die Metadaten wie Name, Erstellungsdatum usw. werden übernommen und mit anderen Daten sowie Informationen verknüpft.

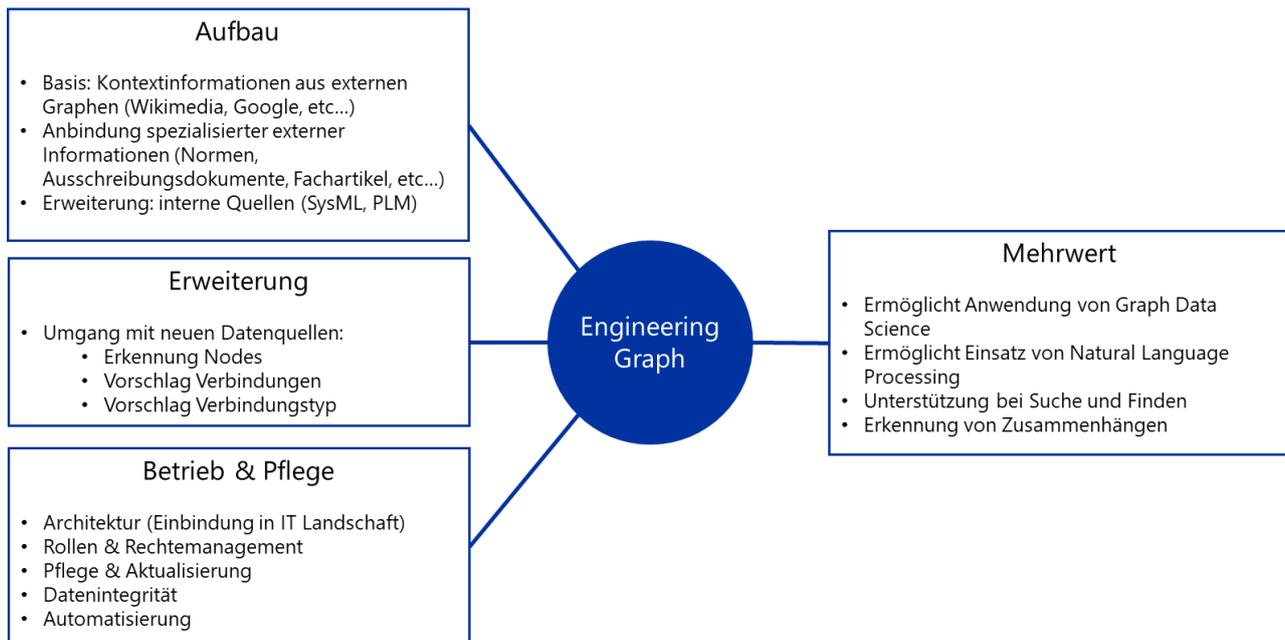


Abbildung 25. Übersicht Engineering-Graph

Abbildung 25 fasst das Konzept des Engineering-Graph zusammen. Im Folgenden wird der Engineering-Graph vorgestellt. Zunächst wird erklärt, wie der initiale Aufbau und die Erweiterung des Graphen durchgeführt werden können. Im Anschluss wird vorgestellt, wie der Engineering-Graph operativ eingesetzt und gepflegt werden kann. Zuletzt wird erklärt, welchen Mehrwert der Einsatz des Engineering-Graph bietet und welche Analysen er ermöglicht. Dazu werden verschiedene Anwendungsfälle dargestellt.

4.2 Aufbau und Erweiterung des Engineering-Graph

In diesem Unterkapitel wird das Vorgehen zum Aufbau und zur Erweiterung des Engineering-Graph beschrieben. Der initiale Aufbau kann bereits vor dem Start eines Entwicklungsprojektes erfolgen. Während der Entwicklung über die Phasen des Lebenszyklus wird der Graph dann um die erstellten Produktdaten erweitert. Abbildung 26 zeigt, wie der Graph zunächst aus externem Wissen besteht, das sukzessiv um das klassische Produktwissen erweitert wird.

4.2.1 Aufbau des Engineering-Graph

Im Folgenden wird das Vorgehen zum initialen Aufbau des Graphen beschrieben. Dabei wird mit frei verfügbarem Kontextwissen und spezialisiertem Industrierwissen begonnen, bevor unternehmensinterne Quellen angebunden werden. Im Anschluss wird erklärt, wie die Verlinkung durchgeführt wird und wie mit weiteren Quellen umgegangen werden kann, die während der Nutzungsphase entdeckt werden.

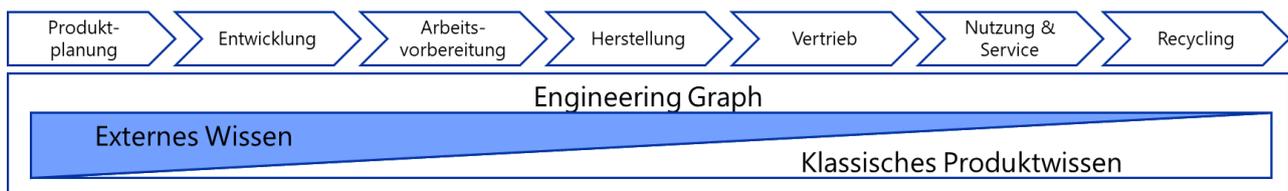


Abbildung 26. Aufbau des Engineering-Graph nach (Schweitzer et al., 2023a)

Zu Beginn eines neuen Entwicklungsprojektes liegen typischerweise noch keine klassischen Produktdaten vor, da diese erst im Laufe der Entwicklung erzeugt werden. Damit der Engineering-Graph Projekte in frühen Phasen unterstützen kann, wird er mit externem Wissen über den Kontext, in dem das zu entwickelnde Produkt später eingesetzt wird, befüllt. Dafür werden relevante Daten aus den beiden höchsten Ebenen des System of Interest, dem ‚Society-Eco-System‘ und dem ‚Industry-System‘, miteinander verbunden. Diese Daten sind frei verfügbar, daher kann gut darauf zugegriffen und ein erster Graph erzeugt werden. Die im Projektverlauf erzeugten Produktdaten können an die geschaffene Basis hinzugefügt werden. Auf diese Art kann der Engineering-Graph direkt zu Beginn eines Projektes einen Mehrwert liefern, da er bereits Daten enthält.

Im ersten Schritt werden die Daten aus dem ‚Society-Eco-System‘ hinzugefügt und miteinander verbunden. Dazu kann auf den Wikimedia Knowledge Graph zurückgegriffen werden (Wikimedia, 2021). Darin sind Objekte der realen Welt und ihre Verbindungen enthalten, wie sie in Wikipedia vorkommen. Außerdem sind die Artikel aus Wikipedia mit den Objekten verknüpft, sodass abgefragt werden kann, ob ein Objekt einen Artikel hat oder nicht. Die Datenbank wird von Forscher_innen gepflegt, erweitert und öffentlich zur Verfügung gestellt. Hier können Abfragen zu einem für das Unternehmen interessanten Objekt erstellt werden. Es kann definiert werden, dass ausgehend von

diesem Objekt alle verbundenen Objekte und ihre Relationen exportiert werden. Diese können genutzt werden, um den Engineering-Graph zu erstellen. Ein Beispiel wird in Kapitel 5.2 in Abbildung 38 vorgestellt.

Mit der Einbeziehung von Wikimedia Daten wird die in Kapitel 3.4 vorgestellte Lücke bei der Erfüllung von Anforderung A4 adressiert, da hier unstrukturierte, frei verfügbare Daten integriert werden, um diese mit Produktdaten zu vernetzen.

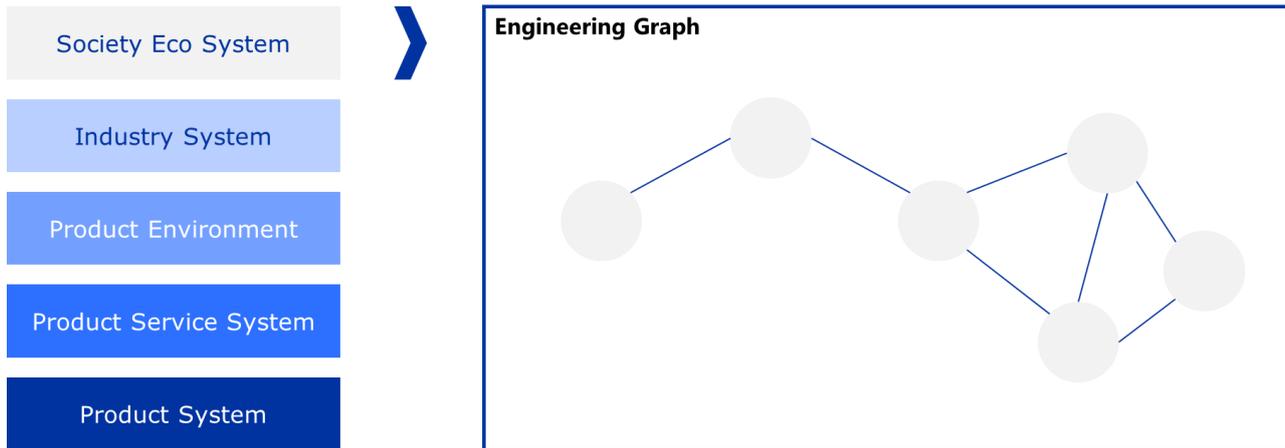


Abbildung 27. Engineering-Graph mit Daten aus dem 'Society-Eco-System'

Weitere frei verfügbare Quellen können angebunden werden, indem auch der Google Knowledge Graph (Google, 2021) abgefragt wird. Auch hier kann der Export durchgeführt werden und die Daten können zum Engineering-Graph hinzugefügt werden. Weitere Quellen aus dem Society-Eco-System wie ökonomische Daten der Weltbank können nach Bedarf hinzugefügt werden. Abbildung 27 zeigt exemplarisch einen Engineering Graph, der aus Daten aus der Ebene 'Society-Eco-System' besteht.

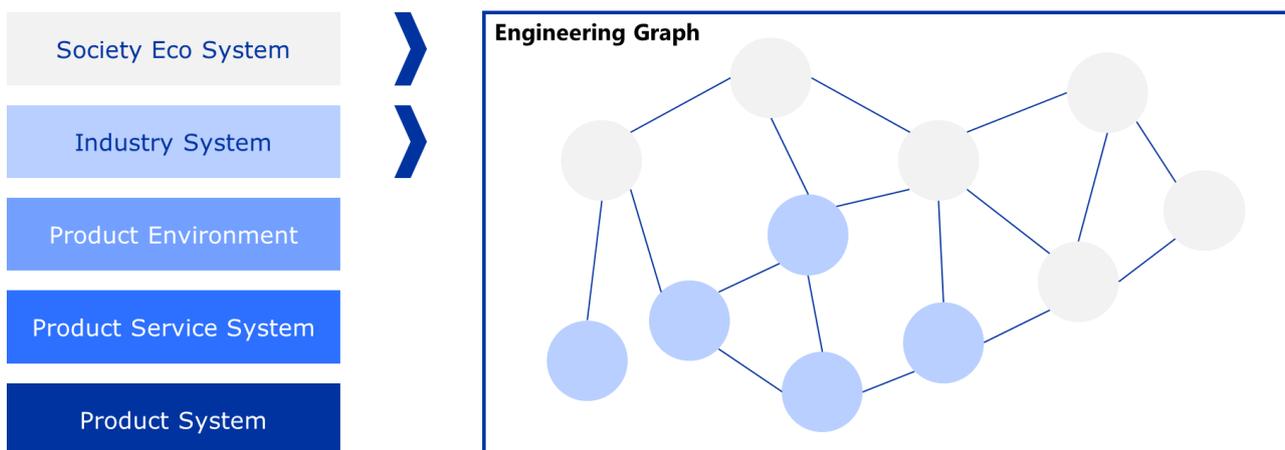


Abbildung 28. Engineering-Graph mit 'Society-Eco-System' und 'Industry-System'

Im zweiten Schritt werden, wie in Abbildung 28 dargestellt, spezialisierte externe Quellen des 'Industry-System' hinzugefügt. Dies sind spezialisierte Quellen, die für die jeweilige Branche, in der ein Unternehmen tätig ist, relevant sind. Im späteren Anwendungsfall, der in Kapitel 5.2 vorgestellt

wird, ist dies die Medizintechnik. Wichtige Quellen der Medizintechnik sind beispielsweise die Health-System-Building-Blocks, die von der WHO veröffentlicht wurden (World Health Organization, 2021). Weitere Quellen können Veröffentlichungen von Krankenkassen oder Ausschreibungsdokumente sein, in denen Krankenhäuser die Anschaffung von neuem Equipment ausschreiben (Tenders Electronic Daily, 2023). Außerdem kann der Engineering-Graph um branchenrelevante Normen erweitert werden. Dies sind beispielsweise Normen der Medizintechnik wie die ISO 62304 (ISO 62304, 2023), Normen der LCSA wie die ISO 14040 (DIN EN ISO 14040, 2009) und ISO 14044 (DIN EN ISO 14044, 2006) sowie Informationen zur Zulassung von der Food and Drug Association (FDA) oder dem TÜV.

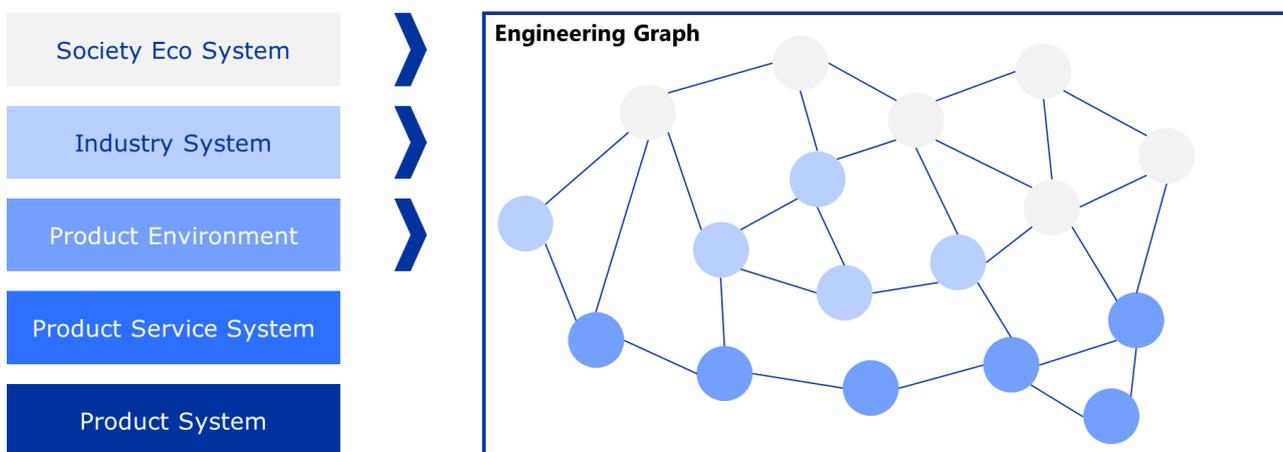


Abbildung 29. Engineering-Graph mit 'Product-Environment'

Der dritte Schritt ist die Integration von Daten aus dem ‚Product-Environment‘, wie in Abbildung 29 dargestellt. Dies können Informationen über die Umwelt am Einsatzort des Produktes sein. So kann abgeschätzt werden, an welchen Orten das Produkt eingesetzt wird und somit z. B. welche Luftdrücke, Luftfeuchtigkeiten, Temperaturen oder Sprachen unterstützt werden sollen. Weitere Informationen aus dieser Ebene sind Produkte und Services anderer Hersteller, mit denen das eigene Produkt in Beziehung steht. Daraus ergeben sich Schnittstellen und mögliche produktübergreifende Arbeitsabläufe und Wertschöpfungsketten, die systematisch in Betracht gezogen werden können.

Der vierte Schritt ist die Anbindung von internen Quellen wie dem PLM-, MES-, IoT- und dem ERP-System. Außerdem werden die vorhandenen Modelle, die z. B. im Rahmen der MBSE-Aktivitäten in SysML erzeugt wurden, angebunden. Das Ziel ist nicht die Replikation der Daten im Engineering-Graph, sondern die Abbildung der Metadaten. Dadurch wird der Fokus auf den Verbindungen der Daten zwischen den Systemen und mit den existierenden, externen Daten gelegt. Abbildung 30 zeigt einen beispielhaften Engineering-Graph mit Daten aus allen Ebenen des System-of-Interest.

Auch Daten aus der Wertschöpfungskette können hinzugefügt werden. Diese sind Daten aus den Systemen der Zulieferer und Partner in der Produktentwicklung. Außerdem kann der Graph durch

Daten aus spezialisierten Datenbanken, wie sie zum Beispiel für ein LCA erforderlich sind (LifeCycleInitiative.org, 2022), erweitert werden.

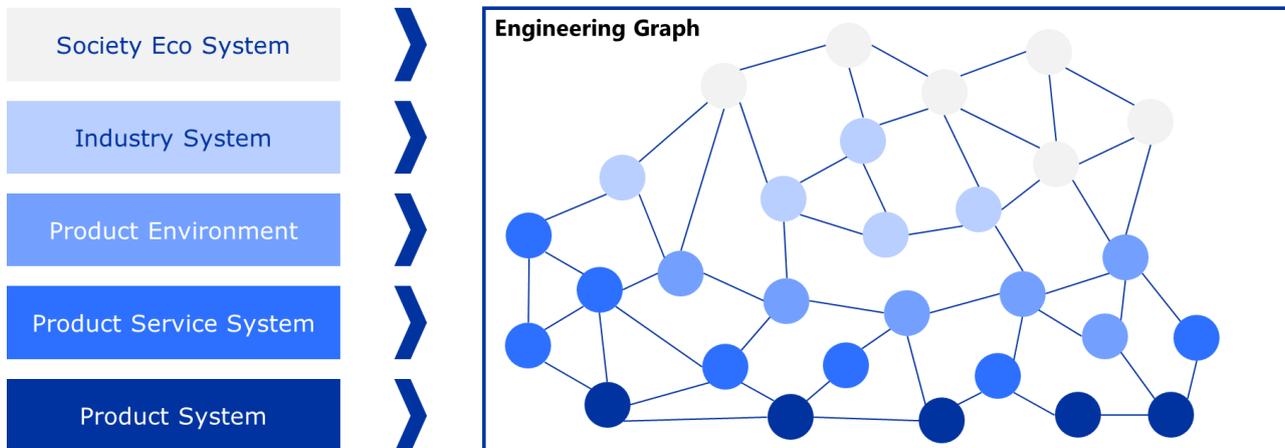


Abbildung 30. Engineering-Graph mit allen Ebenen des System-of-Interest

Durch die Schaffung der Verbindung der Daten und Informationen über alle Ebenen des System-of-Interest wird die in Kapitel 3.4 vorgestellte Lücke bei Anforderung 5 „Das Systemmodell soll frei verfügbares Kontextwissen (wie aus dem Wikimedia Knowledge Graph) mit den bestehenden Daten verbinden können.“ geschlossen. Das frei verfügbare Kontextwissen kann durchgängig mit den Produktdaten verknüpft werden.

4.2.1.1 Erweiterung des Engineering-Graph

Ein generelles Vorgehen, wie weitere Quellen angebunden werden können, wird in Abbildung 31 dargestellt und im Folgenden beschrieben (Schweitzer et al., 2023a).

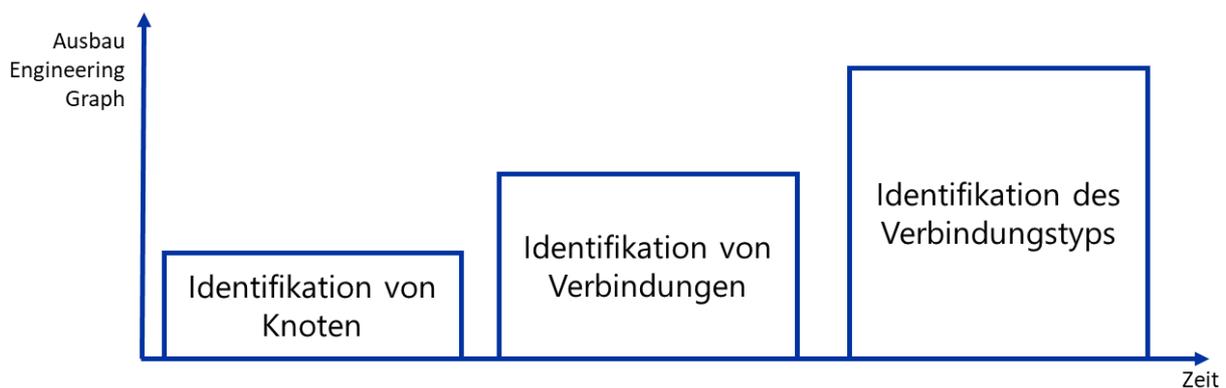


Abbildung 31. Schritte zur Erweiterung des Engineering-Graph nach (Schweitzer et al., 2023a)

Der erste Schritt ist die Identifikation neuer Knoten, die dem Graphen hinzugefügt werden können. Die neue Quelle kann als ein Knoten hinzugefügt werden, was die einfachste Lösung ist. Ein Beispiel wäre, dass eine neue Norm, die hinzugefügt werden soll, durch einen Knoten im Graphen dargestellt wird. Dies kann ausreichend sein, aber in manchen Fällen muss sie weiter untergliedert werden. In

diesem Beispiel können Unterabschnitte der Norm dem Graphen als separate Knoten hinzugefügt werden, die mit dem Hauptknoten in Beziehung stehen. Auf diese Weise ist es möglich, später ein technisches Objekt mit einem bestimmten Abschnitt der Norm in Verbindung zu bringen. Dies ermöglicht tiefgreifendere Analysen, zum Beispiel, wenn eine Norm aktualisiert wird und die Aktualisierung nur in einigen Abschnitten der Norm vorgenommen wird. Wenn die Unterabschnitte mit technischen Objekten verbunden sind, kann automatisch festgestellt werden, welche technischen Objekte von der Aktualisierung betroffen sind.

Der zweite Schritt ist die Identifikation von Verbindungen. Ziel ist es, die ermittelten neuen Knoten mit den bestehenden zu verbinden. Dies kann auf verschiedene Weise geschehen. Die erste Möglichkeit ist eine manuelle Verbindung, die viel Arbeit und detailliertes Fachwissen erfordert, um zu wissen, wie der neue Knoten mit den bestehenden in Beziehung steht. Aus der manuellen Arbeit ergibt sich eine hohe Fehleranfälligkeit, da insbesondere in großen Systemmodellen Verbindungen leicht übersehen werden können. Die zweite Möglichkeit ist automatisiert und basiert auf Schlüsselwörtern. Dies kann mithilfe der Schlüsselwort-Extraktionsfunktion der natürlichen Sprachverarbeitung erfolgen (Beliga, 2014). Nachdem die Schlüsselwörter der neuen Quelle gefunden wurden, kann der Engineering-Graph nach diesen Schlüsselwörtern durchsucht werden und es können Verbindungen zu den Ergebnissen vorgeschlagen werden. Der dritte Weg basiert auf den aktuellen Verbindungen eines Knotens. Die Graph-Datenbank kann maschinelles Lernen nutzen, um zu verstehen, wie die Knoten miteinander verbunden sind, und auf dieser Grundlage neue Verbindungen vorschlagen (Neo4J, 2022c).

