

Andreas Eiden

**Knowledge Graph basiertes
Assistenzsystem zur Unterstützung
der Datenanalyse für das
Engineering Smarter Produkte**

Schriftenreihe

Band 02/2025

Herausgeber:

Prof. Dr.-Ing. Jens C. Göbel

Lehrstuhl für Virtuelle Produktentwicklung (VPE)

Bibliografische Information Der Deutschen Bibliothek

Die Deutsche Bibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.ddb.de> abrufbar.

Bibliographic information published by Die Deutsche Bibliothek

Die Deutsche Bibliothek lists this publication in the Deutsche Nationalbibliografie; detailed bibliographic data is available in the Internet at <http://dnb.ddb.de>.

Berichte aus der Virtuellen Produktentwicklung (VPE)

Wissenschaftliche Schriftenreihe des Lehrstuhls für Virtuelle Produktentwicklung der Rheinland-Pfälzischen Technischen Universität Kaiserslautern-Landau

Herausgeber: Prof. Dr.-Ing. Jens Christian Göbel
Lehrstuhl für Virtuelle Produktentwicklung
Postfach 3049
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
67653 Kaiserslautern

Verlag: Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau

Druck: Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
Dezernat 5 Technik
Abteilung 5.6 Foto-Repro-Druck

D-386

© Lehrstuhl für Virtuelle Produktentwicklung, 2025
Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau
Erwin-Schrödinger-Straße
67663 Kaiserslautern

Alle Rechte vorbehalten, auch das des auszugsweisen Nachdrucks, der auszugsweisen oder vollständigen Wiedergabe (Photographie, Mikroskopie), der Speicherung in Datenverarbeitungsanlagen und das der Übersetzung.

Als Manuskript gedruckt. Printed in Germany.

ISSN 3051 - 9284

Knowledge Graph basiertes Assistenzsystem zur Unterstützung der Datenanalyse für das Engineering Smarter Produkte

Vom Fachbereich Maschinenbau und Verfahrenstechnik
der Rheinland-Pfälzischen Technischen Universität
zur Erlangung des akademischen Grades

Doktor-Ingenieur (Dr.-Ing.)

genehmigte

Dissertation

von

Dipl.-Ing. Andreas Eiden

aus Hermeskeil

Vorsitzender der Prüfungskommission:

1. Berichterstatter:

2. Berichterstatter:

Dekan:

Tag der mündlichen Prüfung:

Prof. Dr.-Ing. Erik von Harbou

Prof. Dr.-Ing. Jens C. Göbel

Prof. Dr.-Ing. Martin Ruskowski

Prof. Dr. rer. nat. Roland Ulber

06. September 2024

Kaiserslautern, 2024

D386

Kurzfassung

Durch die zunehmende Interdisziplinarität der Produkte und Trends wie digitale Zwillinge werden im Engineering-Kontext mehr Daten aus immer mehr Datenquellen genutzt. Gleichwohl fehlt den Nutzern häufig das Wissen welche Daten für ihre Aufgaben relevant sein könnten und wo diese abgespeichert sind.

Aufbauend auf dieser Feststellung ergibt sich die Frage, wie man Abhilfe schaffen könnte durch die Nutzung von Technologien und einem neuen Konzept, welches die Anforderungen erfüllt, welche sich aus der initialen Recherche, Interviews mit Firmen und aus Forschungssicht ergeben. Eine potenzielle Lösungsmöglichkeit sind Metadaten-Repositories – Verzeichnisse, welche einen Überblick über vorhandene Daten bereitstellen und deren Vernetzung ermöglichen. Es zeigt sich weiterhin, dass bereits verschiedene Forschungsansätze verfolgt werden und es Lösungen gibt, welche von Firmen vermarktet werden, die allerdings häufig einen anderen Fokus haben und die hier beschriebenen Anforderungen nicht erfüllen (können).

Deswegen wird in dieser Arbeit ein Konzept vorgeschlagen, welches auf Basis eines Metadaten-Repository neue Prozesse, Methoden und ein IT-Werkzeug zur besseren formalen Beschreibung der Datenbestände in Unternehmen beinhaltet. Methodisch wird ein Knowledge Graph erarbeitet, welcher auf einer abstrakten Ebene die in IT-Systemen gespeicherten, sowie im Metadaten-Repository repräsentiert Daten modelliert. Dies erweitert Metadaten-Repositories, welche nur über eine Modellierung der Daten ohne Abstrahierung verfügen.

Anschließend wird ein Algorithmus genutzt, um Daten aus dem Knowledge-Graph und dem Metadaten-Repository auszuwerten, Ähnlichkeiten zu finden, diese einem Dateningenieur vorzuschlagen und Verbindungen zu ziehen. So entsteht ein semantisch angereichertes Datenmodell, welches eine abstrakte Sicht auf Engineering-Daten (den Knowledge Graph), als auch die IT-Sicht (das Metadaten-Repository) enthält und von Nutzern auf der Suche nach Daten ausgewertet werden kann.

Der Lösungsbaustein des Algorithmus zur Ähnlichkeitssuche kombiniert verschiedenen Strategien und auch verschiedene Suchräume, deren Inhalte verglichen werden, um möglichst viele Übereinstimmungen zwischen den abstrahierten Datenobjekten und den vorhandenen Datentypen aus realen IT-Quellsystemen festzustellen. Auch kann basierend auf den Ergebnissen noch die Sensibilität des Algorithmus durch den Nutzer eingestellt werden, falls zu viele oder zu wenige Übereinstimmungen vorgeschlagen werden.

Zur Realisierung dieses Ansatzes wurde auf Technologien zurückgegriffen, welche im Bereich des Semantic Web eingesetzt werden. Z.B. wurde eine server- und servicebasierte Architektur gewählt, Kommunikation über REST-Schnittstellen gewährleistet und mit einem Demonstrator auch prototypisch umgesetzt. Die Ergebnisse dieser Arbeit wurden anschließend mithilfe des Demonstrators und den erhobenen Anforderungen erfolgreich verifiziert und das Konzept in verschiedenen Interviews mit Forschenden erfolgreich validiert.

Abstract

Due to the increasing interdisciplinarity of products and trends such as the digital twin, more data from more data sources is being used in the engineering context. Nevertheless, users often lack the knowledge of which data could be relevant for their tasks and where it is stored.

Based on this observation, the question arises as to how this could be remedied using technologies and a new concept that fulfills the requirements resulting from initial research, interviews with companies and from a research perspective. One potential solution is metadata repositories - directories that provide an overview of existing data and enable them to be linked. It is also apparent that various research approaches are already being pursued and that there are solutions that are marketed by companies, although these often have a different focus and do not (or cannot) fulfill the requirements described here.

For this reason, this paper proposes a concept based on a metadata repository that includes new processes, methods and an IT tool for better formal description of data stocks in companies. Methodologically, a knowledge graph is developed for this purpose, which models the data stored in the IT systems and represented in the metadata repository on an abstract level. This extends previous metadata repositories, which only have a model of the existing data without any abstraction.

An algorithm is then used to evaluate data from the knowledge graph and the metadata repository, find similarities, suggest these to a data engineer and draw connections. This creates a semantically enriched data model that contains an abstract view of engineering data (the knowledge graph) as well as the IT view (the metadata repository) and can be evaluated by users searching for data.

The solution module of the similarity search algorithm combines different strategies and different search spaces whose contents are compared in order to determine as many matches as possible between the abstracted data objects and the existing data types from real IT source systems. The sensitivity of the algorithm can also be adjusted by the user based on the results if too many or too few matches are suggested.

Various semantic technologies, which are also used in the Semantic Web, were used to implement this approach. Among other things, a server- and service-based architecture was chosen, communication via REST interfaces was ensured and prototypically implemented with an IT demonstrator. The results of this work were then successfully verified using the demonstrator and the requirements collected, and the concept was also successfully validated in various interviews with researchers.

Inhaltsverzeichnis

| | |
|--|-----|
| Kurzfassung | I |
| Abstract | II |
| Abbildungsverzeichnis | VII |
| Tabellenverzeichnis | X |
| Abkürzungsverzeichnis | XI |
| 1. Motivation und Einleitung | 1 |
| 1.1. Aufbau der Arbeit | 5 |
| 1.2. Methodisches Vorgehen | 6 |
| 2. Grundlagen | 7 |
| 2.1. Betrachtete Engineering-Domänen | 7 |
| 2.2. Verständnis industrieller Produkte | 8 |
| 2.2.1. Smartes Produkt-Service System | 9 |
| 2.2.2. Produktstrukturen | 11 |
| 2.3. Prozesse | 12 |
| 2.3.1. Änderungsmanagement | 13 |
| 2.3.2. Freigabemanagement | 15 |
| 2.3.3. Prozessmodellierung | 15 |
| 2.4. Produktlebenszyklus | 23 |
| 2.5. Digitale Objekte entlang des Lebenszyklus | 25 |
| 2.5.1. Produktdaten | 26 |
| 2.5.2. Digitaler Zwilling | 26 |
| 2.5.3. Digitale Verwaltungsschale | 28 |
| 2.6. Digitale Datenverwaltung im Engineering | 29 |
| 2.6.1. PDM/PLM | 30 |
| 2.6.2. Application Lifecycle Management (ALM) | 31 |
| 2.6.3. ERP | 32 |
| 2.6.4. System Lifecycle Management (SysLM) | 33 |
| 2.6.5. BaSys 4.0 | 35 |
| 2.6.6. GAIA-X | 36 |
| 2.7. Wissen | 37 |

| | | |
|---------|---|----|
| 2.7.1. | Explizites/Implizites Wissen..... | 39 |
| 2.7.2. | Schaffung und Umwandlung von Wissen..... | 39 |
| 2.8. | Zusammenfassung..... | 40 |
| 3. | Stand der Forschung und Technik..... | 41 |
| 3.1. | Beschreibung von Daten..... | 41 |
| 3.1.1. | Semantik..... | 41 |
| 3.1.2. | Ontologien..... | 42 |
| 3.1.3. | Dublin Core..... | 45 |
| 3.1.4. | Standards im Bereich Datenbeschreibung..... | 46 |
| 3.1.5. | Mapping..... | 47 |
| 3.1.6. | Graphen..... | 50 |
| 3.1.7. | Knowledge Graphen..... | 51 |
| 3.2. | Nutzung semantisch angereicherter Daten..... | 53 |
| 3.2.1. | Analyse-Frameworks und einfache statistische Methoden..... | 54 |
| 3.2.2. | AKKORD..... | 54 |
| 3.2.3. | Künstliche Intelligenz im Kontext von PLM und Produktentwicklung..... | 55 |
| 3.2.4. | Engineering-Integrationsplattformen..... | 56 |
| 3.2.5. | Linked Data..... | 59 |
| 3.2.6. | Open Services for Lifecycle Collaboration (OSLC)..... | 60 |
| 3.2.7. | Datenintegration mittels IT-System-Schnittstellen..... | 61 |
| 3.2.8. | Erweiterung des PLM für industrielle Produkt-Service Systeme..... | 63 |
| 3.2.9. | Datenzentrierte Architekturen im PLM-Kontext..... | 65 |
| 3.2.10. | LeWiPro..... | 67 |
| 3.2.11. | ENTIME..... | 68 |
| 3.2.12. | International Data Spaces..... | 69 |
| 3.2.13. | SP ² IDER..... | 70 |
| 3.2.14. | Semantic Data Management im Kontext Smarter Produkte..... | 74 |
| 4. | Anforderungen an ein Knowledge Graph basiertes Assistenzsystem..... | 77 |
| 4.1. | Ermittlung der Anforderungen..... | 77 |
| 4.2. | Grundlegende Anforderungen..... | 80 |
| 4.3. | Bewertung Stand der Technik hinsichtlich der grundlegenden Anforderungen..... | 81 |

| | | |
|--------|--|-----|
| 4.4. | Informationstechnische Anforderungen..... | 84 |
| 4.5. | Anforderungen an die Methodik..... | 85 |
| 5. | Konzept für ein Knowledge Graph basiertes Assistenzsystem..... | 87 |
| 5.1. | Knowledge Graph basierte Modellierungsumgebung | 89 |
| 5.1.1. | Beziehungen zwischen den Objekttypen..... | 93 |
| 5.1.2. | Verbindung des Knowledge Graph mit einem Metadaten-Repository..... | 96 |
| 5.2. | Knowledge Graph basierte Lösungsarchitektur..... | 102 |
| 5.2.1. | Neue Ebene im SP ² IDER Prozess- und Datenmodell..... | 102 |
| 5.2.2. | Erweiterung des SP ² IDER Core..... | 104 |
| 5.2.3. | Erweiterung der SP ² IDER-Konnektoren..... | 107 |
| 5.3. | Prozessmodellierung zur Nutzung des Systems..... | 108 |
| 5.3.1. | Geschäftsprozesse analysieren..... | 109 |
| 5.3.2. | Knowledge Graph modellieren..... | 111 |
| 5.3.3. | Knowledge Graph und SP ² IDER-Graph verknüpfen..... | 113 |
| 5.3.4. | Relevante Daten im System identifizieren und verbinden..... | 114 |
| 5.3.5. | User anlegen..... | 115 |
| 5.3.6. | Sichten anlegen..... | 116 |
| 5.3.7. | Automatisierte Erzeugung des Knowledge Graphen..... | 117 |
| 5.4. | Methodik für die Datenverarbeitung im Kontext des Systems..... | 119 |
| 5.4.1. | Mappingvorgänge im SP ² IDER-Graph | 119 |
| 5.4.2. | Assistiertes Finden von entsprechenden Datentypen..... | 122 |
| 5.5. | Rollenkonzept für die industrielle Einführung..... | 126 |
| 5.5.1. | Dateningenieur/experte..... | 126 |
| 5.5.2. | Administrator..... | 126 |
| 5.5.3. | Domäneningenieur | 127 |
| 5.5.4. | allgemeiner Nutzer..... | 127 |
| 5.6. | Sichten auf den Knowledge- und SP ² IDER-Graph als Berechtigungsinstrument..... | 127 |
| 6. | Verifikation und Validierung..... | 131 |
| 6.1. | Anwendungsszenario..... | 131 |
| 6.1.1. | Modellierung des Knowledge Graphen | 132 |
| 6.1.2. | Finden von Datenelementen..... | 138 |

| | | |
|----------|---|-----|
| 6.2. | Verifikation..... | 141 |
| 6.2.1. | Verifikation der grundlegenden und methodischen Anforderungen | 141 |
| 6.2.2. | Verifikation der IT-technischen Anforderungen..... | 143 |
| 6.2.2.1. | Szenario AN-IT03 | 143 |
| 6.2.3. | Ergebnis der Verifikation | 151 |
| 6.3. | Validierung..... | 152 |
| 6.3.1. | Aufbau der Validierungsinterviews | 154 |
| 6.3.2. | Ergebnisse der geführten Interviews..... | 155 |
| 6.3.3. | Quantitative Validierung durch Tests..... | 158 |
| 6.3.4. | Ergebnis der Validierung | 158 |
| 7. | Zusammenfassung und Ausblick..... | 160 |
| 8. | Literaturverzeichnis..... | 163 |
| | Betreute studentische Arbeiten | 186 |
| | Lebenslauf..... | 187 |

Abbildungsverzeichnis

| | |
|--|----|
| Abbildung 1-1: Beispiel-Prozesslandkarte nach [ScSe2020] | 3 |
| Abbildung 1-2: Aufbau der Arbeit | 5 |
| Abbildung 2-1: Grundstruktur eines mechatronischen Systems aus [Rodd2019] | 7 |
| Abbildung 2-2: Ausprägung industrieller Produkt-Service-Systeme nach [MeUK2005] | 10 |
| Abbildung 2-3: Evolution von traditionellen zu smarten Produkten nach [Abra2014] | 11 |
| Abbildung 2-4: Unterschiedliche Produktstrukturen entlang des PLZ aus [EiAS2016] | 12 |
| Abbildung 2-5: Änderungskosten im Laufe des Produktlebenszyklus aus [EiSt2009] | 14 |
| Abbildung 2-6: Auswirkungen von Änderungen nach [Stek2016] | 14 |
| Abbildung 2-7: Zusammenhang Freigabe- und Änderungsmanagement aus [Zarn2007] | 15 |
| Abbildung 2-8: Nutzen von Prozessmodellen aus [BePV2012] | 16 |
| Abbildung 2-9: Prozesslandkarte nach [ScSe2020] | 17 |
| Abbildung 2-10: Beispielhafte Prozessmodellierung mit der eEPK aus [BePV2012] | 19 |
| Abbildung 2-11: Beispiel für einen Prozess in SIPOC Notation nach [MaRe2009] | 20 |
| Abbildung 2-12: BPMN-Elemente nach [ChTr2012] | 21 |
| Abbildung 2-13: Prozessmodellierung mit der OMEGA-Notation nach [Fahr1995] | 22 |
| Abbildung 2-14: Lebenszyklus hybrider Leistungsbündel aus [MeUh2012] | 23 |
| Abbildung 2-15: Phasen des Produktlebenszyklus aus [EiSt2009] | 24 |
| Abbildung 2-16: Der Lebenszyklus smarter Produkte aus [GöEi2020] | 25 |
| Abbildung 2-17: Das Kaiserslautern Digital Twin Modeling Framework aus [GöEi2020] | 27 |
| Abbildung 2-18: Ausprägungsformen Digitaler Zwillinge aus [StAT2020] | 28 |
| Abbildung 2-19: Grobstruktur der Verwaltungsschale aus [AdAB2016] | 29 |
| Abbildung 2-20: Verschiedene PDM-Architekturen nach [JoAH2017] | 30 |
| Abbildung 2-21: Der ALM-Prozess nach [Ross2014] | 31 |
| Abbildung 2-22: Integrationsperspektiven eines IT-Systems nach [Kurb2013] | 32 |
| Abbildung 2-23: Digitized Engineering Framework nach [BiEF2018] | 34 |
| Abbildung 2-24: SysLM-Repository in maximaler Ausbaustufe aus [Sind2022] | 35 |
| Abbildung 2-25: Automatisierungspyramide und BaSyx-Architektur nach [KaHD2021] | 36 |
| Abbildung 2-26: Eine Übersicht über die GAIA-X Architektur nach [Gaia2021] | 37 |
| Abbildung 2-27: Die Wissenstreppe von North, nach [Nort2011] | 38 |
| Abbildung 2-28: Implizites und Explizites Wissen aus [Turk2014] | 39 |
| Abbildung 2-29: Organisationale Wissensbasis aus [Völk2006] | 40 |
| Abbildung 3-1: Semantische Treppe aus [BlPe2006] | 42 |
| Abbildung 3-2: Klassifizierungsansatz für Ontologien aus [JaKR2017] | 43 |
| Abbildung 3-3: Digital Twin Ontologie nach [BaEF2020] | 45 |
| Abbildung 3-4: ECLASS Spezifikation eines Elektromotors aus eclass.eu | 47 |
| Abbildung 3-5: Enterprise Architecture Integration Infrastruktur nach [WaXu2008] | 48 |
| Abbildung 3-6: Visualisierung eines einfachen, ungerichteten Graphen aus [Deo1974] | 51 |
| Abbildung 3-7: Hemmnisse und Erfolgsfaktoren bei der Datenanalyse aus [NoSM2020] | 55 |
| Abbildung 3-8: Auswertung von Ähnlichkeiten in Spezifikationen nach [BrLL2021] | 56 |

| | |
|---|-----|
| Abbildung 3-9: IT-Architektur des CMBC-Systems nach [Erns2016] | 58 |
| Abbildung 3-10: Zentraler und dezentraler Linked Data Ansatz nach [Reic2022] | 60 |
| Abbildung 3-11: Überblick über den OSLC-Core 3.0 nach ²³ | 61 |
| Abbildung 3-12: Direkte und indirekte Schnittstellen nach [EiRZ2014] | 62 |
| Abbildung 3-13: Proprietäre und neutrale Schnittstellen nach [EiRZ2014] | 62 |
| Abbildung 3-14: Traditionelle und Link-Enabling APIs nach [BIPM2022]..... | 63 |
| Abbildung 3-15: Dimensionen des IPSS-LM Konzeptes aus [AbMN2008]..... | 64 |
| Abbildung 3-16: Systematik zum Aufbau der IPSS Top Level Ontologie nach [Dang2017]..... | 65 |
| Abbildung 3-17: Datenzentrierte Architektur aus einer Fallstudie nach [BIPM2022] | 66 |
| Abbildung 3-18: Methode zur Akquise von Lebenszyklusdaten aus [BeGK2015]..... | 67 |
| Abbildung 3-19: Semantische Suche nach Lösungselementen aus [TeJD2013] | 68 |
| Abbildung 3-20: Rollen und Informationsflüsse im IDS-Ökosystem nach [Ota2019] | 70 |
| Abbildung 3-21: Die SP ² IDER-Architektur nach [EiEG2020] | 71 |
| Abbildung 3-22: Ausschnitt des SP ² IDER Prozessmodells nach [EiEG2020] | 72 |
| Abbildung 3-23: SP ² IDER als Datenbacken-System in Analyse-Prozesse aus [EiEG2020]..... | 73 |
| Abbildung 3-24: Datenanalysepipeline basierend auf SP ² IDER nach [EiEG2022]..... | 74 |
| Abbildung 3-25: Semantic Data Management (SDM)-Ansatz nach [AbGD2016]..... | 75 |
| Abbildung 3-26: Komponenten für die Informationsbereitstellung nach [AbGG2016]..... | 76 |
| Abbildung 4-1: Strategien zur Anforderungserhebung..... | 78 |
| Abbildung 4-2: Bewertung von Lösungselementen in Bezug auf die Anforderungen..... | 83 |
| Abbildung 5-1: Grundbausteine des Lösungskonzepts und deren Zusammenhang..... | 88 |
| Abbildung 5-2: OMEGA Notationsobjekte | 90 |
| Abbildung 5-3: Beispielprozess im Engineering aus dem IT-System „Aras Innovator“ | 91 |
| Abbildung 5-4: Enterprise Knowledge Graph-Beispiel des Prozesses „Produkt entwickeln“ | 95 |
| Abbildung 5-5: Ausschnitt aus dem bestehenden SP ² IDER Datenmodell nach [EiEG2022]..... | 96 |
| Abbildung 5-6: Datenmodell der SP ² IDER SourceSystem Collection..... | 98 |
| Abbildung 5-7: Beispiel type_GraphNode im SP ² IDER Metadastore..... | 99 |
| Abbildung 5-8: Kanten, die einen type_GraphNode mit Attributen verknüpfen..... | 100 |
| Abbildung 5-9: SP ² IDER Datenmodell mit verschiedenen Ebenen nach [EiEG2020] | 103 |
| Abbildung 5-10: Erweiterte SP ² IDR-Architektur zur Modellierung des Knowledge Graph..... | 104 |
| Abbildung 5-11: API-Endpunkte des erweiterten SP ² IDER-Core | 105 |
| Abbildung 5-12: Beispielabfrage der DataObjects im Knowledge-Graphen..... | 106 |
| Abbildung 5-13: Prozessmodell der Erweiterung des Metadaten-Repository..... | 109 |
| Abbildung 5-14: Prozessmodelle analysieren | 110 |
| Abbildung 5-15: Knowledge Graph modellieren..... | 112 |
| Abbildung 5-16: Knowledge- und SP ² IDER-Graph verknüpfen..... | 113 |
| Abbildung 5-17: Relevante Daten im System identifizieren und verbinden..... | 115 |
| Abbildung 5-18: User im System anlegen | 116 |
| Abbildung 5-19: Neue Sicht anlegen | 117 |
| Abbildung 5-20: BPMN-Prozessdefinition erstellt mit http://bpmn.io | 118 |

| | |
|--|-----|
| Abbildung 5-21: Mapping-Mechanismen als Teil des SP ² IDER-Konzepts nach [EiEG2020] | 120 |
| Abbildung 5-22: Beschreibung des SP ² IDER Mapping Ansatzes nach [EiEG2021] | 122 |
| Abbildung 5-23: Erster Prototyp des Data Model Canvas aus [EiEG2021] | 122 |
| Abbildung 5-24: Code zur Überprüfung der Ähnlichkeit von Strings | 124 |
| Abbildung 5-25: Illustration der Komplexität in einem Metadaten-Repository | 128 |
| Abbildung 5-26: Nicht ausgewertete Verbindungen im Rollen und Sichtenkonzept | 130 |
| Abbildung 6-1: Template des DMC zum Beginn der Modellierung | 133 |
| Abbildung 6-2: Detailansicht DMC mit modellierten Prozessen | 133 |
| Abbildung 6-3: Detailansicht DMC mit Prozessen und modellierten Abteilungen | 134 |
| Abbildung 6-4: Detailansicht DMC mit Prozessen, Abteilungen und Datenobjekten | 135 |
| Abbildung 6-5: Knowledge Graphen mit allen modellierten Elementen | 136 |
| Abbildung 6-6: Editierfunktion für Kanten im Data Model Canvas | 136 |
| Abbildung 6-7: Benutzereingabe bei mehreren Kanten-Optionen | 137 |
| Abbildung 6-8: Knowledge Graphen mit allen angelegten Kanten | 138 |
| Abbildung 6-9: Importierbare Quellsysteme im rechten Menü des DMC | 139 |
| Abbildung 6-10: Aras als verknüpftes PLM & Change Management System | 139 |
| Abbildung 6-11: Liste aller möglichen Übereinstimmungen | 140 |
| Abbildung 6-12: Manuelle Bestätigung der ermittelten Vorschläge | 140 |
| Abbildung 6-13: Anlegen der Verbindung zwischen den Objekten | 140 |
| Abbildung 6-14: Ist-Situation nach dem Upload eines Datentyps aus einem Quellsystem | 144 |
| Abbildung 6-15: POST-Aufruf zum Erzeugen eines DataObjects | 145 |
| Abbildung 6-16: GET-Aufruf zum Abrufen aller DataObjects | 146 |
| Abbildung 6-17: Lizenzen der Python-Pakete im Projekt | 147 |
| Abbildung 6-18: Lizenzen der eingebundenen Python-Pakete im Projekt „spider_django“ | 148 |
| Abbildung 6-19: Knowledge Graph im Data Model Canvas als Service | 149 |
| Abbildung 6-20: Paare aus DataObject und Type_Graphnode als JSON-Dictionary | 150 |
| Abbildung 6-21: Abfrage zur Bestätigung des Datenmappings | 150 |
| Abbildung 6-22: Knowledge Graph mit Verbindung zu Datentypen aus einem IT-System | 151 |

Tabellenverzeichnis

| | |
|--|-----|
| Tabelle 3-1: Die 15 Core-Elemente des Dublin-Core nach [Alas2009] | 46 |
| Tabelle 3-2: Vergleich von Knowledge Mapping Techniken aus [JaAB2012]..... | 50 |
| Tabelle 3-3: Menge an Instanzen in verschiedenen Knowledge Graphen aus [Paul2016]..... | 52 |
| Tabelle 4-1: Grundlegende Anforderungen | 80 |
| Tabelle 4-2: IT-Anforderungen | 84 |
| Tabelle 4-3: Anforderungen an die Methodik..... | 85 |
| Tabelle 5-1: Attribute der Objekt-Typen..... | 93 |
| Tabelle 5-2: Verbindungen zwischen Objekt-Typen beider-Graphen | 102 |
| Tabelle 5-3: Vergleichsobjekte zum Finden von Datentypen | 123 |
| Tabelle 6-1: Grundlegende und methodische Anforderungen | 141 |
| Tabelle 6-2: IT-Anforderungen | 143 |
| Tabelle 6-3: Grad der Erfüllung der Anforderungen | 152 |
| Tabelle 6-4: Konzeptuelle Anforderungen..... | 153 |

Abkürzungsverzeichnis

| | |
|----------------------|---|
| ALM | Application Lifecycle Management |
| EAI | Enterprise Architecture Integration |
| ERP | Enterprise Resource Planning |
| CMBC | Change-Management Backbone & Control |
| DIN | Deutsche Industrie Norm |
| DT/DZ | Digital Twin/Digitaler Zwilling |
| IoT | Internet of Things |
| IPSS | Industrielles Produkt-Service System |
| PDM | Product Data Management/Produktdatenmanagement |
| PLM | Product Lifecycle Management |
| PLZ | Produktlebenszyklus |
| RDF | Resource Description Framework |
| REST | Representational State Transfer |
| SoS | System of Systems |
| SPARQL | SPARQL Protocol and RDF Query Language |
| SP ² IDER | Semantic Product & Process Information Digitized Engineering Repository |
| SPSS | Smart Product Service System |
| VDA | Verband der deutschen Automobilindustrie |
| VDI | Verein Deutscher Ingenieure |
| WiGeP | Wissenschaftliche Gesellschaft für Produktentwicklung |

1. Motivation und Einleitung

Das heutige Engineering basiert grundlegend auf der Verfügbarkeit von Daten. Wie in jedem anderen Prozess auch, werden im Engineering Inputdaten mithilfe einer Methodik, wie z.B. Pahl/Beitz [FeGr2013], der VDI 2221 [VDI2019] oder der VDI 2206 [VDI/2020] in Ergebnisse, in diesem Fall in ein Konstruktionsmodell umgewandelt. In einfacheren Fällen liegen Anforderungslisten, sowie bspw. Rechercheergebnisse oder technische Berechnungen als Daten für eine Produktentwicklung zugrunde, in komplexeren Produkten und Systemen, insbesondere bei Rekonfigurations- und Weiterentwicklungsaufträgen können die erforderlichen Datenmengen schnell anwachsen und sehr komplex werden. Treiber dieser Entwicklung sind u.a. Innovation allgemein, Kundenwünsche, aber auch regulatorische Rahmenbedingungen, wie z.B. das Lieferkettengesetz¹ oder der EU-Produktpass².

Im Gegensatz zu den immer größeren und heterogener werdenden Datenmengen, haben die zur Verfügung stehenden Strukturen zur Datenspeicherung und -organisation zwar auch eine Weiterentwicklung erfahren, allerdings sind Daten immer noch stark in IT-Silos gefangen, eine interdisziplinäre Verknüpfung existiert nicht in ausreichendem Maße. Gleiches gilt häufig auch für die beteiligten Mitarbeiter, die in ihren gedanklichen Silos arbeiten und Herausforderungen mit den ihnen bekannten Daten und Methoden lösen möchten. Daten werden selten ausgetauscht, oder nicht genutzt, da die Relevanz nicht klar ist und der Bedarf nicht artikuliert wurde.

Der Wandel zu smarten, intelligenten Produkten und Services verschärft diese Problematik nochmals. Zum einen bestehen diese Produkte aus mechanischen, elektronischen und Software-Komponenten, welche unterschiedliche Strukturen aufweisen, in unterschiedlichen Systemen verwaltet werden und trotzdem am Ende in ein Produkt integriert werden müssen. Zum anderen gibt es Interaktionen zwischen Hersteller und Kunden in der Nutzungsphase, sei es durch Updates, den Rückfluss von Daten zur Produktverbesserung oder zur Erbringung von Dienstleistungsaspekten im Produkt (as a Service).

Abramovici und Herzog fordern deswegen neue Prozesse, Methoden und IT-Werkzeuge im Engineering [AbHe2016]. Smarte Produkte erfordern zu Ihrer Realisierung einen „Technology Stack“ [PoHe2015], welcher in der Lage sein muss Daten aus verschiedensten Quellen so zu kombinieren, dass anschließend ein Mehrwert aus ihnen geschaffen werden kann [PoHe2015]. Auch neuartige Arbeitsweisen wie das Crowd Engineering können Teil dieser neuer Prozesse und Methoden sein und sorgen dafür, dass verteilte Teams an unterschiedlichen Standorten und in unterschiedlichen Firmen Datenbedarfe haben, aber evtl. nicht auf die gleichen Daten zugreifen können [EiAE2017].

¹ http://www.bgbl.de/xaver/bgbl/start.xav?startbk=Bundesanzeiger_BGBl&jumpTo=bgbl121s2959.pdf

² https://germany.representation.ec.europa.eu/news/nachhaltige-produkte-sollen-zur-neuen-norm-der-eu-werden-2023-12-05_de

Konkrete Use-Cases für Datenbedarfe aus unterschiedlichsten Datenquellen sind z.B. in der Produktentwicklung die Berücksichtigung von vorhandenen Fertigungshilfsmitteln oder weiteren Randbedingungen in der Fertigung, die Auswertung von Sensorwerten aus vorherigen Produktgenerationen, die Nutzung von User-Feedback anderer Produkte, Marketing-Informationen, z.B. aus Social Media, die Nutzung von Qualitätsdaten aus der Fertigung oder Daten aus dem Industrial Engineering, welche die Ergonomie oder Arbeitszeiten von Bauteilen beinhalten. All diese Daten können in der Produktentwicklung dazu führen, dass ein neues Produkt schneller und günstiger entwickelt werden kann, da es voraussichtlich weniger Änderungen benötigt, in der Fertigung weniger Probleme verursachen wird und anschließend einen höheren Kundennutzen und damit eine bessere Monetarisierung erzeugen kann.

All diese denkbaren Szenarien scheitern in der industriellen Praxis aber häufig an der oben beschriebenen Situation, dass Daten und Ingenieure in gedanklichen und IT-Silos verankert sind, bzw. in unterschiedlichen Systemen verteilt sind, deswegen kein ausreichendes Wissen über die Verfügbarkeit dieser Daten vorhanden ist und eine Nutzung nicht stattfindet.

Die Frage, die dieser Arbeit zugrunde liegt, ist dementsprechend, wie man Ingenieure in ihrer täglichen Arbeit unterstützen kann, um ihnen einen besseren Überblick über vorhandene Daten im Unternehmen zu geben, mit dem Ziel so Produkte und Prozesse zu optimieren.

Aus dieser Motivation heraus sind in den letzten Jahren von verschiedenen Forschungsgruppen, aber auch in der Industrie Ansätze entstanden, um Daten aus verschiedenen Datenverwaltungssystemen zu integrieren, sogenannte Metadaten-Repositories. Teilweise werden dabei Graphen verwendet, welche in der Lage sind, eine Beziehung zwischen Daten auch näher zu beschreiben. Aufgrund der großen Datenmengen, der Vielzahl von unterschiedlichen Objekten und Unterschieden in der Benennung, welche auf den ersten Blick nicht klar sind, ist neben den vernetzten Daten auch noch immenses Wissen bezüglich der Bedeutung der Daten notwendig. Die vorhandenen Metadaten-Repositories bieten so zwar eine Möglichkeit auf verschiedenste Daten zuzugreifen, haben aber selten einen methodischen Unterbau, um die gesuchten Daten zu identifizieren und mit vorhandenem Wissen über Datenbestände zu verknüpfen.

Bereits seit ca. 30 Jahren wird davon gesprochen, dass die Gesellschaft sich von einer industriebasierten hin zu einer wissensbasierten Gesellschaft wandelt [Slau1993]. North [Nort2011] stellt fest, dass wir uns gerade am Beginn einer neuen Welle befinden, welche den Beginn von Wissen als knappem Rohstoff zur Erzeugung von Wohlstand markiert. Pätzold [Pätz2022] erinnert daran, dass beim Aufbau smarter Produkte auch das Schaffen von Informationen und das damit verbundene und benötigte Wissen immer wichtiger werden, aber auch das sich bewusst machen, was durch das Anwenden des Wissens verbessert wurde, um diesen Prozess gezielter zu steuern. Gleichwohl stehen Mitarbeiter in der Produktentwicklung häufig vor der Herausforderung, dass Sie im Internet nach Lösungselementen oder Lösungswissen in unstrukturierter Form suchen müssen [GaTD2013].

Beide Themengebiete – Daten- und Wissensmanagement – können immens davon profitieren, dass sie zusammen gedacht werden und ein neues Konzept nicht das Wissensmanagement in Gänze neu erfindet, sondern mit möglichst wenig zusätzlichem Aufwand Ingenieure in ihrer täglichen Arbeit unterstützt und aus bereits vorhandenen Daten effizient neues Wissen schafft und dieses prozessual eingesetzt werden kann. Dies soll insbesondere mit einem Anwendungsbeispiel verdeutlicht werden, welches an verschiedenen Stellen in dieser Arbeit aufgegriffen wird und insbesondere zur Verifikation genutzt wird. In verschiedenen Publikationen [Gaus2006], [StWH2013], [Luck2014] wird darauf hingewiesen, dass Informationsbeschaffung und Datensuche einen bedeutenden Teil der Arbeitszeit von Ingenieuren ausmachen. Laut Kirchner [Kirc2020] sind ca. 1/3 der Arbeitszeit diesem Thema gewidmet. Als Gründe des steigenden Zeitanteils für Informationsbeschaffung und -suche wurden in einer Mitteilung des Berliner Kreises unter anderem steigende Anforderungen an den Innovationsgrad neuer Produkte und eine generell steigende Komplexität der Produkte genannt [Berl2007].

Der zentrale Verknüpfungspunkt zwischen Daten und Wissen liegt dabei in beschriebenen Unternehmensprozessen, da durch sie definiert wird, wie Informationen verarbeitet werden, um Wert zu schöpfen, z.B. um Produkte zu entwickeln und zu produzieren. Dies trifft insbesondere auf größere Unternehmen zu, welche in der Regel nach ISO 9001 [DIN 2015] zertifiziert sind und die dortige Bedingung eines Prozessmanagementsystems erfüllen müssen.

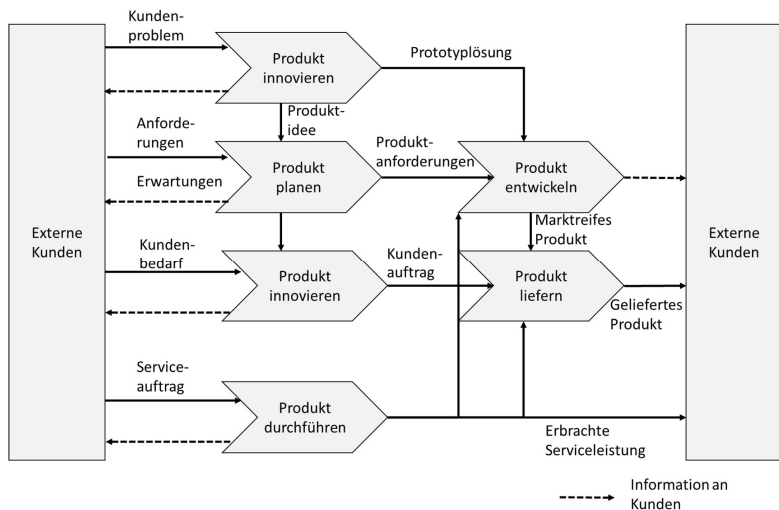


Abbildung 1-1: Beispiel-Prozesslandkarte nach [ScSe2020]

Abbildung 1-1 zeigt exemplarisch die Abbildung einer Prozesslandkarte, die Prozesse wie Produkt planen, Produkt entwickelt und Produkt liefern in den Fokus stellt. Dementsprechend sollte in

den dort beschriebenen Prozessen auch geschildert sein, wie aus Daten als Input ein Output in Form eines Produkts erstellt wird und wer dafür verantwortlich ist.

Auch wenn es auf den ersten Blick scheint, dass die Problematik des Wissens um Daten, Speicherorte und den unternehmerischen Nutzen in Unternehmen praktisch gelöst ist, da es Prozessbeschreibungen gibt, dort die Daten beschrieben sind und Metadaten-Repositories Zugriff auf unterschiedlichste Datentöpfe gewähren, so ist das in der Praxis nicht der Fall. Zum einen sind nicht alle Datenquellen im Metadaten-Repository erfasst und verlinkt, zum anderen fehlt gerade die Verknüpfung zwischen Prozessen und Datenmodellen.

An genau dieser Stelle setzt das Konzept an, welches in dieser Arbeit präsentiert wird. Durch die Kombination der Prozessbeschreibung mit den Datenbeständen wird ein neues Wissensmodell geschaffen, welches den beteiligten Personen ermöglicht festzustellen, ob die erforderlichen Daten bereits im Unternehmen vorhanden sind und wie man darauf zugreifen kann. Oder sich explorativ durch die verfügbaren Datenbestände zu arbeiten, deren Bedeutung zu erfassen und diese Daten anschließend in neue Produkte oder die Optimierung von Prozessen einfließen zu lassen.

Zum Verständnis der Problematik und der erarbeiteten Lösung wird in Kapitel 6 ein Anwendungsbeispiel durchgespielt. Das beispielhafte Szenario behandelt die Überarbeitung eines Produkts durch einen Ingenieur in der Produktentwicklungs-Abteilung. Um Funktionen zu überarbeiten und weitere hinzuzufügen sind Daten aus der Nutzungsphase, aber auch Daten aus der Produktion, wie z.B. vorhandene Fertigungsmittel, Werkzeuge oder Messmittel relevant. Das gleiche gilt für Kundenanforderungen oder Marketingfeedback, um möglichst treffend Anforderungen zu formulieren oder zu bewerten. Das Wissen um diese Informationsquellen und der Zugriff darauf erlauben es, dass der Ingenieur möglichst effizient einen Konstruktionsvorschlag erstellen und umsetzen kann, der alle Kundenbedürfnisse erfüllt, effizient bestehende Fertigungshilfsmittel nutzt und in der Nutzungsphase Feedback für weitere Produktentwicklungen generieren kann.

Im Verlauf des Anwendungsszenarios wird zuerst ein Knowledge Graph aufgebaut, welcher die Zusammenhänge zwischen Abteilungen, Prozessen und Daten beschreibt und anschließend eine Verknüpfung mit der bestehenden Unternehmens-IT durchgeführt. Nachdem die vorhandenen IT-Systeme dem Knowledge-Graph zugeordnet wurden, wird algorithmisch ein Datenmapping durchgeführt, welches dann die benötigten Daten aus dem Knowledge-Graph mit den vorhandenen Daten der IT-Systeme vergleicht und den Nutzern Vorschläge macht, wo sie die für einen besseren Prozess benötigten Daten finden können.

Der Zustand nach Durchlaufen des Anwendungsszenarios ist ein Knowledge-Graph, welcher durchsuchbar entsprechend den Prozessen alle Datenartefakte aufzählt, welche im Rahmen der Engineering-Tätigkeiten benötigt, erzeugt und abgespeichert werden. Von deren Repräsentation im Knowledge-Graph heraus kann direkt eine Brücke geschlagen werden zu den Speicherorten der Daten in ihren jeweiligen IT-Systemen. Sofern der Nutzer für die entsprechenden Systeme Zugriffsrechte besitzt, kann er auch auf die eigentlichen Daten zugreifen und diese in seinem

Workflow benutzen. Sollte er nicht über die Rechte verfügen, so weiß er zumindest um die Existenz der Daten und kann sich die benötigten Zugriffsrechte gewähren lassen.

1.1. Aufbau der Arbeit

Die vorliegende Arbeit gliedert sich in sieben Kapitel, wie in Abbildung 1-2 dargestellt. Nach Motivation und Einleitung werden in Kapitel 2 die Grundlagen der betrachteten Themengebiete erläutert, um einen Überblick über das Thema und seine Zusammenhänge zu geben. Anschließend wird in Kapitel 3 der Stand der Forschung und Technik hinsichtlich der thematischen Blöcke Beschreibung von Daten, sowie Nutzung semantisch angereicherter Daten gezeigt.

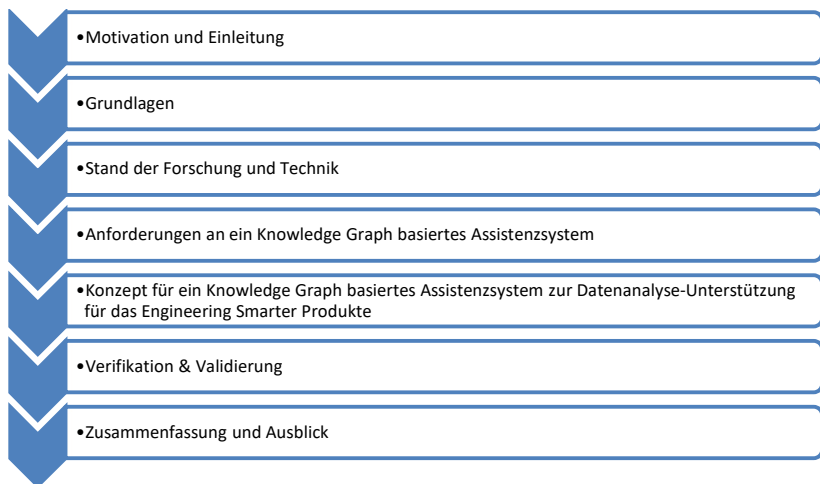


Abbildung 1-2: Aufbau der Arbeit

In Kapitel 4 werden anschließend der Prozess der Anforderungserhebung und anschließend die Anforderungen an das im Rahmen dieser Arbeit entwickelte Knowledge Graph basierte Assistenzsystem beschrieben und die Eignung der vorher präsentierten Prozesse, Methoden und IT-Werkzeugen untersucht.

Kapitel 5 beschreibt das anhand der Anforderungsliste aufgebaute Konzept für ein Knowledge Graph basiertes Assistenzsystem zur Datenanalyse-Unterstützung für das Engineering Smarter Produkte und geht dabei auf die verschiedenen Lösungsbausteine ein, welche notwendig sind, um das System technisch umzusetzen und auch in einem Unternehmen einzusetzen.

Anschließend wird in Kapitel 6 das vorher präsentierte Konzept hinsichtlich der Erfüllung der Anforderungen verifiziert und mithilfe von Experteninterviews auch validiert. Die Verifikation erfolgt anhand eines Anwendungsbeispiels.

1.2. Methodisches Vorgehen

Die Forschung in dieser Arbeit lehnt sich an die Design Research Methodology (DRM) von Blessing et al. an und ist dort dem Typ „Development of Support“ zuzuordnen, was bedeutet, dass die alle vier Phasen der Forschungsarbeit durchlaufen werden: Research Clarification, Descriptive Study I, Prescriptive Study I, Descriptive Study II [BlCh2009].