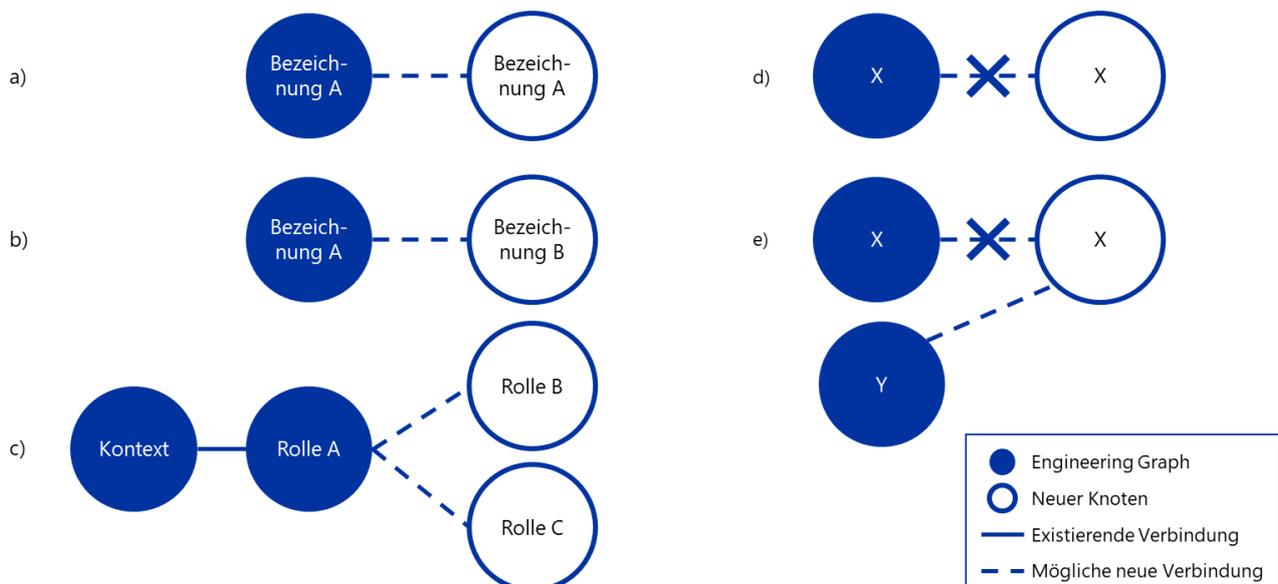


Abbildung 32. Identifikation des Verbindungstyps nach (Schweitzer et al., 2023a)

Der dritte Schritt ist die Identifikation des Verbindungstyps. Der Verbindungstyp gibt die Art der Verbindung zwischen zwei Objekten an. So kann z. B. ein Unterkapitel einer Norm mit der ganzen

Norm durch den Typ „Element von“ oder ein bestimmtes Anforderungsdokument mit dem allgemeinen Knoten „Dokument“ durch den Typ „ist ein“ verbunden sein.

Zur Identifikation des Verbindungstyps ist eine Fallunterscheidung erforderlich, wie in Abbildung 32 dargestellt. Im Fall a) existiert der neu identifizierte Knoten bereits im Engineering-Graph und kann daher mit dem bestehenden Knoten zusammengeführt werden. Beide Knoten haben die gleiche Bezeichnung und repräsentieren das gleiche Objekt. Im Fall b) stellen beide Knoten dasselbe Objekt dar, haben aber unterschiedliche Bezeichnungen. Im Fall c) könnte sowohl Rolle B als auch Rolle C die Rolle A ausfüllen. Eine Verbindung der ersten Ebene muss zusätzlich berücksichtigt werden, um den richtigen Kontext zu liefern. Unter Berücksichtigung des Kontextes kann dann entschieden werden, ob Rolle B oder Rolle C zuzuordnen ist. Im Fall d) haben beide Knoten denselben Namen, stellen aber nicht dasselbe Objekt dar. Hier sollte keine Verbindung hergestellt werden. Fall e) baut darauf auf, dass es einen anderen Knoten im bestehenden Engineering-Graph gibt, der dasselbe Objekt wie der neue Knoten darstellt, aber anders heißt. In allen Fällen kann die Verbindung, nachdem sie geklärt ist, eine neue Eigenschaft erhalten, die den Verbindungstyp angibt.

Durch die vorgestellte flexible und dynamische Erweiterung des Engineering-Graph wird die in Kapitel 3.4 vorgestellte Lücke bei der Erfüllung von Anforderung A7 adressiert. Strukturierte und unstrukturierte Daten können dem Engineering-Graph hinzugefügt werden, um den Graph zu vergrößern und den Nutzen zu erweitern.

Es ist zu beachten, dass die Erweiterung des Engineering-Graph durch Automatisierungswerkzeuge unterstützt werden kann, jedoch mit bestehender Technologie keine vollständige Automatisierung erreicht werden kann. Das System sollte nur Vorschläge machen, die dann von menschlichen Domänenexpert_innen überprüft werden, die die Verbindungen verifizieren und dem Graphen hinzufügen. Auch der Verbindungstyp kann durch das System nur vorgeschlagen werden, die finale Entscheidung muss durch Menschen verifiziert und umgesetzt werden.

Technisch kann das Konzept in jeder Graph-Datenbank umgesetzt werden. Dafür gibt es verschiedene Sprachen und Anbieter, die genutzt werden können. Für diese Arbeit wurde Neo4J verwendet, da das System leicht zugänglich und gut dokumentiert ist (Neo4J, 2023a).

4.3 Betrieb und Pflege des Engineering-Graph

Der Aufbau, der Betrieb und die Pflege des Graphen sind mit Aufwand verbunden. Sie müssen von Personen, die bestimmte Rollen im Unternehmen bekleiden, ausgeführt werden. Damit der Engineering-Graph eingesetzt wird, muss er sich für Unternehmen ökonomisch rentieren. Dies bedeutet, dass der Mehrwert, den der Graph liefert, größer sein muss als der Aufwand, der betrieben werden muss, um den Graphen zu erstellen und zu pflegen. In diesem Unterkapitel werden die

Aufgaben beschrieben, die für den Betrieb und die Pflege des Engineering-Graph notwendig sind. Im folgenden Unterkapitel wird dann der Nutzen vorgestellt, bevor im finalen Unterkapitel eine wirtschaftliche Betrachtung vorgenommen wird.

Technisch muss der Engineering-Graph in die bestehende IT-Landschaft des Unternehmens eingebunden werden. Für Unternehmen, die ihre Engineering-Systeme auf eigenen Servern betreiben, ist es sinnvoll, auch den Engineering-Graph auf eigenen Servern zu betreiben. Andere Unternehmen, die auf Cloud-Anbieter zurückgreifen, um ihre Systeme zu betreiben, werden den Engineering-Graph auch dort betreiben wollen. Da der Engineering-Graph eine Graph-Datenbank ist, die für diese Arbeit prototypisch in Neo4J umgesetzt wurde, sind beide Varianten technisch möglich. Die Entscheidung, welche Variante eingesetzt werden soll, muss im individuellen Einzelfall in Zusammenarbeit mit den für die IT-Architektur verantwortlichen Personen getroffen werden.

Voraussetzungen für den Einsatz des Engineering-Graph sind technische Schnittstellen zu allen unternehmensinternen Systemen, die angebunden werden sollen. Dies wird benötigt, um die Daten abzugleichen und Änderungen einpflegen zu können. Eine weitere Voraussetzung ist die Schnittstelle zu den Autorensystemen, in denen die Informationen und Hilfestellungen aus dem Engineering-Graph angezeigt werden sollen.

Um die Aufgaben auszufüllen, die zur Erstellung und Nutzung des Engineering-Graph notwendig sind, wird die Einführung einer neuen Rolle vorgeschlagen: des Graph Architect. Diese soll die in Kapitel 3.1.2.1 vorgestellten MBSE-Rollen erweitern. Abbildung 33 zeigt die Aufgaben der neuen Rolle sowie deren Zusammensetzung aus Systems Engineering und Informatik. Wichtig ist dabei, dass nicht eine Person eine Rolle ausfüllen muss, sondern dass eine Rolle von verschiedenen Personen übernommen werden kann oder eine Person mehrere Rollen ausfüllen kann. Im Folgenden werden die Aufgaben und Verantwortlichkeiten bei der Erstellung sowie Pflege des Engineering-Graph zusammengefasst. Zunächst werden die Aufgaben zur Erstellung und Erweiterung des Engineering-Graph beschrieben, dann die Aufgaben der Pflege und des Rollenmanagements, die während der Nutzung erforderlich sind.

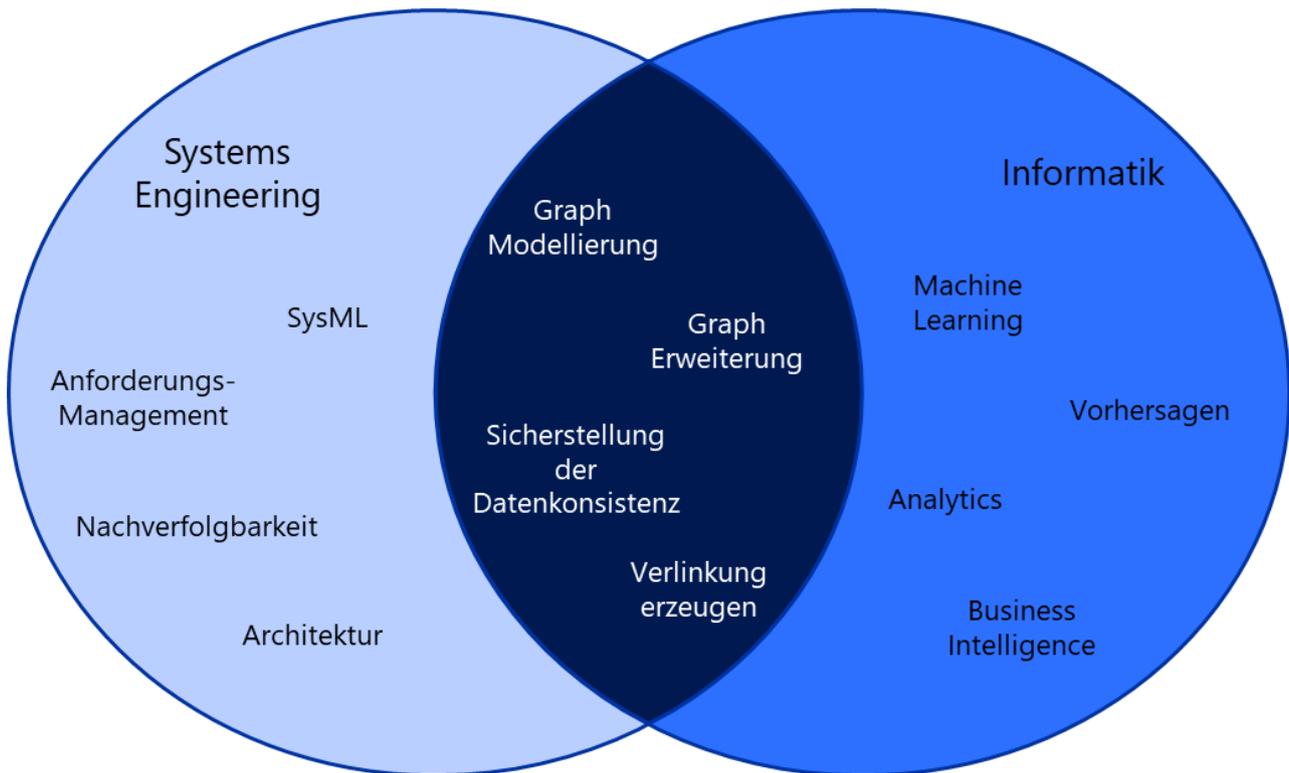


Abbildung 33. Aufgaben der neuen Rolle des Graph Architects

Die Hauptaufgabe von Graph Architects ist die Verbindung der Daten zur Erstellung des Engineering-Graph. Der erste Schritt ist die Verbindung der Daten aus den IT-Systemen, die in der Produktentwicklung eingesetzt werden. Wie oben beschrieben, sind die Engineering-Daten derzeit in verschiedenen Systemen wie PLM, ERP, IoT und domänenspezifischen Tools, zum Beispiel für die Simulation, gespeichert. Darüber hinaus gibt es verschiedene Modelle, die bei der Entwicklung verwendet werden. Dies können SysML-Produktmodelle aus MBSE, Simulationen in MATLAB oder Failure-Mode-and-Effects-Analysis-Modelle für die geometrische Konstruktion sein. Die Aufgabe von Graph Architects besteht darin, die Verbindungen zwischen den Datenpunkten in diesen Datenbanken herzustellen, ohne die Informationen zu kopieren.

Während der produktiven Nutzung des Engineering-Graph müssen die Informationen bei Bedarf aktualisiert werden. Dies kann beispielsweise der Fall sein, wenn eine neue Version einer Norm veröffentlicht wird oder wenn eine Non-Government-Organization (NGO) wie die WHO oder die Weltbank einen aktualisierten Datensatz herausgibt. Darüber hinaus müssen auch Daten, die aus Wikimedia oder dem Google Knowledge Graph importiert wurden, regelmäßig aktualisiert werden. Die Aufgabe von Graph Architects ist es, die richtigen Aktualisierungsintervalle für das jeweilige Unternehmen zu definieren und die Aktualisierungen wie definiert durchzuführen.

Bei der Aktualisierung von Informationen müssen regelmäßig Datenqualitätsprüfungen erfolgen. Es muss sichergestellt werden, dass der erstellte Graph gut aufgebaut und vernetzt ist. Dies kann

mithilfe der Gemeinschafts- und Zentralitätsalgorithmen der Graph-Data-Science erfolgen. Darüber sollen Graph Architects mit ihrem menschlichen Verständnis der Zusammenhänge Plausibilitätsprüfungen durchführen. Werden dadurch Unstimmigkeiten erkannt, können entsprechende Maßnahmen definiert und durchgeführt werden.

Der Graph sollte kontinuierlich verbessert werden, indem neue Beziehungen erkannt und hinzugefügt werden. Dies kann durch maschinelles Lernen, wie oben beschrieben, geschehen. Zusätzlich können Dokumente, die dem Graphen hinzugefügt werden, mit NLP analysiert werden, um die Schlüsselwörter zu finden. Dann kann der Graph nach diesen Schlüsselwörtern durchsucht werden und es können neue Verbindungen zwischen Dokumenten oder Objekten vorgeschlagen werden, bei denen gleiche oder ähnliche Schlüsselwörter gefunden wurden. Graph Architects führen diese Analysen durch und prüfen die Vorschläge des Systems, um zu entscheiden, welche neuen Verbindungen dem Graphen hinzugefügt werden sollen und welche nicht. Dabei definieren Graph Architects auch den Verbindungstyp.

Schließlich muss eine Benutzer- und Rollenverwaltung eingerichtet und gepflegt werden. Das Ingenieurteam, das die Produktentwicklung vornimmt, sollte Lesezugriff haben und die Informationen erhalten, die der Engineering-Graph bietet. Die Graph Architects sollten Lese- und Schreibrechte haben, damit sie ihre Arbeit am Engineering-Graph durchführen können. Es muss individuell im Unternehmen entschieden werden, welche weiteren Rollen oder Personengruppen eine Art des Zugriffs auf den Engineering-Graph benötigen. Dabei muss das Rollen- und Berechtigungskonzept des jeweiligen Unternehmens beachtet und eingehalten werden.

Mitarbeiter_innen, die die vorgeschlagene neue Rolle des Graph Architect ausfüllen, benötigen eine Kombination von Fähigkeiten aus den Bereichen Systems-Engineering und Informatik. Sie arbeiten mit technischen Daten aus SysML-Modellen, PLM-Datenbanken, ERP-Systemen und disziplinspezifischen Werkzeugen und Datenbanken, die für Simulation und Tests verwendet werden. Daher müssen sie diese Daten verstehen und wissen, wie sie aus verschiedenen Silos miteinander verbunden sind.

Hinsichtlich der Informatik benötigen sie Fähigkeiten in den Bereichen Datenanalyse, Business-Intelligence und maschinelles Lernen. Diese Fähigkeiten werden eingesetzt, um den Graphen weiter zu verbessern, NLP anzuwenden und Qualitätsprüfungen durchzuführen.

Um diese vielfältigen Fähigkeiten zu erlangen, können Graph Architects entweder Systems Engineers sein, die eine spezielle Ausbildung in Datenanalyse erhalten haben, oder Informatiker_innen, die die notwendige Ausbildung in Systems-Engineering erhalten. Diese Ausbildung kann *on the job* erfolgen, indem Teams aus Systems Engineers und Informatiker_innen gebildet werden, die offen und

interessiert sind, das jeweils andere Fachgebiet kennenzulernen. Außerdem können Zertifizierungsprogramme für Systems-Engineering und Informatik genutzt werden.

Die Arbeit an der Schnittstelle zweier Fachgebiete erfordert ein tiefes Verständnis für die Ziele, die das Team erreichen will, und eine offene Einstellung, um zu lernen und zu wachsen, wenn man mit Menschen aus einem anderen Berufsfeld zusammenarbeitet.

Wird der Graph in einem Unternehmen langfristig eingesetzt, können einige Schritte automatisiert werden, um die Erweiterung, den Betrieb und die Pflege zu vereinfachen. Dadurch verringert sich der manuelle Aufwand und der ökonomische Nutzen erhöht sich. Im Folgenden werden Möglichkeiten der Automatisierung aufgezeigt.

Eine erste Möglichkeit der Automatisierung ist das Finden von neuen Quellen. Die manuelle Arbeit wäre die Bearbeitung von vorgeschlagenen Quellen, die Ingenieur_innen noch benötigen oder als wünschenswert betrachten. Eine Automatisierung könnte durch die systematische Suche im Web nach bestimmten Schlagwörtern erfolgen. Dadurch könnten Listen erzeugt werden, die von den Graph Architects im Hinblick auf die Frage bewertet werden, ob die Quellen hinzugefügt werden sollen oder nicht. Die Schlagwörter könnten stark vernetzte Knoten sein, wo Stärken des Netzes ausgebaut werden sollen, oder schwach vernetzte Knoten, wenn Schwächen des Netzes beseitigt werden sollen.

Eine zweite Möglichkeit der Automatisierung ist die Anbindung neuer Quellen. Nachdem diese identifiziert wurden, könnten die Daten automatisch zum Graphen hinzugefügt werden. Eine initiale Verlinkung könnte zu dem Knoten erfolgen, über den die Suche gestartet wurde. Ein erster Schritt könnte darin bestehen, einen Knoten einer hohen Ebene zum Netz hinzuzufügen. Dies könnte zum Beispiel ein Knoten sein, der eine neue Norm enthält.

Die Konkretisierung eines neu hinzugefügten Knotens einer hohen Ebene könnte auch automatisiert werden. Dazu könnte zum Beispiel die Norm in ihre Kapitel zerlegt werden. Neue Knoten würden dann jeweils ein Kapitel repräsentieren. Diese wären mit dem ‚Norm-Knoten‘ über eine Beziehung verbunden.

Die Erweiterung des Graphen durch stärkere Vernetzung, also das Hinzufügen weiterer Verbindungen, könnte auch automatisiert werden. Dazu wurden oben bereits zwei Konzepte vorgestellt, wie dies geschehen könnte. Eines könnte durch maschinelles Lernen auf Basis der aktuellen Verbindungen neue vorschlagen und das zweite könnte durch NLP semantisch nahe Knoten finden und eine Verbindung vorschlagen.

Die Aktualisierung der Quellen, die bereits in den Engineering-Graph aufgenommen wurden, könnte ebenfalls automatisiert werden. Dabei könnten entweder periodisch alle Quellen auf neue Versionen

untersucht werden oder die Veröffentlichung einer neuen Version könnte die Aktualisierung direkt auslösen.

Das aktuelle Konzept des Engineering-Graph beinhaltet noch keine Automatisierung. Auch die prototypische Umsetzung, die in Kapitel 5 vorgestellt wird, beinhaltet noch keine Automatisierung. Ziel in dieser Arbeit ist das Aufzeigen der Möglichkeit zur Schaffung eines Systemmodells zum Einsatz von KI zur Unterstützung von Ingenieur_innen bei der Bewältigung der Komplexität.

Ein weiteres bedeutendes Thema während des Betriebs ist die Sicherstellung der Datenintegrität. Es muss gewährleistet sein, dass der Engineering-Graph ein korrektes Modell der Umwelt beinhaltet. Dies bedeutet, dass es ein großes Risiko ist, neue Quellen ungeprüft hinzuzufügen. Insbesondere wenn für den initialen Aufbau des Graphen Modelle von Wikimedia oder Google zum Einsatz kommen, könnten diese Fehler beinhalten. Allerdings muss auf solche Modelle zurückgegriffen werden, da ansonsten der Aufwand zu hoch sein würde.

Um die Datenintegrität zu gewährleisten, sollten bei allen Automatisierungen immer Vorschläge automatisch erstellt werden, die dann von den Graph Architects geprüft und freigegeben werden. Dadurch ist gewährleistet, dass ein Mensch die Entscheidung trifft, die Verantwortung übernimmt und eine gewisse Prüfung vornimmt.

4.4 Nutzung des Engineering-Graph

In diesem Unterkapitel wird der Einsatz des Engineering-Graph beschrieben. Zunächst werden die verschiedenen Analysen, die durch die Graph-Datenbank ermöglicht werden, erklärt. Im Anschluss wird die Rolle von NLP vorgestellt. Abschließend werden die Möglichkeiten bei der Suche und der Erkennung von Zusammenhängen erörtert.

Eine Graph-Datenbank ermöglicht den Einsatz von Graph-Data-Science (Neo4J, 2022d). Abbildung 14 zeigt vier Kategorien, die damit analysiert werden können: Gemeinschaft, Zentralität, Vorhersage und Analyse der Verbindungen. Die im Folgenden gezeigten Beispiele sind aus der Anwendung entnommen, die in Kapitel 5.3.5 vorgestellt wird.

Gemeinschaft

Weakly Connected Components

```
plm-conf$ CALL gds.wcc.stats('plm-conf1') YIELD componentCount
```

componentCount
1

-> 1 Komponente gefunden

Label Propagation

```
plm-conf$ CALL gds.labelPropagation.stats('plm-conf1') YIELD communityCount, ranIterations, didConverge
```

communityCount	ranIterations	didConverge
7	4	true

-> 7 Gemeinschaften gefunden

Abbildung 34. Graph-Data-Science: Gemeinschaft

Die erste Kategorie sind die Analysen zum Thema Gemeinschaft. Abbildung 34 bietet eine Übersicht der hier eingesetzten Algorithmen. Diese haben zum Ziel, die Struktur des Graphen zu analysieren (Neo4J, 2022b). Durch die Analyse der Struktur kann abgeleitet werden, ob der Graph korrekt erstellt wurde und ein Verständnis für den Aufbau und die enthaltenen Themenfelder gewonnen wird. Dazu werden zunächst die ‚Weakly Connected Components‘ berechnet. Dabei wird betrachtet, ob jeder Knoten im Graphen mindestens eine Verbindung zu einem weiteren Knoten hat. Zudem wird untersucht, ob es sich um einen verbundenen Graphen handelt oder ob es mehrere nicht miteinander verbundene Subgraphen gibt. Als Nächstes kann durch Label-Propagation berechnet werden, aus wie vielen Gemeinschaften ein Graph besteht. Dabei wird jedem Knoten ein zufällig erstelltes Label gegeben. Bei jeder Iteration nehmen die Knoten das Label an, das die meisten ihrer Nachbarn haben. Dadurch erhalten stark vernetzte Knoten schnell das gleiche Label, während das Label sich bei schwächer vernetzten Knoten nicht durchsetzen kann. Eine Gemeinschaft ist dabei eine endliche Menge stark miteinander verbundener Knoten, die zwar mit anderen Gemeinschaften vernetzt sind, allerdings weniger als untereinander. Durch diese Analyse können Themenfelder entdeckt werden, aus denen ein Graph besteht.

Zentralität

Pagerank

```
plm-conf$ CALL gds.pageRank.stream('plm-conf1') YIELD nodeId, score RETURN gds.util.asNode(nodeId).name AS name, score ORDER BY score DESC, name ASC
```

	name	score
1	"Sustainable Development Goals"	4.223685101433106
2	"Climate Action"	2.3925707075683595
3	"Reference"	1.9958044723388675
4	"Norm"	1.8687000000000002
5	"Human Health"	1.4433184794921878

-> „Sustainable Development Goals“ ist der am stärksten verknüpfte Knoten

Abbildung 35. Graph-Data-Science: Zentralität

Abbildung 35 gibt eine Übersicht der zweiten Kategorie: Zentralität (Neo4J, 2022a). Diese gibt Aufschluss auf den zentralen Knoten des Graphen. Ein Knoten wird als zentral gewertet, wenn er der am stärksten vernetzte Knoten ist. Der am häufigsten verwendete Algorithmus zur Berechnung der Zentralität ist Pagerank, der von Larry Page, einem der Google-Gründer, entwickelt wurde (Brin & Page, 1998). Die Analyse kann auf den gesamten Graphen oder auf eine der durch Label-Propagation gefundene Gemeinschaft angewendet werden. Der Pagerank-Wert gibt Aufschluss darüber, welche der Knoten am stärksten vernetzt sind. Aus der starken Vernetzung lässt sich ableiten, dass diese Knoten mit hoher Wahrscheinlichkeit eine hohe Relevanz haben.