Nach der initialen Klärung der Forschungsidee wurde im Rahmen von Recherchen die theoretische Grundlage des Themas und der aktuelle Stand der Forschung und Technik in allen relevanten Teilbereichen geklärt und beschrieben. Basierend auf der Recherche, aus Gesprächen und eigenen Überlegungen ergeben sich Anforderungen an das anschließende Konzept. Dieses Konzept wurde anschließend explorativ theoretisch erarbeitet und ein Demonstrator aufgebaut. Die einzelnen methodischen Phasen Research Clarification, Descriptive Study I und Prescriptive Study I wurden vollständig durchlaufen, während anhand des entwickelten Demonstrators nur eine initiale Verifikation und Validierung durchgeführt wurde.

2. Grundlagen

Der betrachtete Gegenstand der Daten im Engineering ist ein sehr stark interdisziplinäres Thema, zum einen, weil Engineering-Prozesse aus Sicht der IT betrachtet werden, zum anderen aber auch, weil das Engineering an sich in den letzten Jahrzehnten immer stärker interdisziplinär ausgerichtet wurde. Zu den betrachteten Grundlagen gehören deswegen die unterschiedlichen Domänen mit ihren jeweiligen Eigenschaften. Daneben spielen auch die Veränderungen im Charakter der Produkte durch Interdisziplinarität und smarte Eigenschaften eine große Rolle und werden in diesem Kapitel geschildert. Eine weitere Folge dieser mittlerweile smarten Produkte sind auch mehr Daten entlang des Produktlebenszyklus, welche in unterschiedlichen Systemen verwaltet werden und dabei auch von den richtigen Personen zur richtigen Zeit gefunden werden und anschließend mit Fachwissen interpretiert und bearbeitet werden. Dementsprechend schildert dieses Kapitel auch verbreitete Datenverwaltungstools im Produktlebenszyklus und zuletzt, welchen Einfluss Wissen darauf hat.

2.1. Betrachtete Engineering-Domänen

Die oben beschriebene Interdisziplinarität zeigt sich direkt in den Disziplinen, die notwendig sind, um ein Produkt zu entwickeln und auf den Markt zu bringen. Diese umfassen in erster Linie die Mechanik, Elektrik/Elektronik und Software, werden aber zusehends von Services als vierter Disziplin ergänzt. Häufig wird im Zusammenhang mit interdisziplinären Produkten auch von Mechatronik gesprochen [Rodd2019].

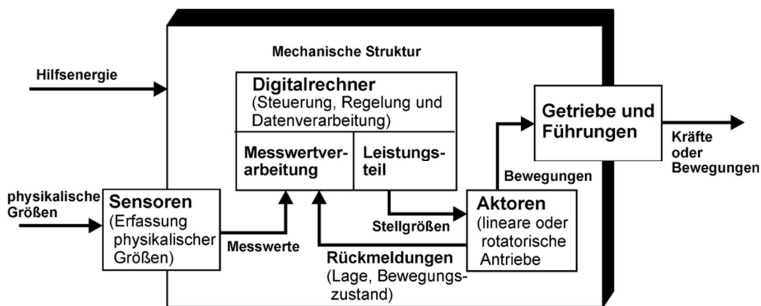


Abbildung 2-1: Grundstruktur eines mechatronischen Systems aus [Rodd2019]

Abbildung 2-1 zeigt die Grundstruktur eines mechatronischen Systems, aus der sich direkt die involvierten Disziplinen ablesen lassen. Ein mechanisches Grundsystem wird mithilfe von Sensoren und Aktoren in die Lage versetzt mit seiner Umgebung zu interagieren. Zur Vernetzung dieser mechanischen Komponenten dienen elektrische Komponenten, welche durch eine Kombination aus Software und elektronischen Komponenten in Form eines Digitalrechners gesteuert werden [Rodd2019].

Mechanische Systeme sind Ausgangspunkt der gesamten Entwicklung in Richtung der Elektronik. Sie beschreiben die Gesamtheit der Erzeugung von Bewegungen, dem Übertragen von Kräften oder Drehmomenten. Um diese Arbeit zu verrichten sind sie auf Steuerungen und von außen zugeführter Energie angewiesen. Hierbei können die Steuerungen entweder mechanisch oder elektrisch sein, ebenso die zugeführte Energie. Ihre Hauptkomponenten sind unter anderem Maschinenelemente, Maschinen oder Feingerätetechnik [Iser2008].

Elektrische bzw. elektronische Systeme zeichnen sich durch Komponenten wie Mikroelektronik, Leistungselektronik, Messtechnik und Aktorik aus. Sie ergänzen häufig mechanische Systeme, indem Mess- und Führungsgrößen in Form von elektrischen Signalen steuernd oder regelnd auf das mechanische Grundsystem einwirken. In mechanisch-elektronischen Systemen ist dies insbesondere dadurch gekennzeichnet, dass neben einem Energie- auch ein Informationsstrom durch das System fließt [Iser2008]

Software sind die Programme, die für einen Rechner verfügbar und ausführbar sind, sowie alle dazugehörigen Daten. Software die speziell dazu dient Bedürfnisse ihrer Nutzer zu ermöglichen wird auch als Anwendungssoftware bezeichnet [GeGa2022]. Software, welche im Kontext von physischen Systemen zur Steuerung oder Regelung eingesetzt wird, wird als „embedded software“ [Lee2002] bezeichnet. Im Gegenzug zu Software, die auf einem PC ausgeführt wird, stellen sich höhere Anforderungen an embedded Software, wie z.B. Echtzeitanforderungen, parallelen Prozessen oder einer enormen Stabilität [Lee2002]. In mechatronischen Systemen dient die Software dazu auch höherwertige Funktionen als nur die Regelung zu bedienen, was sich bis hin zu intelligenten mechatronischen Systemen entwickeln kann [Iser2008].

Im Gegensatz zu materiellen Produkten beschreiben Services eine immaterielle Komponente, welche eine Interaktion zwischen einem Service Provider und einem Kunden voraussetzt. Services können hierbei entweder ein materielles Produkt begleiten, oder stehen als immaterielles Wirtschaftsgut für sich [AlDE2019]. Hierbei kann ein Wandel der Geschäftsmodelle ausgemacht werden, in denen Kunden ein Produkt nicht mehr käuflich erwerben, sondern lediglich das Produkt als Service nutzen und hierbei eine Gebühr zahlen [KoER2018].

2.2. Verständnis industrieller Produkte

Industrielle Produkte sind Ergebnisse eines Prozesses [Jura1993], [DIN 2015]. Sie zeichnen sich durch Eigenschaften aus, die Kundenbedürfnisse erfüllen und Zufriedenheit erreichen sollen [Jura1993]. In der Produktionswirtschaft sind sie insbesondere die finale Ausbringungsmenge einer Produktion [Cors2004]. Das Ziel der Herstellung von Produkten ist es sie zu verkaufen oder für die Nutzung Geld zu verlangen [Send2009].

Produkte lassen sich in unterschiedliche Kategorien einteilen. Man unterscheidet z.B. materielle und immaterielle Produkte [Cors2004]. Eine ähnliche Unterscheidung beschreibt auch Juran, der sie in Güter und Dienstleistungen einteilt [Jura1993]. In der DIN EN ISO 9000:2005 wird darüber

hinaus eine vierteilige Kategorisierung gewählt, die Produkte in Dienstleistungen, Software, Hardware und verfahrenstechnische Produkte ordnet [DIN 2015].

Innovative Produkte erfordern ein interdisziplinäres Zusammenwirken von Maschinenbau, Elektrotechnik und Informationstechnik. Der Begriff „Mechatronik“ bringt dies zum Ausdruck [VDI2004]. Die Einteilung eines solchen Produktes in Kategorien geschieht letztlich durch den überwiegenden Anteil einer Disziplin, z.B. ist ein Auto in erster Linie der Kategorie „Hardware“ zuzuordnen, obwohl auch Software und verfahrenstechnische Produkte vorhanden sind [DIN 2015].

2.2.1. Smartes Produkt-Service System

Laut Porter und Heppelmann bestehen smarte Produkte aus drei Komponenten: physische Komponenten, welche das mechanische Grundsystem abbilden, smarte Komponenten, welche vor allem Sensoren und die Steuerung umfassen, sowie Konnektivitätskomponenten, welche über Hard- und Software eine Verbindung zu anderen Komponenten oder dem Internet herstellen [PoHe2014].

Müller et al. betonen den Charakter smarter Produkte als Teil eines größeren Systems, welches durch eine Lernfähigkeit möglichst ideales Verhalten für den Kunden schafft, da ein Produktwechsel mit einer neuen Anlernphase verbunden wäre [MüLG2019]. Ähnlich sieht Mühlhäuser smarte Produkte als Objekte, welche sich selbst-organisierend in verschiedene Umgebungen einbetten und über Einfachheit und Offenheit einen Benefit für den Nutzer und andere Produkte schaffen [Mühl2008].

Im Sinne der sich wandelnden Geschäftsmodelle und ansteigender Serviceanteile an den Umsätzen betrachten Uhlmann und Meier [UhMe2017] ein gewandeltes Produktverständnis und prägen in diesem Sinne den Begriff „Industrielles Produkt-Service-System“ (IPSS), also eine stark B2B-zentrierte Sicht auf Produkte, welche aus Sach- und Dienstleistungen bestehen und sich direkt in die Wertschöpfungsprozesse der Kunden integrieren. Zur Gestaltung dieser IPSS bieten sich dem Anbieter eine Reihe von Ausprägungen, die er wählen kann [UhMe2017]. Ein solches IPSS, oder alternativ ein hybrides Leistungsbündel (HLB) hat für Anbieter und Kunde weitere Vorteile. Der Kunde kann durch den Dienstleistungsanteil Knowhow outsourcen und sich auf seine Kernkompetenzen konzentrieren, für den Anbieter bietet der Dienstleistungsanteil einen schwer zu plagierenden Teil, der das eigene Knowhow schützt [MeUh2012].

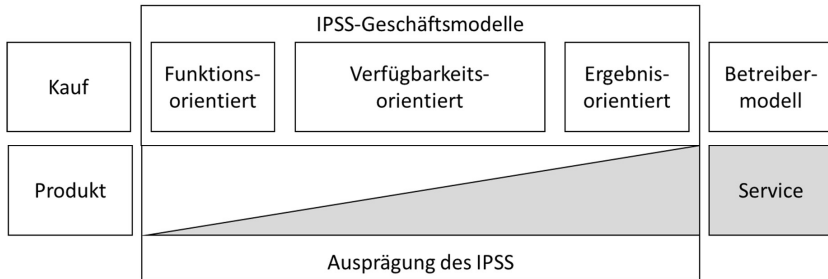


Abbildung 2-2: Ausprägung industrieller Produkt-Service-Systeme nach [MeUK2005]

Welche Ausprägungen smarte Produkte, bzw. Industrielle Produkt-Service Systeme – besitzen, zeigt Abbildung 2-2. Ein IPSS kann auf einer Skala von 100% produktorientiert bis hin zu 100% serviceorientiert sein. Je nachdem kommen dabei verschiedene Geschäftsmodelle wie Funktionsorientierung, Verfügbarkeitsorientierung oder Ergebnisorientierung in Frage. Dies hat auch Auswirkungen darauf, ob ein IPSS gekauft wird, oder in Form eines Betreibermodells beim Kunden genutzt werden kann [UhMe2017].

Kagermann et al. definieren smarte Produkte vor allem über die Intelligenz, welche durch den Einbau von Hochleistungscomputern in physische Produkte entsteht. Eine weitere wichtige Eigenschaft der smarten Produkte ist die Vernetzung, welche durch Plattformen mit Informationen aus den smarten (physischen) Produkten gespeist wird [KaRH2014].

Abramovici [Abra2014] definiert cyber-physische Produkte bzw. Systeme (CPS), als solche, welche internetbasiert Services integrieren und nutzen, um notwendige Funktionen zu erbringen. Hierbei sind CPS als „intelligente“ mechatronische Produkte bzw. Systeme mit Kommunikationsfähigkeit zu anderen CPS definiert. Aufgrund der großen Bandbreite an industriellen Produkten wird der Begriff „System“ oder „Produkt-System“ eher für komplexere Produkte verwendet, was sich teilweise auch historisch als Entwicklung darstellt, wenn Produkte immer komplexere Funktionen beherrschen und über mehrere Generationen dann Intelligenz und Kommunikationsfähigkeiten hinzugewinnen [Abra2014]. Eine ähnliche Definition prägen Gausemeier et al. [GaTD2013], die der „intelligenten technischen Systeme“, welche sich im Gegensatz zur Mechatronik dadurch auszeichnen, dass die Kopplung zwischen Sensorik und Aktorik nicht mehr starr ist, sondern sich aufgrund der Aufgaben wandeln kann, wobei auch eine Kooperation mit anderen Systemen in einem Verbund stattfinden kann [GaTD2013].

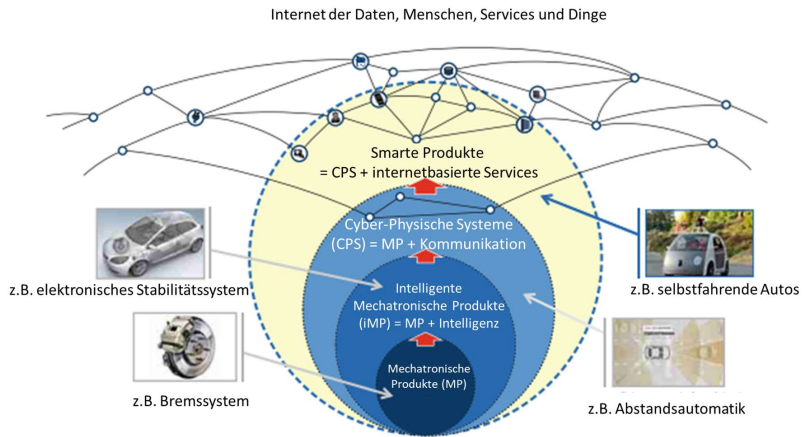


Abbildung 2-3: Evolution von traditionellen zu smarten Produkten nach [Abra2014]

In einer Expertenbefragung durch Abramovici und Herzog wurde von Seiten verschiedener Fachexperten vor allem die Kommunikations- und Vernetzungsfähigkeit als zentrale Komponente smarter Produkte hervorgehoben. Eigene Intelligenz und Autonomie der Produkte, sowie weitere Eigenschaften wurden nicht so eindeutig als zentrales Merkmal angesehen. Dementsprechend sind vor allem auch Kommunikationsschnittstellen, Sensoren sowie Software die laut Studienteilnehmern wichtigsten Komponenten smarter Produkte [AbHe2016].

Neben den Komponenten des eigentlichen smarten Produkts erfordern diese auch den Umbau der firmeneigenen IT-Infrastruktur zum Management der smarten Produkte und der von dort bereitgestellten Informationen. Aus den Komponenten und dem Zusammenspiel mit der firmeneigenen IT-Infrastruktur ergeben sich laut Porter und Heppelmann auch die Fähigkeiten smarter Produkte, welche in aufsteigender Komplexität zuerst sich selbst oder ihre Umgebung monitoren und das Ergebnis mitteilen können, anschließend eine Steuerungsmöglichkeit durch eine nicht vor Ort befindliche Instanz, die Fähigkeit sich selbst zu optimieren und zuletzt eine Autonomiefähigkeit. Für Firmen ergeben sich aus dem Einsatz smarter Produkte erheblich mehr strategische Fragen als aus der Produktentwicklung klassischer Erzeugnisse [PoHe2014]. Durch die strategische Ebene – ausgelöst aus der Vielzahl der neuen Fähigkeiten, aber auch durch die Vielzahl der Implikationen in allen Richtungen führt dazu, dass durch den Wandel zu smarten Produkten auch die Firmen und ihre Wertschöpfungsnetzwerke einem umfassenden Wandel unterliegen [PoHe2015].

2.2.2. Produktstrukturen

Eng verbunden mit der Natur eines Produktes ist auch die Produktstruktur, welche im Engineering festgelegt und später in der Produktion umgesetzt wird. Historisch gesehen gibt es hier je nach Disziplin unterschiedliche Strukturen. So sind in mechanischen oder elektrischen/elektronischen

Systemen eher hierarchisch strukturierte Stücklisten anzutreffen, während im Bereich der Software netzartige Strukturen aus der Versionskontrolle vorherrschen. Smarte Produkte müssen über die „klassischen“ Disziplinen Maschinenbau, Elektrik/Elektronik und Software aber auch noch Modelle im Bereich Systemarchitekturen und Services integrieren und erfordern daher neuartige Produktstrukturen [EiAS2016].

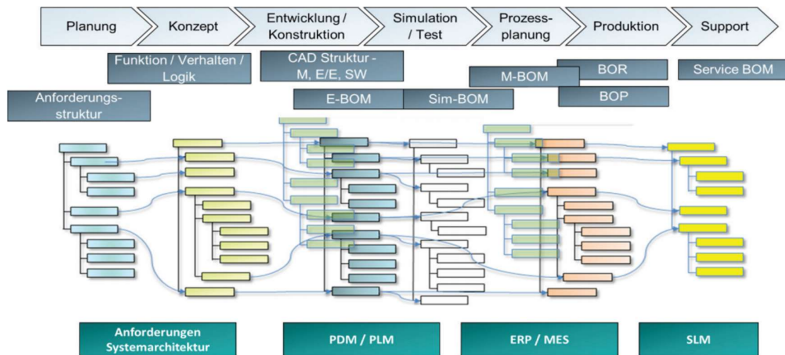


Abbildung 2-4: Unterschiedliche Produktstrukturen entlang des PLZ aus [EiAS2016]

Eine weitere Problematik innerhalb der Produktstrukturen ergibt sich aus unterschiedlichen Anforderungen, historischen Pfadabhängigkeiten und IT-Systembrüchen (vgl. Abbildung 2-4). So werden innerhalb des Produktlebenszyklus abgeleitet aus der 3D-CAD Konstruktion zuerst Konstruktionsstücklisten (Engineering-BOMs, E-BOMs) erstellt und anschließend in produktions- und montageorientierte Fertigungsstücklisten (Manufacturing-BOMs, M-BOMs) umgewandelt, welche häufig auch in unterschiedlichen IT-Systemen verwaltet werden. Beide Strukturen beschreiben das gleiche Produkt, sind allerdings unterschiedlich und teilweise inkompatibel, was vor allem im Bereich des Änderungsmanagements zu Problemen führt. Ansätze wie funktional substituierbare Baukästen zur lieferanten- und standortunabhängigen Darstellung einer Fertigungsstruktur können bei einer Ausnutzung von Datenverwaltungssystemen wie z.B. Product Lifecycle Management (PLM) Systemen als Single Source of Truth das Problem der unterschiedlichen Strukturen und der IT-Brüche lösen, lassen sich allerdings nur schwer mit Bestandsdaten in Unternehmen realisieren [MrLR2022], [LeRM2021].

2.3. Prozesse

Es existiert eine Vielzahl von Definitionen für Prozesse. Z.B. ist ein Prozess als „Gesamtheit von aufeinander einwirkenden Vorgängen in einem System, durch die Materie, Energie oder Information umgeformt, transportiert oder gespeichert wird“ [DIN 2014] definiert. Prozesse müssen nicht für sich allein stehen, sondern können kombiniert mit weiteren Prozessen auch Unterprozesse oder übergreifende Prozesse bilden [DIN 2014].

In der DIN EN ISO 9000:2005 ist die Definition von Prozess allgemein gefasst. So ist ein Prozess eine Menge von wechselwirkenden Tätigkeiten, welche Eingaben umwandelt in Ergebnisse. In aufeinander aufbauenden Prozessen dienen die Ergebnisse des Einen als Eingabe des Anderen [DIN 2015]. Die DIN 69901 definiert Prozess allgemein über seine Bestandteile: Input, Umsetzung und Output und teilt diese in verschiedene Kategorien ein, vor allen in Wertschöpfungsprozesse, welche dem eigentlichen fachlichen Ziel dienen, sowie Unterstützungs-, Projektmanagement-, sowie Führungsprozesse, welche übergeordnet stattfinden [DIN2009].

Im PLM-Umfeld werden Prozesse als „Abbildungen von technischen/organisatorischen Geschäftsabläufen“ [EiSt2009] verstanden und unterscheiden zwischen langfristig stabilen Prozessen, sowie sich dynamisch und kurzfristigen oder temporären Prozessen. Zu den Beispielen der langfristigen Prozesse zählen im PLM-Umfeld z.B. das Freigabe- und Änderungswesen, zu den kurzfristigen Prozessen eher das Delegieren von Aufgaben und Workflows [EiSt2009]. Allgemeiner sind „Geschäftsprozesse [...] betriebliche Abläufe, die sich entlang einer Wertschöpfungskette identifizieren lassen, unmittelbar auf den Erfolg am Markt ausgerichtet sind und durch einen messbaren Input eine Wertschöpfung und einen messbaren Output gekennzeichnet sind“ [Gier2000].

2.3.1. Änderungsmanagement

Ein in der Praxis besonders wichtiger Prozess ist das Änderungsmanagement. Ihm kommt so hohe Bedeutung zu, da Änderungskosten im Verlauf eines Produkts exponentiell steigen, wie Abbildung 2-5 zeigt [EiSt2009]. Änderungen sind dabei definiert als „alle nachträglichen Änderungen an freigegebenen (d.h. verbindlich festgelegten) Arbeitsergebnissen innerhalb eines zusammenhängenden technischen Entwicklungsprozesses [...]. Sie beinhalten immer eine Änderung der technischen Dokumentation bzw. Datenbasis, schließen aber auch alle damit zusammenhängenden Produkt- und Prozessänderungen ein“ [Conr1997].

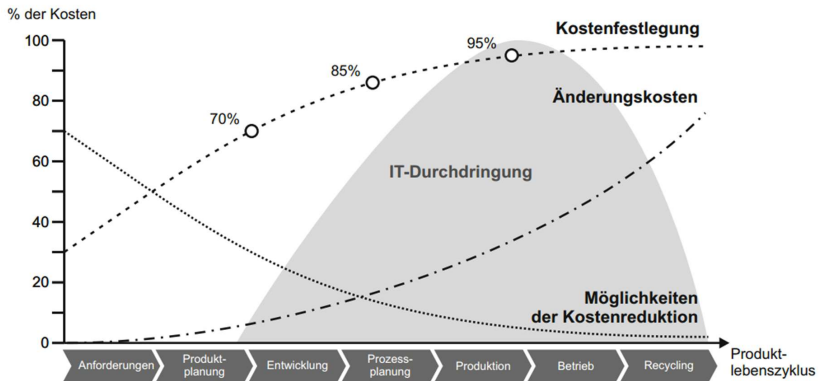


Abbildung 2-5: Änderungskosten im Laufe des Produktlebenszyklus aus [EiSt2009]

Ein gut beherrschter Änderungsmanagement-Prozess wird als entscheidender Erfolgsfaktor für Unternehmen gesehen [JaCE2005]. Die Bedeutung unterstreicht auch eine am Lehrstuhl VPE durchgeführte Studie, welche zeigt, dass bis zu 34% der durchschnittlichen Arbeitszeit einer Abteilung mit Änderungen verbracht werden, jährlich im Durchschnitt ca. 6.000 Änderungen durchgeführt werden, in manchen Unternehmen sogar mehr als 15.000 [Quir2015].

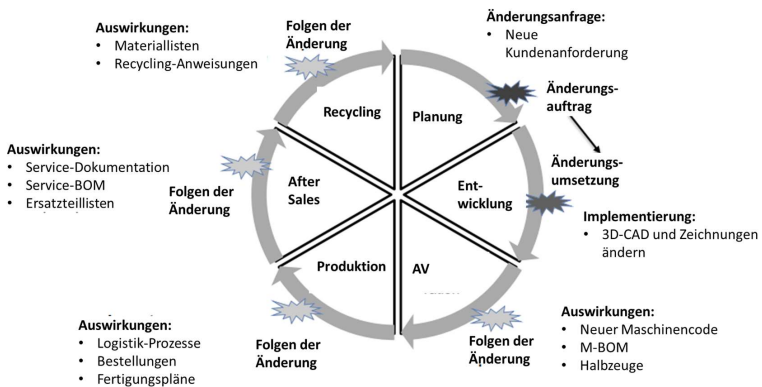


Abbildung 2-6: Auswirkungen von Änderungen nach [Stek2016]

Abbildung 2-6 zeigt wie vielfältig die Auswirkungen von Änderungen auf verschiedene Dokumente und Objekte entlang des Produktlebenszyklus sind und wie viele Bereiche von einer Änderung betroffen sein können [Stek2016]. Entsprechend seiner Wichtigkeit gibt es eine ganze Reihe von definierten Prozessen, die das Änderungsmanagement organisieren. Z.B. die Richtlinie VDA 4965 [VDA2010], [VDA2010], Kooperatives Änderungsmanagement [AuZa2015], ADVICE [KoAk2010], oder Change Management Backbone and Control (vgl. Kapitel 3.2.4) [Erns2016].

2.3.2. Freigabemanagement

Neben dem Änderungsmanagement ist auch das Freigabemanagement ein wichtiger Prozess im Engineering, Abbildung 2-7 zeigt in welchem Zusammenhang beide Prozesse zueinanderstehen.

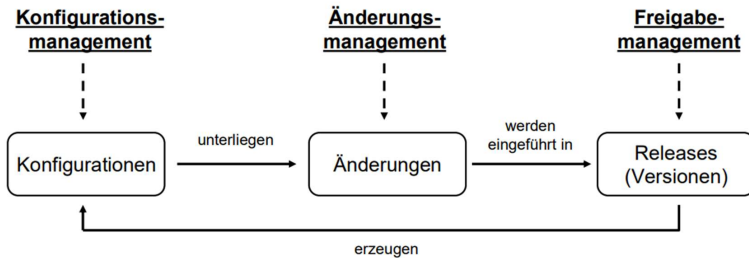


Abbildung 2-7: Zusammenhang Freigabe- und Änderungsmanagement aus [Zarn2007]

Das Freigabemanagement ist als komplexer Prozess eingebettet in Datenstrukturen und Organisation vor allem der Produktentwicklung. Die Freigabe erfolgt immer anlassbezogen, z.B. sobald ein Meilenstein erreicht ist. Der Prozess selbst ist oftmals durch teilautomatisierte und manuelle Schritte gekennzeichnet. Informationen müssen von verschiedenen Personen aus verschiedenen IT-Systemen zusammengetragen und weiterverarbeitet werden. Zur Unterstützung der Freigabeprozesse werden Anforderungen wie ein unterstützendes IT-System, welches die Datensammlung vereinfacht, eine bessere Prozessunterstützung und hohe Usability der IT-Werkzeuge gestellt [MüHH2006].

2.3.3. Prozessmodellierung

Besonderes Augenmerk liegt im Bereich der Prozesse nicht nur darauf, dass diese existieren, sondern auch, dass diese nachvollziehbar dokumentiert sind, um mit diesen zu arbeiten. Abbildung 2-8 zeigt beispielhaft, was man alles mit diesen Prozessmodellen an Tätigkeiten im Bereich Management, Optimierung und Dokumentation machen kann [BePV2012].

Laut Gierhake [Gier2000] besteht die Modellierung aus mehreren Schritten. Zuerst soll der Problembereich möglichst präzise identifiziert und beschrieben werden. Anschließend wird dieser Problembereich so modelliert, dass eine Problemlösung am Modell erfolgen kann. Zuletzt wird dann die Problemlösung vom Modell auf die Realität übertragen und umgesetzt [Gier2000].

Modelle sind hierbei „zielgerichtete Abbildungen eines realen Systems oder Systemausschnitts [...], welche besonders die in dem gegebenen Zusammenhang für eine Problemlösung als wichtig erachteten Aspekte unter Vernachlässigung anderer, als weniger wichtig angesehener Gesichtspunkte darstellt“ [Gier2000]. Dieser Modellbegriff deckt sich auch mit Stachowiak [Stac1973], der die Merkmale der Abbildung, der Verkürzung und des Pragmatismus im Hinblick auf Modelle formuliert [Stac1973].

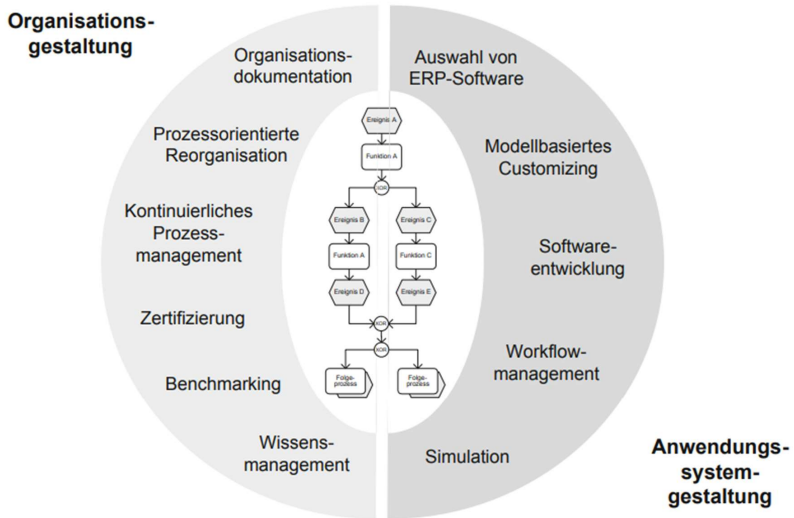


Abbildung 2-8: Nutzen von Prozessmodellen aus [BePV2012]

Je nach Ziel und Nutzen der Modellierung, kann diese auf unterschiedlichen Ebenen stattfinden. Prozesse sind keine immer gleichförmige Struktur, sondern unterliegen einer Hierarchie, welche sich von der obersten Ebene der Prozesslandkarte, welche ein ganzes Unternehmen beschreiben kann, über Prozesse, Teilprozesse, einzelne Schritte bis hin zu Aktivitäten zieht [ScSe2020].

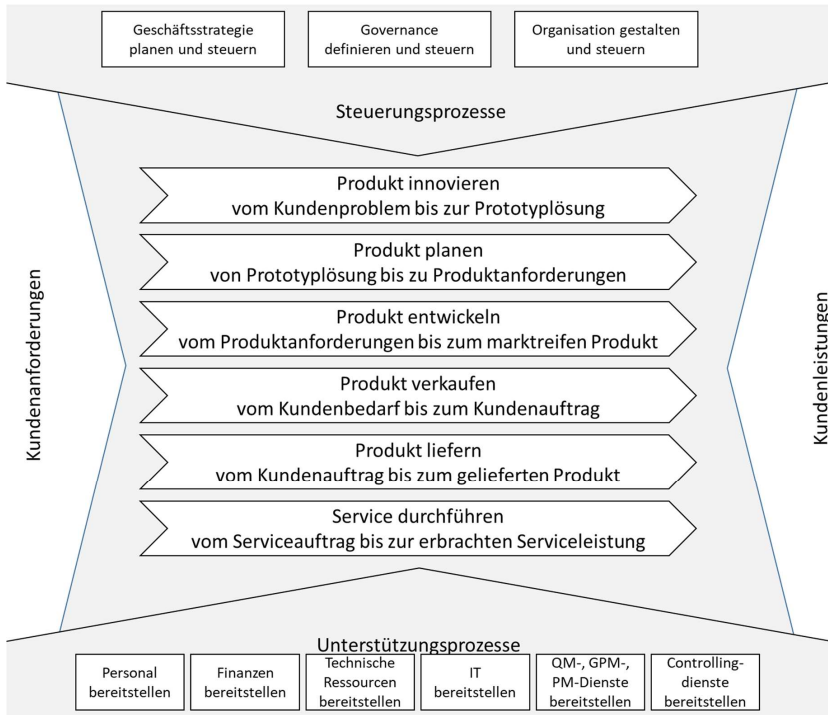


Abbildung 2-9: Prozesslandkarte nach [ScSe2020]

Abbildung 2-9 zeigt eine Prozesslandkarte als Modellierungstechnik für Prozesse auf der obersten Ebene. Sie umfasst klassischerweise mehrere Bereiche, die sich in Steuerungs-, Unterstützungs- und Kernprozesse aufteilen lassen. Allen gemein ist, dass auf der obersten Ebene die Prozessbeschreibung sehr allgemein und umfassend ist, da sich alle Prozesse noch untergliedern lassen [ScSe2020].

Um einzelne Prozesse oder Workflows zu modellieren, existiert eine Vielzahl von Modellierungssprachen, IT-Werkzeuge und Unterstützung, wie z.B. die EPK, BPMN, OMEGA oder Petri-Netze. Die Vielzahl der Modellierungssprachen und kein allgemeinverbindlicher Standard, der verfügbar ist, führen dazu, dass immer neue Symbole eingeführt werden und so die Komplexität weiter erhöhen [Rose2006].

Die EPK – ereignisgesteuerte Prozesskette – wurde 1992 am Institut für Wirtschaftsinformatik in Saarbrücken entwickelt. Sie zielt darauf ab, eine ganzheitliche Modellierung für Daten und Verhalten innerhalb von Prozessen zu schaffen. Hierzu wird methodisch ARIS (Architektur integrierter Informationssysteme) [Sche1992] genutzt. Die Ergebnisse der Modellierung können

genutzt werden, um die Prozesse zu verbessern und eine einheitliche Datengrundlage und Datenbank zu schaffen [KeNS1992].

Die Grundelemente der EPK bestehen aus Funktionen, Ereignissen und Informationsobjekten. Funktionen transformieren einen Input in einen Output und beschreiben „was“ hierzu getan werden muss, ohne genauer auf das „wie“ einzugehen. „Eine Funktion beschreibt auf der Fachkonzeptebene die Durchführung eines betrieblichen Vorgangs, der zur Erfüllung eines Unternehmensziels beiträgt. Sie ist somit eine semantische Verarbeitungsregel, die einen Eingangszustand in einen Zielzustand (Output) umwandelt“ [KeNS1992].

Ein Ereignis beschreibt einen definierten Zustand, welche aufgrund von Funktionen oder externen Bedingungen eingetreten ist. Ein Ereignis kann an dieser Stelle wiederum Funktionen auslösen, es sei denn, es ist ein „Systemereignis“, welches keine Funktion nach sich zieht. Es können weiterhin Beziehungen zwischen Informationsobjekten und Ereignissen bestehen [KeNS1992].

Informationsobjekte sind solche abstrakten oder realen Dinge, welche eine Information tragen, die für die Bearbeitung von Funktionen sinnvoll ist. „Ein Informationsobjekt ist ein von den Aktionsträgern semantisch zu beschreibender und identifizierbarer Sachverhalt. Informationsobjekte stellen Mengen realer oder abstrakter Dinge dar, die für ein Unternehmen von Interesse sind“ [KeNS1992].

Neben den Kernelementen der EPK gibt es noch Verknüpfungsoperatoren, um die Elemente logisch miteinander zu verbinden. Es gibt drei unterschiedliche Operationen, die eine „Und“-Verknüpfung, eine „Entweder Oder“-Verknüpfung und zuletzt eine „Und/Oder“-Verknüpfung symbolisieren. Je nach gewählter Verknüpfung muss mindestens eine (bei „Und/Oder“), exklusiv eine (bei „Entweder oder“) oder beide Bedingungen erfüllt sein (bei „Und“) [KeNS1992].

Abbildung 2-10 zeigt einen beispielhaften Prozess, welcher mit der erweiterten EPK modelliert wurde. Diese Erweiterung zielt vor allem darauf ab, noch weitere neue Konstrukte hinzu. So ist trotz seiner Beschreibung das Informationsobjekt kein Teil einer ereignisgesteuerten Prozesskette. Genauso werden noch weitere Objekte wie „organisatorische Rolle“, „organisatorische Stelle“ und „Anwendungssystem“ hinzugefügt [BePV2012].

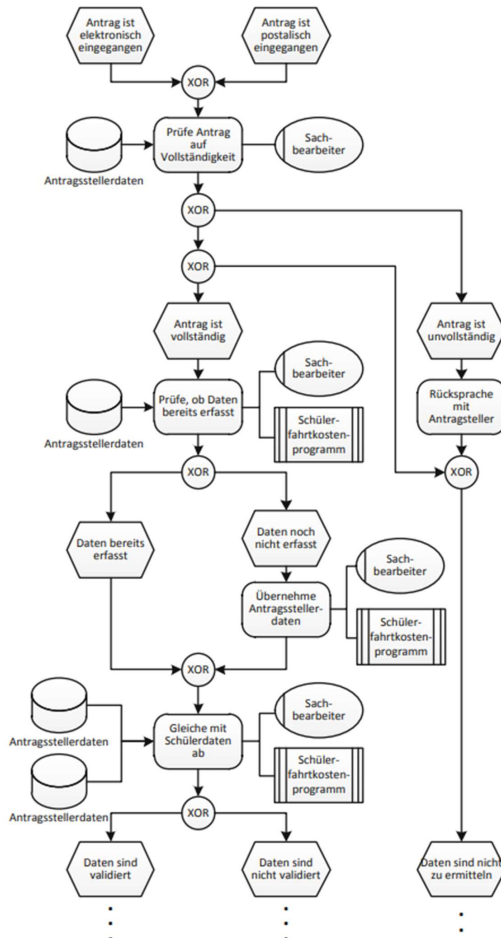


Abbildung 2-10: Beispielhafte Prozessmodellierung mit der eEPK aus [BePV2012]

Eine weitere Art der Prozessmodellierung ist die SIPOC-Methode, welche aus dem Bereich Qualitätsmanagement bzw. Six Sigma entstammt. Der Name ist ein Akronym und steht für die einzelnen Bestandteile, welche hier modelliert werden. Supplier, Input, Process, Output, Customer. Also Lieferanten, Eingaben, Prozesse, Ausgaben und Kunden [MaRe2009].

| Lieferant | Input | Prozess | Output | Kunde |
|--|---|---|---|--|
| <ul style="list-style-type: none"> • Endkunde • Material-Lieferant • Software-Lieferant | <ul style="list-style-type: none"> • Modelle • Anforderungen • Software | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Druck- vorbereitung</div> | <ul style="list-style-type: none"> • Designvorschlag | <ul style="list-style-type: none"> • Druckabteilung |
| <ul style="list-style-type: none"> • Druckvorbereitung • Material-Lieferant • Drucker-Lieferant | <ul style="list-style-type: none"> • Designvorschlag • Tinte • Pappe | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Druck</div> | <ul style="list-style-type: none"> • Gedruckte Objekte | <ul style="list-style-type: none"> • Nachbereitung |
| <ul style="list-style-type: none"> • Druckvorbereitung • Druckabteilung • Material-Lieferant | <ul style="list-style-type: none"> • Gedruckte Objekte • Designvorschlag • Equipment | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Nach- bereitung</div> | <ul style="list-style-type: none"> • Fertiggestellter Auftrag | <ul style="list-style-type: none"> • Versand |
| <ul style="list-style-type: none"> • Versand | <ul style="list-style-type: none"> • Fertiggestellter Auftrag • Verpackung | <div style="border: 1px solid black; padding: 5px; width: fit-content; margin: 0 auto;">Versand</div> | <ul style="list-style-type: none"> • Fertiggestellter Auftrag wurde versandt | <ul style="list-style-type: none"> • Endkunde |

Abbildung 2-11: Beispiel für einen Prozess in SIPOC Notation nach [MaRe2009]

Die SIPOC-Modellierung zeichnet sich durch eine besondere Einfachheit aus, da wenige Zusammenhänge nach einer festen Syntax modelliert werden müssen. Informationen werden vor allem als Texte in die verschiedenen Formularfelder gefüllt. Hierdurch ist eine automatisierte Weiterverarbeitung dieser Diagramme allerdings schwierig.

Die BPMN (Business Process Model and Notation), eine weitere Prozessmodellierungssprache ist sogar international standardisiert durch die OMG (Object Management Group), die entsprechende Spezifikation³ kann im Internet eingesehen werden. Die BPMN deckt verschiedene Anwendungen ab. Neben der reinen Prozessbeschreibung kann sie auch für Simulationen und Prozessausführungen verwendet werden. Hierfür werden Erweiterungen wie XPD L und WS-BPEL⁴ genutzt. Diese Erweiterungen dienen zur Speichern entweder aller (bei der XPD L) oder der zur automatisierten Ausführung relevanten (bei WS-BPEL) Aspekte der BPMN [ChTr2012]. Die BPMN existiert im Laufe der Jahre in verschiedenen Versionen, wie der BPMN 1.x und der BPMN 2.x [WhMi2008], [FiSS2012].

Die BPMN kann mit ihrer reichhaltigen Symbolik (vgl. Abbildung 2-12) Prozesse sehr genau darstellen. Ereignisse können dargestellt werden, welche einen Prozess starten oder beenden, bzw. während des Prozesses eintreten können. Unterschiedlichste Aktivitäten können modelliert werden, mit Hilfe von Konnektoren und Gateways können die einzelnen Ereignisse und Aktivitäten verknüpft werden, um eine logische Reihenfolge darzustellen. Verschiedene weitere Objekte wie Datenobjekte können als Input oder Ergebnis eines Prozesses dargestellt werden.

³ <https://www.omg.org/spec/BPMN/>

⁴ <http://docs.oasis-open.org/wsbpel/2.0/wsbpel-v2.0.html>

Zusätzlich ist eine Zuordnung zu ausführenden oder verantwortlichen Personen über Swimlanes und Pools durchführbar [ChTr2012].

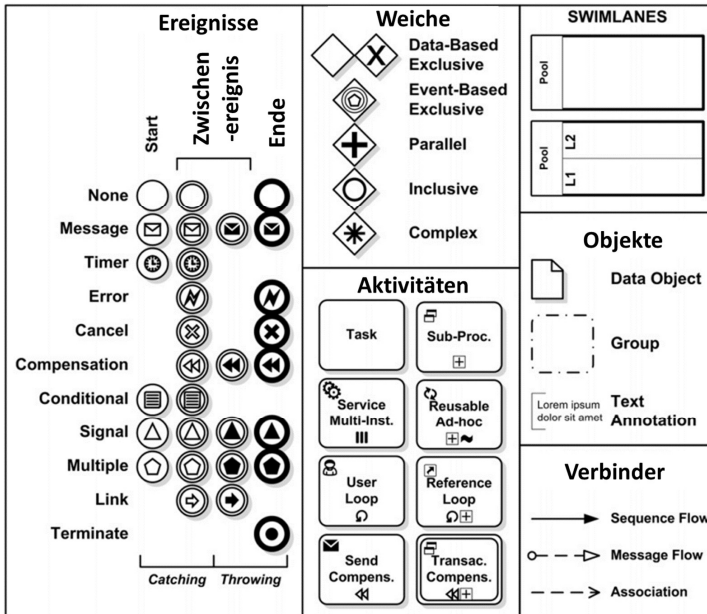


Abbildung 2-12: BPMN-Elemente nach [ChTr2012]

Aufgrund der Standardisierung der BPMN kann diese auch in anderem Kontext zur modellbasierten Erstellung eines Smarten Produkt-Service-Systems eine Modellierung der gewünschten Prozesse in BPMN erfolgen, während Elemente der Produktmodellierung in UML⁵ [BoRJ1999] oder SysML⁶ [Alt2012] erfolgen. Anschließend können beide Modelle miteinander verbunden werden [BoHH2019], [Eign2021].

Eine weitere Modellierungssprache ist OMEGA, die objektorientierte Methode für die Geschäftsprozessmodellierung und -Analyse, welche 1995 im Rahmen einer Dissertation entwickelt wurde. Die Zielsetzung von OMEGA liegt nicht in der Modellierung von Prozessen, sondern vor allem in der Analyse mit einem klaren Fokus auf der Informationsverarbeitung und nicht auf Materialflüssen. Weiterhin soll die Methode möglichst IT-unterstützt ablaufen und das Business Process Re-Engineering unterstützen [Fahr1995].

⁵ UML: Unified Modeling Language, <https://www.omg.org/spec/UML/>

⁶ SysML: System Modeling Language, <https://www.omg.org/spec/SysML>

Das vorgeschlagene Modell besteht aus Elementen der Typen Geschäftsprozesse, externe Objekte, technische Ressourcen, Bearbeitungsobjekten, sowie Organisationseinheiten und Kommunikationsbeziehungen. Geschäftsprozesse stellen hierbei die Aktivitäten dar, externe Objekte sind relevante Dinge außerhalb des eigenen Unternehmens. Technische Ressourcen sind aufgeteilt in verschiedene Kategorien wie IT-Applikationen, Papier- und Materialspeicher. Bearbeitungsobjekte sind der In- und Output des Geschäftsprozesses und können IT-Objekte, Papierinformationen, mündliche Informationen, Material oder Gruppen von Objekten sein. Organisationseinheiten sind Elemente der Aufbauorganisation eines Unternehmens und beschreiben Stellen, Teams oder Abteilungen. Kommunikationsbeziehungen beschreiben den Transfer von In- und Output-Objekten in Geschäftsprozesse bzw. aus Geschäftsprozessen [Fahr1995].

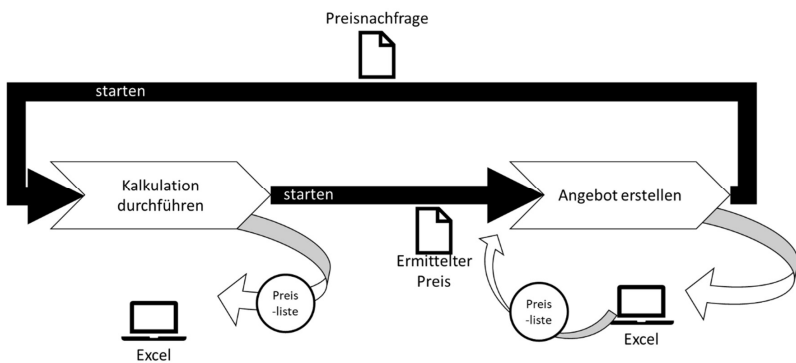


Abbildung 2-13: Prozessmodellierung mit der OMEGA-Notation nach [Fahr1995]

Abbildung 2-13 zeigt einen beispielhaften Geschäftsprozess, in welchem Papier- und IT-Objekte involviert sind und welcher mit der OMEGA-Methodik analysiert werden kann. Eine der wichtigsten Analysen ist dabei die Analyse der Informationsverarbeiten, welche hier beschrieben werden soll [Fahr1995].