Vorhersage

Vorhersage neuer Verbindungen

```
plm-conf$ CALL gds.alpha.ml.linkPrediction.predict.stream('plm-conf1_undirected', { relationshipTypes: ['connect'], modelName: 'lp-model', topN: 50,
```

	Node1	Node2	probability
1.	"Product Family"	"Change Request"	0.49950002516525244
2.	"Product Family"	"Process"	0.49950002516525244
3.	"Product Family"	"Role"	0.49950002516525244
4.	"Product Family"	"Content"	0.49950002516525244
5.	"Product Family"	"Deliverable"	0.49950002516525244

1. Modell wird mit aktuellem Graph trainiert
2. Modell auf Graph anwenden und Vorschläge für neue Verbindungen erhalten

Abbildung 36. Graph-Data-Science: Vorhersage

Die dritte Kategorie wird in Abbildung 36 vorgestellt und umfasst Vorhersagen (Neo4J, 2022c), um neue Verbindungen zu finden. Dazu kann maschinelles Lernen verwendet werden. Ein Modell wird auf Basis des aktuellen Graphen trainiert und das Modell kann Vorhersagen treffen, an welcher Stelle weitere Verlinkungen sein sollten. So können Verbindungen gefunden werden, die nicht im Graphen repräsentiert sind, aber repräsentiert sein sollten. Des Weiteren können neue Verbindungen aufgedeckt werden, die allein durch menschliches Denken nicht aufgedeckt worden wären. Dies kann großen Mehrwert bei der Entdeckung und Aufdeckung von neuen Wirkzusammenhängen bieten.

Eine weitere Einsatzmöglichkeit des Graphen ist die Unterstützung der Suche von Informationen. Internetsuchmaschinen haben einen Graphen im Hintergrund, da dieser mit geringem Rechenaufwand durchlaufen werden kann, um Informationen und Zusammenhänge zu finden. In der Produktentwicklung ist dies ebenfalls von hoher Bedeutung, da Informationen häufig in verschiedenen Systemen abliegen, die daher von Ingenieur_innen durchsucht werden müssen. Der Engineering-Graph bietet eine Möglichkeit, diese Informationen miteinander vernetzt an einer Stelle durchsuchen zu können und zu finden.

Der Engineering-Graph ermöglicht den Einsatz von NLP zur semantischen Analyse der Daten. Dies bedeutet, dass Ähnlichkeiten zwischen Knoten erkannt werden können, nicht nur auf Basis ihrer Verlinkungen (wie oben beschrieben), sondern auch auf Grundlage der semantischen Bedeutung. Diese Analyse ist nicht nur für die Knoten anwendbar, sondern auch für die Verbindungen zwischen den Knoten. Da die Verbindungen semantischer Natur sind, das heißt, sie haben eine Beschreibung des Verbindungstyps, können sie auch durch NLP analysiert werden. Dies ermöglicht die Analyse, welcher Art ein Einflussfaktor sein könnte. Dadurch kann bereits maschinell vorselektiert werden, wie

groß der wahrscheinliche Einfluss auf eine Zielgröße ist und welche Einflussfaktoren damit den größten Hebel zum Erreichen eines gewünschten Wertes haben.

Die Erkennung von Zusammenhängen ist ein weiterer Nutzen des Engineering-Graph. Abbildung 37 zeigt beispielhaft, welche Faktoren in der Medizintechnik einen Einfluss auf das zu entwickelnde Bauteil haben können. Dabei wurden bewusst firmeninterne Einflüsse von benachbarten Bauteilen ausgelassen, um sich auf die höheren Ebenen des System of Interest zu konzentrieren.

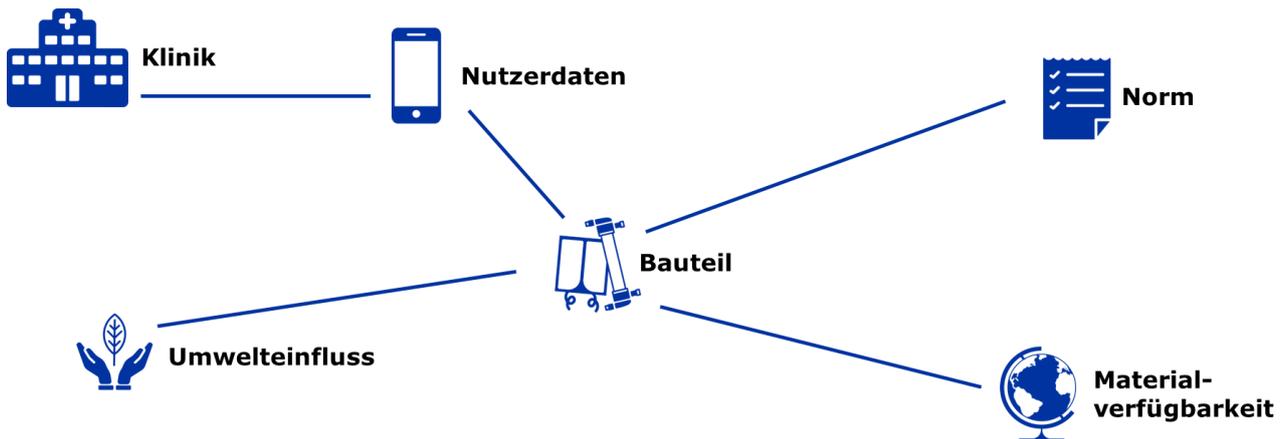


Abbildung 37. Erkennung von Zusammenhängen mit dem Engineering-Graph

Ein zu entwickelndes Bauteil wird durch Normen zur Zulassung, Produktentwicklung und Nachhaltigkeit beeinflusst. Die anzuwendenden Normen müssen während der Entwicklung bekannt sein und Änderungen müssen berücksichtigt werden. Der Graph kann Ingenieur_innen dabei unterstützen, die richtigen Normen und die anzuwendenden Passagen innerhalb der Normen zu identifizieren. Des Weiteren wird ein physisches Bauteil durch das Material beeinflusst, aus dem es hergestellt werden soll. Dabei sind sowohl die Verarbeitbarkeit in der Produktion als auch die Materialverfügbarkeit und die Liefertreue möglicher Zulieferer zu beachten. Auch diese Informationen können durch den Engineering-Graph identifiziert werden. Ein weiteres Feld umfasst die Umwelteinflüsse eines Bauteils. Diese können sozialer, ökonomischer oder ökologischer Art sein. Durch die große Vernetzung des Graphen mit externen Informationen können diese Einflüsse identifiziert werden.

Während der Nutzungsphase eines Produktes fallen Nutzerdaten an. Diese können über den Graphen mit den Produkt- und weiteren Marktdaten vernetzt werden, was eine tiefere Analyse erlaubt. Ist zum Beispiel bekannt, in welcher Klinik die Nutzerdaten erzeugt wurden, können diese über die Region, in der die Klinik betrieben wird, mit demographischen Daten abgeglichen werden. Weitere Einflüsse können, wie oben beschrieben, maschinell erkannt und ausgewertet werden, die allein durch Menschen übersehen worden wären.

Das Ziel ist, dass Ingenieur_innen durch den Engineering-Graph die für sie relevanten Informationen zum richtigen Zeitpunkt angezeigt bekommen. Der Graph enthält alle relevanten Informationen sowie die Verbindungen dieser Informationen untereinander und kann dadurch auf möglicherweise relevante Informationen hinweisen, die ansonsten übersehen worden wären.

Der Engineering-Graph bietet außerdem eine Datengrundlage zur Anwendung von KI zur Optimierung der gesamten Produktentwicklung und zur Identifikation von Potenzialen zur Verbesserung von Produkten. Dadurch können Ingenieur_innen in der komplexen und dynamischen Informationslage unterstützt werden.

4.5 Wirtschaftliche Betrachtung

In diesem Unterkapitel wird eine wirtschaftliche Betrachtung des Engineering-Graph vorgenommen. Die bisher vorgestellten Schritte des Aufbaus und der Pflege des Engineering-Graph haben das Potenzial, einen großen Mehrwert für Unternehmen zu bieten. Allerdings sind sie auch mit einem hohen manuellen Aufwand verbunden. Daher wird im Folgenden beschrieben, in welchen Fällen der Einsatz des Engineering-Graph wirtschaftlich sinnvoll sein kann.

Der Engineering-Graph schafft eine Wissensbasis für den Einsatz von künstlicher Intelligenz in der Produktentwicklung. Es können Verbindungen von Daten aus allen Ebenen des System-of-Interest analysiert werden, um Ingenieur_innen in der Produktentwicklung zu unterstützen. Durch diese Unterstützung erhoffen sich Unternehmen eine Steigerung der Effizienz und damit eine Senkung der Kosten der Produktentwicklung.

Im Fokus liegt die Beherrschung der Komplexität. Es können Einflüsse gefunden und beachtet werden, die ohne maschinelle Unterstützung verborgen geblieben wären. Dadurch kann die Produktqualität gesteigert werden, was sich in einer höheren Preiserzielung am Markt oder in gesunkenen Servicekosten zeigen kann.

Des Weiteren wird eine Impact Analyse über die Ebenen des System-of-Interest unterstützt werden. Dadurch kann der Einfluss sich verändernder Umweltfaktoren oder sozioökonomischer Bedingungen auf das Produkt analysiert werden. So können frühzeitig Maßnahmen definiert und ergriffen werden, um diesen Einflüssen zu begegnen und für jede Situation das richtige Produkt oder die richtige Veränderung in der Produktentwicklung umzusetzen.

Der Aufbau und Einsatz des Engineering-Graph sind mit hohem manuellem Aufwand verbunden. Die in Kapitel 4.2 und 4.3 vorgestellten Aktivitäten müssen durch eigene Mitarbeiter_innen durchgeführt werden. Dadurch können diese in dieser Zeit keine Produkte entwickeln oder an anderen Themen arbeiten.

Für den Aufbau, die Erweiterung und die Nutzung des Engineering-Graph ist der Aufbau neuer Fähigkeiten notwendig. Die Rolle des Graph Architect sollte geschaffen werden und die Personen, die diese Rolle ausfüllen sollen, müssen die beschriebenen Fähigkeiten erwerben. Diese Weiterbildung ist sowohl mit einem zeitlichen Aufwand verbunden als auch mit finanziellen Kosten, wenn Schulungen in Anspruch genommen werden. Des Weiteren muss bei den Mitarbeiter_innen die Bereitschaft bestehen, neue Fähigkeiten zu erwerben und anzuwenden.

Neben der Bereitschaft und der Fähigkeit zum Erwerb neuer Fähigkeiten muss auch die Bereitschaft vorliegen, die aktuelle Arbeitsweise zu verändern. Bei der Änderung der Arbeitsweise und der Einführung neuer Technologien ist ein professionelles organisationales Änderungsmanagement wichtig. Mitarbeiter_innen müssen informiert und geschult werden und diese Informationen und Schulungen müssen vorher konzipiert und erstellt werden. Diese Aktivitäten sind mit zeitlichen und finanziellen Aufwänden verbunden, die in Betracht gezogen werden müssen.

Zuletzt gibt es auch die Möglichkeit, dass neues Personal eingestellt werden muss. Alternativ dazu können die Lastspitzen während der initialen Erstellung des Engineering-Graph auch über externe Berater abgefangen werden. Dies bedingt das Risiko der Abhängigkeit. Wenn keine eigenen Fähigkeiten aufgebaut werden und die Aktivitäten nicht selbst durchgeführt werden, könnte eine Abhängigkeit von externen Partnern entstehen, die langfristig zu hohen Kosten führt. Außerdem wird dadurch ein niedrigeres eigenes Verständnis der Systematik aufgebaut, wodurch auch die Potenziale nicht optimal genutzt werden können.

Eine Gegenüberstellung der Vorteile und Aufwände des Einsatzes des Engineering-Graph ist individuell und muss von jedem Unternehmen einzeln durchgeführt werden. Dabei muss bewertet werden, ob die Potenziale des Einsatzes die Aufwände überwiegen. Außerdem sollte berücksichtigt werden, ob das Unternehmen die Veränderung der Arbeitsweise zu diesem Zeitpunkt verarbeiten kann.

Es ist zu vermuten, dass der Engineering-Graph erst ab einer gewissen Unternehmensgröße ökonomisch sinnvoll ist, da dann die Effizienzgewinne die Aufwände des Aufbaus und der Pflege aufwiegen. Gleichzeitig muss das Produkt des Unternehmens eine gewisse Komplexität haben, da es ansonsten auch mit herkömmlichen Methoden gut zu entwickeln ist.

5 Anwendung des Engineering-Graph in der Medizintechnik

In diesem Kapitel wird der Einsatz des Engineering-Graph in einem Medizintechnik-Unternehmen beschrieben. Der Graph wurde für verschiedene Anwendungsfälle eingesetzt. Nach einer Vorstellung des Unternehmens und einer Erklärung zum Aufbau des Prototyps wird zunächst allgemein beschrieben, wie die Anwendungsfälle kategorisiert werden können. Anschließend werden diese detailliert beschrieben.

5.1 Vorstellung Fresenius Medical Care

Fresenius Medical Care ist ein weltweit führender Anbieter von Produkten und Dienstleistungen für Menschen mit Nierenerkrankungen. Der Konzern hat über 125.000 Beschäftigte und versorgt über 340.000 Patient_innen pro Jahr. Diese werden in mehr als 4.000 Dialysekliniken behandelt (Fresenius Medical Care AG & Co. KGaA, 2023a).

Im Geschäftsjahr 2022 hat das Unternehmen 19,3 Mrd. € Umsatz gemacht. 13,5 Mrd. € davon wurden in der Region Nordamerika, 2,8 Mrd. € in der Region Europa, Mittlerer Osten und Afrika, 2,1 Mrd. € in der Region Asien-Pazifik und 0,7 Mrd. € in der Region Lateinamerika erzielt (Fresenius Medical Care AG & Co. KGaA, 2023a). Daran ist die globale Ausrichtung des Konzerns zu erkennen.

Die Organisation von Fresenius Medical Care ist in die Bereiche „Care Enablement“ und „Care Delivery“ unterteilt. Im Bereich „Care Enablement“ werden die Produkte zusammengefasst, im Bereich „Care Enablement“ die Durchführung der Therapie der Patient_innen, also die Dienstleistungen. Dabei wurden im Geschäftsjahr 2022 15,4 Mrd. € des Umsatzes durch Dienstleistungen und 3,9 Mrd. € durch Produkte erzielt (Fresenius Medical Care AG & Co. KGaA, 2023a).

Die Produkte der Fresenius Medical Care drehen sich um die Versorgung von Menschen mit Nierenerkrankungen. Diese sind in die Bereiche Zentrumsdialyse, Heimdialyse und Akutdialyse unterteilt. Für alle Bereiche werden spezialisierte Dialysemaschinen angeboten. Neben den Dialysemaschinen, die für die verschiedenen Bereiche unterschiedliche Anforderungen erfüllen müssen, werden auch weitere Produkte angeboten, die zur Durchführung einer Dialyse notwendig sind. Dazu zählen Schläuche, Filter und Beutel mit verschiedenen Lösungen (Fresenius Medical Care AG & Co. KGaA, 2023b).

Neben den Produkten werden auch Dienstleistungen angeboten. Dazu zählen die Durchführung der Dialysebehandlung in eigenen Kliniken sowie der Betrieb von Krankenhäusern.

Fresenius Medical Care ist ein sehr großer Konzern, der in vielen Ländern mit diversen Produkten und Dienstleistungen tätig ist. Die Produkte und Dienstleistungen werden in vielen verschiedenen Kontexten eingesetzt. Es ist notwendig, diese verschiedenen Einsatzmöglichkeiten bei der

Produktentwicklung in Betracht zu ziehen. Damit ist Fresenius Medical Care für eine prototypische Umsetzung des Engineering-Graph gut geeignet.

5.2 Aufbau des Prototyps

In diesem Unterkapitel wird der initiale Aufbau des Prototyps des Engineering-Graph bei einem Medizintechnik-Unternehmen beschrieben. Dabei wurde, wie in Kapitel 4.2 dargelegt, bei den hohen Ebenen des System of Interest begonnen. Weitere Informationen wurden sukzessive hinzugefügt.

Um Informationen der höchsten Ebene des System of Interest hinzuzufügen, werden zunächst Daten aus dem Wikimedia Knowledge Graph abgefragt und hinzugefügt. Dafür wird die API von Wikimedia abgefragt und die zurückgespielte Antwort in Neo4J zum Engineering-Graph hinzugefügt (Neo4J, 2023b). Als Start der Abfrage wurden die beiden Elemente ‚Hemodialysis‘ und ‚Medical Device‘ verwendet. Die API gibt dann diese Knoten und alle weiteren, die eine Verbindung dazu haben, zurück. Abbildung 38 zeigt einen Auszug der auf diese Art eingelesenen Daten.

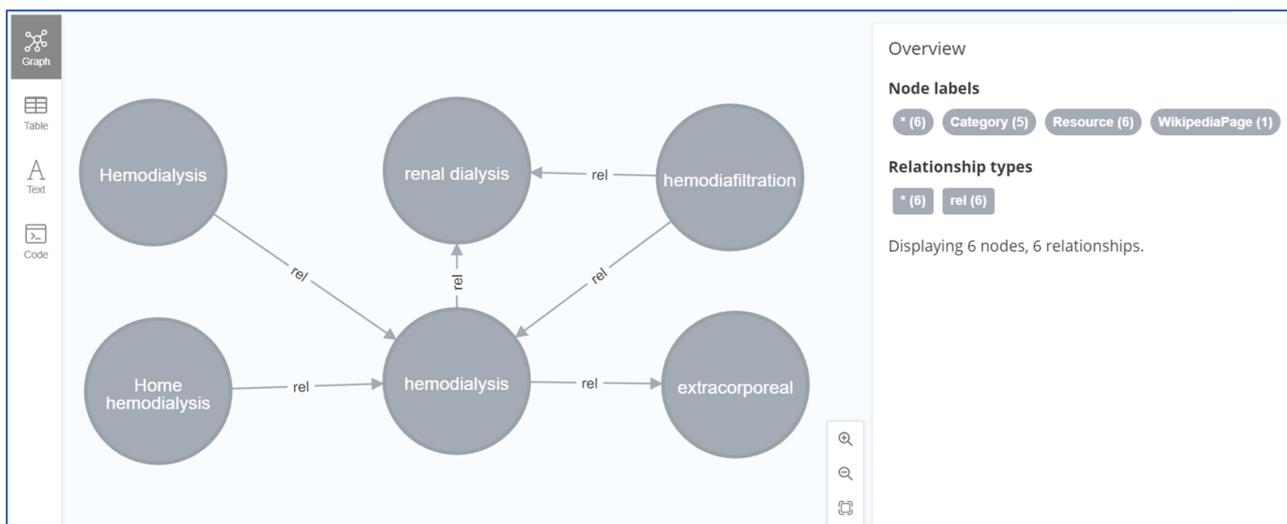


Abbildung 38. Engineering-Graph mit Wikimedia-Daten

Erste Informationen der zweiten Ebene des System of Interest, des Health-System, werden in Form von Fachartikeln hinzugefügt. Hier wird dies am Beispiel von Artikeln des Johner Instituts gezeigt. Dazu wurde manuell die Webadresse von möglicherweise interessanten Artikeln in einer Datei gespeichert. Diese können dann automatisiert abgefragt und zum Prototyp hinzugefügt werden. Name, Inhalt und Webadresse wurden als Attribute der Knoten, die den Artikel repräsentieren, gespeichert. Abbildung 39 zeigt eine beispielhafte Darstellung, bei der die Attribute auf der rechten Seite zu finden sind.

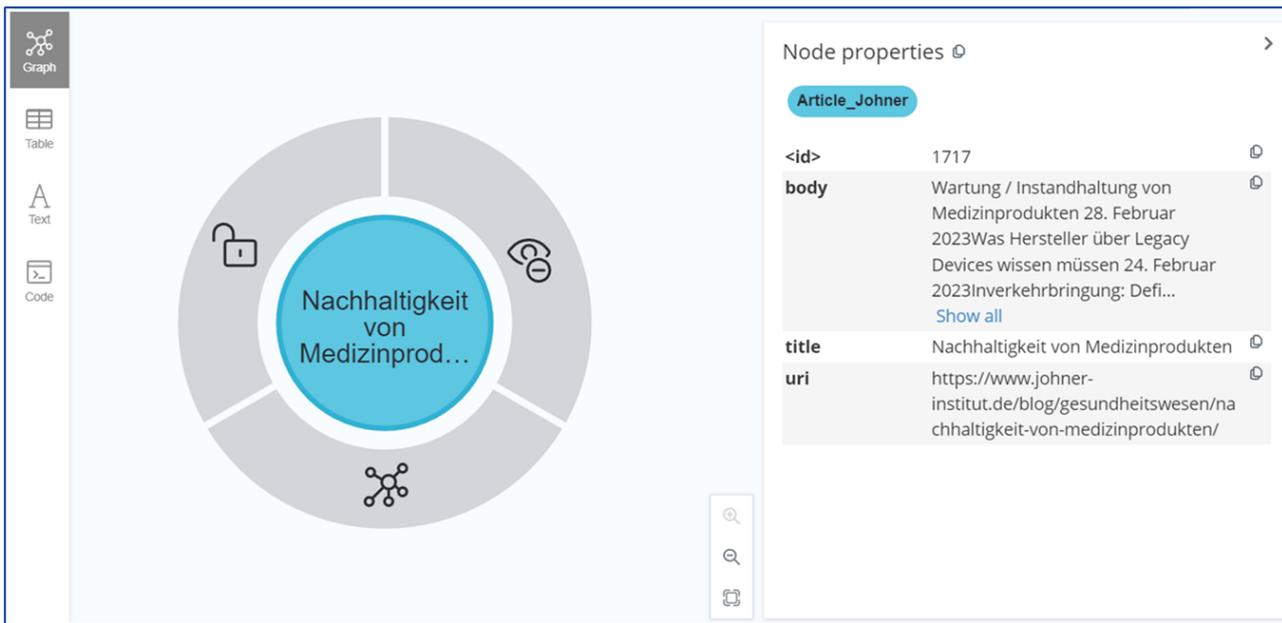


Abbildung 39. Engineering-Graph mit Artikel

Im nächsten Schritt werden weitere Informationen aus der Health-System-Ebene hinzugefügt. Dies sind die ‚Health System Building Blocks‘ der WHO (World Health Organization, 2021) und LCSA-Informationen. Diese können auch miteinander verbunden werden. Abbildung 40 zeigt eine Darstellung, wie die Building-Blöcke mit den Midpoints und Endpoints aus der LCSA sowie den Sustainable Development Goals der Vereinten Nationen (UN General Assembly, 2021) verbunden sind.

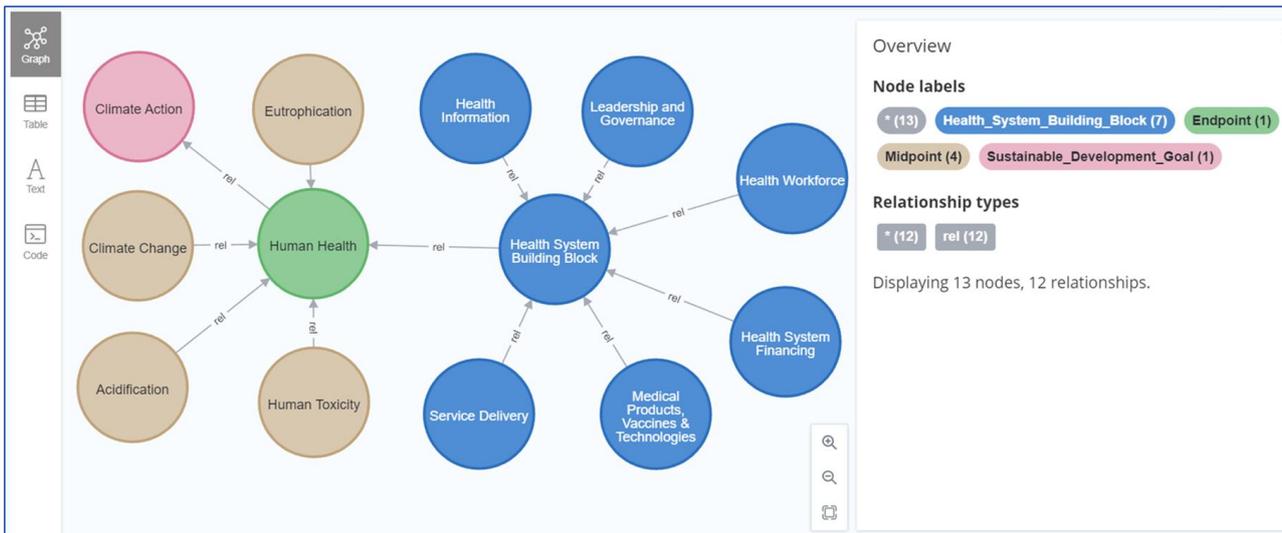


Abbildung 40. Engineering-Graph mit WHO- und LCSA-Daten

Informationen der Ebene ‚Therapy-Treatment-System‘ liegen bereits in der Form von SysML-Modellen aus MBSE vor. Diese können über XML-Schnittstellen zum Engineering-Graph hinzugefügt werden.