Eine wichtige Rolle in der Informationsverarbeitung spielen Medienbrüche, also der mangelhafte Übergang von einem Medium in das andere, beispielsweise weil Dokumente nur in Papierform vorliegen, anschließend aber digital verarbeitet werden müssen. Oder Dokumente liegen in Datenbank A, der nächste Prozess kann allerdings nur Datenbank B ansteuern. Ein weiterer Analysegegenstand sind Redundanzen und fehlende Informationsobjekte, wenn Informationen doppelt vorliegen oder an anderen Stellen fehlen, so dass Prozesse gestört werden können. Zuletzt können Prozesse dahingehend untersucht werden, wie sie miteinander gekoppelt werden, ob direkt oder über Speicher als verbindende Elemente [Fahr1995].

2.4. Produktlebenszyklus

Grundlage dieser Arbeit ist der Wunsch nach einer übersichtlichen Datenverwaltung von Engineering-Daten entlang des Produktlebenszyklus (PLZ). Hierbei gibt es auch verschiedene Definitionen und Ausprägungen von Produktlebenszyklen. Abbildung 2-14 zeigt beispielsweise den Lebenszyklus eines hybriden Leistungsbündels, welches sich durch seine Multidimensionalität von reinen Produkten unterscheidet [MeUh2012].

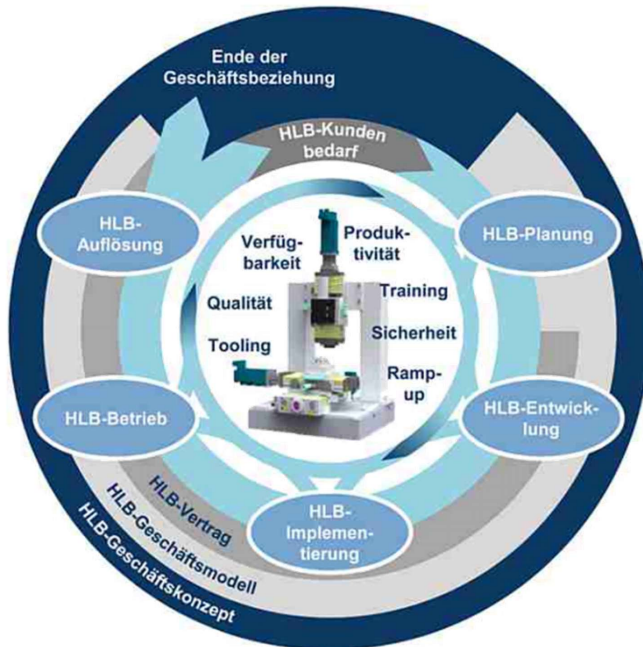


Abbildung 2-14: Lebenszyklus hybrider Leistungsbündel aus [MeUh2012]

Eine eher klassische Sichtweise auf den Produktlebenszyklus zeigt Abbildung 2-15, in der der Produktlebenszyklus beginnend bei Anforderungen linear über die Produktplanung, Produktentwicklung, Prozessplanung und Produktion in die Betriebsphase mündet und mit dem Recycling abschließt. Jeder Phase des Produktlebenszyklus sind dabei typische Aufgaben zugeordnet, welche abgearbeitet werden müssen, bevor die nächste Phase erreicht werden kann. Die Betriebs- und Recycling-Phasen sind in diesem Modell noch nicht so stark von IT durchdrungen, wie einige Jahre später [EiSt2009].

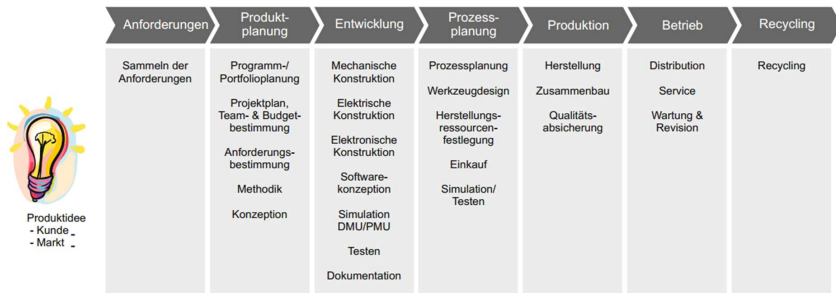


Abbildung 2-15: Phasen des Produktlebenszyklus aus [EiSt2009]

Im Gegensatz dazu zeigt Abbildung 2-16 eine modernere Interpretation des Produktlebenszyklus. Sind die Phasen im Wesentlichen ähnlich, beginnend mit der Entwicklung, anschließend der Produktion, der Nutzung und des End-of-Life, zeigen sich im Detail entscheidende Veränderungen. So ist unter anderem die Nutzungsphase verlängert, indem die Produkte rekonfiguriert und an die aktuellen Erfordernisse angepasst werden können [GöEi2020].

Ebenfalls wird am Ende des Lebenszyklus nicht mehr nur von Recycling gesprochen, sondern das End-of-Life als Input für weitere Iterationen des Produkts angesehen, was insbesondere auch durch die neu hinzugekommenen Digitalen Zwillinge ermöglicht wird. Diese liefern einen permanenten Informationsfluss aus unterschiedlichen Phasen und können zur Optimierung der Produkte während des Betriebs, aber auch als Feedbackmechanismus in der nächsten Produktiteration genutzt werden [GöEi2020].

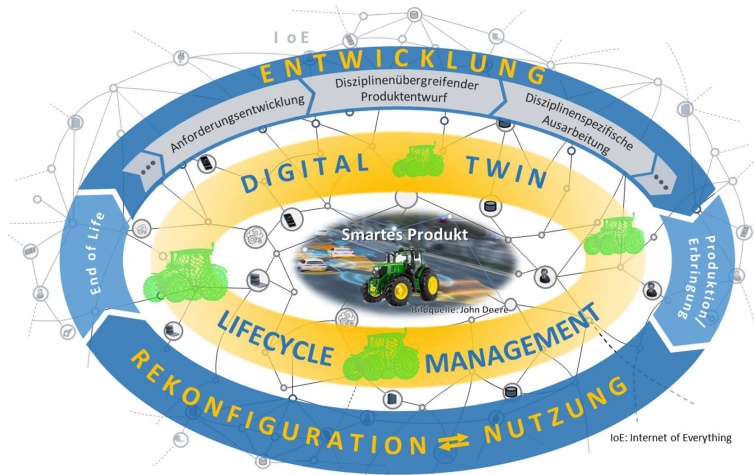


Abbildung 2-16: Der Lebenszyklus smarter Produkte aus [GöEi2020]

Der Wandel hin zu smarten und vernetzten Produkten bewirkt nicht nur die oben beschriebenen Auswirkungen auf den Produktlebenszyklus, sondern sorgt auch dafür, dass der eine isolierte Betrachtung eines Produkts nicht mehr zielführend ist. Vielmehr müssen smarte Produkte auch in ihrem Ökosystem eines System of Systems (SoS) betrachtet werden [FoGD2021]. Eine weitere Ausprägung moderner Lebenszyklen zeigt sich in der immer engeren und schnelleren Rückkopplung von Ergebnissen einer späteren Phase nach vorne in die nächste Iteration des Produkts. In diesem Fall wird auch vom „Engineering 4.0 Lifecycle“ [FoEW2022] gesprochen, welcher auch auf neuartige Methoden der Kollaboration im Engineering zurückgreift [FoEW2022].

Die Nutzung von Datenrückflüssen durch den Produktlebenszyklus und die Analyse dieser bringen neue Herausforderungen mit sich. Entlang der üblichen Prozessschritte zur Datenanalyse sind vor allem die eigentlichen Daten eine Herausforderung. Beginnend bei der Verfügbarkeit der Daten, der Vielfalt, Qualität, Transparenz, Beschaffung und Speicherung der Daten bis hin zur Nutzung in Algorithmen in Prozessen [MeGA2022].

2.5. Digitale Objekte entlang des Lebenszyklus

Auch wenn der Produktlebenszyklus eng mit dem Inverkehrbringen eines physischen Produkts verknüpft ist, wie in Kapitel 2.2.1 beschrieben, so sind dafür eine ganze Reihe von Dokumenten notwendig. Bis zum Start der Produktion ist die gesamte Produktdokumentation in erster Linie digital vorhanden. Digitale Objekte werden häufiger geändert als physische Objekte angepasst werden können. Die Auswirkungen dessen beschreibt Abbildung 2-5 anhand der Kostenfestlegung und der Änderungskosten. Die Summe an digitalen Objekten wird auch das intellektuelle Produkt genannt, im Gegensatz zum physischen Produkt [EiSt2009]. Dieses

Unterkapitel gibt einen Überblick über die wichtigsten digital vorhandenen Daten, bzw. Datentypen, welche entlang des Produktlebenszyklus erstellt, verändert und genutzt werden.

2.5.1. Produktdaten

Produktdaten bezeichnen alle Daten, die in einem logischen Zusammenhang mit der Entwicklung und Erzeugung eines Produkts entstehen. Sie dienen der Identifikation, des Verständnisses über die Konfiguration eines Produkts und beschreiben es auch ökonomisch, z.B. durch seinen Preis. Insbesondere bei rein digitalen Objekten kann eine mangelhafte Identifizierbarkeit der Produktdaten schnell zu Doppelarbeit und erhöhten Kosten führen [Schi2002].

Neben der reinen Beschreibung eines Objekts ist es auch von enormer Bedeutung die Verknüpfungen zwischen verschiedenen Objekten bzw. Produktdaten herzustellen, z.B. in Form einer Stückliste [Schi2002]. Entscheidend ist vor allem auch, dass ein sinnvolles Management dieser Daten stattfindet [CrAD2003]. Produktdaten selbst können dabei sehr unterschiedlich sein, je nachdem in welcher Disziplin und in welcher Branche sie entstehen [EiAS2016].

Vor besondere Herausforderungen stellen daher insbesondere variantenreiche Produkte, welche über äußerst komplexe Strukturen und daher besonders viele Daten verfügen. Für ihr Management braucht es spezielle Konzepte [Zage2006].

2.5.2. Digitaler Zwilling

Der Begriff digitaler Zwilling geht zurück auf die technische Roadmap der NASA im Jahr 2010, die einen digitalen Zwilling als Simulation der Fahrzeuge propagiert, welcher möglichst viele Subsysteme eines Flugzeuges oder Weltraumfahrzeugs abbildet und der eine Mischung aus Sensordaten des physischen Zwillings und Simulationsergebnissen des rein digitalen Zwillings vereint. Die Einsatzzwecke wurden definiert als Vorhersage des Flugverhaltens während laufender Missionen, die komplette Simulation weiterer Missionen, die forensische Untersuchung im Falle eines Unfalls des physischen Zwillings und zuletzt als Plattform für Simulationen durch veränderte Missionsparameter [ShCD2010].

Bereits kurze Zeit später wurde in der NASA ein Digital Twin Paradigma spezifiziert, welches die Anwendungsfälle noch breiter fasst und unter anderem die Auslegung von Konstruktionen, aber auch die Auslegung von Wartungsintervallen umfasst. Dieses Paradigma umfasst vor allem eine Anwendung zur Verifikation und Validierung von technischen Systemen [GLSt2013].

Im Laufe der Jahre sind Definitionen, Einsatzgebiete und Ausprägungen des digitalen Zwillings immer weiter angewachsen und differenziert. Grieves z.B. sieht im digitalen Zwilling mehrere Ausprägungen. So gibt es unter anderem Digital Twin Prototypes, welche alle Produkte, die theoretisch fertigbar sind bezeichnen, Digital Twin Instances, welche alle gerade in der Fertigung befindlichen Produkte bezeichnen und zuletzt Digital Twin Aggregates, welche alle Produkte, welche bereits gefertigt sind bezeichnen [Grie2021]. Göbel und Eickhoff zeigen, dass ein digitaler Zwilling nicht nur auf technische Aspekte wie Sensordaten begrenzt betrachtet werden sollte, sondern auch weitere Gestaltungs-komponenten in den Bereichen Konzeption, IT und

Organisation betrachtet werden sollten. Abbildung 2-17 zeigt das aus dieser Betrachtung erwachsene Kaiserslautern Digital Twin Modeling Framework (KDTM), welches als Ausgangsbasis auch davon ausgeht, dass der Digitale Zwilling eine Abbildung eines existierenden physischen Produkts ist und mithilfe der verschiedenen Bereiche so designt werden kann, dass er unterschiedliche, aber jeweils zu definierende Anwendungsbereiche unterstützt [GöEi2020]. Hierbei muss auch beachtet werden, dass ein solcher Digitaler Zwilling nicht isoliert existiert, sondern auch über Verbindungen zu weiteren Strukturen, meist hierarchischer Art in Form von Produktstrukturen verfügt [EiAD2019]. Anwendungsfälle in diesem Sinne bietet der Digitale Zwilling z.B. zum Ausfall von Komponenten im Bereich eines Kartoffelvollrnters [SiAB2019] oder als Unterstützung durch Handlungsanweisungen für Servicetechniker im Servicefall von Landmaschinen [EiAA2019]. Die Daten aus Digitalen Zwillingen können auch für weitergehende Analysen verwendet werden, z.B. im Bereich der Kabelbaumanalyse [ThWE2018], in der Qualitätssicherung von Automobilen [DeRE2020] oder zum Erreichen von operationeller Exzellenz in der Fertigung [GöMü2020]. Eine weitere Anwendung ist die virtuelle Absicherung von Produktentwicklungsergebnissen in einer virtuellen Produktionsumgebung durch Simulation und Kombination von Daten aus realen Systemen und Simulationen bzw. digitalen Mockups [LuDa2019].



Abbildung 2-17: Das Kaiserslautern Digital Twin Modeling Framework aus [GöEi2020]

Auch Eigner et al. heben besonders die Verbindung zwischen Digitalem Zwilling, Digitalem Master und Digitalem Schatten/Thread hervor. Gerade in Produkten mit langen Lebenszyklen wie der Flugzeugindustrie wird bereits verstärkt mit Digitalen Zwillingen gearbeitet. Den langen Zyklen entsprechend betrachten sie mehrere Phasen, in denen der Digitale Zwilling existiert und jeweils verschiedene Aufgabengebiete erfüllt. Hierzu erfolgt eine Einteilung in die Phasen „as designed“, „as built“ und „as maintained“. Die „as designed“ Phase schließt sich der Produktentwicklung an

und verknüpft die physischen Prototypen mit ihren Digitalen Zwillingen und liefert Ergebnisse für iterative Verbesserungen des Produkts. Nach der Serienfreigabe schließt sich die „as built“ Phase an, in der die Zwillinge zum ersten Mal auch nach einem expliziten Kundenwunsch aus dem digitalen Master im PLM-System abgeleitet werden. Nach dem Durchlaufen der Produktion beginnt die „as maintained“ Phase, in der das physische Produkt vom Kunden genutzt wird und die Informationen über diese Nutzung im Digitalen Zwilling vorgehalten werden, um über eine IoT-Plattform für weitere Verbesserungen genutzt zu werden [EiDS2019], [Eign2020].

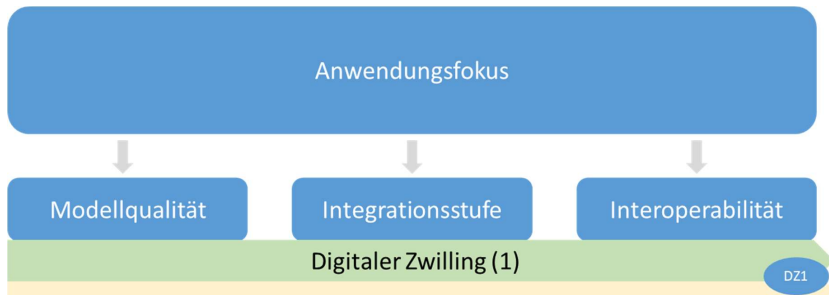


Abbildung 2-18: Ausprägungsformen Digitaler Zwillinge aus [StAT2020]

In einem Positionspapier der WiGeP wurde versucht die verschiedenen Ausprägungsformen von Digitalen Zwillingen in ein Modell zu überführen, welches auch Abbildung 2-18 zeigt. Der Digitale Zwilling muss entsprechend seinem Anwendungsfokus Anforderungen an Modellqualität, Integrationsstufe und Interoperabilität erfüllen. Das heißt in Bezug auf die Modellqualität, dass ein Digitaler Zwilling realitätsgetreu den physischen Zwilling abbilden sollte, aus dessen Digitalem Master er abgeleitet wurde. Die Integrationsstufe bezieht sich auf die Verbindung des Digitalen Zwillinges mit der realen Umgebung. Ob z.B. eine Kommunikation zwischen dem Digitalen Zwilling eines Produkts mit der ihn produzierenden Fertigungslinie möglich ist. Die Interoperabilität ist ähnlich zu verstehen und betrachtet die Kommunikations- und Vernetzungsfähigkeit Digitaler Zwillinge untereinander [StAT2020].

2.5.3. Digitale Verwaltungsschale

Im Kontext von Automatisierung und Industrie 4.0 gibt es ein ähnliches Konzept im Vergleich zum Digitalen Zwilling, die Verwaltungsschale der I4.0-Komponente. Sie ist im RAMI 4.0 definiert als „virtuelle digitale und aktive Repräsentanz eines Assets in einem I4.0 System“ [DIN2016]. Hierbei kann die Verwaltungsschale sowohl in der Komponente selbst abgebildet werden, oder auch in anderen IT-Systemen gespeichert werden. Sie umfasst verschiedene Inhalte, unter anderen Teilmodelle aus Bereichen wie Engineering, Konfiguration, Kommunikation, Identifikation, Lebenslauf oder Condition Monitoring [DIN2016]. Ein beispielhafter Anwendungsfall der Digitalen Verwaltungsschale, welcher sowohl die digitale Abbildung des physischen Produkts, aber auch den anders gerichteten Weg der Steuerung des physischen Assets durch seine Digitale Verwaltungsschale zeigt, wird in [GrSt2020] beschrieben.

Aufgrund der vielfältigen Anwendungsszenarien von digitalen Verwaltungsschalen muss diese unterschiedliche Merkmale abspeichern, welche nicht immer relevant sind. Deswegen dienen verschiedene Basis-Sichten zur Zuordnung von Merkmalen und zur Vereinfachung der verschiedenen Anwendungsfälle. Durch das hohe Level an Komplexität ergeben sich eine Reihe von Anforderungen an die Verwaltungsschale im Allgemeinen, aber auch an einzelne Teilelemente wie z.B. die enthaltenen Informationselemente. Informationen in der Verwaltungsschale müssen z.B. so strukturiert sein, dass sie in Typen oder Instanzen einsortiert werden können, hierarchisch und abzählbar sein, Referenzen möglich sind und Sicherheitskriterien erfüllen können [AdAB2016].

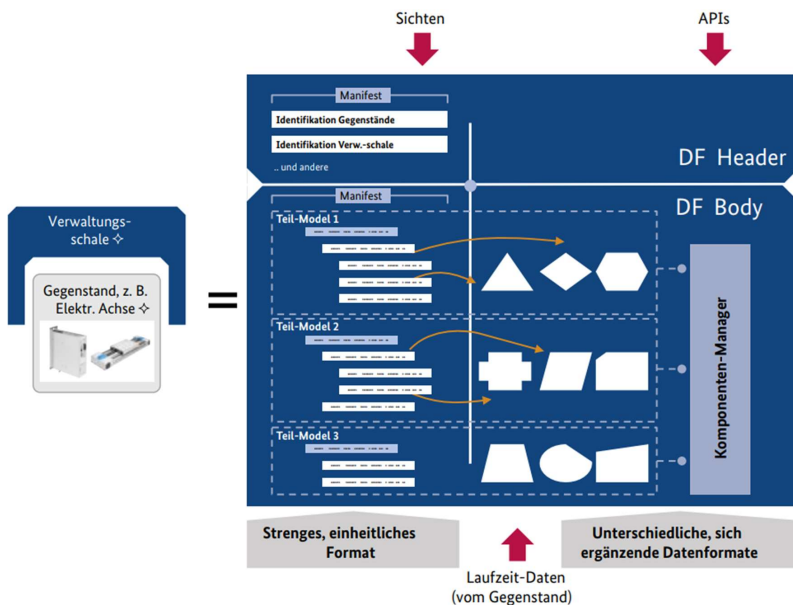


Abbildung 2-19: Grobstruktur der Verwaltungsschale aus [AdAB2016]

Abbildung 2-19 zeigt die Struktur der Verwaltungsschale. Wichtige Teilelemente der Struktur sind unter anderem der Header, welcher für eine Authentifizierung sorgt, sowie der Body, in welchem alle relevanten Merkmale für die definierten Sichten zusammengefasst sind. Die Verwaltungsschale kann dabei mit Live-Daten durch den realen Gegenstand versorgt werden und alle Daten über APIs nach Außen zur Verfügung stellen [AdAB2016].

2.6. Digitale Datenverwaltung im Engineering

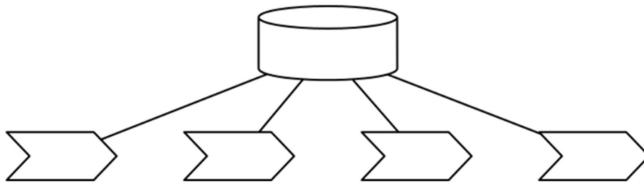
Bereits seit den 1980er Jahren gibt es Software, die Engineering-Artefakte verwalten soll [EiHS1991]. Viele der damals eingeführten Techniken sind immer noch im Einsatz, die

grundlegenden Konzepte der Datenverwaltung haben sich nur leicht weiterentwickelt. Hier soll ein Überblick über die Software gegeben werden, die im Engineering-Umfeld relevant ist und die im Rahmen eines Metadaten-Repository betrachtet wird.

2.6.1. PDM/PLM

PDM/PLM-Systeme sind als zentraler Datenspeicher gut geeignet, um auch verteilte Engineering-Prozesse über physische Grenzen hinweg zu orchestrieren. Neben dem klassischen Ansatz eines gemeinsamen Datenspeichers gibt es auch verteilte Peer-2-Peer (P2P)-Ansätze, um eine Zusammenarbeit zu ermöglichen. Abbildung 2-20 zeigt schematisch den Unterschied zwischen der klassischen und der P2P-Architektur. Die Schwierigkeit solcher verteilten Ansätze ist die Wahrung der Datenkonsistenz, die als eine der Hauptaufgaben eines PDM/PLM-Systems anzusehen ist [JoAH2017].

Schema einer Standard-PDM-Architektur



Schema einer Peer-to-Peer PDM Architektur

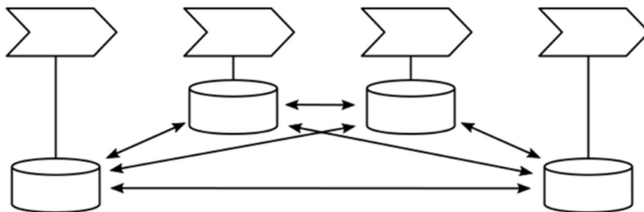


Abbildung 2-20: Verschiedene PDM-Architekturen nach [JoAH2017]

Jordan et al. [JoAH2017] haben aus diesem Grund ein Konzept entwickelt, welches auf die Fähigkeiten von Git⁷ als zentrales Repository. In diesem Konzept wird vor allem auf zwei Funktionen von Git als Software-Versionskontrolltool gesetzt. Zum einen können unkompliziert Neue Zweige einer Entwicklung angelegt werden, welche zu einem späteren Zeitpunkt wieder der Hauptentwicklung hinzugefügt werden können. Zum anderen können über cherry-picks sehr detailliert Teilentwicklungen zwischen diesen Strängen übertragen werden [JoAH2017].