Die Produktdaten der Ebenen ‚Therapy-Product-System‘ und ‚Product-System‘ werden aktuell in den PLM-Systemen verwaltet. Die Metadaten der Produktdaten können als ‚comma separated values‘ (csv) aus dem PLM-System exportiert und zum Engineering-Graph hinzugefügt werden. Für den ersten Prototyp wurden Parts, Change-Requests und Dokumente exportiert und hinzugefügt. Da die Metadaten weitere Informationen enthalten, wie zum Standort, an dem die Objekte erzeugt wurden, oder zur Produktfamilie, zu denen die Objekte gehörten, können daraus weitere Nodes der Typen ‚Standort‘ und ‚Produktfamilie‘ erstellt werden. Die Engineering-Objekte werden dann automatisch mit dem neuen Node verknüpft, zu dem sie gehören. Abbildung 41 zeigt eine Darstellung des Engineering-Graph mit PLM-Daten.

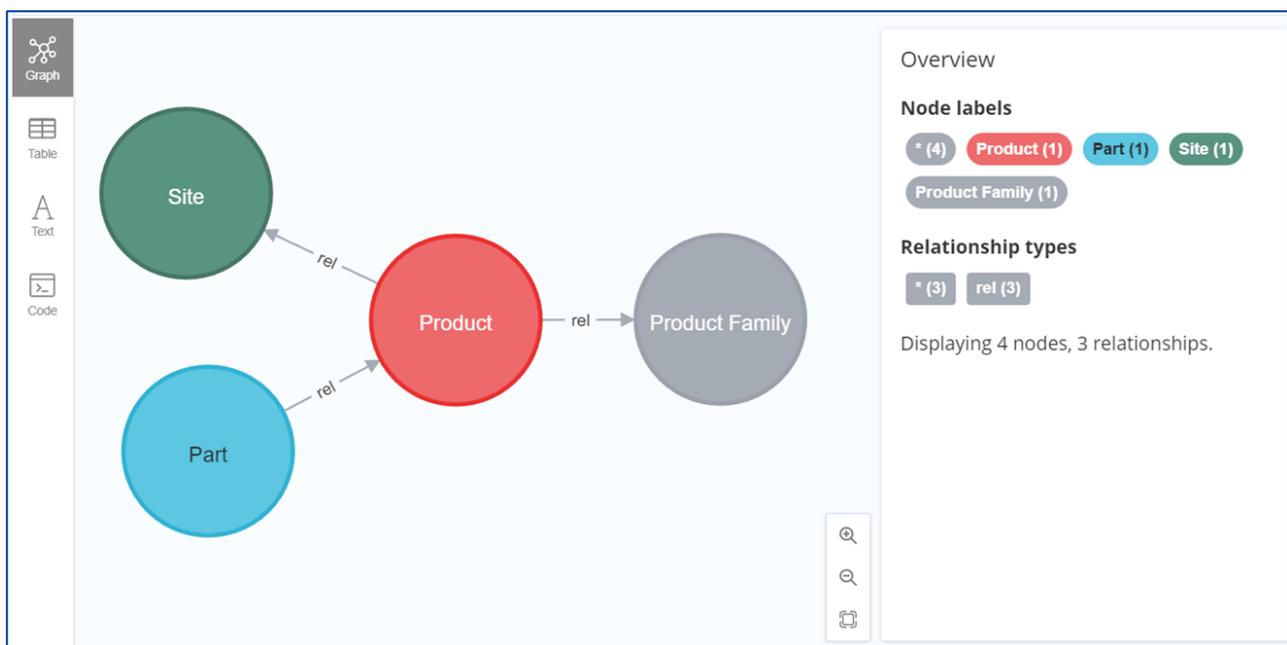


Abbildung 41. Engineering-Graph mit PLM-Daten

Dieser initiale Aufbau des Engineering-Graph ist die kleinstmögliche Implementierung eines Prototyps, in dem alle Ebenen des System of Interest verbunden werden. Auf die gleiche Art können weitere Informationen, die für einen bestimmten Anwendungsfall relevant sind, hinzugefügt und verlinkt werden. Im nächsten Unterkapitel werden die verschiedenen Anwendungsfälle beschrieben, für die dieser Prototyp bereits eingesetzt wurde.

5.3 Engineering-Graph in der Praxis

In diesem Unterkapitel wird beschrieben, wie der Engineering-Graph in der Praxis eingesetzt wird. Dazu wird zunächst vorgestellt, wie die Anwendungsfälle klassifiziert und ausgewählt werden können, bevor sie exemplarisch vorgestellt werden.

5.3.1 Klassifizierung und Auswahl der Anwendungsfälle

Anwendungsfälle, für die der Engineering-Graph geeignet ist, können nach zwei Kriterien klassifiziert werden: der Beziehung und dem Einfluss. Der Einfluss einer Information auf das System of Interest kann entweder statisch oder dynamisch sein. Dabei ist ein statischer Einfluss ein Einfluss einer konstanten Größe, die sich nicht verändert. Ein dynamischer Einfluss dagegen entsteht plötzlich durch eine Veränderung. Das zweite Kriterium ist die Beziehung, die entweder direkt oder indirekt sein kann. Eine direkte Beziehung bedeutet, dass die Zielgröße direkt von den berechneten Größen abhängt, während eine indirekte Beziehung bedeutet, dass weitere moderierende Größen dazwischen liegen. Abbildung 42 stellt die möglichen Anwendungen dar.

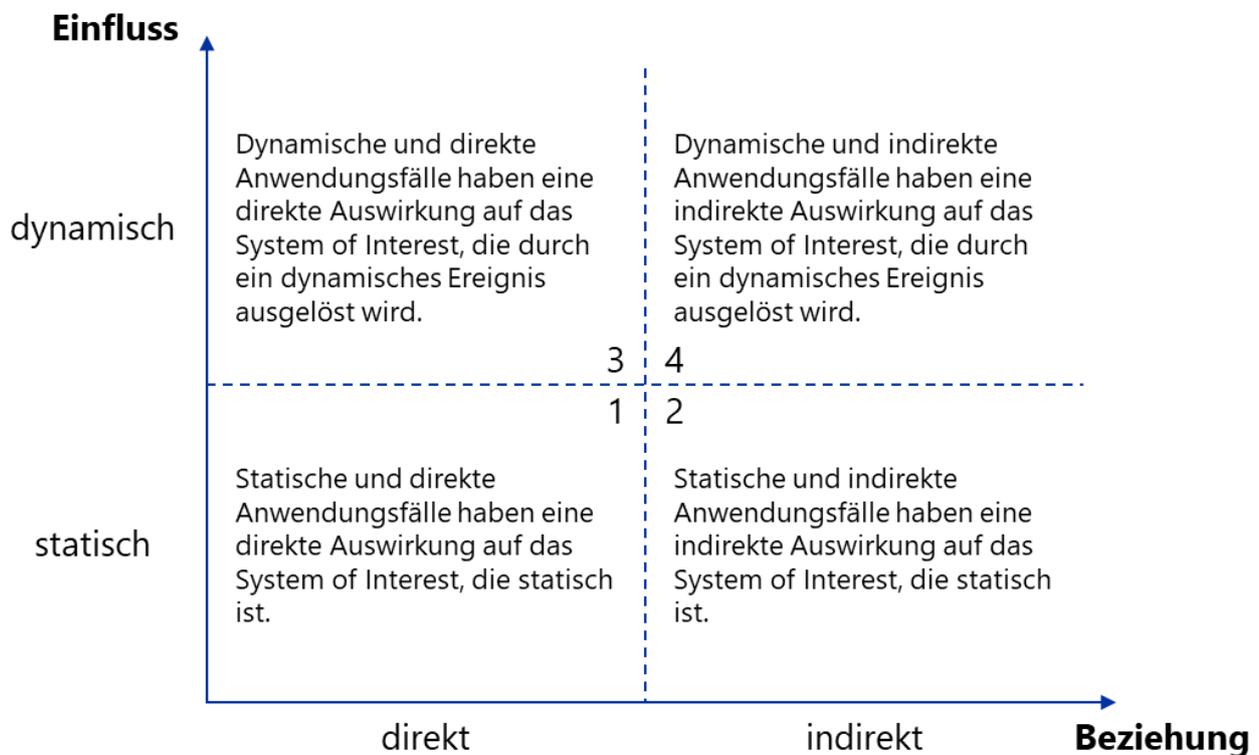


Abbildung 42. Allgemeine Klassifizierung der Anwendungsfälle

Bei Anwendungsfällen, die in den Quadranten direkt und statisch fallen, liegt eine konstante Einflussgröße vor, die direkt auf das System of Interest wirkt. Eine statische Einflussgröße mit direktem Einfluss auf das Produkt können Materialeigenschaften von im Produkt verwendeten Materialien sein.

Im zweiten Quadranten liegt auch ein statischer Einfluss vor, allerdings wirkt dieser indirekt auf das System of Interest. Einen indirekt wirkenden statischen Einfluss können Materialeigenschaften von Materialien darstellen, die in anderen Produkten eingesetzt wurden, mit denen das System of Interest interagieren muss.

Der dritte Quadrant beinhaltet Anwendungsfälle, in denen ein dynamischer Einfluss vorliegt, der direkt auf das System of Interest wirkt. Neue rechtliche oder regulatorische Rahmenbedingungen sind ein dynamischer Einfluss auf das System of Interest und haben eine direkte Auswirkung.

Im vierten Quadranten befinden sich die Anwendungsfälle, bei denen ein dynamischer Einfluss indirekt wirkt. Sich verändernde Zinsen in einem dynamischen ökonomischen Umfeld können viele Größen beeinflussen, zum Beispiel die Produktkosten und damit die angestrebte Gewinnmarge.

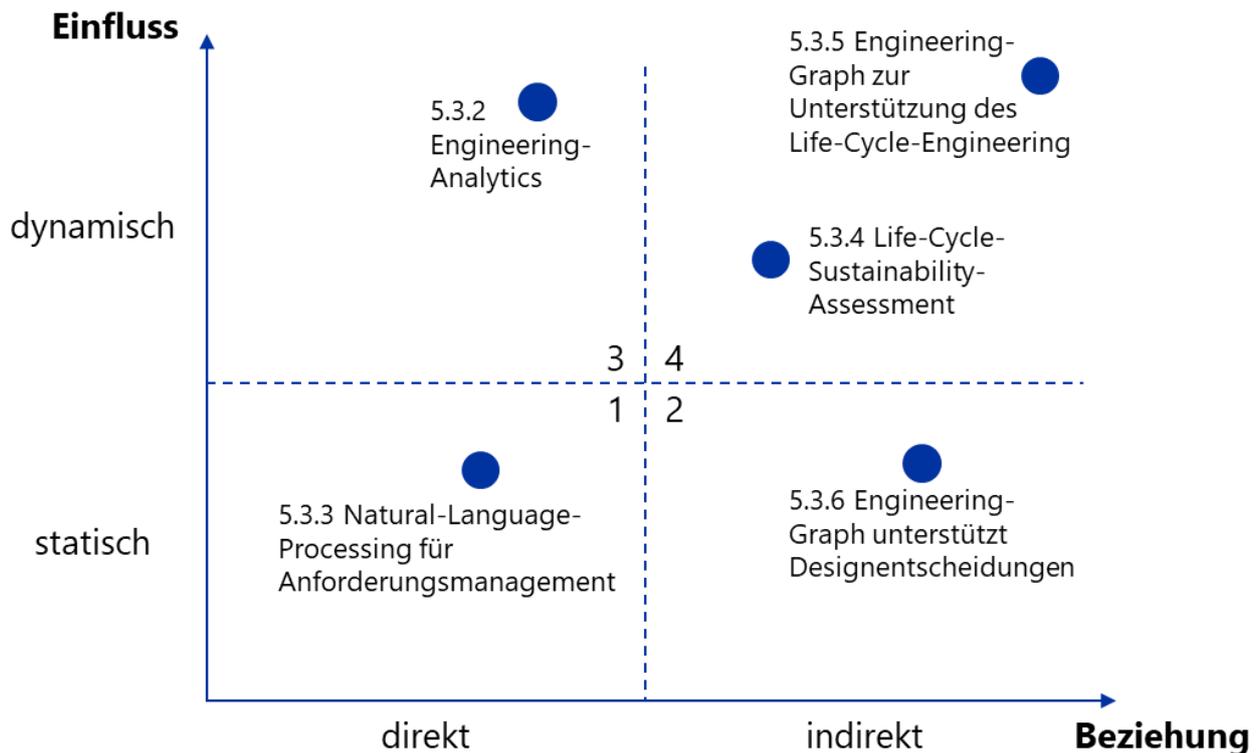


Abbildung 43. Einordnung der Anwendungsfälle in die Quadranten

Im Folgenden werden verschiedene Einsatzmöglichkeiten des Engineering-Graph beschrieben. Diese sind Engineering-Analytics, Unterstützung des Anforderungsmanagements durch NLP, Durchführung eines LCSA, Unterstützung von LCE und von Designentscheidungen. Eine Einordnung in die Quadranten zeigt Abbildung 43.

5.3.2 Engineering-Analytics

In diesem Unterkapitel wird das erste Anwendungsbeispiel vorgestellt. Dabei werden Data-Science-Technologien eingesetzt, um ein Engineering-Dashboard zu erstellen. Ziel des Dashboards ist die Berechnung und visuelle Darstellung von Kennzahlen, mit denen die Produktentwicklung eines Unternehmens repräsentiert wird.

Der erste Schritt ist die Identifikation der Kenngrößen, die für das Unternehmen relevant sind. Diese sollten die Leistung der Entwicklungsabteilung, die Datenqualität der Engineering-Objekte und den Status der IT-Systeme, die in der Produktentwicklung eingesetzt werden, abbilden.

Der Status der IT-Systeme kann durch Kennzahlen wie Verfügbarkeit, Antwortgeschwindigkeit und ‚Mean Time between Errors‘ gemessen werden. Letzteres gibt an, wie viel Zeit durchschnittlich zwischen zwei Fehlermeldungen vergeht. Ziel sollte es sein, diese Zeit zu maximieren. Es ist essenziell, diese Kennzahlen zu messen, da Ingenieur_innen ohne ausreichende Verfügbarkeit und Antwortgeschwindigkeit der IT-Systeme nur eingeschränkt in der Lage sind, Produkte zu entwickeln.

Die Datenqualität sollte gemessen werden, um sicherzustellen, dass die Daten richtig gepflegt sind. Dies erhöht die Wiederverwendbarkeit und die Möglichkeit, Daten zu finden. Beispiele für Kennzahlen dieser Kategorie sind Prüfungen, wie die Verlinkung der Dokumente zu Projekten, der Anteil an positiv durchgeführten Tests und der Anteil an Freigabeprozessen mit positivem Ausgang.

Des Weiteren können Kennzahlen, die die Leistung der Entwicklungsabteilung messen, berechnet werden. Dabei ist ein gemeinsames Verständnis, dass diese Kennzahlen die Leistung der Prozesse und nicht die Leistung der Mitarbeiter_innen messen, wichtig. Hier steht im Vordergrund, dass Ineffizienzen in Prozessen gefunden und datengetrieben behoben werden können. Ein Beispiel dafür wird in Abbildung 44 dargestellt. Das Beispiel zeigt die Berechnung der Durchlaufzeit eines Änderungsprozesses. Ziel ist die Identifikation von Wartezeiten oder anderen Ineffizienzen, die datengetrieben behoben werden können, um die Durchlaufzeit zu verringern.

Abbildung 44 zeigt des Weiteren die möglichen Reifegrade der Kennzahlen. Zunächst steht die Erkundung im Vordergrund. Die Durchlaufzeit soll berechnet werden, um eine objektive Kenngröße zu ermitteln, mit der weitergearbeitet werden kann. Im nächsten Schritt steht die Erklärung im Vordergrund. Dabei werden die Einflussfaktoren auf die Kenngröße berechnet. Damit kann analysiert werden, warum ein bestimmter Wert erreicht wird, und es gibt erste Hinweise auf Möglichkeiten, diesen Wert zu verbessern. Die dritte Stufe ist die Vorhersage. Sind die Einflussfaktoren bekannt, kann basierend darauf vorhergesagt werden, wie viel Zeit ein neuer Änderungsauftrag voraussichtlich beanspruchen wird. Außerdem kann der Einfluss von Maßnahmen zur Verringerung der Durchlaufzeit berechnet werden. Dafür können neben statistischen Methoden wie einer Multifaktoranalyse auch KI-Technologien wie maschinelles Lernen zum Einsatz kommen.

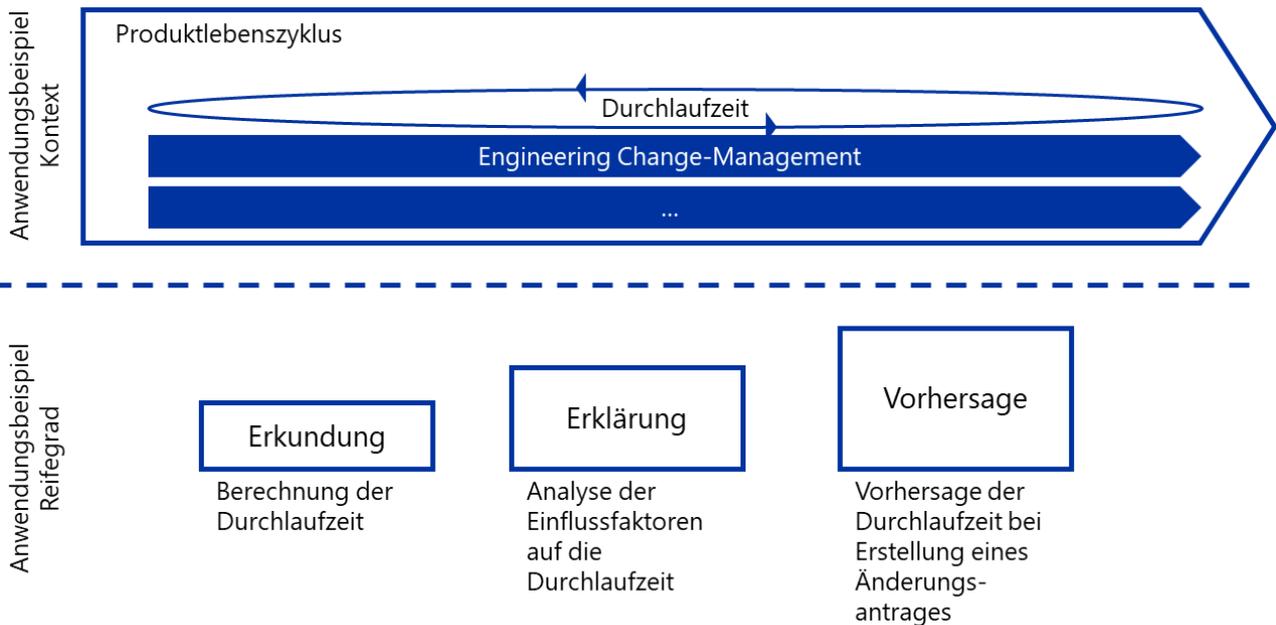


Abbildung 44. Kontext und Reifegrad der Kennzahl nach (Schweitzer et al., 2020)

Die Kennzahlen der drei Kategorien können auf einem Engineering-Dashboard dargestellt werden, um für Mitarbeiter_innen, Projektleitung und Management verfügbar zu sein. Durch die Schaffung einer gemeinsamen Datenbasis kann ein gemeinsames Verständnis entstehen, sodass die Produktentwicklung datengetrieben optimiert werden kann. Abbildung 45 zeigt ein exemplarisches Dashboard mit Beispielen für die drei Bereiche Status der IT-Systeme, Datenqualität der Engineering-Objekte und Leistung der Entwicklungsabteilung. Für den Status der IT-Systeme steht hier die Kennzahl Verfügbarkeit. Diese gibt den prozentualen Anteil der Zeit an, zu dem die IT-Systeme den Ingenieur_innen zur Verfügung standen. Für Datenqualität steht die Grafik der Engineering-Objekte pro Projekt. Diese gibt an, wie viele Objekte den Projekten zugewiesen sind. Eine hohe Anzahl an Objekten ohne Projekt kann für eine niedrige Datenqualität sprechen. Die Leistung der Entwicklung kann mit der Durchlaufzeit, zum Beispiel eines Änderungsantrages, gemessen werden. In der Grafik wird aufgezeigt, wie viele Tage ein Änderungsantrag benötigt, in Abhängigkeit der Kalenderwoche, in der er erstellt wurde. Kennzahlen wie diese können genutzt werden, um den realen Einfluss von Prozessveränderungen auf die Leistung der Produktentwicklung zu messen. Außerdem können die Kennzahlen bei der Analyse von Problemen und Ableitung von Maßnahmen zur Problemlösung dienen. Das Dashboard wurde im Rahmen der Dissertation mit diesen und weiteren Kennzahlen realisiert. Aus Gründen der Geheimhaltung kann hier nur ein Mock-Up gezeigt werden.

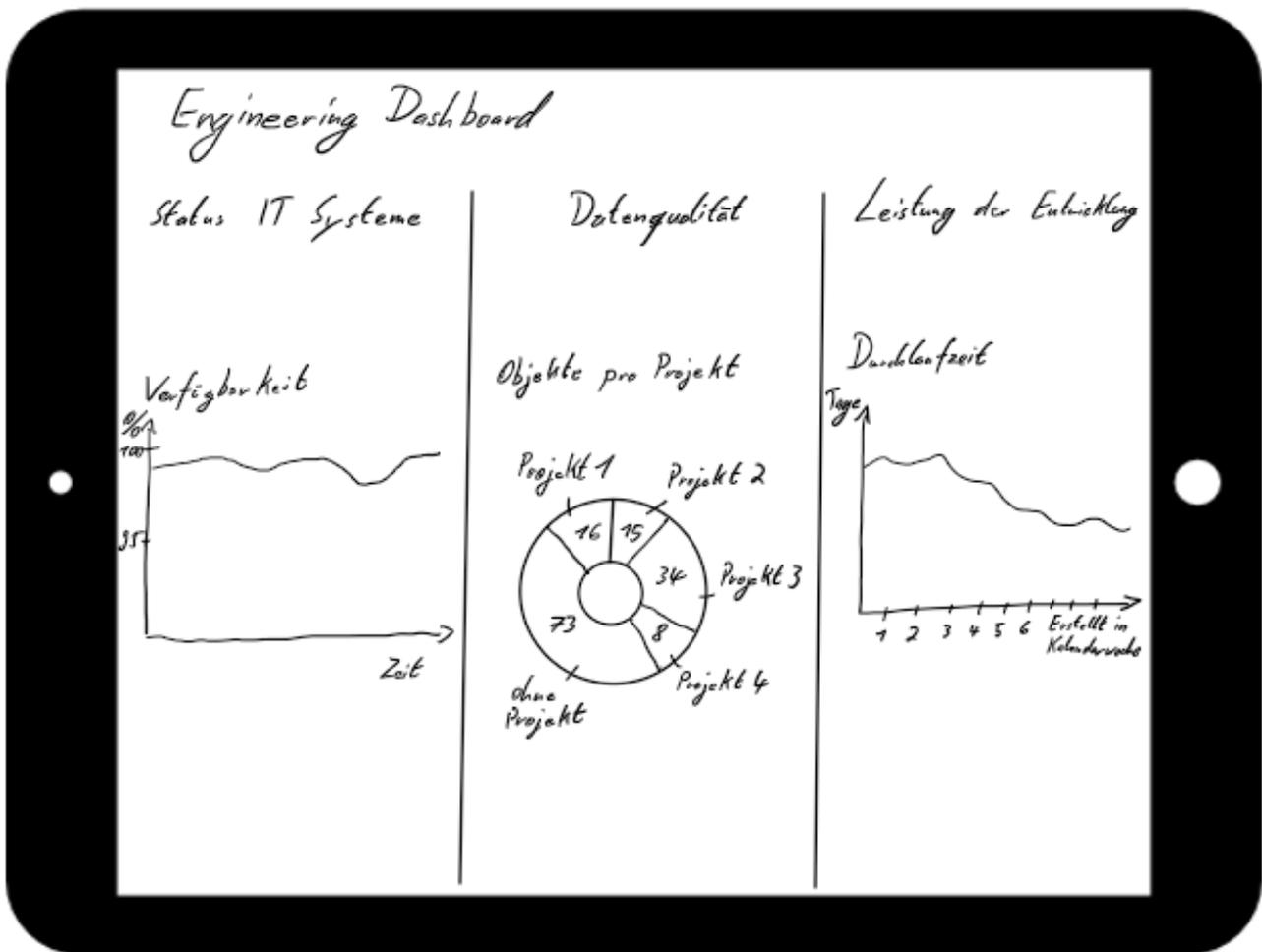


Abbildung 45. Mock-up eines Engineering-Dashboards nach (Schweitzer et al., 2020)

Der Engineering-Graph unterstützt diesen Anwendungsfall, indem er es ermöglicht, ein Engineering-Objekt über den gesamten Prozess hinweg zu verfolgen. Zum Beispiel kann ein Änderungsprozess mehrere IT-Systeme beinhalten. So wird der Antrag möglicherweise in einem ERP-System gestellt und freigegeben, die Daten liegen dann in einem PLM-System und die Bearbeitung findet in einem speziellen Autorensystem statt. Hier Durchlaufzeiten zu berechnen, ohne die Systeme verbunden zu haben, ist praktisch unmöglich. Der Engineering-Graph schafft diese Verbindung und ermöglicht somit die Betrachtung des Prozesses über verschiedene IT-Systeme hinweg.

5.3.3 Natural-Language-Processing für Anforderungsmanagement

In diesem Unterkapitel wird beschrieben, wie NLP eingesetzt werden kann, um das Anforderungsmanagement zu unterstützen. Dies ist ein Beispiel für einen Anwendungsfall, in dem KI-Technologie eingesetzt wird, um einen speziellen Schritt im Produktentwicklungsprozess zu unterstützen.

Eine Marktrecherche hat ergeben, dass es für diesen Anwendungsfall bereits kommerzielle Lösungen am Markt gibt. Analysiert wurden hier vor allem QVscribe (QRA Corp, 2023) und reQlab (IT Designers, 2023). Beide Systeme wurden bereits in Kapitel 3.2.2.2 genannt. Nach der Analyse der Fähigkeiten der Systeme wurde sich entschieden, eine eigene Lösung zu entwickeln. Dadurch können die eigenen Anforderungen in größerem Maße erfüllt werden und es können Erfahrungen im Einsatz von KI-Technologie gesammelt werden.

Die notwendigen Schritte zum Einsatz von NLP für Anforderungen werden in Abbildung 46 zusammengefasst. Am Anfang steht der Input in Form von geschriebenem Text, also ganzen Sätzen oder einzelnen Wörtern. Dieser wird im ersten Schritt vorbereitet, indem unerwünschte Elemente entfernt werden. Dies kann notwendig sein, wenn zu viele Füllwörter verwendet wurden oder der Input in gemischter Sprache (zum Beispiel Deutsch und Englisch) vorliegt. Danach kann der Input von einem Encoder vektorisiert werden. Dabei werden die Buchstaben in einen Zahlenvektor umgewandelt, mit dem ein Computer später Berechnungen durchführen kann. Die Wahl der Vektorisierung kann einen großen Einfluss auf das Ergebnis haben, da es hier unterschiedliche Herangehensweisen gibt. So können verschiedene Mengen an Kontextinformationen über die vorangegangenen oder nachstehenden Worte in den Vektor eines Wortes oder Satzes aufgenommen werden. Mit diesem Vektor können dann durch Algorithmen Berechnungen angestellt werden. In diesem Beispiel wurde die semantische Ähnlichkeit durch eine Kosinus-Ähnlichkeit berechnet. Der daraus resultierende Output ist dann eine Liste von Texten, absteigend sortiert nach der Ähnlichkeit.

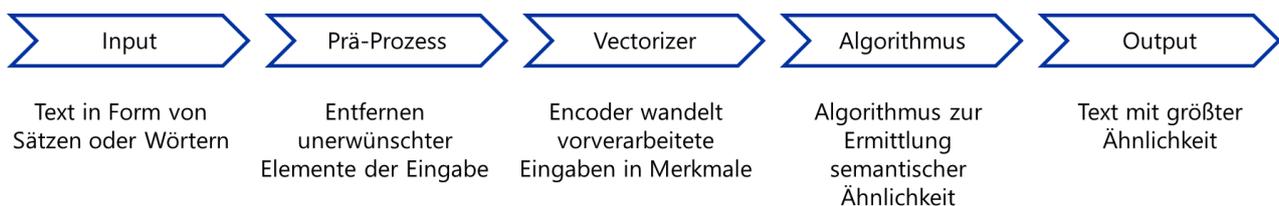


Abbildung 46. Schritte zum Einsatz von Natural-Language-Processing

Dieses Verfahren wurde angewandt, um drei Anwendungsfälle zu bearbeiten. Diese werden in Abbildung 47 zusammengefasst. Der erste ist die Analyse einzelner Anforderungen nach bestimmten Gütekriterien, der zweite die Berechnung der semantischen Ähnlichkeit und der dritte die Wiederverwendung existierender Anforderungen.

Die Analyse einzelner Anforderungen zur Erhöhung der Qualität ist vor allem sinnvoll, um unerfahrenen oder fachfremden Mitarbeiter_innen die Möglichkeit zu geben, direkt qualitativ hochwertige Anforderungen zu schreiben. Dabei untersucht das System insbesondere geschriebene Anforderungen nach vorher definierten Qualitätskriterien und zeigt mögliche Verbesserungen an. Dies kann bereits ohne den Einsatz von KI umgesetzt werden; KI wird dann benötigt, um die

geschriebene Anforderung mit erlernten idealen Anforderungen zu vergleichen. Das System kann einen Score geben, wie gut die Anforderung geschrieben ist.

Der zweite Anwendungsfall ist die Erkennung von semantischer Ähnlichkeit. Dabei wird die Kette wie oben beschrieben durchlaufen. Bei der Erstellung von Anforderungen können so semantisch ähnliche Anforderungen gefunden und angezeigt werden. Anforderungen, die ein bestimmtes Thema zum Inhalt haben, werden tendenziell semantisch ähnlich sein im Vergleich zu Anforderungen zu einem weniger verwandten Thema. Daher kann die Erkennung ähnlicher Anforderungen verhindern, dass Anforderungen doppelt erstellt werden oder dass widersprüchliche Anforderungen auftauchen.

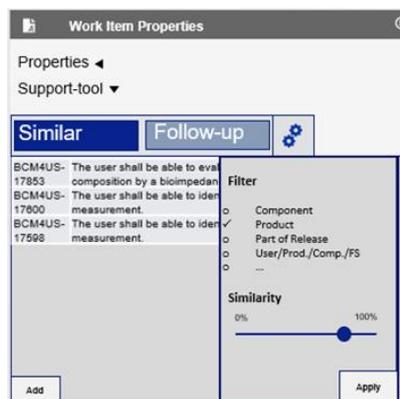
Analyse einzelner Anforderungen

Hervorhebung bestimmter Qualitätsprobleme und Verbesserungsvorschläge



Semantische Ähnlichkeit

Vergleich der Anforderungen und Aufzeigen möglicher Konflikte oder Wiederholungen



Wiederverwendung

Wiederverwendung bereits geschriebener Anforderungen aus früheren Projekten



Abbildung 47. Natural-Language-Processing für Anforderungsmanagement

Der dritte Fall ist die Wiederverwendung von Anforderungen. Hier können wieder Anforderungen aus vergangenen Produkten oder anderen Entwicklungsprojekten über semantische Ähnlichkeit gefunden werden, die an dieser Stelle verwendet werden können. Die Wiederverwendung hat den Vorteil, dass (1) Änderungen nicht mehrfach gepflegt werden müssen, falls Anforderungen mehrere Produkte betreffen, und dass (2) möglicherweise auch weitere Engineering-Objekte wie verbundene Tests wiederverwendet werden können. Dabei muss eine individuelle und manuelle Prüfung durchgeführt werden.

Der Engineering-Graph ermöglicht hier den letzten Anwendungsfall, da Anforderungen aus parallelen Entwicklungsprojekten oder vergangenen Produkten eventuell nicht im selben Autorensystem geschrieben und verwaltet wurden. Außerdem wird das Auffinden und Anzeigen von

verbundenen weiteren Engineering-Objekten ermöglicht, da auch diese in anderen Systemen abliegen.

5.3.4 Life-Cycle-Sustainability-Assessment

Wie in Kapitel 3.1.3 dargestellt, liegen die Hauptprobleme beim derzeitigen LCSA in der zunehmenden Produktkomplexität und den immer stärker globalisierten Lieferketten, die beide zu vielen zusammenhängenden und miteinander verbundenen Ursache-Wirkungs-Ketten führen, die zu einem Endpunkt führen. Diese Beziehungen und Mechanismen sind derzeit nur teilweise in Modellen enthalten, die Ingenieur_innen bei der Durchführung eines LCSA unterstützen. Außerdem ist die Erweiterung dieser Modelle mit neuen Daten ein manueller und langsamer Prozess (Schweitzer et al., 2022).

Um die oben genannten Probleme zu lösen, würden Ingenieur_innen von einem System profitieren, das (a) Daten aus verschiedenen Quellen verbindet, (b) sich auf die Verbindung von Datenpunkten konzentriert, (c) die Flexibilität hat, mehr Quellen mit unbekannter Struktur einzubeziehen, und (d) die Erforschung des erweiterten Systems ermöglicht. Daher ist ein Engineering-Graph, wie er in Kapitel 4 beschrieben wird, geeignet, um diese Funktionalität bereitzustellen und das Engineering bei der Durchführung eines LCSA zu unterstützen.

Um den Anwendungsfall der Durchführung eines LCSA zu unterstützen, können spezielle Daten, zum Beispiel aus den LCI-Datenbanken, die für das LCSA erforderlich sind, auch über die APIs der spezifischen LCI-Datenbank hinzugefügt werden. Nach der Anbindung dieser kann der Graph um weitere LCSA-spezifische Daten wie verschiedene Mittel- und Endpunkte erweitert werden, wie in Abbildung 48 dargestellt. Diese sind ebenfalls miteinander verknüpft und können bei der Verknüpfung von Lebenszyklus-Auswirkungsfaktoren wie CO₂-Emissionen mit dem Mittelwert ‚Klimawandel‘ verwendet werden, der bereits mit ‚Ressourcen‘, ‚Gesundheit des Ökosystems‘ und ‚Gesundheit des Menschen‘ verknüpft ist. Diese Verbindungen können für jedes LCSA für jedes Produkt wiederverwendet werden. Der Engineering-Graph kann durch neue Mittelpunkte, Endpunkte oder Beziehungen erweitert werden, wenn diese erforscht und veröffentlicht werden. Diese neuen spezifischen Informationen werden dann mit existierenden Informationen verlinkt (Schweitzer et al., 2022).

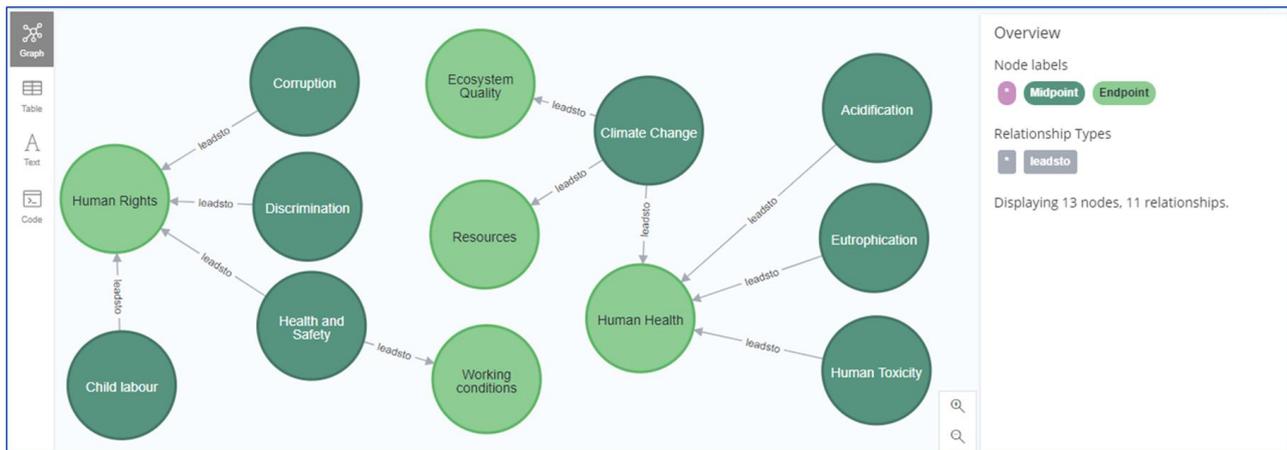


Abbildung 48. Engineering-Graph mit LCSA-Daten

Der Engineering-Graph mit den LCSA-relevanten Daten kann mit Data-Science-Technologie analysiert werden. Teile des Graphen können visualisiert werden, sodass Ingenieure benachbarte Knoten untersuchen und neue mögliche Ursache-Wirkungs-Beziehungen entdecken können. Diese können erkannt werden, indem eine stärker integrierte Sicht auf zuvor vernachlässigte Teile des Lebenszyklus genutzt wird, was eine ganzheitlichere Sicht auf das Produkt ermöglicht und die Bewertung der Nachhaltigkeit des Lebenszyklus unterstützen kann.

Die Erkennung sozialer Auswirkungen, bei denen die Komplexität sehr hoch ist und die Gesellschaften sich von Land zu Land unterscheiden, profitiert von der Integration semantischer Webquellen. Dies können beispielsweise Änderungen des Arbeitsrechts, der sozioökonomischen Bedingungen oder der Bildung sein.

Die Gewinnung quantitativer Erkenntnisse, wie die Identifikation der größten Verursacher von Emissionen, erfordern, dass die Knoten des Graphen um quantitative Eigenschaften erweitert werden. Auf diese Weise kann der Engineering-Graph dabei helfen, die größten Verursacher von Emissionen im Lebenszyklus zu identifizieren und sie zu beseitigen, um den ökologischen Fußabdruck schnell und effizient zu verringern. Für jedes Produkt können die Maßnahmen mit den größten Auswirkungen zur Verbesserung der Nachhaltigkeit identifiziert und umgesetzt werden. Darüber hinaus können Wechselwirkungen zwischen den Säulen der Nachhaltigkeit identifiziert und berücksichtigt werden. Selbst wenn die Interdependenz hinter mehreren Ecken versteckt ist, kann ein Engineering-Graph den schnellsten Weg zwischen den Säulen finden. Auf diese Art können versteckte Abhängigkeiten und Zusammenhänge explizit gemacht werden.

Eine weitere Möglichkeit ist die Entdeckung alternativer Materialien für medizinische Anwendungen. Diese Materialien könnten den ökologischen Fußabdruck der Produkte verbessern, ohne die medizinische Anwendbarkeit zu beeinträchtigen. Der Engineering-Graph könnte Materialien zeigen,

die in ähnlichen Anwendungsfällen verwendet werden, und Ingenieure könnten über die Anwendbarkeit entscheiden.

5.3.5 Engineering-Graph zur Unterstützung des Life-Cycle-Engineering

Life-Cycle-Engineering umspannt nach seiner Definition den gesamten Lebenszyklus. Hier werden Anwendungsfälle betrachtet, in denen das Produkt holistisch in seinem Kontext analysiert wird. Daher hat ein Systemmodell wie der Engineering-Graph im LCE einen besonderen Wert.

Die Analyse des Graphen über den gesamten Produktlebenszyklus mithilfe der Methoden, die in Kapitel 4.3 vorgestellt wurden, kann zur Aufdeckung unbekannter und unerwarteter Beziehungen führen. Dadurch können Ursache-Wirkungs-Ketten erkannt werden, die ansonsten unerkannt geblieben wären. Diese können genutzt werden, um die ökonomischen Kosten, die Zeit zur Markteinführung oder den ökologischen Fußabdruck eines Produktes zu minimieren oder den Nutzen des Produktes für Endanwender_innen zu optimieren.

Die in Kapitel 3.1.3.1 vorgestellten Herausforderungen beim LCE können durch den Engineering-Graph gelöst werden. Die Nutzung vorhandener Modelle erhöht die Geschwindigkeit (Problem Nr. 1) durch die Vermeidung doppelter Arbeit. Dadurch wird das Problem der übermäßig vereinfachten Modelle (Problem Nr. 2) und des mangelnden Umfangs (Problem Nr. 3) gelöst, da die Modelle aus SE und PLM sehr umfangreich sind. Transparenz wird erhöht, da alle Modelle und Daten an einem Ort gemeinsam durchsucht werden können und damit auch verfügbar gemacht wurden (Problem Nr. 4). Die Integration zwischen den Umgebungen der wichtigsten Ingenieurdisziplinen (Problem Nr. 5) wird durch die direkte Integration von Daten aus diesen verschiedenen Systemen in den Engineering-Graph verbessert (Schweitzer et al., 2023b).

Der ökologische Fußabdruck oder die Kosten über den gesamten Lebenszyklus werden in frühen Entwurfsphasen vordefiniert. Der Engineering-Graph bietet die Möglichkeit, Konstruktionsentscheidungen in frühen Phasen zu unterstützen, um LCE-Maßnahmen durch die frühzeitige Bereitstellung großer Datenmengen zu verbessern. Im folgenden Unterkapitel wird die Unterstützung von Designentscheidungen beschrieben.

5.3.6 Engineering-Graph unterstützt Designentscheidungen

In diesem Unterkapitel wird die Unterstützung von Designentscheidungen durch den Engineering-Graph vorgestellt. Dafür wird zunächst aufgezeigt, in welchen Phasen der Produktentwicklung dies von besonderer Relevanz ist, und anschließend, bei welchen Fragestellungen der Engineering-Graph konkret unterstützen kann.

Der Engineering-Graph bringt eine umfangreiche Datenbank mit verknüpften Informationen aus verschiedenen Quellen in die frühen Entwurfsphasen, in denen Designentscheidungen die größten

Auswirkungen haben. Abbildung 49 zeigt den Zusammenhang von Systemwissen, Designflexibilität und Ressourceneinsatz (Boy, 2019). Daraus kann geschlossen werden, dass in den frühen Phasen mit der höchsten Flexibilität Designentscheidungen die größten Auswirkungen haben. Der Engineering-Graph erweitert das Systemwissen und verbindet es mit Kontextinformationen, die bereits in frühen Phasen zur Verfügung stehen. Dieses Wissen kann genutzt werden, um eine Verbesserung der Designentscheidungen zu erzielen.

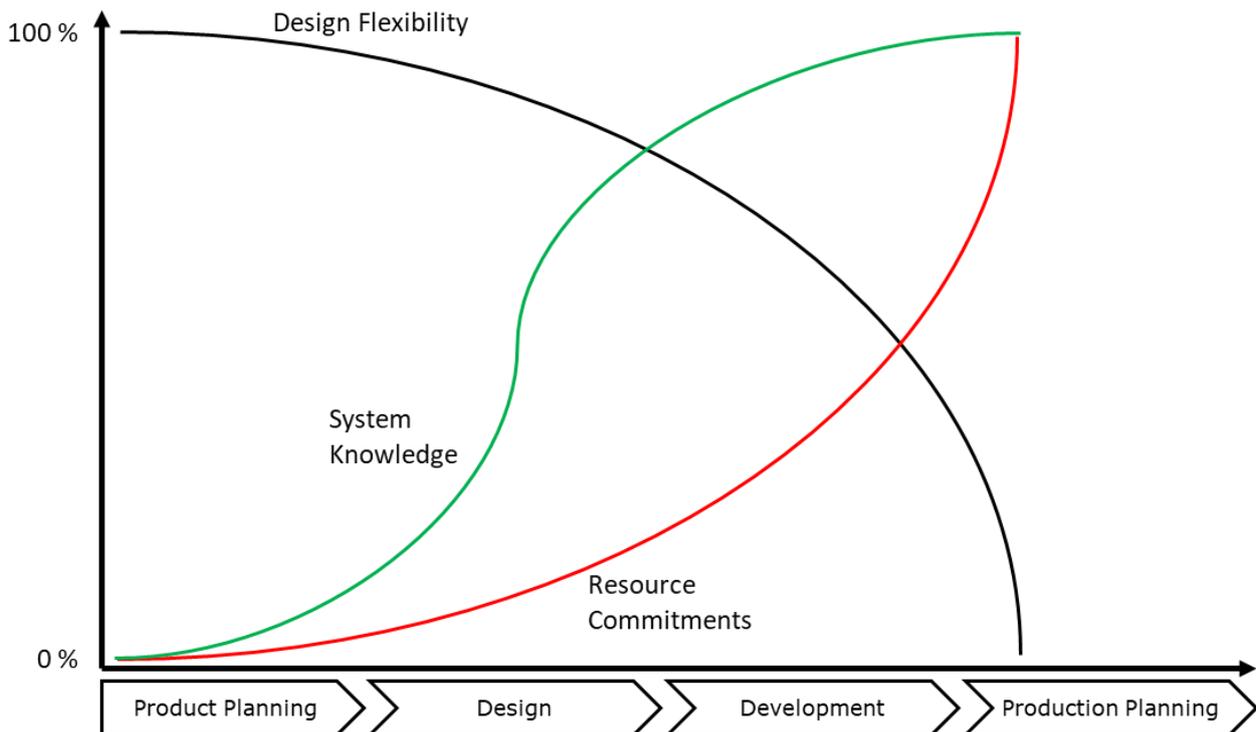


Abbildung 49. Beziehung zwischen Systemwissen, Designflexibilität und Ressourceneinsatz nach (Boy, 2019)

In der Literatur werden zwei Klassen von Designentscheidungen unterschieden (Kaul & Rao, 1995): erstens Entscheidungen über die Positionierung eines Produkts und zweitens Entscheidungen darüber, wie diese Positionierung in die Tat umgesetzt werden kann. Bei der ersten Klasse von Designentscheidungen geht es um die Einrichtung eines Entwicklungsprojekts. Die zweite Klasse von Designentscheidungen wird dann innerhalb eines Entwicklungsprojektes getroffen.

Die erste Klasse von Entscheidungen, die der Engineering-Graph unterstützt, ist das Aufsetzen eines Entwicklungsprojektes. Zunächst müssen Entscheidungen zur Projektstrategie und -planung getroffen werden. Diese können durch die Bereitstellung von Marktinformationen und deren Verknüpfung mit vorhandenen Produktinformationen über andere Produkte und Dienstleistungen, die ein Unternehmen anbietet, unterstützt werden, um eine wertvolle Nische zu finden, die ein neues Produkt ausfüllen kann. Zweitens muss die Organisation des Produktentwicklungsprojektes festgelegt werden. Hier kann der Engineering-Graph unterstützen, indem er Informationen über die Fähigkeiten der Mitarbeiter und Zeitpläne anderer Entwicklungsprojekte liefert, um die Verfügbarkeit

bestimmter Schlüsselmitarbeiter_innen zu planen. Drittens werden Entscheidungen bezüglich des Projektmanagements unterstützt. Der Engineering-Graph kann hier bei der Festlegung der Entwicklungsprioritäten unterstützen, indem er das benötigte Kontextwissen bereitstellt.