⁷ <http://git-scm.com>

Es existieren verschiedene Lösungsansätze hinsichtlich der Datenverwaltung. So können Daten in einem monolithischen System gespeichert werden, oder die Daten werden föderiert in verschiedenen Systemen und sind erreichbar, von wo sie benötigt werden. Weiterhin stellt sich die Frage, ob eine Lösung „out of the box“ oder eine stark angepasste Lösung die bessere Wahl für ein Unternehmen ist [BIPM2022].

Neben der weit verbreiteten Verwendung des Begriffs PLM als IT-System, wird Product Lifecycle Management auch verwendet, um die Steuerung aller Prozesse im Produktentstehungsprozess zu beschreiben. Dies umfasst neben der Entscheidung wo und welche Daten verwaltet werden, auch Entscheidungen wie methodisch und prozessorientiert Anforderungen, Zulieferer, Themen der Arbeitsverwaltung oder allen After Sales Vorgängen umgegangen werden soll [EiSt2009]. Aufgrund der Komplexität und der Vielzahl der betrachteten Aspekte im Bereich PLM sind zur optimalen Planung und Einführung, sowie zur IT-Unterstützung auch methodische Hilfestellungen sinnvoll [Bitz2008].

2.6.2. Application Lifecycle Management (ALM)

Für Application Lifecycle Management existieren verschiedene Definitionen. Unter anderem umfasst es alle Tätigkeiten eines Software-Entwicklers neben dem Erstellen des Software-Codes. Unter anderem die Mitarbeit bei der Produktplanung, das Anforderungsmanagement, die Erstellung der Software-Architektur, das Erstellen einer Umgebung für die Software-Entwicklung, die Verwaltung des eigentlichen Codes, die Analyse von Code und weitere Tätigkeiten [ChFo2013].

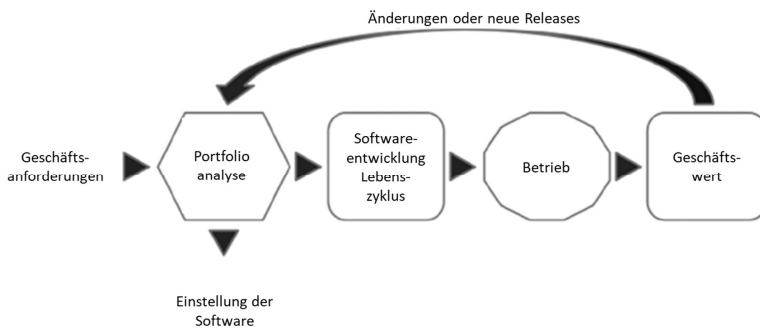


Abbildung 2-21: Der ALM-Prozess nach [Ross2014]

Abbildung 2-21 zeigt die Gesamtübersicht über den ALM-Prozess, welcher sich von der initialen Idee bis hin zur Abkündigung der Software zieht. Die einzelnen Tätigkeiten müssen so durchgeführt und gesteuert werden, dass ein optimales Endergebnis erzielt werden kann. Im ALM-Sinn kann dies insbesondere durch die Verwaltung der entsprechenden Prozessartefakte gesteuert werden. Hierzu gibt es von verschiedenen Herstellern komplette Tool-Ökosysteme, welche aus Verwaltungssoftware für Anforderungen, Aufgaben, Projektmanagement und Wissen,

sowie weiteren Elementen bestehen. Die Tools, z.B. von Atlassian, IBM oder Microsoft zeichnen sich weiterhin durch eine gute Integrierbarkeit hinzu Open Source Lösungen wie z.B. Git zur Code-Verwaltung aus. Die Tools verknüpfen die entsprechenden Elemente durch Links, um z.B. eine Anforderung oder einen Änderungswunsch mit einem Code-Schnipsel zu verknüpfen [Ross2014].

2.6.3. ERP

Enterprise Resource Planning (ERP)-Systeme bilden neben PLM-Systemen ein wichtiges Fundament der IT-Strukturen in (produzierenden) Unternehmen. ERP-Systeme werden auch als IT-Backbone der Firma bezeichnet. Sie sind historisch aus der Materialplanung und der Steuerung der Fertigungsressourcen (MRP bzw. MRP II) heraus gewachsen, umfassen mittlerweile allerdings das Management nahezu aller betrieblichen Ressourcen und kombinieren dies mit einer Steuerung der entsprechenden Prozesse [Kurb2013].

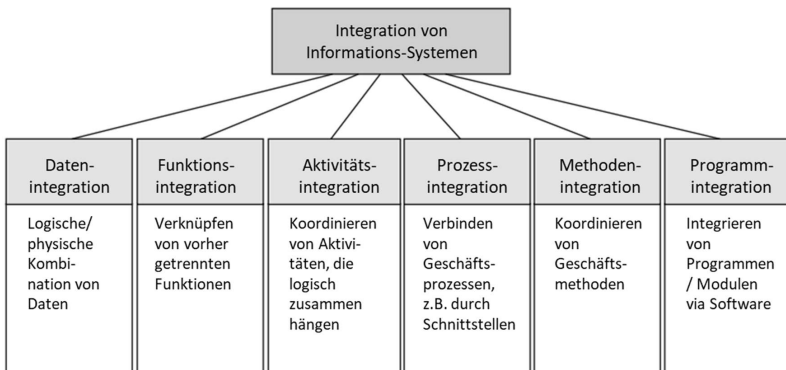


Abbildung 2-22: Integrationsperspektiven eines IT-Systems nach [Kurb2013]

Abbildung 2-22 zeigt die verschiedenen Perspektiven hinsichtlich Integrationen, welche in IT-Systemen abgebildet werden können. ERP-Systeme sind hierbei besonders herausfordernd, da sie aufgrund der Breite der Integration durch das gesamte Unternehmen hindurch die höchsten Anforderungen an alle Aspekte der Integration stellen. Dies zeigt sich an typischen Funktionsbereichen, welche durch das ERP unterstützt werden: Einkauf, Warenwirtschaft, Inventar, Produktionsplanung, Qualitätsmanagement, Marketing, Finanzen, Personal, um nur einige zu nennen. ERP-Systeme sind daher meistens modular aufgebaut [Kurb2013].

Eine weitere Herausforderung im Bereich ERP ist die Integration mit PLM-Systemen zur nahtlosen Datenweitergabe aus der Konstruktion in die Fertigung [LeLH2011] bzw. die parallele Entwicklung der Produktionssysteme und der Produkte [LeEZ2012]. Kern der Datenweitergabe sind Stücklisten, welche in einem PLM-System erzeugt und in einem ERP-System weitergenutzt werden. Während eine im PLM erzeugte Engineering-BOM die Bedürfnisse einer hierarchischen

Konstruktion erfüllt, muss sich eine Manufacturing-BOM in einem ERP System an den Anforderungen der Fertigung ausrichten [LeLH2011].

2.6.4. System Lifecycle Management (SysLM)

Der Wandel von klassischen Produkten hin zu cyber-physischen Systemen (CPS) oder Smarten Produkt-Service-Systemen erfordert auch eine Änderung und Erweiterung in den dafür vorgesehenen Datenverwaltungssystemen. Abbildung 2-23 zeigt das Umfeld, in welchem diese Entwicklung stattfindet. Getrieben von der Integration in Wertschöpfungsnetzwerke entwickeln sich Firmen zu immer digitaleren Einheiten, während auch die Kunden immer größere Anforderungen an SPSS oder CPS formulieren. In diesem Spannungsfeld vollzieht sich ein Wandel der Engineering-Prozesse hin zu einem Digitized Engineering, welches auf neuen Konzepten, wie u.a. dem System Lifecycle Management (SysLM) aufbaut [BiEF2018]. Für Sendler beschreibt der Begriff bereits den Wandel an sich, ob auch die IT-Werkzeuge sich speziell in bisherige PLM- und neuartige SysLM-Software aufspalten, oder diese neuen Konzepte in bestehender Software abgebildet werden sei nicht entscheidend [Send2013].

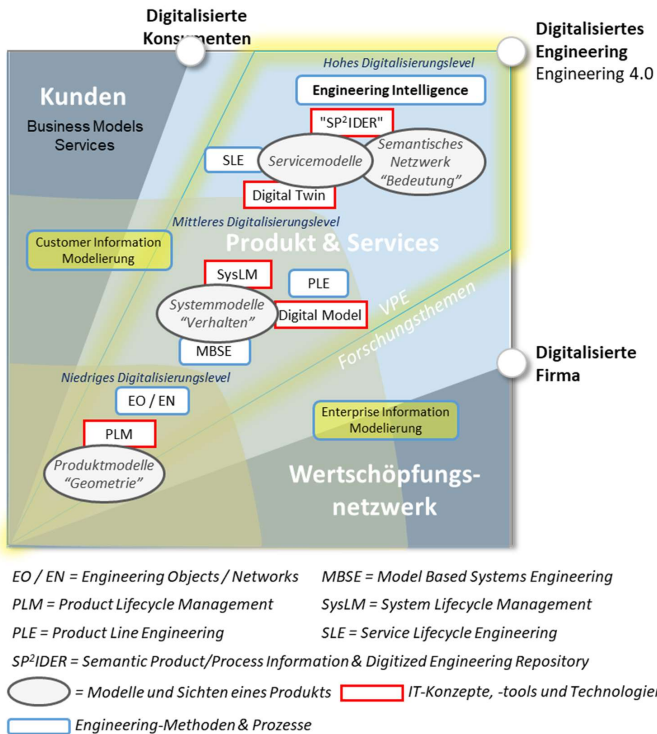


Abbildung 2-23: Digitized Engineering Framework nach [BiEF2018]

System Lifecycle Management laut Eigner ist die Erweiterung von PLM, so dass neben Dokumenten auch Modelle im System verwaltet werden können. Wesentliches Ziel ist die erhöhte Traceability durch die gleichzeitige Verwaltung in einem IT-Werkzeug und die Möglichkeit der direkten Vernetzung von Funktionen auf Baugruppe und Bauteile, so dass im Änderungsfall alle Auswirkungen nachvollziehbar sind. Die funktionale Beschreibung dient hierbei als verknüpfendes Element zwischen den Anforderungen auf der einen und der Stückliste auf der anderen Seite [Eign2013], [Giltz2014]. Ein initialer Lösungsansatz wird in [GiEi2013] demonstriert. Ein mögliches Datenschema für eine SysLM-Lösung wird in [EiGZ2012] gezeigt. Die immer weiter zunehmenden Herausforderungen, vor denen das Engineering steht und welche auch unter dem Schlagwort Engineering 4.0 zusammengefasst werden [AbHe2016], [GaSM2019], bringt eine nochmalige Erweiterung des System Lifecycle Management Ansatzes auch um die Use-Case Phase mit sich, welche initial nicht betrachtet wurde [Eign2021].

Die so realisierten Funktionserweiterungen von SysLM kann z.B. genutzt werden, um mithilfe einer Funktionsbeschreibung und modellierten Use-Cases in Kombination mit der verknüpften

Stückliste eine Abschätzung der Umweltauswirkungen eines Entwicklungsprojekts abzuschätzen und zu verbessern [EiDA2014], [EISA2014].

Es existiert auch ein weitergehender Ansatz eines SysLM-Backbones, welcher auf dem modellbasierten Grundkonzept beruht und dieses nutzt, um SysLM mit einen Linked-Data Ansatz zu kombinieren mit dem Ziel der Nachvollziehbarkeit von Engineering-Artefakten und dem Aufbau eines Digital Threads. An dieser Stelle wird aus dem SysLM-Konzept ein hybrider Ansatz, welcher ähnlich einer Integrationsplattform funktioniert [Sind2022].

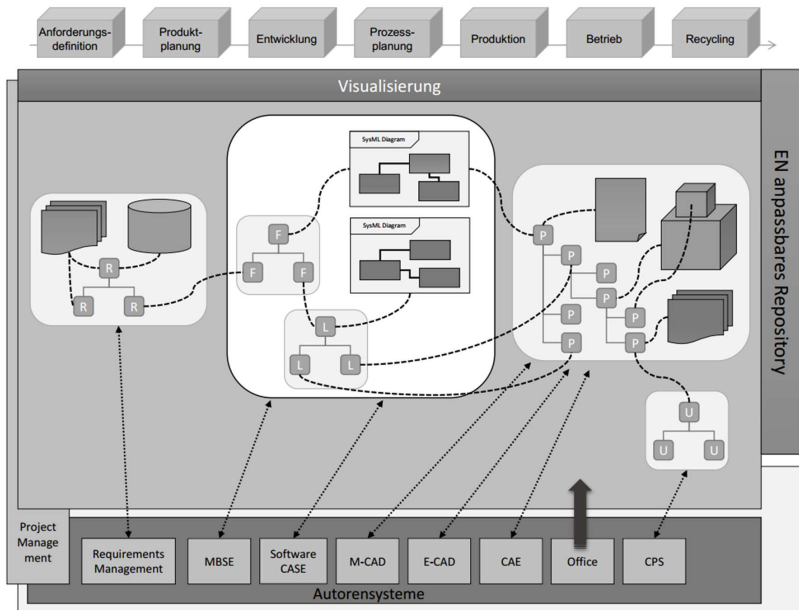


Abbildung 2-24: SysLM-Repository in maximaler Ausbaustufe aus [Sind2022]

Abbildung 2-24 zeigt hierbei ein Beispiel für einen hybriden Ansatz in seiner maximalen Ausbaustufe, bei dem die Modelle vollständig im SysLM-Backbone verwaltet werden und eine Referenzierung auf Engineering-Artefakte aus anderen Datenverwaltungs- und Autorensystemen vollständig möglich ist [Sind2022].

2.6.5. BaSys 4.0

Im Bereich der Automatisierungstechnik wurde mit dem Referenzarchitekturmodell RAMI 4.0 eine datentechnische Beschreibung des Lebenslaufs von technischen Gegenständen geschaffen [DIN2016].

Eine der größten Schwierigkeiten im Bereich der Automatisierungstechnik ist die Kommunikation zwischen verschiedenen Assets auf der physischen Ebene, der MES-Ebene, sowie der

übergeordneten ERP-Ebene. Viele Anlagen können zwar kommunizieren, allerdings sind hierfür verschiedene Standards notwendig, an manchen Stellen kann eine Kommunikation auch nur in der horizontalen oder vertikalen Ebene funktionieren. Zur Lösung dieser Probleme wurde die Middleware BaSys4.0 bzw. deren softwaretechnische Entsprechung BaSyx erschaffen [KaHD2021].

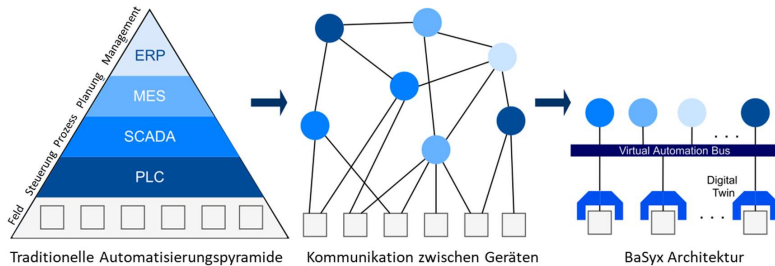


Abbildung 2-25: Automatisierungspyramide und BaSyx-Architektur nach [KaHD2021]

Abbildung 2-25 zeigt den Übergang der traditionellen Automatisierungspyramide bis hin zu einer Bus-Struktur durch den Virtual Automation Bus, welcher Kommunikation ermöglicht. Während klassischerweise Geräte auf der Feldebene nur mit der direkt übergeordneten Ebene kommunizieren können, werden sie nun durch eine Verwaltungsschale bzw. Asset Administration Shell (AAS) gekapselt, welche nach außen über standardisierte Kommunikationsprotokolle angesprochen werden kann. BaSys stellt hierfür das entsprechende Protokoll [KaHD2021].

Durch diese Protokolle erreicht BaSys4.0 insbesondere Flexibilität und Wandelbarkeit durch eine dynamische Handhabung von Komponenten. Einzelne Elemente können im laufenden Betrieb hinzugefügt, ausgetauscht oder entfernt werden [GrWE2018].

2.6.6. GAIA-X

GAIA-X ist ein europäisches Projekt für einen gemeinsamen Datenraum, in welchem alle Nutzer zum gegenseitigen Vorteil Daten speichern, austauschen und darauf aufbauend neue Geschäftsmodelle entwickeln können. Zu den Kernelementen von GAIA-X gehören zum einen die europäische Cloudsouveränität, um die Daten im eigenen Hoheitsgebiete und mit eigener Infrastruktur abspeichern zu können und zum anderen ein Konzept, welches die sichere Weitergabe und gemeinsame Nutzung von Daten entsprechend des International Data Spaces Referenzarchitekturmodells [OtST2019] gewährleistet [BrFR2021].

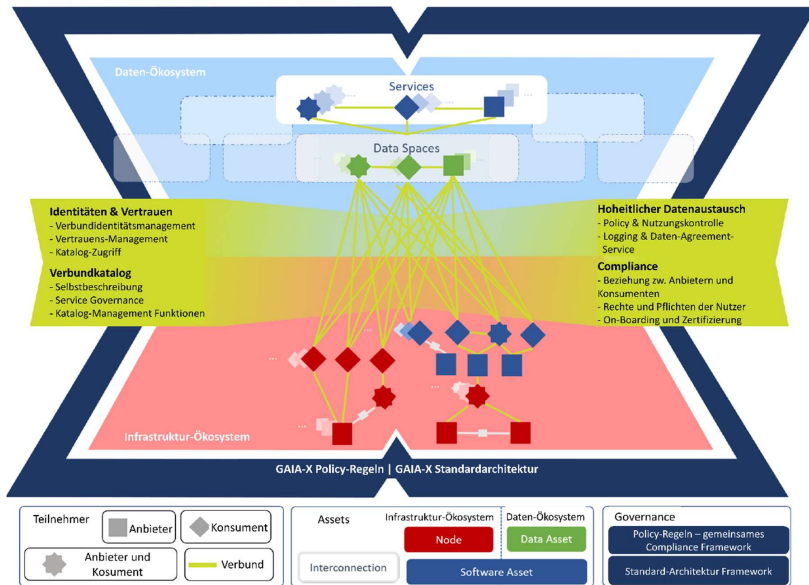


Abbildung 2-26: Eine Übersicht über die GAIA-X Architektur nach [Gaiä2021]

Abbildung 2-26 zeigt eine Übersicht über das GAIA-X Architektursystem. Es besteht aus den zwei Teilen des Infrastrukturokosystems, sowie des Daten-Ökosystems. Ersteres ist darauf ausgelegt, dass Provider eine Compliance-gerechte Datenspeicher-Struktur schaffen, welche dann genutzt werden kann, um im zweiten Teil, des Daten-Ökosystems Daten in dieser Infrastruktur abzulegen und smarte Services zu erschaffen, welche diese Daten nutzen [BrFR2021].

2.7. Wissen

Wissen als Grundlage unserer Wertschöpfung und Wohlstandes wird immer wichtiger. Bereits 2011 wurde von North postuliert, dass wir uns am Beginn eines-Zyklus befinden, in welchem Arbeit und Kapital als ausschlaggebende Faktoren für den Wohlstand unserer Gesellschaft durch Wissen abgelöst werden [Nort2011]. Um diesen Umstand bewusst zu nutzen, ist es notwendig sich anzuschauen in welchem Zusammenhang Wissen mit weiteren Begriffen steht, wie Wissen geschaffen und genutzt wird. Diesen Zusammenhang zeigt exemplarisch die Wissenstreppe von North in Abbildung 2-27. Wissen entsteht aus Daten und Informationen, ist allerdings auch immer im Zusammenhang mit einer Person zu verstehen, d.h. ohne Personenbezug kann kein Wissen existieren [Nort2011][Völk2006].

Zeichen sind die unterste Stufe der Wissenstreppe. Wenn mehrere Zeichen in einen Zusammenhang gebracht werden, dann bilden diese Daten, welche allerdings ohne einen Kontext noch nicht interpretierbar sind [Völk2006]. Z.B. ist „4“ nur ein Zeichen, wenn mehrere kombiniert

werden, z.B. „Gebäude 44“ spricht man von Daten, allerdings erst durch einen Kontext werden diese Daten relevant, z.B. „Der Lehrstuhl VPE befindet sich in Gebäude 44.“ Wissen wäre die individuelle Einordnung, warum diese Information relevant ist, z.B. durch eine Person, welche das Angebot des Lehrstuhls kennt und nutzen will und deswegen aus dieser Information Wissen schöpfen kann.

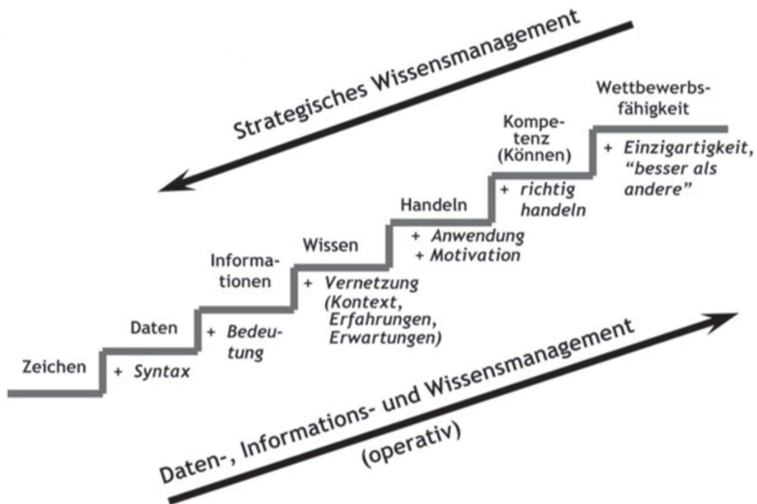


Abbildung 2-27: Die Wissenstreppe von North, nach [Nort2011]

Im Gegensatz zu diesem stufenweisen Prinzip, was auch zu einer Trennung und unterschiedlichen Betrachtung der verschiedenen Ebenen führt, schlägt Probst vor, dass man eher von einem kontinuierlichen Prozess, bzw. Qualitätswandel zwischen Daten und Wissen den entgegengesetzten Polen spricht [Prob2012]. Im Gegensatz zu westlich geprägten Ansätzen beschreiben Nonaka und Takeuchi Wissen als „Konsequenz einer bestimmten Einstellung“ [NoTa1997] und zeigen auf, dass Vorstellungen, Engagement und Handeln im Zusammenhang mit Wissen stehen, da Wissen immer zweckgerichtet ist [NoTa1997].

Die unterschiedlichen Sichtweisen auf Wissen fasst Turki zusammen und spricht von Wissen als vernetzter Information im Sinne Norths, Wissen als Entscheidungs-, Problemlösungs- und Handlungsfähigkeit im Sinne Probsts, Wissen als Rohstoff für Information im Sinne Kuhlens, sowie Wissen als plausible begründete Aussage [Turk2014].