Die zweite Klasse von Entscheidungen wird während eines Entwicklungsprojektes unterstützt. Während der Konzeptentwicklung kann der Engineering-Graph die Anforderungen mit bereits entwickelten Teilen aus dem Unternehmen verbinden und bei der Erstellung der Produktarchitektur und der Entscheidung, welche Komponenten in verschiedenen Varianten des Produkts gemeinsam genutzt werden können, helfen. Beispielsweise ermöglicht die Verknüpfung von Transportrouten und geopolitischen Informationen mit dem Graphen die Unterstützung des Supply-Chain-Designs durch die Bereitstellung von Daten für Risikoanalysen und Make-or-Buy-Entscheidungen. Das Produktdesign wird direkt durch die Verbindung von Anforderungen mit Materialinformationen sowie Entwicklungs- und Zulassungsnormen unterstützt, die eine verbesserte Materialauswahl ermöglichen.

Die Erweiterung des Systemwissens durch die Verbindung von Systemdaten mit Kontextinformationen ist in jeder Entwurfsphase nützlich. Dynamische Änderungen der Kontextinformationen können im Graphen reflektiert werden, indem er erweitert und aktualisiert wird, was die Entscheidungsfindung und Überarbeitung in Echtzeit unterstützen kann.

Konkret werden im Folgenden zwei Anwendungsfälle aus der Medizinproduktindustrie vorgestellt. Im ersten Fall wird ein bereits entwickeltes und zugelassenes Medizinprodukt von einem Kontinent auf einen anderen gebracht und dort mit einer neuen benannten Stelle zugelassen. Der zweite Anwendungsfall ist die Entwicklung eines neuen Medizinprodukts.

Der **erste Anwendungsfall** ist ein zugelassenes medizinisches Gerät, das derzeit in Europa verkauft wird und in den Vereinigten Staaten von Amerika (USA) auf den Markt gebracht werden soll. Daher muss das Produkt von der Food and Drug Administration (FDA) zugelassen werden, da diese die Zulassung von Medizinprodukten in den USA regelt. Der Engineering-Graph kann hier hilfreich sein, da die gesamten Produktdaten bereits Teil des Graphen sind. Nun können die für den neuen Markt spezifischen Informationen wie Krankenhausstandorte, Höhe, Temperaturen, Luftdruck usw. sowie die Anforderungen für die Zulassung durch die FDA hinzugefügt werden. Diese Informationen werden mit den vorhandenen Produktdaten verknüpft. Damit können die Fachexpert_innen unterstützt werden, die analysieren, ob Konstruktionsänderungen erforderlich sind, um die zusätzlichen Anforderungen zu erfüllen. Durch die Unterstützung der Ingenieur_innen bei der Analyse der Auswirkungen und der Durchführung von Konstruktionsänderungen wird die Zeit bis zur Markteinführung erheblich verkürzt und die Produktqualität verbessert, da die Bewältigung der komplexen Anforderungen des neuen Marktes durch die Daten unterstützt wird.

Der **zweite Anwendungsfall** ist eine neue Produktentwicklung. Hier werden Informationen der Weltbank über Märkte, Demografie und wirtschaftliche Bedingungen mit Informationen der WHO über den Krankenversicherungsschutz in verschiedenen Ländern kombiniert, um den Kontext abzubilden, in dem das neue Produkt eingesetzt werden soll. Verbunden mit einer Bedarfsanalyse aus Ausschreibungsunterlagen von Kliniken auf verschiedenen Kontinenten und Zulassungsanforderungen von benannten Stellen ergibt sich ein Datensatz, der die Entscheidung unterstützt, welche Märkte erschlossen werden können und sollen. Die Suche nach dem richtigen Markt und die Kenntnis der Bedingungen in dem Land führen zur richtigen Strategie und zur Aufstellung des richtigen Entwicklungsprojektes. Zuletzt können Produktdaten aus vergangenen Projekten hinzugefügt werden, um Übernahmeteile zu finden und aus den Erfahrungen zu lernen.

Abbildung 50 zeigt einen beispielhaften Auszug aus dem Engineering-Graph, der verdeutlicht, wie Daten aus verschiedenen externen Quellen mit den Produktdaten verbunden werden können. Die von den Vereinten Nationen veröffentlichten Ziele für nachhaltige Entwicklung (UN General Assembly, 2021) sind grün hervorgehoben, End- und Mittelpunkt aus einem LCSA (Bare et al., 2012) sind mit dem Ziel ‚Climate Action‘ verbunden und in Rot und Orange dargestellt. Die Bausteine des Gesundheitssystems, die von der WHO (World Health Organization, 2021) veröffentlicht wurden, sind mit dem Endpunkt ‚Human Health‘ verknüpft und die Produktfamilie des Unternehmens ist mit dem Baustein ‚Medical Products, Vaccines & Technologies‘ verbunden. Durch diese Kette sind die Informationen der UN, des LCSA und der WHO miteinander verbunden und direkt mit den Produktdaten verknüpft. Dies ermöglicht es Menschen und Maschinen, das Produkt in seinem weiteren Kontext zu verstehen.

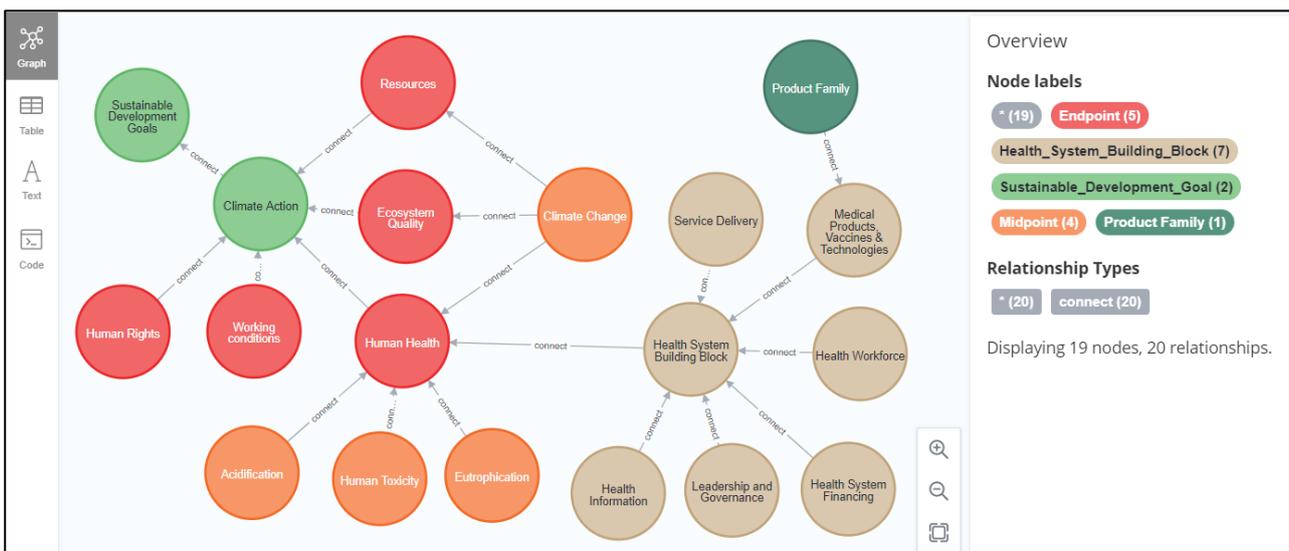


Abbildung 50. Auszug aus dem Engineering-Graph zur Unterstützung von Designentscheidungen

Die beiden genannten Beispiele zeigen, wie die Bereitstellung eines umfangreichen Datensatzes in den frühen Entwurfsphasen Entwurfsentscheidungen unterstützen kann. Die richtigen Entscheidungen in frühen Phasen zu treffen, ist nicht nur aus wirtschaftlicher Sicht notwendig, sondern trägt auch zur Verringerung des ökologischen Fußabdrucks bei, da die Auswirkungen späterer Phasen bereits berücksichtigt werden können.

5.4 Bewertung der Arbeit an den Anforderungen

In diesem Unterkapitel wird das in Kapitel 4 vorgestellte Konzept anhand der in Kapitel 2.3.2 aufgestellten Anforderungen bewertet. Dabei wird der Erfüllungsgrad der Anforderungen nach der Anwendung des Engineering-Graph mit dem Erfüllungsgrad aus der Literatur verglichen. Tabelle 12 zeigt die Erfüllungsgrade im Überblick.

Tabelle 12. Erfüllungsgrad der Anforderungen nach Einführung des Engineering-Graph

#	Anforderung	Erfüllungsgrad Literatur	Erfüllungsgrad Engineering-Graph
A1	Das Systemmodell soll verschiedene Datenbanken aus dem Bereich Produktentwicklung (zum Beispiel PLM, ERP, ALM, IoT etc.) miteinander verbinden können.	●	●
A2	Das Systemmodell soll verschiedene Modelle (zum Beispiel MBSE, Simulation etc.) aus dem Bereich Produktentwicklung miteinander verbinden können.	◐	●
A3	Das Systemmodell soll Modelle und Datenbanken miteinander verbinden können.	◐	●
A4	Das Systemmodell soll unstrukturierte, frei verfügbare Daten (wie Normen, Ausschreibungsdokumente, Fachartikel) mit den bestehenden Daten verbinden können.	◑	●
A5	Das Systemmodell soll frei verfügbares Kontextwissen (wie Wikimedia Knowledge Graph) mit den bestehenden Daten verbinden können.	○	●

A6	Das Systemmodell soll semantische Beziehungen zwischen den Daten abbilden können.	●	●
A7	Das Systemmodell soll um strukturierte und unstrukturierte Daten flexibel und dynamisch erweiterbar sein.	◐	●

Der Erfüllungsgrad von Anforderung 1 hat sich nicht verändert, da diese Anforderung bereits durch die Konzepte aus der Literatur vollständig erfüllt ist. Auch der Engineering-Graph besitzt die Fähigkeit, Informationen aus verschiedenen Datenbanken miteinander zu vernetzen, und erfüllt damit die Anforderung vollständig.

Anforderung 2 wird durch den Engineering-Graph nahezu vollständig erfüllt. Dies liegt an den Fähigkeiten des Graphen, Informationsobjekte jeglichen Datenformats miteinander zu vernetzen. In der prototypischen Umsetzung, die in diesem Kapitel vorgestellt wurde, findet diese Vernetzung manuell statt. Damit konnte gezeigt werden, dass die Vernetzung möglich ist.

Anforderung 3 ist erfüllt, da Datenbanken und Modelle durch den Graphen miteinander verbunden werden können. Diese Vernetzung wurde in der prototypischen Umsetzung des Graphen manuell durchgeführt.

Frei verfügbare Daten können im Engineering-Graph sehr gut eingebunden werden. Dies wurde unter anderem in Kapitel 4.2 vorgestellt. Diese Informationen können mit internen Daten aus Datenbanken oder Modellen verknüpft werden. Für die prototypische Umsetzung des Engineering-Graph wurde diese Verknüpfung manuell durchgeführt.

Kontextwissen, das bereits im Wikimedia oder Google Knowledge Graph modelliert ist und vorliegt, wird von den aktuellen Konzepten der Literatur kaum beachtet. Der Engineering-Graph nutzt dieses Wissen und schafft eine Verbindung zu den internen Datenbanken und Modellen der Produktentwicklung. Damit ist Anforderung 5 vollständig erfüllt.

Die Relationen zwischen den Datenpunkten des Engineering-Graph können quantitative und qualitative Attribute haben, in denen semantische Beziehungen abgebildet werden können. Daher kann Anforderung 6 als erfüllt angesehen werden. Es existieren bereits mehrere Konzepte in der Literatur, die semantische Beziehungen zwischen Daten im Engineering nutzen.

Die Technologie-Graph-Datenbank, auf der der Engineering-Graph beruht, erlaubt die Vernetzung verschiedener Datenpunkte unabhängig vom Schema der Daten. Daher kann der Engineering-Graph flexibel und dynamisch während der Nutzungsphase erweitert werden. Auch neue Datenquellen mit

unvorhergesehenem Schema können einbezogen und genutzt werden. Anforderung 7 kann also als weitestgehend erfüllt betrachtet werden.

Insgesamt kann der Engineering-Graph bei fünf von sieben Anforderungen eine Verbesserung erzielen. Die verbliebenen beiden waren bereits durch bestehende Konzepte vollständig erfüllt. Diese werden auch vom Engineering-Graph vollständig erfüllt. Der Engineering-Graph kann damit als Konzept für ein Systemmodell, das für zukünftige KI-Anwendungen die Grundlage schafft, verwendet werden. Die aufgestellten Anforderungen werden vollständig erfüllt. Allerdings wurden die einzelnen Schritte, die zum Aufbau und zur Verlinkung des Graphen notwendig sind, in der prototypischen Umsetzung manuell durchgeführt. Damit konnte gezeigt werden, dass die Anforderungen erfüllt sind und das Konzept grundsätzlich geeignet ist, allerdings ist es in dieser Form zum produktiven und wirtschaftlichen Einsatz im Unternehmen noch nicht bereit. In Folgearbeiten könnten die Möglichkeiten der Automatisierung untersucht werden. Kapitel 6 wird einen Ausblick auf mögliche zukünftige Forschung geben, in der die IT-Umsetzung automatisiert, die Methodik weiterentwickelt und weitere Anwendungsfälle untersucht werden können.

6 Zusammenfassung und Ausblick

In diesem Kapitel folgen nach der Zusammenfassung die Limitationen der Arbeit und ein Fazit. Abschließend wird mögliche zukünftige Forschung vorgestellt.

6.1 Zusammenfassung

In dieser Arbeit wird der Engineering-Graph vorgestellt. Er verbindet Daten aus verschiedenen Datenbanken der Produktentwicklung mit Daten aus Modellen und unternehmensexternen Daten wie dem Wikimedia oder Google Knowledge Graph, spezialisierten Quellen wie Normen und Regularien, Blogbeiträgen und Fachartikeln sowie Veröffentlichungen von NGOs wie der WHO oder der Weltbank. Dieses menschen- und maschinenlesbare Systemmodell dient als Quelle für Kontextinformationen. Mithilfe dieser Kontextinformationen können mögliche KI-Anwendungen trainiert werden, um Ingenieur_innen bei der Beherrschung der Komplexität in der Produktentwicklung zu unterstützen.

In Kapitel 1 wurde die Motivation der Arbeit aufgezeigt. Diese ist die zunehmende Komplexität und Dynamik in der Produktentwicklung, die Unterstützung von Ingenieur_innen erforderlich macht. Diese Unterstützung könnte durch KI-Technologie geleistet werden. In Kapitel 2 werden die für diese Arbeit relevanten Begriffe aus den Ingenieurwissenschaften und der Informatik vorgestellt sowie die Anforderungen an diese Arbeit abgeleitet. Anschließend wird in Kapitel 3 der aktuelle Stand der Technik vorgestellt und der Handlungsbedarf aufgezeigt. Hier wird verdeutlicht, dass KI-Technologie geeignet ist, die Beherrschung der Komplexität zu erleichtern, und welche Voraussetzungen dafür erfüllt sein müssen. Diese Voraussetzungen sind aktuell in den Ingenieurwissenschaften nicht erfüllt, da ein Systemmodell fehlt, das alle Informationen aus der Produktentwicklung sowie dem Kontext zusammenfasst. In Kapitel 4 wird der Engineering-Graph beschrieben, der diese Lücke schließen soll. Es wird gezeigt, wie die Erstellung und Pflege des Graphen durchgeführt werden können und wie der Graph Anwendungsfälle in der Produktentwicklung unterstützen kann. In Kapitel 5 werden verschiedene Anwendungsfälle vorgestellt, bei denen der Engineering-Graph erfolgreich verprobt wurde.

6.2 Limitationen

Eine Limitation der Arbeit ist die Umsetzung des Konzeptes, die nur prototypisch vorgenommen wurde. Es wurden nicht alle möglichen Elemente des Konzeptes umgesetzt. Das Rollenkonzept, eine Benutzeroberfläche, die Anbindung weiterer Quellen und viele weitere Möglichkeiten wurden noch nicht realisiert. Im Fokus dieser Arbeit stand die Entwicklung des Konzeptes und die Validierung der Kernelemente. Die vollständige Umsetzung des Konzeptes könnte in Folgearbeiten realisiert werden.

Mit jeder Erweiterung des Einsatzes in der industriellen Praxis werden neue Erkenntnisse gewonnen, die zur Erhöhung der Reife des wissenschaftlichen Konzeptes und der industriellen Praxis genutzt werden können.

Es wurden nicht alle Schnittstellen zu IT-Werkzeugen, die Unternehmen in der Produktentwicklung einsetzen, realisiert und getestet. Die Vielzahl der spezialisierten Werkzeuge, die in einer modernen Entwicklungsabteilung eingesetzt werden, macht dies aus Kosten und Zeitgründen nahezu unmöglich. Daher wurden die Anbindung und Vernetzung nur prinzipiell beschrieben und beispielhaft umgesetzt.

Eine weitere Limitation ist die fehlende Anbindung von weiteren Modellen. In dieser Arbeit wurde gezeigt, wie SysML-Modelle aus dem Bereich Systems-Engineering angebunden werden können. In der Produktentwicklung und in Unternehmen allgemein wird allerdings eine Vielzahl weiterer Modelle erstellt und eingesetzt. Die Beschreibung dieser Modelle und eine Anbindung an den Engineering-Graph konnten im Rahmen dieser Arbeit nicht durchgeführt werden. Das Konzept ist grundsätzlich offen und kann auch weitere Modelle aufnehmen, jedoch fehlt eine konkrete Beschreibung für weitere Modelle.

Eine Anbindung von zusätzlichen unternehmensexternen Quellen der Ebenen 4 und 5 des System of Interest wurde nicht durchgeführt. Im Rahmen dieser Arbeit wurde gezeigt, dass der Engineering-Graph ein offenes Konzept ist, das verschiedene Datentypen und -quellen vernetzen kann. Allerdings lag der Fokus auf der Medizintechnik und daher wurden vor allem Quellen, die für diese Branche wichtig sind, angebunden. Es wurde grundsätzlich aufgezeigt, wie verschiedene Datentypen wie Webseiten, PDF-Dokumente und quantitative Daten durch den Engineering-Graph vernetzt werden können. Die konkrete Umsetzung wurde jedoch nur an wenigen Beispielen durchgeführt. Daher ist unklar, ob es bei der Anbindung von neuen Quellen in der Praxis noch zu unvorhergesehenen Schwierigkeiten kommen kann.

Die Aktualisierung der Daten innerhalb des Engineering-Graph wurde nur konzeptionell beschrieben, aber nicht prototypisch umgesetzt. Daher kann hier keine Aussage getroffen werden, ob das Konzept so in der Praxis tatsächlich funktionieren wird. Es ist möglich, dass hier Anpassungen am Konzept notwendig sein werden, wenn die Aktualisierung der Daten besser erprobt sein wird.

Zur praktischen Umsetzung des Engineering-Graph im Unternehmen muss dieser in die IT-Landschaft eingebunden werden. Die IT-Landschaft wird in Unternehmen durch die IT-Strategie geprägt. So nutzen einige Unternehmen Cloud-Lösungen, während andere dies vermeiden. Eine Einbindung des Engineering-Graph in die IT-Landschaft wurde im Rahmen dieser Arbeit nicht beschrieben und durchgeführt. Da das Konzept auf Graph-Datenbanken basiert, könnte es flexibel genug sein, um sich in verschiedene IT-Landschaften einzugliedern. Zukünftige Forschung ist hier

notwendig, um die Einbindung des Engineering-Graph in verschiedenen IT-Landschaften zu beschreiben und zu erproben.

6.3 Fazit anhand der Forschungsfragen

Der Engineering-Graph ist das erste Konzept, das eine Verbindung von unternehmensinternen und -externen Daten erlaubt. Dies ist ein wichtiger Schritt in die Richtung eines besseren Verständnisses des Kontextes, in dem Produktentwicklung durchgeführt wird, und des Kontextes, in dem ein Produkt eingesetzt wird.

Forschungsfrage 1, wie KI in der Produktentwicklung zur Unterstützung von Ingenieur_innen bei der Bewältigung der Komplexität eingesetzt werden kann, wurde beantwortet, indem die vier Unterfragen systematisch beantwortet wurden. Zunächst wurden durch eine Literaturrecherche die Voraussetzungen für den Einsatz von KI untersucht und in Kapitel 3.2.2 beschrieben. Die Anforderungen von KI an ein Systemmodell wurden in Kapitel 2.3.2 vorgestellt. Das Konzept des Engineering-Graph, das in Kapitel 4 eingeführt wurde, nutzt MBSE-Modelle als eine wichtige Basis. Als Modellierungssprache und -methode wurden Graph-Datenbanken ausgewählt. Ein genauer Aufbau des Systemmodells im Engineering-Graph wurde in Kapitel 4.2 beschrieben. Zur Beantwortung der Forschungsfrage 2, wie die Erstellung und Pflege des Systemmodells in die Produktentwicklung integriert werden können, wurden die Rollen im Systems-Engineering durch den Graph Architect erweitert. In Kapitel 4.3 wurden Kompetenzen und Verantwortlichkeiten der neuen Rolle vorgestellt, um den Engineering-Graph in die Produktentwicklung zu integrieren. Damit wurden in der Arbeit alle Forschungsfragen erfolgreich beantwortet.

Eine abschließende Bewertung des Konzeptes ist mit dem aktuellen Forschungsstand nicht möglich. Dafür müsste das Konzept umfassender umgesetzt und erprobt werden. Die in dieser Arbeit aufgestellten Anforderungen werden durch das Konzept erfüllt. Im folgenden Unterkapitel wird beschrieben, welche Folgearbeiten zur umfassenden Umsetzung und Erprobung mit dem aktuellen Wissensstand denkbar sind.

6.4 Ausblick auf zukünftige Forschung und Folgearbeiten

Zukünftige Forschung und Umsetzungsarbeiten im Bereich dieser Arbeit ist in vielen Bereichen möglich. Zunächst könnte das in dieser Arbeit vorgestellte Konzept vollständig umgesetzt werden. Dies würde eine ganzheitliche Bewertung und Weiterentwicklung des Konzeptes ermöglichen.

Der Aufbau und die Pflege des Engineering-Graph sind aktuell mit manuellen Aufwänden verbunden, sodass sich der Einsatz hauptsächlich für große Unternehmen lohnt. Es sollte untersucht werden, wie

das Konzept auch für kleine und mittelständische Unternehmen (KMUs) ökonomisch erfolgreich eingesetzt werden kann. Dies könnte zum Beispiel durch eine Verschlinkung des Konzeptes erfolgen.

Eine weitere Möglichkeit ist die Erstellung und Pflege des Engineering-Graph durch ein spezialisiertes Unternehmen, das den Engineering-Graph als Service für andere Unternehmen anbietet. So könnten die Kosten insgesamt gesenkt werden, da die frei verfügbaren Kontextinformationen nur einmal miteinander verbunden werden müssen. Diese können dann genutzt werden, um die individuelle Lösung bei einem Unternehmen durch Anbindung der spezifischen Produktdaten und Modelle zu erzeugen. Auf diese Art würden alle Unternehmen von der Aktualisierung und Erweiterung des Kontextwissens durch das anbietende Unternehmen profitieren, ohne eigene Ressourcen einsetzen zu müssen.

Eine Automatisierung des Aufbaus und der Pflege des Engineering-Graph wurde im aktuellen Konzept noch nicht berücksichtigt. Hier gibt es noch große Potenziale, die zukünftig genutzt werden könnten, um manuelle Aufwände zu reduzieren. Durch die Reduktion des Aufwandes wird der Einsatz des Engineering-Graph sowohl in großen Unternehmen als auch in KMUs wahrscheinlicher. Mit einem breiteren Einsatz kann auch das Konzept weiterentwickelt werden und die oben beschriebenen Lücken können geschlossen werden.

Der in dieser Arbeit vorgestellte Engineering-Graph bietet eine Grundlage, auf der IT-Werkzeuge zur Unterstützung von Ingenieur_innen in der Produktentwicklung bei der Beherrschung der Komplexität erzeugt werden können. Diese konkreten Anwendungen für verschiedene Rollen im Produktentwicklungsprozess müssen in Zukunft entwickelt und erprobt werden. Eine solche Erprobung findet auch durch andere Arbeiten und Organisationen statt. So werden in dem in Kapitel 3.2.2.2 vorgestellten KI Marktplatz Anbieter und Kunden von KI Lösungen für die Produktentwicklung zusammengebracht. Eine Vernetzung von Engineering Objekten wird in dem in Kapitel 3.2.3.2 vorgestellten Bosch Semantic Stack entwickelt und kommerziell angeboten.