2.7.1. Explizites/Implizites Wissen

Häufig wird im Zusammenhang mit Wissen unterschieden, ob dieses Wissen explizit oder implizit vorliegt. Abbildung 2-28 zeigt typische Merkmale von explizitem und implizitem Wissen [Turk2014].

| Implizites Wissen | Explizites Wissen |
|--|--------------------------------------|
| Subjektiv | Objektiv |
| Erfahrungswissen (Körper) | Verstandeswissen (Geist) |
| Gleichzeitiges Wissen (hier und jetzt) | Sequenzielles Wissen (da und damals) |
| Analoges Wissen (Praxis) | Digitales Wissen (Theorie) |
| Unbewusste Aneignung | Bewusste Aneignung |
| Intuitiv | Wenig kontextgebunden |
| Nicht oder nur schwer artikulierbar | Artikulierbar |
| Personengebunden | Unabhängig vom Produzent |

Abbildung 2-28: Implizites und Explizites Wissen aus [Turk2014]

Implizites Wissen ist dabei zwangsweise persönliches Wissen, welches für andere Personen unzugänglich ist, während explizites Wissen auch anderen Personen zugänglich gemacht werden kann, allerdings nicht zwingend zugänglich sein muss. Entsprechend dem Zugang kann man auch unterscheiden, ob Wissen individuell oder kollektiv verfügbar ist, wobei vor allem explizites Wissen hierzu genutzt werden kann, z.B. Regeln, Verfahrensrichtlinien, kristallisiertes Wissen oder organisatorische Routinen [Turk2014].

2.7.2. Schaffung und Umwandlung von Wissen

Gerade die Tatsache, dass Wissen oftmals nicht explizit und abrufbar vorliegt, sondern auf einer persönlichen Ebene und häufig implizit vorliegt, macht es Unternehmen sehr schwer, eine organisationale Wissensbasis zu schaffen und diese zu nutzen, da hierzu das Wissen in expliziter und transparenter Form vorliegen muss, auf welches die Organisation zugreifen kann, vgl. Abbildung 2-29 [Völk2006].

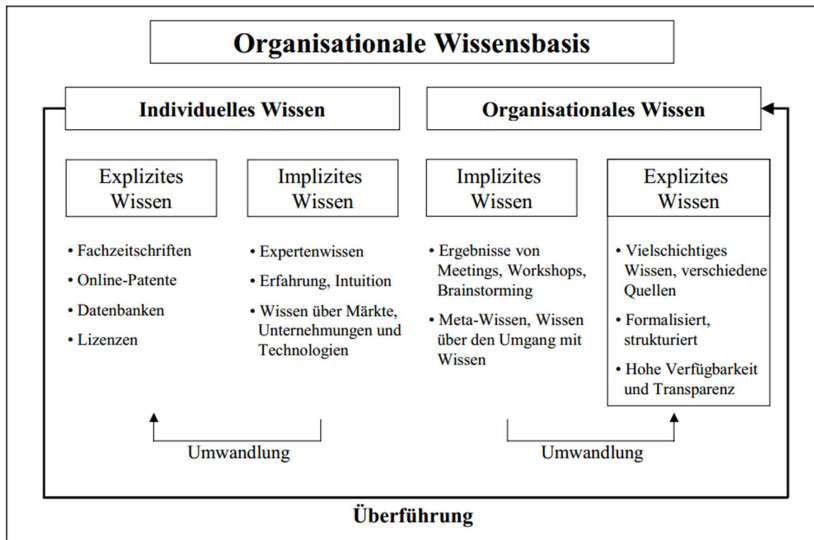


Abbildung 2-29: Organisationalen Wissensbasis aus [Völk2006]

Mit der letzten Überarbeitung der DIN EN ISO 9001 zum Thema Qualitätsmanagementsysteme [DIN 2015] wurde das „Wissen der Organisation“ sogar als explizite Ressource aufgenommen, welche in der Firma sichergestellt werden muss, um die organisationalen Prozesse zu beherrschen [DIN 2015]. Die Wichtigkeit des Wissens steigt an, Firmen müssen sich damit auseinandersetzen, wie sie das Wissensmanagement in ihrer Organisation verankern können [GrBU2015]. Mögliche Optionen zur Verwaltung des Wissens sind vielfältig und erfüllen unterschiedliche Anforderungen, Anwendungsszenarien und unterschiedliche Kostenrahmen und müssen daher individuell von Firmen auf ihre jeweilige Tauglichkeit untersucht werden [GrKL2014].

2.8. Zusammenfassung

Der grundlegende Wandel der Produkte hin zu interdisziplinären, intelligenten und vernetzten Smart Product Service Systems hat beträchtliche Auswirkungen auf die gesamte Wertschöpfungskette bzw. auf Wertschöpfungsnetzwerke. Verschiedene Disziplinen haben unterschiedliche Datenstrukturen und Prozesse, verwenden unterschiedliche IT-Werkzeuge, die jeweils auf die speziellen Domänenanforderungen zugeschnitten sind. Gleichzeitig werden Prozesse interdisziplinärer und müssen digitale Objekte aus verschiedenen Domänen und auch Lebenszyklusphasen miteinander in Beziehung setzen, um eine informierte Entscheidung zu treffen. Nicht zuletzt erfordert dies ein enormes prozessuales und Fachwissen aller Beteiligten, welches einerseits erkannt und aufbereitet werden muss, andererseits auch wieder von unterschiedlichen Akteuren identifiziert und verwendet werden muss.

3. Stand der Forschung und Technik

Die wichtigsten Themen dieser Arbeit, die Beschreibung von Daten, sowie die semantische Anreicherung dieser und die daraus folgenden Nutzungsszenarien sind seit langem intensiv im Bereich der Forschung, aber auch in verschiedenen Unternehmen breit diskutiert. Dieses Kapitel soll einen Überblick über die wichtigsten Entwicklungen geben, die aktuell entweder erforscht oder in der Industrie umgesetzt werden. Als inhaltliche Leitposten werden dabei die Kategorien „Beschreibung von Daten“, sowie „Nutzung semantisch aufgewerteter Daten“ gewählt, welche wesentliche Pfeiler eines Knowledge Graph basierten Assistenzsystems zur Datenanalyse-Unterstützung für das Engineering Smarter Produkte darstellen. Vorhandene Prozesse, Methoden, IT-Werkzeuge und Technologien wurden geclustert und in diese Kategorien einsortiert.

3.1. Beschreibung von Daten

Um Daten besser analysieren und im betrieblichen Umfeld nutzen zu können, ist es essenziell die Bedeutung der Daten zu erfassen und für Nutzer, aber auch für Maschinen sicht- und auslesbar zu gestalten. Dies ist noch umfassender als die reine Datenmodellierung, bei welcher Experten, in der Regel Informatiker, die Strukturen von Datenbanken erstellen oder verändern und die Datenflüsse durch Anwendungen und Prozesse hindurch festlegen. Auch wenn dies ein wichtiger erster Schritt ist, ist dieser meistens dem Anwender verborgen, da er weder über den Code noch Zugänge dazu, außerdem meist nicht über z.B. Entity-Relationship-Diagramme verfügt. Um dem Nutzer trotzdem die Bedeutung von Daten sichtbar zu machen, ist eben eine Beschreibung von Daten sinnvoll. Hierzu bedarf es einer Methodik, um die Bedeutung von Daten sichtbar zu machen und darauf aufbauend Zusammenhänge und Ähnlichkeiten zu visualisieren. Dieses Unterkapitel gibt zuerst einen Überblick über grundlegendes Wissen im Bereich Semantik und Ontologien, bevor die Themen Mapping, Graphen und Knowledge Graphen vorgestellt werden.

3.1.1. Semantik

Semantik beschreibt die Bedeutung von syntaktischen Ausdrücken und ist eng mit der Pragmatik verbunden, welche sich damit befasst, warum und wozu jemand diesen Ausdruck einsetzt [Deng2012]. Als Verbindung zwischen Symbolen bzw. Daten und ihrer Bedeutung steht die Semantik im Gegensatz zur Syntax, welche die grammatikalische Kombinierbarkeit betrachtet, allerdings keine Aussage über Bedeutungen trifft [IEEE1990]. Eine vollständige Betrachtung umfasst die Syntax, das heißt die Kombinierbarkeit, die Semantik, die Bedeutung von Ausdrücken und die Pragmatik, was meint, dass man Bezeichnungen nicht als Selbstzweck verwendet, sondern sich diese ihre Gültigkeit in einem bestimmten Kontext, durch einen Sender und einen Empfänger haben [BlPe2006].

Um Semantik im Kontext von Informationstechnologie sinnvoll einzusetzen, ist es wichtig, dass ein gewisser Grad an semantischer Reichhaltigkeit durch das genutzte Modell erreicht wird. Abbildung 3-1 verdeutlicht dies anhand der semantischen Treppe.

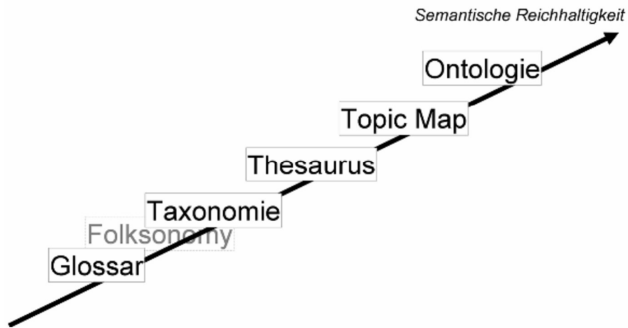


Abbildung 3-1: Semantische Treppe aus [BlPe2006]

Die semantische Treppe zeigt die Entwicklung der semantischen Reichhaltigkeit. Während auf den unteren Stufen mit einem Glossar Begriffe unstrukturiert gesammelt werden, sind im mittleren Bereich angeordnete Thesauri, welche einen Zusammenhang zwischen ähnlichen Begriffen erläutern und so verschiedene Begriffe eines Glossars miteinander in Beziehung setzen. Eine weitere semantische Aufwertung ist eine Topic Map, welche verschiedene Begriffe zu Themen zuordnet und so Sachzusammenhänge nicht nur zwischen ähnlichen, aber thematisch verwandten Begriffen aufzeigt. An der Spitze der semantischen Treppe steht die Ontologie, welche anhand von Klassen und Relationen Objekte strukturiert und miteinander verknüpft [BlPe2006].

Semantiken bilden eine wichtige Grundlage für Interoperabilität, wenn Objekte zwischen mehreren Systemen ausgetauscht werden und deren Bedeutung verstanden wird, so dass direkt eine korrekte Klassifizierung möglich ist. Insbesondere im Semantic Web wurden dafür verschiedene Standards festgelegt, unter anderem das Document Object Model (DOM) für die Strukturierung von Dokumenten oder das Resource Description Framework (RDF) zur Beschreibung von Objekten [Gali2006].

3.1.2. Ontologien

Seinen begrifflichen Ursprung hat das Wort Ontologie im Griechischen und bedeutet die Lehre vom Seienden. Es sollen Strukturen und Unterscheidungen von Wirklichem beschrieben werden. Dieses Konzept wurde in der Neuzeit übernommen, um im Zuge von Informationsrepräsentation Informationen und Objekte zu beschreiben [JaKR2017].

Ontologien sind eng verbunden mit der Kontextualisierung von Informationen. Objekte existieren nicht einfach so, sondern stehen in einem bestimmten Zusammenhang mit anderen Objekten und Konzepten. Dieses häufig vereinfachte Konzept, welches einen Teil der Realität beschreibt, kann formal mit einer Ontologie beschrieben werden. Eine Ontologie ist daher die explizite Spezifikation einer Konzeptualisierung [Grub1995].

Zu den Hauptbestandteilen einer Ontologie gehören verschiedene Elemente wie Klassen, Attribute, Eigenschaften, Relationen und Bedingungen. Klassen sind hierbei abstrakte Gruppen, in die Objekte einsortiert werden können. Hierbei kann eine Klasse sowohl Objekte, als auch weitere Klassen umfassen. Zur weiteren Beschreibung der Objekte in einer Klasse dienen die Attribute oder Eigenschaften, welche ein Objekt dieser Klasse besitzt. Eine Relation beschreibt die Verbindung, in der Klassen oder Objekte zueinander stehen. Z.B. die Relation, dass eine Klasse die Unterklasse einer Oberklasse ist. Zuletzt sind Bedingungen zu nennen, welche z.B. aufgrund der Zuordnung zu einer Klasse die Zuordnung zu einer weiteren Klasse ausschließen können [JaKR2017].

Beim Erstellen einer Ontologie sollten laut Gruber [Grub1995] folgende Regeln beachtet werden:

- Klarheit – Die Ontologie sollte in ihrer Bedeutung klar und objektiv verfasst sein
- Kohärenz – Es sollten in der Ontologie keine sich selbst widersprechenden Definitionen genutzt werden
- Erweiterbarkeit – Ontologien beziehen sich auf spezielle Teilgebiete, insofern kann bei der Erstellung höchstwahrscheinlich nicht jeder mögliche Anwendungsfall berücksichtigt werden. Sollten sich weitere Fälle ergeben, sollte die Ontologie erweiterbar sein.
- Minimaler Kodier-Bias – Es sollte zur Nutzung der Ontologie kein spezielles Wissen hinsichtlich der Symbolik erforderlich sein
- Minimale ontologische Verpflichtung – Um eine Nutzung sicherzustellen, sollte die Ontologie so einfach wie möglich nutzbar sein, indem eine Beschreibung und das Befolgen der Regeln nicht zu kompliziert sein sollen.

Ontologien beschreiben immer nur einen Teil der Wirklichkeit. Eine Ontologie, welche sämtliche Aspekte der Realität beschreiben sollte, wäre viel zu komplex. Dementsprechend sind Ontologien klassifizierbar anhand ihres Zweckes und ihres Umfangs [JaKR2017].

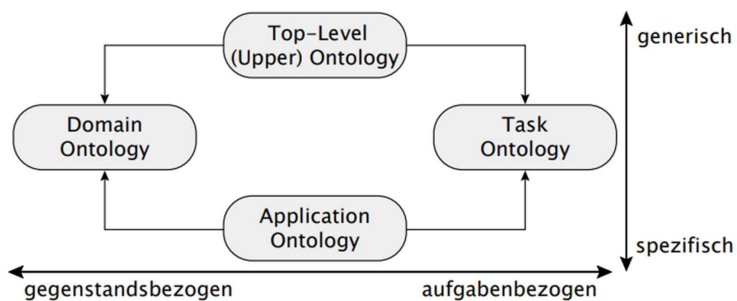


Abbildung 3-2: Klassifizierungsansatz für Ontologien aus [JaKR2017]

Abbildung 3-2 zeigt eine mögliche Klassifizierung von Ontologien. Top-Level Ontologien versuchen dabei sehr allgemeine Ontologien zu erschaffen, welche eine möglichst große

Gültigkeit haben. Eine Domain Ontologie beschreibt hingegen nur Begriffe aus einem bestimmten Teilbereich der Realität. Eine Task Ontologie tut das gleiche mit Tätigkeiten. Kombiniert man die letzteren beiden, so kann dies eine Application Ontologie ergeben, welche Tätigkeiten im Kontext einer bestimmten Domäne beschreibt [JaKR2017].

Eine weitere Kategorisierung ist die Unterscheidung in Lightweight und Heavyweight Ontologien. Erstere bestehen vor allem aus Wörterbüchern, Glossaren oder Thesauri, welche der Kategorisierung dienen, während letztere zusätzlich sehr formale Bedingungen an die Objekte und ihre Beschreibung stellen und dadurch auch besser maschinenlesbar ausgewertet und gedeutet werden können [GóFC2004].

Generell erfordert die maschinelle Auswertbarkeit von Ontologien, dass diese auch formal in einer dokumentierten Sprache beschrieben werden müssen. Hierzu sind vor allem im Bereich des Semantic Web eine Vielzahl von Sprachen wie SHOE, XOL, RDF(S), OWL und weitere entstanden [GóFC2004]. Neben den Sprachen gibt es auch entsprechende Editoren, mit denen der Anwender bei der Beschreibung seiner Ontologie unterstützt wird, wie z.B. der Editor Protégé, mit dem Ontologien in der Web Ontology Language (OWL) erstellt werden können [Horr2011].

Die Anzahl an Ontologien für bestimmte Einsatzzwecke ist groß, Beispiele finden sich u.a. in [BaEF2020], [KüKS2018], [HiKK2020], [HiTB2018], [StRR2018], [ELER2020]. Interessant ist vor allem die Ontologie von Barth et al. [BaEF2020], da hier auch Datenquellen direkt Teil der Ontologie sind, wie in Abbildung 3-3 ersichtlich. Auch können Ontologien eine Hilfestellung bei der Datenintegration sein, wie in [WoHS2015] beschrieben.

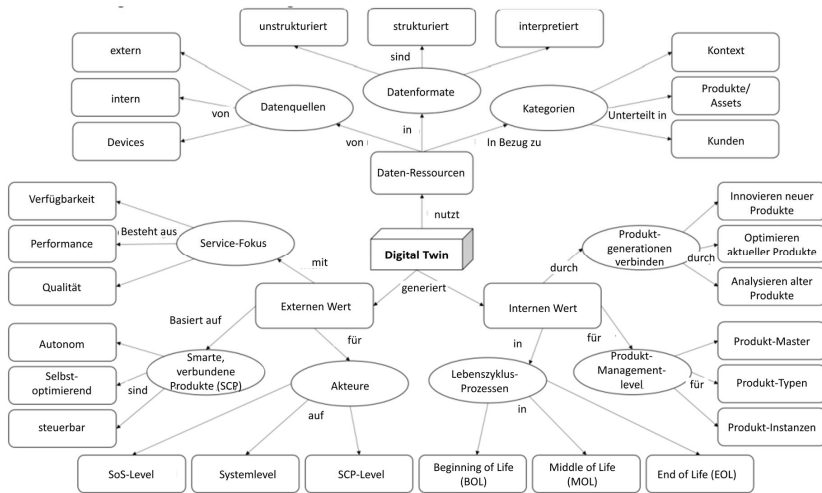


Abbildung 3-3: Digital Twin Ontologie nach [BaEF2020]

Ontologien zur Informationsbeschaffung haben oftmals den Nachteil der großen Modellierungsdauer, bis sie einsetzbar sind, weswegen auch Ansätze entwickelt wurden, um teilautomatisiert Wissen zu extrahieren und in Form einer Ontologie abzuspeichern. Nach einer initialen, manuellen Erstellung der ersten Ontologie können weitere Texte und Dokumente automatisiert gescannt und mit den beschriebenen Klassen, Attributen und Verbindungen verglichen und im Erfolgsfall zugeordnet werden [LiRR2007].

Neben dem reinen Vorhandensein von Ontologien ist es auch wichtig, dass diese strukturiert aufgebaut werden und entsprechende Methoden für diese Tätigkeit bereitstehen [HiTC2018], [Milk1998].

3.1.3. Dublin Core

Der Dublin Core ist eine Initiative⁸, die sich zum Ziel gesetzt hat, ein möglichst einfaches Set an Metadaten zu entwickeln, welches in der Lage ist, im Internet erreichbare Dokumente identifizierbar zu beschreiben [ISO2017]. Ausgehend von diesen Kern an Metadaten, welcher auch in Tabelle 3-1 dargestellt ist, wurde eine Erweiterung geschaffen, die weitere Metadaten eines Dokuments beschreibt [ISO2019].

⁸ <http://dublincore.org>

Tabelle 3-1: Die 15 Core-Elemente des Dublin-Core nach [Alas2009]

| Element | Definition |
|--------------|--|
| Titel | Der Name der Ressource |
| Inhalt | Das Thema des Inhalts der Ressource |
| Beschreibung | Ein Auszug des Inhalts der Ressource |
| Typ | Die Art des Inhalts der Ressource |
| Quelle | Eine Referenz auf die Ressource, aus der die hier vorliegende Ressource abgeleitet ist |
| Beziehungen | Beziehungen zu verwandten Ressourcen |
| Abdeckung | Die Breite bzw. der Fokus des Inhalts der Ressource |
| Ersteller | Die Entität, welche die Ressource zur erstellt hat |
| Herausgeber | Die Entität, welche die Ressource zur Verfügung stellt |
| Beitragende | Entitäten, welche zum Inhalt der Ressource beitragen |
| Rechte | Informationen über Rechte an der Ressource |
| Daten | Daten, welche im Lebenszyklus der Ressource anfallen |
| Formate | Die physische oder digitale Manifestation der Ressource |
| ID | Eine eindeutige Referenz auf die Ressource im jeweiligen Kontext |
| Sprache | Sprache(n) des Inhalts der Ressource |

Die 15 Elemente des Dublin Core Metadata Schema lassen sich in mehrere Bereiche auf-teilen. Informationen über das Dokument selbst, Informationen über Personen, die an der Erstellung des Dokuments beteiligt sind [ISO2017].

3.1.4. Standards im Bereich Datenbeschreibung

Es existieren verschiedene Definitionen für Standards, z.B. die „Vereinheitlichung von Gütern, von Produktionsmethoden oder anderem“ [Wies1990] oder Spezifikationen, die weite Verwendung finden [Pico2001], im Sinne einer IT-Integration aber insbesondere die Voraussetzung für die Vernetzung betrieblicher Anwendungssysteme, indem sie einheitliche Regeln definieren, welche vor einer Kommunikation bekannt sind und die diese Kommunikation steuern [Müll2005]. Gerade der Datenaustausch wäre ohne Standards nur erschwert möglich, da Daten an jeder Schnittstelle mit großem Aufwand in ein neues Format übersetzt werden müssten und bei einer Anpassung die Schnittstelle nicht mehr funktionieren würde, so wie bisher. Dementsprechend werden fehlende Standards auch als häufiges Problem bei der Digitalisierung genannt, insbesondere in Industrieunternehmen [DeER2016], [Enge2017].

Beispiele für diese Standards sind z.B. ECLASS⁹, Schema.org¹⁰ und weitere. ECLASS wird vor allem in den Bereichen Vertrieb und Einkauf genutzt, um Produkte zu klassifizieren und sie so schneller

⁹ ECLASS: <https://eclass.eu/>

¹⁰ Schema.org: <https://schema.org>

auffindbar zu machen. Weiterhin dient der Standard als Semantik für das Internet of Things [IWCO2019].



Abbildung 3-4: ECLASS Spezifikation eines Elektromotors aus eclass.eu

Bei der Nutzung des ECLASS Standards werden alle Produkte in Kategorien einsortiert, welche sich in einem immer weiter verfeinerten Baum befinden. Den unterschiedlichen Kategorien, wie oben 44080909 Elektromotor (KFZ, Sonstige, nicht spezifiziert) Abbildung 3-4 sind dann Attribute zugeordnet, die dieses Produkt weiter beschreiben.

Schema.org wurde im Jahr 2011 von verschiedenen Suchmaschinenbetreibern gegründet, mit dem Ziel der Vereinheitlichung semantischer Informationen zu unterschiedlichen Themen, welche häufig Ziel von Suchanfragen sind. Diese dienen z.B. als Informationsquelle für die Infobox, welche verschiedene Suchmaschinen direkt einblenden, können aber auch genutzt werden, um Informationen in Emails zu verstehen und z.B. eine Benachrichtigung zu triggern [GuBM2016], [Pate2014].