Das Konzept des Engineering-Graph bietet viele Anwendungsmöglichkeiten und kann somit vielfältig eingesetzt werden. Da es sich um ein Konzept mit einem sehr großen Scope handelt, hat sich diese Arbeit auf die Entwicklung und prototypische Validierung der fundamentalen Grundlagen fokussiert. Zukünftige Forschung wird das Konzept weiterentwickeln, erproben und dazu beitragen, erfolgreich Produkte zu entwickeln.

Abkürzungsverzeichnis

BPMN – Business Process Modeling Notation
CAD – Computer Aided Design
CAE – Computer Aided Engineering
CAM – Computer Aided Manufacturing
COMPASS – Comprehensive Modeling for Advanced SoS
CSV – comma separated values
DRM – Design Research Methodology
ERP – Enterprise Resource Planning
IoT – Internet of Things
IoT – Internet of Things
IT – Information Technology
KI – Künstliche Intelligenz
KMU – kleine und mittelständische Unternehmen
LCA – Life Cycle Assessment
LCC – Life Cycle Costing
LCE – Life Cycle Engineering
LCSA – Life Cycle Sustainability Assessment
LML – Lifecycle Modeling Language
LPG – Label Property Graph
MBSE – Model Based Systems Engineering
NLP – Natural Language Processing
NGO – Non Government Organization
PLM – Product Lifecycle Management
RDF – Resource Description Framework
SE – Systems Engineering
S-LCA – Social-Life Cycle Assessment
SysML – System Modeling Language

UML – Universal Modeling Language

WHO – World Health Organization

Abbildungsverzeichnis

Abbildung 1. Forschungsziele.....	3
Abbildung 2. Forschungsfragen.....	4
Abbildung 3. Vorgehen bei dieser Dissertation.....	5
Abbildung 4. Aufbau dieser Dissertation	5
Abbildung 5. Bestandteile der Künstlichen Intelligenz (Paschen et al., 2019).....	18
Abbildung 6. Framework zur Ableitung von Anforderungen der Künstlichen Intelligenz an ein Systemmodell (Schweitzer et al., 2021).....	23
Abbildung 7. Einsatz von Produktlebenszyklusmanagement nach (Abramovici & Schulte, 2004).....	29
Abbildung 8. V-Modell nach (Gausemeier & Moehring, 2002).....	31
Abbildung 9. MBSE-Dreieck nach (Dumitrescu et al., 2015)	32
Abbildung 10. Ebenen des System-of-Interest nach (Bitzer et al., 2022).....	33
Abbildung 11. Mehrwert von Model-based-Systems-Engineering nach (Cloutier, 2019b).....	35
Abbildung 12. Anzahl der Publikationen mit ‚Data Science‘ in Titel oder Abstract nach (dimensions.ai, 2023).....	38
Abbildung 13. Erhoffte Ergebnisse durch Data-Science nach (World Data Science Initiative, 2022).40	
Abbildung 14. Übersicht von Graph-Data-Science Algorithmen nach (Neo4J, 2022e).....	41
Abbildung 15. Übersicht der Forschung zur Künstlichen Intelligenz nach (Wahlster, 2019).....	42
Abbildung 16. Künstliche Intelligenz und ihre Hauptfunktionen.....	44
Abbildung 17. SWOT-Analyse der aktuellen Fähigkeiten von Künstlicher Intelligenz.....	46
Abbildung 18. Visualisierung einer Graph-Datenbank und einer relationalen Datenbank nach (BI Connector Team, 2021).....	49
Abbildung 19. Neutrale Dateninterfaces (Eigner & Sindermann, 2014).....	55
Abbildung 20. Aufbau des SP ² IDER-Graph (Eickhoff et al., 2020).....	57
Abbildung 21. CACDA-Vorschlagessystem für Designregeln (Huet et al., 2021).....	59
Abbildung 22. Semantisches Datenmanagement (Abramovici et al., 2016).....	60
Abbildung 23. Architektur der Kooperation von PTC und Microsoft (Möring & Feldmann, 2022)	62
Abbildung 24. Engineering-Graph verbindet Informationen über alle Ebenen des System-of-Interest nach (Schweitzer et al., 2023b).....	67

Abbildung 25. Übersicht Engineering-Graph.....	67
Abbildung 26. Aufbau des Engineering-Graph nach (Schweitzer et al., 2023a).....	68
Abbildung 27. Engineering-Graph mit Daten aus dem 'Society-Eco-System'	69
Abbildung 28. Engineering-Graph mit 'Society-Eco-System' und 'Industry-System'	69
Abbildung 29. Engineering-Graph mit 'Product-Environment'.....	70
Abbildung 30. Engineering-Graph mit allen Ebenen des System-of-Interest.....	71
Abbildung 31. Schritte zur Erweiterung des Engineering-Graph nach (Schweitzer et al., 2023a)	71
Abbildung 32. Identifikation des Verbindungstyps nach (Schweitzer et al., 2023a).....	72
Abbildung 33. Aufgaben der neuen Rolle des Graph Architects	75
Abbildung 34. Graph-Data-Science: Gemeinschaft.....	78
Abbildung 35. Graph-Data-Science: Zentralität.....	79
Abbildung 36. Graph-Data-Science: Vorhersage	80
Abbildung 37. Erkennung von Zusammenhängen mit dem Engineering-Graph	81
Abbildung 38. Engineering-Graph mit Wikimedia-Daten.....	85
Abbildung 39. Engineering-Graph mit Artikel.....	86
Abbildung 40. Engineering-Graph mit WHO- und LCSA-Daten	86
Abbildung 41. Engineering-Graph mit PLM-Daten.....	87
Abbildung 42. Allgemeine Klassifizierung der Anwendungsfälle.....	88
Abbildung 43. Einordnung der Anwendungsfälle in die Quadranten.....	89
Abbildung 44. Kontext und Reifegrad der Kennzahl nach (Schweitzer et al., 2020).....	91
Abbildung 45. Mock-up eines Engineering-Dashboards nach (Schweitzer et al., 2020).....	92
Abbildung 46. Schritte zum Einsatz von Natural-Language-Processing.....	93
Abbildung 47. Natural-Language-Processing für Anforderungsmanagement.....	94
Abbildung 48. Engineering-Graph mit LCSA-Daten.....	96
Abbildung 49. Beziehung zwischen Systemwissen, Designflexibilität und Ressourceneinsatz nach (Boy, 2019)	98
Abbildung 50. Auszug aus dem Engineering-Graph zur Unterstützung von Designentscheidungen	100

Tabellenverzeichnis

Tabelle 1. Definitionen Künstlicher Intelligenz (Russell et al., 2016)	15
Tabelle 2. Einsatzgebiete Künstlicher Intelligenz.....	16
Tabelle 3. Rollen im Systems-Engineering nach (Sheard, 1996).....	34
Tabelle 4. Herausforderungen im Life-Cycle-Engineering	36
Tabelle 5. Herausforderungen bei einem Life-Cycle-Sustainability-Assessment	37
Tabelle 6. Bewertung ausgewählter Modellierungssprachen nach (Kutaganahalli, 2022).....	54
Tabelle 7. Erfüllungsgrad der Anforderungen neutrale Dateninterfaces.....	56
Tabelle 8. Erfüllungsgrad der Anforderungen SP ² IDER Konzept.....	58
Tabelle 9. Erfüllungsgrad der Anforderungen CACDA	59
Tabelle 10. Erfüllungsgrad der Anforderungen semantisches Datenmanagement	61
Tabelle 11. Erfüllungsgrad der Anforderungen an diese Arbeit	63
Tabelle 12. Erfüllungsgrad der Anforderungen nach Einführung des Engineering-Graph.....	101

Literaturverzeichnis

- Abramovici, M., Göbel, J. C., & Dang, H. B. (2016). Semantic data management for the development and continuous reconfiguration of smart products and systems. *CIRP Annals*, 65(1), 185–188. <https://doi.org/10.1016/j.cirp.2016.04.051>
- Abramovici, M., & Schulte, S. (2004). *Benefits of PLM-Nutzenpotentiale des Product Lifecycle Managements in der Automobilindustrie, Benchmark Studie*. IBM Verlag, Frankfurt.
- Ackoff, R. L. (1989). From data to wisdom. *Journal of applied systems analysis*, 16(1), 3–9.
- Aguwa, C., Olya, M. H., & Monplaisir, L. (2017). Modeling of fuzzy-based voice of customer for business decision analytics. *Knowledge-Based Systems*, 125, 136–145. <https://doi.org/10.1016/j.knosys.2017.03.019>
- AI HLEG. (2019). *A definition of Artificial Intelligence: Main capabilities and scientific disciplines | Shaping Europe's digital future*. <https://digital-strategy.ec.europa.eu/en/library/definition-artificial-intelligence-main-capabilities-and-scientific-disciplines>
- Al-Fedaghi, S., & Shbeeb, D. (2019). Petri nets and machines of things that flow. *Intelligent Systems and Applications: Proceedings of the 2018 Intelligent Systems Conference (IntelliSys) Volume 2*, 238–252.
- Alves, J., Lima, T. M., & Gaspar, P. D. (2023). Is Industry 5.0 a Human-Centred Approach? A Systematic Review. *Processes*, 11(1), Article 1. <https://doi.org/10.3390/pr11010193>
- Anderl, R. (1998). *Produktdatentechnologie III – Produktdatenmanagement. Vorlesungsskript*. Fachgebiet Datenverarbeitung in der Konstruktion.
- Angles, R. (2018). The Property Graph Database Model. *AMW*.

- Angles, R., & Gutierrez, C. (2005). Querying RDF Data from a Graph Database Perspective. In A. Gómez-Pérez & J. Euzenat (Hrsg.), *The Semantic Web: Research and Applications* (S. 346–360). Springer. https://doi.org/10.1007/11431053_24
- Angles, R., & Gutierrez, C. (2008). Survey of graph database models. *ACM Computing Surveys*, 40(1), 1:1-1:39. <https://doi.org/10.1145/1322432.1322433>
- Annarelli, A., Battistella, C., & Nonino, F. (2016). Product service system: A conceptual framework from a systematic review. *Journal of Cleaner Production*, 139, 1011–1032. <https://doi.org/10.1016/j.jclepro.2016.08.061>
- Autodesk. (2023). *Autodesk empowers innovators everywhere to make the new possible*. <https://www.autodesk.com/>
- Aziz, H., Gao, J., Maropoulos, P., & Cheung, W. M. (2005). Open standard, open source and peer-to-peer tools and methods for collaborative product development. *Computers in Industry*, 56(3), 260–271.
- Bare, J., Hofstetter, P., Pennington, D., & Haes, H. (2012). Midpoints Versus Endpoints: The Sacrifices and Benefits. *Int. J. Life Cycle Ass.*, 5, 319–326. <https://doi.org/10.1007/BF02978665>
- Bates, M., & Weischedel, R. M. (2006). *Challenges in Natural Language Processing*. Cambridge University Press.
- Beckett, D., Berners-Lee, T., Prud'hommeaux, E., & Carothers, G. (2014). RDF 1.1 Turtle. *World Wide Web Consortium*, 18–31.
- Beliga, S. (2014). Keyword extraction: A review of methods and approaches. *Journal of Information and Organizational Sciences*.
- Bellman, R. (1978). *An introduction to artificial intelligence: Can computers think?* Thomson Course Technology.

- Benoît, C., & Mazijn, B. (2013). *Guidelines for social life cycle assessment of products*. United Nations Environment Programme, CIRAI, Interuniversity Research Centre for the Life Cycle of Products, P. and S.
- Benoît Norris, C., Traverso, M., Neugebauer, S., Ekener, E., Schaubroeck, T., Russo Garrido, S., Berger, M., Valdivia, S., Lehmann, A., Finkbeiner, M., & Arcese, G. (2020). *Guidelines for Social Life Cycle Assessment of Products and Organizations*.
- Bentley, P. (2018). The three laws of artificial intelligence: Dispelling common myths. *Should we fear artificial intelligence*, 6–12.
- Bertoni, A., Larsson, T., Larsson, J., & Elfsberg, J. (2017). Mining data to design value: A demonstrator in early design. *DS 87-7 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 7: Design Theory and Research Methodology, Vancouver, Canada, 21-25.08.2017*, 021–029.
- BI Connector Team. (2021). *Knowledge Graph vs Traditional Database*. BI Connector Blog. <https://www.biconnector.com/blog/knowledge-graph-vs-relational-database-differences/>
- Bitzer, M., Eigner, M., Faißt, K.-G., Muggeo, C., & Eickhoff, T. (2017). Framework of the evolution in virtual product modelling and model management towards digitized engineering. *DS 87-6 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 6: Design Information and Knowledge, Vancouver, Canada, 21-25.08.2017*.
- Bitzer, M., Kleiner, S., Michels, N., Bauch, M., Kaspar, J., & Zingel, C. (2022). Systems Engineering at Fresenius Medical Care – Starting a transformation. *TdSE*, 50–54.
- Bitzer, M., & Vielhaber, M. (2016). Product Lifecycle Management – How to adapt PLM to support changing product development processes in industry? *DS 85-1: Proceedings of NordDesign 2016, Volume 1, Trondheim, Norway, 10th - 12th August 2016*.

<https://www.designsociety.org/publication/39313/Product+Lifecycle+Management+%E2%80%93+How+to+adapt+PLM+to+support+changing+product+development+processes+in+industry%3F>

Blanchard, B. S., & Fabrycky, W. J. (2006). *Systems engineering and analysis* (4th ed). Pearson Prentice Hall. <http://books.google.com/books?id=KnNRAAAAMAAJ>

Blessing, L. T. M., & Chakrabarti, A. (Hrsg.). (2009). DRM: A Design Research Methodology. In *DRM, a Design Research Methodology* (S. 13–42). Springer. https://doi.org/10.1007/978-1-84882-587-1_2

Booch, G., Rumbaugh, J., & Jacobson, I. (2005). *Unified Modeling Language User Guide, The, 2nd Edition* (2nd Aufl.). Addison-Wesley Professional. Part of the Addison-Wesley Object Technology Series series. <https://www.informit.com/store/unified-modeling-language-user-guide-9780321267979>

Boy, G. (2019). From Rigid to Flexible – From Virtual to Tangible an Evolution of Human-Centered Design. *Proceedings of the 20th Congress of the International Ergonomics Association*, 54–63. https://doi.org/10.1007/978-3-319-96071-5_6

Bracht, U., Geckler, D., & Wenzel, S. (2018). *Digitale Fabrik: Methoden und Praxisbeispiele* (2., aktualisierte und erweiterte Auflage). Springer Vieweg. <https://doi.org/10.1007/978-3-662-55783-9>

Brin, S., & Page, L. (1998). The anatomy of a large-scale hypertextual Web search engine. *Computer Networks and ISDN Systems*, 30(1), 107–117. [https://doi.org/10.1016/S0169-7552\(98\)00110-X](https://doi.org/10.1016/S0169-7552(98)00110-X)

- Bundeszentrale für politische Bildung. (2020). *Verschobene Machtverhältnisse: Handelskrieg zwischen China und den USA*. bpb.de. <https://www.bpb.de/themen/asien/china/322240/verschobene-machtverhaeltnisse-handelskrieg-zwischen-china-und-den-usa/>
- Cerdas, F., Thiede, S., & Herrmann, C. (2018). Integrated Computational Life Cycle Engineering—Application to the case of electric vehicles. *CIRP Annals*, 67(1), 25–28. <https://doi.org/10.1016/j.cirp.2018.04.052>
- Charniak, E., & McDermott, D. (1985). Introduction to artificial intelligence. Addison. Reading, MA.
- Chen, P.-H. (2020). Essential Elements of Natural Language Processing: What the Radiologist Should Know. *Academic Radiology*, 27(1), 6–12. <https://doi.org/10.1016/j.acra.2019.08.010>
- Chiarello, F., Belingheri, P., & Fantoni, G. (2021). Data science for engineering design: State of the art and future directions. *Computers in Industry*, 129, 103447. <https://doi.org/10.1016/j.compind.2021.103447>
- Chiprianov, V., Falkner, K., Gallon, L., & Munier, M. (2014). Towards modelling and analysing non-functional properties of systems of systems. *2014 9th International Conference on System of Systems Engineering (SOSE)*, 289–294.
- Chowdhury, M., & Sadek, A. W. (2012). Advantages and limitations of artificial intelligence. *Artificial intelligence applications to critical transportation issues*, 6(3), 360–375.
- Christensen, K., Nørskov, S., Frederiksen, L., & Scholderer, J. (2017). In Search of New Product Ideas: Identifying Ideas in Online Communities by Machine Learning and Text Mining. *Creativity and Innovation Management*, 26(1), 17–30. <https://doi.org/10.1111/caim.12202>
- Cloutier, R. (2019a). 2018 MBSE survey results. *Proceedings of the 2019 INCOSE MBSE Workshop, presented at the INCOSE 2019 International Workshop, Torrance, CA, USA*. https://sebokwiki.org/wiki/Model-Based_Systems_Engineering_Adoption_Trends_2009-2018

- Cloutier, R. (2019b, Januar). *2019 JPL MBSE Survey Results*. 2019 Jet Propulsion Lab (JPL) MBSE Workshop, Pasadena, CA, USA.
- Conweaver. (2023). *Connected Engineering*. <https://www.conweaver.com/solutions/de-connected-engineering>
- Cook, D. J., & Holder, L. B. (2006). *Mining Graph Data*. John Wiley & Sons.
- Costa, D., Quinteiro, P., & Dias, A. C. (2019). A systematic review of life cycle sustainability assessment: Current state, methodological challenges, and implementation issues. *Science of The Total Environment*, 686, 774–787. <https://doi.org/10.1016/j.scitotenv.2019.05.435>
- Dassault. (2023). *Die 3DEXPERIENCE Plattform*. Dassault Systèmes. <https://www.3ds.com/de/3dexperience>
- DataHub. (2023). *A Metadata Platform for the Modern Data Stack | DataHub*. <https://datahubproject.io/>
- Department of Defence. (1998). *DoD Modeling and Simulation (M&S) Glossary*. <https://apps.dtic.mil/sti/citations/ADA349800>
- Devlin, J., Chang, M.-W., Lee, K., & Toutanova, K. (2019). BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding. *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, 4171–4186.
- Dilger, N., Kaluza, A., Kiesewetter, A., Cerdas, F., Blume, S., Zellmer, S., & Herrmann, C. (2021). Definition and Reference Framework for Life Cycle Technologies in Life Cycle Engineering—A Case Study on All Solid State Traction Batteries. *Procedia CIRP*, 98, 217–222. <https://doi.org/10.1016/j.procir.2021.01.033>

dimensions.ai. (2023). *Anzahl der Publikationen mit „Data Science“ in Titel oder Abstract.*

Dimensions.ai.

https://app.dimensions.ai/analytics/publication/overview/timeline?search_mode=content&search_text=%22data%20science%22&search_type=kws&search_field=text_search

DIN 19266:2015-05, *pH-Messung_- Referenzpufferlösungen zur Kalibrierung von pH-Messeinrichtungen.* (2015). Beuth Verlag GmbH. <https://doi.org/10.31030/2307479>

DIN EN ISO 14040. (2009). *DIN EN ISO 14040:2009-11, Umweltmanagement_- Ökobilanz_- Grundsätze und Rahmenbedingungen (ISO_14040:2006); Deutsche und Englische Fassung EN_ISO_14040:2006.* Beuth Verlag GmbH. <https://doi.org/10.31030/1555059>

DIN EN ISO 14044. (2006). *DIN EN ISO 14044:2006, Environmental management—Life cycle assessment—Requirements and guidelines.* <https://www.iso.org/cms/render/live/en/sites/isoorg/contents/data/standard/03/84/38498.html>

Duden. (2022). *Duden | Management | Rechtschreibung, Bedeutung, Definition, Herkunft.* Duden | Management | Rechtschreibung, Bedeutung, Definition, Herkunft. <https://www.duden.de/rechtschreibung/Management>

Dumitrescu, R., Albers, A., Riedel, O., Stark, R., & Gausemeier, J. (2021). *Engineering in Deutschland – Status quo in Wirtschaft und Wissenschaft, Ein Beitrag zum Advanced Systems Engineering.* Fraunhofer IEM, Paderborn. <https://www.hni.uni-paderborn.de/pub/10234>

Dumitrescu, R., Bremer, C., Kühn, A., Trächtler, A., & Friebe, T. (2015). Model-based development of products, processes and production resources. *at-Automatisierungstechnik*, 63(10), 844–857.

Eickhoff, T., Eiden, A., Göbel, J. C., & Eigner, M. (2020). A Metadata Repository for Semantic Product Lifecycle Management. *Procedia CIRP*, 91, 249–254.

- Eigner, M. (2016). Das Industrial Internet. In U. Sendler (Hrsg.), *Industrie 4.0 grenzenlos* (S. 137–168). Springer Vieweg. http://dx.doi.org/10.1007/978-3-662-48278-0_9
- Eigner, M., & Sindermann, S. (2014). Enabling System Lifecycle Management through neutral data interfaces—News and background on Digitalization, Artificial Intelligence and PLM. *PLM Portal*. <https://www.plmportal.org/en/research-detail/enabling-system-lifecycle-management-through-neutral-data-interfaces.html>
- Eigner, M., & Stelzer, R. (2013). *Product Lifecycle Management: Ein Leitfaden für Product Development und Life Cycle Management* (2., neu bearb. Aufl.). Springer. <https://link.springer.com/book/10.1007/b93672>
- Engels, G., Gogolla, M., Hohenstein, U., Hülsmann, K., Löhr-Richter, P., Saake, G., & Ehrich, H.-D. (1992). Conceptual modelling of database applications using an extended ER model. *Data & Knowledge Engineering*, 9(2), 157–204.
- Entringer, T. C., Ferreira, A. da S., & Nascimento, D. C. de O. (2021). Comparative analysis of the main business process modeling methods: A bibliometric study. *Gestão & Produção*, 28.
- Estefan, J. A. (2007). Survey of model-based systems engineering (MBSE) methodologies. *IncoSE MBSE Focus Group*, 25(8), 1–12.
- Eurostat. (2023). *Globale Wertschöpfungsketten—Wirtschaftliche Globalisierung*. <https://ec.europa.eu/eurostat/de/web/economic-globalisation/globalisation-in-business-statistics/global-value-chains>
- Fettke, P. (2009). How conceptual modeling is used. *Communications of the Association for Information Systems*, 25(1), 43.
- Flach, P. A. (2012). *Machine learning: The art and science of algorithms that make sense of data*. Cambridge University Press.

- Frank, U. (2013). Domain-specific modeling languages: Requirements analysis and design guidelines. *Domain engineering: Product lines, languages, and conceptual models*, 133–157.
- Fresenius Medical Care AG & Co. KGaA. (2023a). *Geschäftsbericht 2022*. https://factsandfigures.freseniusmedicalcare.com/fileadmin/2022/fme_geschaeftsbericht_2022_de.pdf
- Fresenius Medical Care AG & Co. KGaA. (2023b). *Produkte*. Fresenius Medical Care. <https://www.freseniusmedicalcare.com/de/medien/multimedia/bilder/produkte>
- Friedenthal, S., Griego, R., & Sampson, M. (2007). INCOSE model based systems engineering (MBSE) initiative. *INCOSE 2007 symposium*, 11.
- Gausemeier, J., Lanza, G., & Lindemann, U. (2012). *Produkte und Produktionssysteme integrativ konzipieren: Modellbildung und Analyse in der frühen Phase der Produktentstehung*. <https://mediatum.ub.tum.de/1167170>
- Gausemeier, J., & Moehring, S. (2002). VDI 2206- A New Guideline for the Design of Mechatronic Systems. *IFAC Proceedings Volumes*, 35(2), 785–790. [https://doi.org/10.1016/S1474-6670\(17\)34035-1](https://doi.org/10.1016/S1474-6670(17)34035-1)
- Gerhard, D. (2019). *Digital Engineering -von CAD zum Digital Twin*. <https://doi.org/10.13140/RG.2.2.27592.06401>
- Goldberg, Y. (2017). Neural Network Methods for Natural Language Processing. *Synthesis Lectures on Human Language Technologies*, 10(1), 1–309.
- Google. (2021). *Google Knowledge Graph Search API*. Google Developers. <https://developers.google.com/knowledge-graph?hl=de>

- Gopsill, J. A., McAlpine, H. C., & Hicks, B. J. (2011). Trends in technology and their possible implications on PLM: Looking towards 2020. *Proceedings... of International Conference on Product Lifecycle Management*.
- Haifeng Liu, Youliang Huang, Wee-Keong Ng, Bin Song, Xiang Li, & Wen-Feng Lu. (2007). Deriving configuration knowledge and evaluating product variants through intelligent techniques. *2007 6th International Conference on Information, Communications Signal Processing*, 1–5. <https://doi.org/10.1109/ICICS.2007.4449767>
- Hart, L. E. (2015). Introduction to model-based system engineering (MBSE) and SysML. *Delaware Valley INCOSE Chapter Meeting*, 30.
- Haskins, C., Forsberg, K., Krueger, M., Walden, D., & Hamelin, D. (2006). *Systems engineering handbook* (Bd. 9). International Council on Systems Engineering Seattle.
- Haugeland, J. (1989). *Artificial Intelligence: The Very Idea*. MIT Press.
- Hauschild, M. Z., Herrmann, C., & Kara, S. (2017). An integrated framework for life cycle engineering. *Procedia CIRP*, 61, 2–9.
- Hauschild, M. Z., Rosenbaum, R. K., & Olsen, S. I. (2018). *Life Cycle Assessment: Theory and Practice*. <https://link.springer.com/book/10.1007/978-3-319-56475-3>
- Hein, A. M., & Condat, H. (2018). Can Machines Design? An Artificial General Intelligence Approach. In M. Iklé, A. Franz, R. Rzepka, & B. Goertzel (Hrsg.), *Artificial General Intelligence* (S. 87–99). Springer International Publishing. https://doi.org/10.1007/978-3-319-97676-1_9
- Helms, M. M., & Nixon, J. (2010). Exploring SWOT analysis – where are we now? A review of academic research from the last decade. *Journal of Strategy and Management*, 3(3), 215–251. <https://doi.org/10.1108/17554251011064837>

- Henkel, R., Wolkenhauer, O., & Waltemath, D. (2015). Combining computational models, semantic annotations and simulation experiments in a graph database. *Database, 2015*(bau130). <https://doi.org/10.1093/database/bau130>
- Holt, J., Perry, S., Brownsword, M., Cancila, D., Hallerstedde, S., & Hansen, F. O. (2012). Model-based requirements engineering for system of systems. *2012 7th International Conference on System of Systems Engineering (SoSE)*, 561–566.
- Huet, A., Pingué, R., Véron, P., Segonds, F., & Fau, V. (2020). *Knowledge Graph of Design Rules for a Context-Aware Cognitive Design Assistant* (S. 334–344). https://doi.org/10.1007/978-3-030-62807-9_27
- Huet, A., Segonds, F., Pingué, R., Veron, P., Guegan, J., & Mallet, A. (2021). Context-aware cognitive design assistant: Implementation and study of design rules recommendations. *Advanced Engineering Informatics, 50*, 101419. <https://doi.org/10.1016/j.aei.2021.101419>
- Hunkeler, D., & Rebitzer, G. (2003). Life Cycle costing—Paving the road to sustainable development? *The International Journal of Life Cycle Assessment, 8*(2), 109–110. <https://doi.org/10.1007/BF02978435>
- INCOSE. (2022). *Systems Engineering Vision 2035*.
- Ingram, C., Payne, R., Fitzgerald, J., & Couto, L. D. (2015). Model-based Engineering of Emergence in a Collaborative SoS: Exploiting SysML & Formalism. *INCOSE International Symposium, 25*(1), 404–419.
- Iqbal, T., Elahidoost, P., & Lúcio, L. (2018). A Bird’s Eye View on Requirements Engineering and Machine Learning. *2018 25th Asia-Pacific Software Engineering Conference (APSEC)*, 11–20.
- ISO 15288. (2015). *ISO/IEC/IEEE 15288:2015*.
- ISO 62304. (2023). *IEC 62304:2006*. ISO. <https://www.iso.org/standard/38421.html>

- IT Designers. (2023). *reQlab*. Polarion Extensions. <https://extensions.polarion.com/extensions/333-reqlab-automated-requirements-quality-check-with-state-of-the-art-artificial-intelligence>
- its owl. (2023). *AI Marketplace*. KI-Marktplatz. <https://ki-marktplatz.com/en/>
- Jatana, N., Puri, S., Ahuja, M., Kathuria, I., & Gosain, D. (2012). A survey and comparison of relational and non-relational database. *International Journal of Engineering Research & Technology*, 1(6), 1–5.
- Kaluza, A., Gellrich, S., Cerdas, F., Thiede, S., & Herrmann, C. (2018). Life Cycle Engineering Based on Visual Analytics. *Procedia CIRP*, 69, 37–42. <https://doi.org/10.1016/j.procir.2017.11.128>
- Kaul, A., & Rao, V. R. (1995). Research for product positioning and design decisions: An integrative review. *International Journal of Research in Marketing*, 12(4), 293–320. [https://doi.org/10.1016/0167-8116\(94\)00018-2](https://doi.org/10.1016/0167-8116(94)00018-2)
- Kloepffer, W. (2008). Life cycle sustainability assessment of products: (With Comments by Helias A. Udo de Haes, p. 95). *The International Journal of Life Cycle Assessment*, 13(2), 89–95. <https://doi.org/10.1065/lca2008.02.376>
- Korthals, K., Auricht, M., & Felten, M. (2020). *Systems Engineering Solution Lab—Experience Model based Systems Engineering at CLAAS*. Prostep ivip Symposium 2020, 02.09.2020.
- Kotler, P., Burton, S., Adam, S., Brown, L., & Armstrong, G. (2007). *Marketing*. Pearson Education Australia.
- Koudal, P. (2005). *Mastering Innovation: Exploiting Ideas for Profitable Growth*.
- Kurzweil, R., Richter, R., Kurzweil, R., & Schneider, M. L. (1990). *The age of intelligent machines* (Bd. 580). MIT press Cambridge.
- Kutaganahalli, A. (2022). *Investigation of modeling languages and approaches for lifecycle engineering for medical devices*. [unveröffentlicht].

- Lashin, G., Willmann, C., & Andrae, R. (2020, Juni 9). *Success Metrics Driven by PLM: Vaillant Group's Digital Transformation*. Liveworx 2020.
- Learned, E. P., Andrews, K. R., Christensen, C. R., & Guth, W. D. (1969). *Business policy: Text and cases*. RD Irwin.
- Lee, I., & Lee, K. (2015). The Internet of Things (IoT): Applications, investments, and challenges for enterprises. *Business horizons*, 58(4), 431–440.
- Lee, J. M., & Wong, E. Y. (2021). Suez Canal blockage: An analysis of legal impact, risks and liabilities to the global supply chain. *MATEC Web of Conferences*, 339, 01019.
<https://doi.org/10.1051/mateconf/202133901019>
- Lifecycle Modeling Organization. (2022). *LML Specification 1.4*.
<https://www.lifecyclemodeling.org/specification>
- LifeCycleInitiative.org. (2022). Interactive map of LCA databases. *Life Cycle Initiative*.
<https://www.lifecycleinitiative.org/applying-lca/lca-databases-map/>
- Lunze, J. (2016). *Künstliche Intelligenz für Ingenieure* (3rd updated edition). Walter de Gruyter GmbH.
<https://www.beck-shop.de/lunze-de-gruyter-studium-kuenstliche-intelligenz-ingenieure/product/16271387>
- Mahesh, B. (2020). Machine learning algorithms-a review. *International Journal of Science and Research (IJSR)*. [Internet], 9, 381–386.
- Maier, M. W. (1998). Architecting principles for systems-of-systems. *Systems Engineering: The Journal of the International Council on Systems Engineering*, 1(4), 267–284.
- Microsoft. (2023). *Datenvisualisierung | Microsoft Power BI*. <https://powerbi.microsoft.com/de-de/>
- Microstrategy. (2020). *2020 Global State of Enterprise Analytics*.

- Modelica Association. (2023). *Modelica® – A Unified Object-Oriented Language for Systems Modeling—Language Specification*. <https://modelica.org/modelicalanguage.html>
- Möring, F., & Feldmann, R. (2022, Oktober 13). *Digital Engineering and Manufacturing*. PTC Dach Talks.
- Neo4J. (2022a). *Centrality—Neo4j Graph Data Science*. Neo4j Graph Data Platform. <https://neo4j.com/docs/graph-data-science/1.8/algorithms/centrality/>
- Neo4J. (2022b). *Community detection—Neo4j Graph Data Science*. Neo4j Graph Data Platform. <https://neo4j.com/docs/graph-data-science/1.8/algorithms/community/>
- Neo4J. (2022c). *Link prediction—Neo4j Graph Data Science*. <https://neo4j.com/docs/graph-data-science/current/algorithms/ml-models/linkprediction/>
- Neo4J. (2022d). *Similarity—Neo4j Graph Data Science*. Neo4j Graph Data Platform. <https://neo4j.com/docs/graph-data-science/1.8/algorithms/similarity/>
- Neo4J. (2022e). *The Neo4j Graph Data Science Library Manual v2.0—Neo4j Graph Data Science*. Neo4j Graph Data Platform. <https://neo4j.com/docs/graph-data-science/2.0/>
- Neo4J. (2023a). *Neo4j Graph Data Platform – The Leader in Graph Databases*. Neo4j Graph Data Platform. <https://neo4j.com/>
- Neo4J. (2023b). *Tutorial: Build a Knowledge Graph using NLP and Ontologies - Developer Guides*. Neo4j Graph Data Platform. <https://neo4j.com/developer/graph-data-science/build-knowledge-graph-nlp-ontologies/>
- Nielsen, C. B., Larsen, P. G., Fitzgerald, J., Woodcock, J., & Peleska, J. (2015). Systems of systems engineering: Basic concepts, model-based techniques, and research directions. *ACM Computing Surveys (CSUR)*, 48(2), 1–41.
- Nilsson, N. J. (1998). *Artificial intelligence: A new synthesis*. Morgan Kaufmann.

OMG. (2020). *What is SysML?* <http://www.omg-sysml.org/what-is-sysml.htm>

Oxford Advanced Learner's Dictionary. (2020a). *Definition Artificial*. Oxford Advanced Learner's Dictionary.

<https://www.oxfordlearnersdictionaries.com/definition/english/artificial?q=artificial>

Oxford Advanced Learner's Dictionary. (2020b). *Definition Artificial Intelligence*. Oxford Advanced Learner's Dictionary.

<https://www.oxfordlearnersdictionaries.com/definition/english/artificial-intelligence?q=artificial+intelligence>

Oxford Advanced Learner's Dictionary. (2020c). *Definition Intelligence*. Oxford Advanced Learner's Dictionary.

<https://www.oxfordlearnersdictionaries.com/definition/english/intelligence?q=intelligence>

Pang, A. (2022). Top 10 PLM and Engineering Software Vendors, Market Size and Market Forecast 2021-2026. *Apps Run The World*. <https://www.appsruntheworld.com/top-10-product-lifecycle-management-engineering-software-vendors-and-market-forecast/>

Paschen, J., Kietzmann, J., & Kietzmann, T. C. (2019). Artificial intelligence (AI) and its implications for market knowledge in B2B marketing. *Journal of Business & Industrial Marketing*, 34(7), 1410–1419. <https://doi.org/10.1108/JBIM-10-2018-0295>

PE-International. (2021). *GaBi Databases*. <https://gabi.sphera.com/international/databases/gabi-databases/>

Pereira, J. L., & Silva, D. (2016). Business process modeling languages: A comparative framework. *New Advances in Information Systems and Technologies*, 619–628.

Pinquié, R., Véron, P., Segonds, F., & Croué, N. (2016). Natural Language Processing of Requirements for Model-Based Product Design with ENOVIA/CATIA V6. In A. Bouras, B. Eynard, S. Foufou,

& K.-D. Thoben (Hrsg.), *Product Lifecycle Management in the Era of Internet of Things* (S. 205–215). Springer International Publishing.

Pinquié, R., Véron, P., Segonds, F., & Croué, N. (2018). A requirement mining framework to support complex sub-systems suppliers. *Procedia CIRP*, 70, 410–415.
<https://doi.org/10.1016/j.procir.2018.03.228>

Poole, D., Mackworth, A., & Goebel, R. (1998). *Computational Intelligence*.

QRA Corp. (2023). *QVscribe*. <https://qracorp.com/welcome-to-the-future-of-requirements-writing-lp/>, <https://qracorp.com/welcome-to-the-future-of-requirements-writing-lp/>

Ramos, A. L., Ferreira, J. V., & Barceló, J. (2011). Model-based systems engineering: An emerging approach for modern systems. *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, 42(1), 101–111.

Rawat, D. S., & Kashyap, N. K. (2017). Graph database: A complete GDBMS survey. *International journal for innovative research in science & technology*, 3(12), 217–226.
<http://www.ijirst.org/articles/IJIRSTV3I12047.pdf>

Rechtin, E. (1991). *Systems architecting: Creating and building complex systems*. Prentice Hall.

Rich, E., & Knight, K. (1991). *Artificial Intelligence*. Mc Graw Hill, Columbus.

Robert Bosch Manufacturing Solutions GmbH. (2023). *Bosch Semantic Stack*. Bosch Connected Industry. <https://www.bosch-connected-industry.com/de/de/portfolio/bosch-semantic-stack/>

Roques, P. (2016). MBSE with the ARCADIA Method and the Capella Tool. *8th European Congress on Embedded Real Time Software and Systems (ERTS 2016)*.

- Rückemann, C.-P., Kovacheva, Z., Schubert, L., Lishchuk, I., Gersbeck-Schierholz, B., & Hülsmann, F. (2016). *Best Practice and Definitions of Data-centric and Big Data: Science, Society, Law, Industry, and Engineering*.
- Russell, S. J., Norvig, P., Davis, E., & Edwards, D. (2016). *Artificial intelligence: A modern approach* (Third edition, Global edition). Pearson.
- Sakao, T., Funk, P., Matschewsky, J., Bengtsson, M., & Ahmed, M. U. (2021). AI-LCE: Adaptive and Intelligent Life Cycle Engineering by applying digitalization and AI methods – An emerging paradigm shift in Life Cycle Engineering. *Procedia CIRP*, 98, 571–576. <https://doi.org/10.1016/j.procir.2021.01.153>
- Samuel, A. L. (1959). Machine learning. *The Technology Review*, 62(1), 42–45.
- Schmidt, M. M., & Stark, R. (2020). *Model-Based Systems Engineering (MBSE) as computer-supported approach for cooperative systems development*. https://doi.org/10.18420/ecscw2020_ep04
- Schweitzer, G., Bitzer, M., & Vielhaber, M. (2020). Produktentwicklung: KI-ready? *Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 115(12), 873–876. <https://doi.org/10.3139/104.112464>
- Schweitzer, G., Bitzer, M., & Vielhaber, M. (2021). Artificial intelligence in engineering: Evolution of virtual product development in the context of medical device industry. *Procedia CIRP*, 100, 349–354. <https://doi.org/10.1016/j.procir.2021.05.081>
- Schweitzer, G., Bitzer, M., & Vielhaber, M. (2023a). Engineering Graph as an approach to support design decisions in product development. *Proceedings of the International Conference on Engineering Design (ICED23)*, 3, 1625–1634. <https://doi.org/10.1017/pds.2023.163>
- Schweitzer, G., Bitzer, M., & Vielhaber, M. (2023b). *Lifecycle Engineering in the Context of a Medical Device Company—Leveraging MBSE, PLM and AI*. 557–566. https://doi.org/10.1007/978-3-031-25182-5_54

- Schweitzer, G., Mörsdorf, S., Bitzer, M., & Vielhaber, M. (2022). Detection of cause-effect relationships in Life Cycle Sustainability Assessment based on an Engineering Graph. *DESIGN 2022*. <https://doi.org/10.1017/pds.2022.115>
- Semantic PDM GmbH & Co. KG. (2023). *Semantic PDM*. semantic-PDM.com. <http://www.semantic-pdm.com/>
- semanticPDM. (2023). *Leistungen und Produkt*. semantic-PDM.com. <http://www.semantic-pdm.com/de/produkt/>
- Shabestari, S. S., Herzog, M., & Bender, B. (2019). A Survey on the Applications of Machine Learning in the Early Phases of Product Development. *Proceedings of the Design Society: International Conference on Engineering Design*, 1(1), 2437–2446. <https://doi.org/10.1017/dsi.2019.250>
- Sheard, S. A. (1996). Twelve Systems Engineering Roles. *INCOSE International Symposium*, 6(1), 478–485. <https://doi.org/10.1002/j.2334-5837.1996.tb02042.x>
- Shuman, E. A. (2010). Understanding executable architectures through an examination of language model elements. *Proceedings of the 2010 Summer Computer Simulation Conference*, 483–497.
- Siemens. (2023). *Industrie-Software* [Mobility-portfolio-access]. Siemens Deutschland. <https://new.siemens.com/de/de/produkte/automatisierung/industrie-software.html>
- Sillitto, H., Martin, J., McKinney, D., Griego, R., Dori, D., Krob, D., Godfrey, P., Arnold, E., & Jackson, S. (2019). *Systems Engineering and System Definitions*. 18.
- Spur, G., & Krause, F.-L. (1997). *Das virtuelle Produkt: Management der CAD-Technik*. Hanser München.
- statista. (2021). *Total data volume worldwide 2010-2025*. Statista. <https://www.statista.com/statistics/871513/worldwide-data-created/>

statista. (2022a). *Global IoT and non-IoT connections 2010-2025*. Statista.

<https://www.statista.com/statistics/1101442/iot-number-of-connected-devices-worldwide/>

statista. (2022b). *Infographic: How Many Websites Are There?* Statista Infographics.

<https://www.statista.com/chart/19058/number-of-websites-online/>

statista. (2022c). *Wikipedia—Anzahl der Artikel weltweit 2022*. Statista.

<https://de.statista.com/statistik/daten/studie/195081/umfrage/anzahl-der-artikel-auf-wikipedia-weltweit/>

Süddeutsche Zeitung. (2011). *Wie gefährlich der Ausbruch wirklich war*. Süddeutsche.de.

<https://www.sueddeutsche.de/wissen/islands-chaos-vulkan-wie-gefaehrlich-eyjafjallajoekull-wirklich-war-1.1088849>

Tableau. (2023). *Tableau: Software für Business Intelligence und Analytics*. Tableau.

<https://www.tableau.com/de-de/node/62770>

Tegmark, M. (2017). *Life 3.0: Being human in the age of artificial intelligence*. Knopf.

Tenders Electronic Daily. (2023). *Search result—TED Tenders Electronic Daily*.

<https://ted.europa.eu/TED/search/searchResult.do>

Terzi, S., Bouras, A., Dutta, D., Garetti, M., & Kiritsis, D. (2010). Product lifecycle management—From its history to its new role. *Int. J. of Product Lifecycle Management*, 4, 360–389.

<https://doi.org/10.1504/IJPLM.2010.036489>

UN General Assembly. (2021). *Transforming our world: The 2030 Agenda for Sustainable Development*.

Refworld. <https://www.refworld.org/docid/57b6e3e44.html>

Valdivia, S., Backes, J. G., Traverso, M., Sonnemann, G., Cucurachi, S., Guinée, J. B., Schaubroeck, T., Finkbeiner, M., Leroy-Parmentier, N., Ugaya, C., Peña, C., Zamagni, A., Inaba, A., Amaral, M., Berger, M., Dvarioniene, J., Vakhitova, T., Benoit-Norris, C., Prox, M., ... Goedkoop, M. (2021).

- Principles for the application of life cycle sustainability assessment. *The International Journal of Life Cycle Assessment*. <https://doi.org/10.1007/s11367-021-01958-2>
- van der Maas, A. A., Ter Hofstede, A. H., & Ten Hoopen, A. J. (2001). Requirements for medical modeling languages. *Journal of the American Medical Informatics Association*, 8(2), 146–162.
- Vicknair, C., Macias, M., Zhao, Z., Nan, X., Chen, Y., & Wilkins, D. (2010). A comparison of a graph database and a relational database: A data provenance perspective. *Proceedings of the 48th Annual Southeast Regional Conference*, 1–6. <https://doi.org/10.1145/1900008.1900067>
- von Drachenfels, N., Cerdas, F., & Herrmann, C. (2020). Towards knowledge based LCE of battery technologies. *Procedia CIRP*, 90, 683–688. <https://doi.org/10.1016/j.procir.2020.01.125>
- Voulodimos, A., Doulamis, N., Doulamis, A., & Protopapadakis, E. (2018). Deep Learning for Computer Vision: A Brief Review. *Computational Intelligence and Neuroscience*, 2018, e7068349. <https://doi.org/10.1155/2018/7068349>
- VPE. (2023). Lehrstuhl für Virtuelle Produktentwicklung (VPE). *We move it* -. <https://wemoveit.rlp.de/vpe-tu-kl/>
- Wahlster, W. (2017). Künstliche Intelligenz als Treiber der zweiten Digitalisierungswelle. *IM+io Fachmagazin*, 2.
- Wahlster, W. (2019, April 4). *KI als Speerspitze der Digitalisierung*. Executive Dialog 2019, Berlin.
- Walden, D. D., Roedler, G. J., & Forsberg, K. (2015). INCOSE systems engineering handbook version 4: Updating the reference for practitioners. *INCOSE International Symposium*, 25(1), 678–686.
- Waller, M. A., & Fawcett, S. E. (2013). Data Science, Predictive Analytics, and Big Data: A Revolution That Will Transform Supply Chain Design and Management. *Journal of Business Logistics*, 34(2), 77–84. <https://doi.org/10.1111/jbl.12010>

- Wang, Y., & Zhang, J. (2017). Bridging the semantic gap in customer needs elicitation: A machine learning perspective. *DS 87-4 Proceedings of the 21st International Conference on Engineering Design (ICED 17) Vol 4: Design Methods and Tools, Vancouver, Canada, 21-25.08. 2017*, 643–652.
- Weilkiens, T. (2016). *SYSMOD - The Systems Modeling Toolbox—Pragmatic MBSE with SysML*. Lulu.com.
- Whittlestone, J., Nystrup, R., Alexandrova, A., & Cave, S. (2019). The Role and Limits of Principles in AI Ethics: Towards a Focus on Tensions. *Proceedings of the 2019 AAAI/ACM Conference on AI, Ethics, and Society*, 195–200. <https://doi.org/10.1145/3306618.3314289>
- Wikimedia. (2021). *Wikimedia Foundation*. Wikimedia Foundation. <https://wikimediafoundation.org/>
- Winston, P. H. (1992). *Artificial intelligence*. Addison-Wesley Longman Publishing Co., Inc.
- Wolf, D., & Heckman, S. (2019, Juni 10). *Johnson & Johnson Leverages ThingWorx, Product Lifecycle Intelligence (PLI) and Machine Learning in Support of New Regulations*. Liveworx 2019, Boston.
- Woodcock, J., Cavalcanti, A., Fitzgerald, J., Larsen, P., Miyazawa, A., & Perry, S. (2012). Features of CML: A formal modelling language for Systems of Systems. *2012 7th International Conference on System of Systems Engineering (SoSE)*, 1–6. <https://doi.org/10.1109/SYSoSE.2012.6384144>
- World Data Science Initiative. (2022). *Why Data Science is the most in-demand skill now and how can you prepare for it?* <https://www.worlddatascience.org/blogs/why-data-science-is-the-most-indemand-skill-now-and-how-can-you-prepare-for-it>
- World Health Organization. (2021). *Health System Building Blocks*. <https://extranet.who.int/nhptool/Default.aspx>

Xu, B., Zhao, T., Zheng, D., & Wang, S. (2010). Product features mining based on Conditional Random Fields model. *2010 International Conference on Machine Learning and Cybernetics*, 6, 3353–3357. <https://doi.org/10.1109/ICMLC.2010.5580679>

ZEIT ONLINE. (2023). *Ein Jahr Krieg in der Ukraine*. ZEIT ONLINE. <https://www.zeit.de/thema/krieg-in-ukraine>

Zhang, C., Kwon, Y. P., Kramer, J., Kim, E., & Agogino, A. M. (2017). Concept clustering in design teams: A comparison of human and machine clustering. *Journal of Mechanical Design*, 139(11).