3.1.5. Mapping

Neben der reinen semantischen Beschreibung einzelner Datenelemente, spielt das Mapping eine wichtige Rolle, indem es gleiche Dinge miteinander verbindet. Dies kann entweder durch Menschen, idealerweise Domänenexperten oder durch Algorithmen stattfinden. Hierbei können verschiedene Techniken und Methoden angewendet werden, um die Gleichheit von verschiedenen Dingen festzustellen. Z.B. kann eine linguistische Analyse eingesetzt werden, bei der Ähnlichkeiten in Texten verglichen werden, eine strukturelle Analyse, bei der die Form von z.B. Datenstrukturen und abgeleiteten Bäumen verglichen wird oder eine Analyse bezogen auf Randbedingungen wie z.B. Datentypen oder Wertfelder, welche die Daten annehmen können. Darüber hinaus können für das Mapping nicht nur die Informationen selbst, sondern auch weitere Informationen aus anderen Quellen eine Rolle spielen und betrachtet werden [J.PE2001].

Mapping von Daten

Das Mappen von Daten, sprich das Verstehen und Sichtbarmachen von Beziehungen zwischen unterschiedlichen Objekten und Daten ist eine Disziplin, welche bereits seit Jahren existiert. Ziel ist es, eine möglichst redundanzfreie IT-Architektur zu schaffen, in der Zusammenhänge klar werden, in welcher die Daten dort gespeichert sind, wo sie anfallen und [LeNa2007].

Aufgrund der Vielzahl der Engineering Applikationen und der Vielzahl an Daten, welche in verschiedenen Programmen bearbeitet und gespeichert werden, spielt das Mapping von Daten und Parametern eine große Rolle. Hier gilt es eine integrierte Enterprise Architecture Integration Infrastruktur aufzubauen, in welcher Mapping-Prozesse ablaufen können, wie in Abbildung 3-5 gezeigt [WaXu2008].

Mapping kann sehr einfach über standardisierte Austauschformate stattfinden, wobei dann sichergestellt werden muss, dass diese Austauschformate auch sämtliche Bedürfnisse an Informationen erfüllen. Es erfordert domänenspezifische Mapping-Tabellen, wie u.a. in diesem Paper für den Austausch von BIM (Building Information Modeling) gezeigt [ChDG2021].

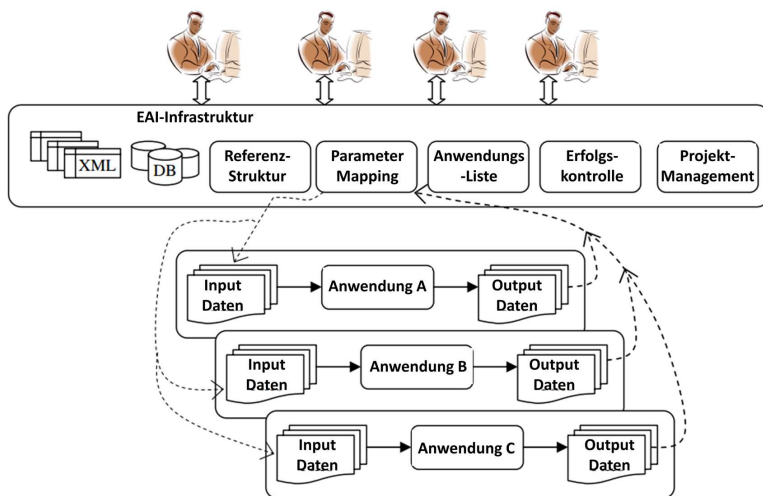


Abbildung 3-5: Enterprise Architecture Integration Infrastruktur nach [WaXu2008]

Zum Aufbau des Mapping-Systems FiLiPo (Finding Linkage Points) wurden verschiedene Methoden getestet, welche gleiche Daten in unterschiedlichen Datenbanken finden sollen. Am Beispiel einer internen Wissensbasis für Literatur, welche mit diversen externen Schnittstellen abgeglichen wird, wurden bis zu 48 Methoden identifiziert, welche Zeichenketten miteinander vergleichen. Diese Methoden lassen sich in drei grundlegende Kategorien einteilen: Methoden, welche eine 1:1 Entsprechung finden, Methoden, die die Ähnlichkeit anhand der notwendigen

Operationen zählen, die benötigt werden, um die eine Zeichenkette in die andere umzuwandeln, sowie Token-basierte Methoden [ZeSc2020].

Bei der Anwendung von FiLiPo zeigen sich im Detail die Probleme des Mappings von Daten. So können z.B. Strukturen so unterschiedlich sein, dass manche Methoden der Ähnlichkeitssuche nicht funktionieren können, wie z.B. ein direkter Vergleich, während eine Ersetzen-Methoden wie z.B. die Levenshtein-Distanz zu ungenau ist. Um mit diesen Unsicherheiten umzugehen ist es immer erforderlich, dass manuelle Anpassungen vorgenommen werden und je nach Anwendungsfall die Werte für eine positive Beurteilung der Ähnlichkeit angepasst werden können, z.B. durch einstellbare Koeffizienten [ZeSc2020].

Mapping von Wissen

Kernaufgabe des Mappings von Wissen ist es, dieses so aufzubereiten, dass bestimmte Bereiche sichtbar werden, die anschließend möglichst transparent und nutzbar visualisiert werden. Hierfür bedarf es eines integrierten Ansatzes bestehend aus Prozessen, Methoden und IT-Werkzeuge zum Mapping. Tabelle 3-2 zeigt einen Vergleich der zur Verfügung stehenden Ansätze [JaAB2012]. Im Kontext der hier angedachten Nutzung des Mappings von Wissen sind die Ansätze „Yellow Pages“, mit dessen Hilfe die Personen identifiziert werden, welche Wissensträger in einem Unternehmen sind, sowie das „Process Knowledge Mapping“, in welchem Prozesse analysiert werden, um zu schauen welches Wissen in einem Prozess benötigt wird, wer es bereitstellen kann und wo es abgespeichert werden kann [JaAB2012].

Einen anderen, automatisierten Ansatz verfolgen [GoHT2020], die das Mapping mithilfe eines Co-Word Graphen automatisieren, um Wissen darzustellen. Co-Word Analyse dient dazu, ähnliche Wörter zu finden und solche Paare zu speichern. Die Methode wurde vor allem genutzt, um in wissenschaftlichen Texten relevante Wörter und Begriffe zu identifizieren [GoHT2020].

Tabelle 3-2: Vergleich von Knowledge Mapping Techniken aus [JaAB2012]

| | „Gelbe Seiten“ | Informationsflussanalyse | Soziale Netzwerk-Analyse | Prozesswissens-mapping | Funktionales Wissens-mapping |
|----------------------------------|---|---|---|---|---|
| Tools zur Daten-aggregation | Frage/Antwort-Katalog, Fähigkeits-Listen und Reporte | Interviews, Fähigkeitslisten, Befragungen, Flussdiagramme | Fragebögen, Soziogramme, Graphen | Brainstorming, Interviews mit Prozess-Ownern | Fragebögen und Interviews |
| Tools zur Evaluation des Wissens | Fähigkeitsliste | Befragungen, Interviews und Templates | InFlow, Krackplot, NetMiner | - | Interviews, Beobachtungen, Reporte |
| Ziele | Transparenz hinsichtlich Wissen in der Organisation schaffen durch Kompetenzdatenabnken | Transparenz hinsichtlich der Nutzung (wann und wie oft) von Informations-Ressourcen | Interaktive Muster zwischen Personen finden | Transparenz hinsichtlich benötigtem Wissen, Meilensteinen, vorhandenem Wissen, Wege dieses zu heben, Fähigkeitslücken feststellen und beheben | Bereiche identifizieren, in denen Wissen benötigt wird, wissensintensive Stellen finden |
| Mapping-Ansatz | Projekt-basiert | Beziehungs-basiert | Beziehungs-basiert | Prozess-basiert | Prozess-basiert |
| Statisch/dynamisch | statisch | statisch | dynamisch | dynamisch | Dynamisch |
| Implizit/explicit | explizit | implizit | implizit | Implizit, explizit | Implizit, explizit |

3.1.6. Graphen

Graphen sind eine Modellierungstechnik, in der Informationen in Form von Knoten bzw. „vertices“ und Kanten bzw. „edges“ vorliegen. Kanten können dabei Knoten auch mit sich selbst verbinden oder doppelte Verbindungen darstellen, siehe auch Abbildung 3-6 [Deo1974]. Eine einfache Art der Repräsentation solcher Zusammenhänge lässt sich mit der Subjekt - Prädikat - Objekt Logik beschreiben. Diese oder alternativ Entity - Relationship - Entity Logik lässt sich ideal mit RDF¹¹-Tripeln beschreiben [HiKR2008]. Diese Zuordnungen finden auch jetzt schon in Methoden wie dem Quality Function Deployment oder der Design Structure Matrix Anwendung. Besonders vorteilhaft an einer Graph-basierten Darstellung der Engineering-Daten ist es, dass Zusammenhänge schnell sichtbar werden und Analysen, sogenannte Product Lifecycle Analytics verwendet werden können Zu diesen Anwendungen gehören z.B. auch Qualitätsservices, welche

¹¹ <https://www.w3.org/RDF/>

Daten auf Inkonsistenz überprüfen, eine Suche in den vernetzten Daten ermöglichen, isolierte Knoten auffinden, welche keine Verbindungen zu weiteren Elementen haben [HeKi2016].

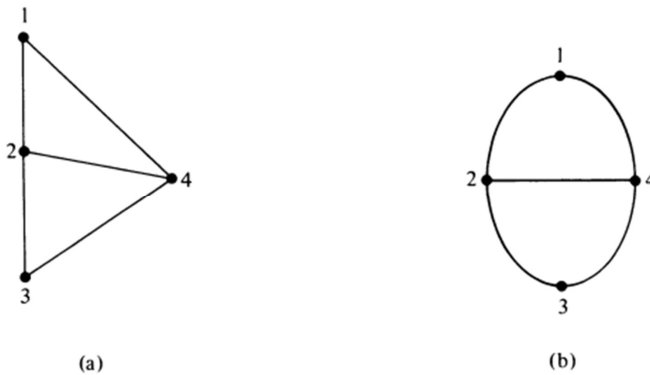


Abbildung 3-6: Visualisierung eines einfachen, ungerichteten Graphen aus [Deo1974]

Ebenfalls können mithilfe von Graphen auch strukturelle Analysen im Bereich des Engineering, z.B. zur Vorhersage von Änderungen im Bereich des Produkts oder auch zu Änderungen der notwendigen Produktionsressourcen im Falle einer Produkt-Rekonfiguration [Bied2015].

Der größte Vorteil von Graphen ist das schnelle Erfassen von Zusammenhängen. So wurde z.B. in einer Fallstudie ein Graph aufgebaut, welcher Informationen aus verschiedenen Quellen miteinander vernetzt. Die Produktstruktur eines Pedelecs mit den Zulieferern, welche die einzelnen Baugruppen herstellen, sowie die Kundenbeschwerden, welche sich eindeutig einer Baugruppe zuordnen ließen. Auf diese Art und Weise konnte über den Umweg der Produktstruktur eine Zuordnung zwischen den Kundenbeschwerden und den einzelnen Zulieferern gefunden werden, um das Qualitätsproblem einzugrenzen [HeKi2016]. Zur Speicherung dieser Graphstrukturen gibt es spezielle Datenbanken, welche sich von relationalen Datenbanken unterscheiden [BaTy2012], [RoWE2015].

3.1.7. Knowledge Graphen

Eine Spezialform des Graphen ist ein Enterprise Knowledge Graph. Er repräsentiert das Unternehmenswissen in Form von vernetzten Daten und Artefakten. Die Besonderheit von Enterprise Knowledge Graphen im Vergleich zu anderen Graphmodellen ist der Wunsch einer universellen Einsetzbarkeit im gesamten Unternehmen [Keen2020]. Ähnlich gelagert wie der Enterprise Knowledge Graph ist auch ein Engineering Knowledge Graph, welcher zum Ziel hat, das Wissen im Bereich des Engineerings abzubilden, z.B. um Wissen und Zusammenhänge von Patenten abzubilden [SiBW2022]. Die Ursprünge der Knowledge Graphen gehen zurück auf den

Google Knowledge Graph¹², welcher aufgebaut wurde, um Suchergebnisse zu verbessern und zu erkennen, wenn eine Suchanfrage sich auf ein bestimmtes Objekt bezieht, um nicht nur Links anzuzeigen, sondern auch weitere Kontextinformationen und verlinkte Objekte zu finden [FeŞA2020]. Während viele Graphen nicht für die Öffentlichkeit zugänglich sind, gibt es auch open source Projekte, auf die über eine API zugegriffen werden kann, wie z.B. ConceptNet¹³ [SpCH2017], DBpedia [LeIJ2015] oder YAGO³¹⁴ [MaBS2015].

Tabelle 3-3: Menge an Instanzen in verschiedenen Knowledge Graphen aus [Paul2016]

| Name | Instanzen | Datenobjekte | Typen | Beziehungen |
|------------------------|-------------|----------------|---------|-------------|
| DBpedia (Englisch) | 4.806.150 | 176.043.129 | 735 | 2.813 |
| YAGO | 4.595.906 | 25.946.870 | 488.469 | 77 |
| Freebase | 49.947.845 | 3.041.722.635 | 26.507 | 37.781 |
| Wikidata | 15.602.060 | 65.993.797 | 23.157 | 1.673 |
| NELL | 2.006.896 | 462.845 | 285 | 425 |
| OpenCyc | 118.499 | 2.413.894 | 45.153 | 18.526 |
| Google Knowledge Graph | 570.000.000 | 18.000.000.000 | 1.500 | 35.000 |
| Google Knowledge Vault | 45.000.000 | 271.000.000 | 1.100 | 4.469 |
| Yahoo! Knowledge Graph | 3.443.743 | 1.391.054.990 | 250 | 800 |

Tabelle 3-3 zeigt die Größe, die verschiedene Knowledge Graphen bereits 2016 erreicht haben. Diese großen Knowledge Graphen werden insbesondere automatisiert erstellt, z.B. durch das automatisierte Erfassen und Auswerten von Artikeln aus der Wikipedia in verschiedenen Sprachen, da diese über eine enorme Faktenbasis verfügt, welche eine gute Basis darstellt. Eine besondere Herausforderung stellen dabei die verschiedenen Sprachen und Versionen von Artikeln dar, da die Zuordnungen von Fakten zueinander sich dort unterscheiden können und so eine manuelle Nachprüfung notwendig sein kann [MaBS2015]. Ein wichtiges Zusammenspiel besteht zwischen Knowledge Graphen und Ontologien (vgl. [GóFC2004]). Die Einschränkung der Struktur aus Subjekt, Prädikat und Objekt auf eine vorher festgelegte Ontologie schafft es erwartbare Ergebnisse bei der Nutzung eines Knowledge Graphen zu fördern, entweder durch die Nutzung einer offiziell zugänglichen Ontologie wie z.B. schema.org oder durch eine selbstdefinierte [FeŞA2020].

¹² <https://blog.google/products/search/introducing-knowledge-graph-things-not/>

¹³ <https://conceptnet.io/>

¹⁴ <https://yago-knowledge.org/>

Nach dem Aufbau eines Knowledge Graphen liegt einer seiner Hauptanwendungsfälle im Reasoning, d.h. im automatisierten Ableiten von neuem Wissen. Aus den Zuordnungen „Der VPE ist ein Teil der RPTU“ und „Die RPTU liegt in Kaiserslautern“ lässt sich der – für Menschen logische – Schluss ziehen, dass also auch der VPE in Kaiserslautern liegt. Für einen Algorithmus muss allerdings beides in Form von auswertbaren Statements vorliegen, um diesen Schluss ziehen zu können. Auf diese Weise kann mithilfe eines Knowledge Graphen auch neues Wissen automatisiert geschaffen werden [ChJX2020]. Beispiele für die Anwendung von (Engineering) Knowledge Graphen zur Informationsaggregation und -nutzung gibt es vielfach, z.B. in [ScMB2022] wird ein Engineering Knowledge Graph aufgebaut, welcher bei der Lebenszyklusanalyse von medizinischen Geräten unterstützt und so die Komponenten findet, welche den größten Impact im Bereich der Umwelt haben und auf die Konstrukteure einen genauen Blick haben sollten.

Bliesinger et al. [BIPM2022] fassen einige Herausforderungen, die sich speziell im Bereich des Engineerings und dem Einsatz von Knowledge Graphen in diesem Bereich ergeben zusammen. So ist gerade das häufige Ändern von Artefakten, die ausgeprägte Zugriffskontrolle und die Verfügbarkeit und Heterogenität in den Datenstrukturen der Artefakte ein Problem [BIPM2022].

3.2. Nutzung semantisch angereicherter Daten

Die initiale Beschreibung und semantische Aufwertung von Daten bieten zwar bereits Vorteile, wie z.B. eine einfachere Auffindbarkeit und Nutzbarkeit von Daten, einen größeren Vorteil können Unternehmen aber insbesondere durch tieferegehende Analysen ihrer Datenbestände generieren. Dieses Unterkapitel gibt einen Einblick in die Nutzung von Datenanalyse, sowie künstlicher Intelligenz, wobei der Fokus auf solchen Techniken liegt, die zum Aufbau eines Knowledge Graph basierten Assistenzsystems genutzt werden können oder die im PLM-Kontext zum Einsatz kommen.

Die semantische Aufwertung von Daten ist kein reiner Selbstzweck, sondern dient immer der besseren Usability beim Nutzen der Daten. Hier gibt es bereits vielfältige Forschungsergebnisse, sowie auch marktreife Produkte, die zur Nutzung solcher Daten geeignet sind. Semantisch aufgewertete Daten sind insbesondere dann wertvoll, wenn aufgrund der beschreibenden Metadaten eine Integration auf Daten- oder Plattformebene stattfinden kann.

Datenintegration, also die Möglichkeit von miteinander kommunizierenden Anwendungen hat bereits eine historische Entwicklung hinter sich, ausgehend von Medienbrüchen durch den Transfer von Informationen auf Papier und das manuelle Eingeben, über SUN Remote Procedure Calls, Java RMI und CORBA bis hin und service-orientierten Architekturen (SOA) und Web-Services [Tied2006].

Die Integration kann mittlerweile auf verschiedenen Wegen stattfinden, beispielsweise durch Daten-Replikation, durch Daten-Föderation sowie durch datenorientierte Schnittstellen [Tied2006].

3.2.1. Analyse-Frameworks und einfache statistische Methoden

Durch die in den letzten Jahren immer stärker gewachsenen Datenbestände ergibt sich die Frage, welche Erkenntnisse sich daraus ziehen lassen. Standardisierte Datenanalyseprozesse wie CRISP-DM (CRoss-Industry Standard Process for Data Mining) [AzSa2008] oder ASUM-DM (The Analytics Solutions Unified Method for Data Mining/Predictive Analytics) [AnLM2018] sind eine Möglichkeit, sich aus den Daten Erkenntnisse generieren zu lassen.

Weiterhin kennen beide Datenanalyseprozesse umfangreiche Phasen im Bereich des Data Understandings, um festzustellen, welche Datentypen und Datenquellen es gibt, die in einem speziellen Analysefall genutzt werden können, um Schlüsse zu ziehen. Insofern kann das hier zu entwickelnde Konzept sowohl auf Datenanalyse als Methode zurückgreifen, als auch eine Verbesserung zukünftiger Datenanalyseprozesse sein.

Vergleich von Zeichenketten

Eine Möglichkeit, um innerhalb der vorhandenen Datenmengen ähnliche Datentypen zu finden, ist ein systematischer Vergleich. Hierzu stehen algorithmisch verschiedene Mechanismen zur Verfügung. Historisch wurden die Ansätze hierfür entwickelt, um eine Fehlerkorrektur in binärem Software-Code zu ermöglichen. Der sogenannte Hamming-Abstand beschreibt, wie viele Unterschiede zwischen zwei Zeichen eines binären Codes existieren. Ein Abstand von 1 bedeutet, dass an einer Stelle ein Bit unterschiedlich ist [Hamm1950].

Eine Erweiterung der Hamming-Distanz ist die Levenshtein-Distanz [Leve1966]. Sie ist eine mögliche Lösung für Probleme im Bereich der Verarbeitung von Zeichenketten, um hier Fehler aufzufinden, wenn man Strings mit einem Wörterbuch vergleicht [Dame1964]. Darüber hinaus gibt es noch andere Metriken, die z.B. die Länge der Strings einbeziehen, welche mittels der Levenshtein-Distanz verglichen werden. Eine so aufgewertete Metrik ist die normalisierte Levenshtein-Distanz, welche Werte zwischen 0 und 1 annehmen kann, je nachdem wie ähnlich die zu vergleichenden Strings sind [YuBo2007].

3.2.2. AKKORD

Das Projekt AKKORD befasst sich mit der Schaffung eines Referenzbaukastens und einer AKKORD-Toolbox zur Vereinfachung der industriellen Datenanalyse in Wertschöpfungsnetzwerken. Zu diesem Zweck sollen verschiedene Bereiche gestaltet werden, die im Zusammenspiel das übergeordnete Ziel erreichen. Diese sind Kompetenzen und Handlungsempfehlungen, Geschäftsmodelle und Kollaboration, Analysemodule und ein Daten-Backend-System [MaWS2019]. Maßgeblich für die Möglichkeit der Datenanalyse ist es, dass entsprechende Datenvorräte identifiziert und der Analyse zugeführt werden können. Hierbei gilt es auch ähnliche Daten zu identifizieren und entsprechend zu markieren [EIGE2020].

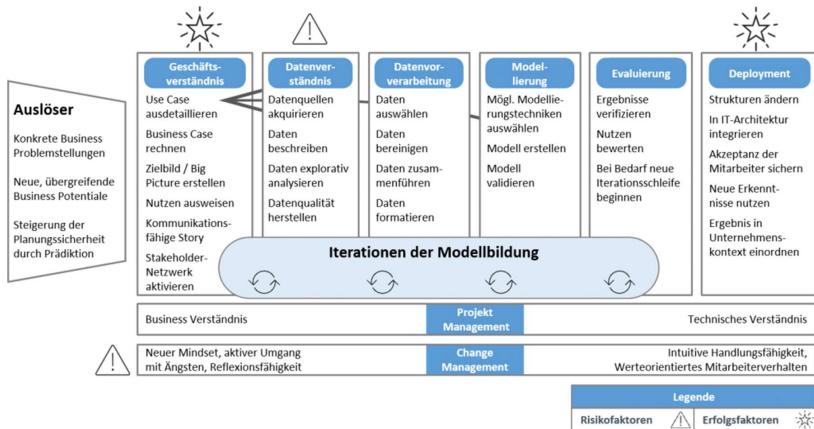


Abbildung 3-7: Hemmnisse und Erfolgsfaktoren bei der Datenanalyse aus [NoSM2020]

Abbildung 3-7 zeigt exemplarisch Hemmnisse und Erfolgsfaktoren bei der Datenanalyse entlang der einzelnen Schritte des CRISP-DM. So gilt es in den initialen Phasen Datenverständnis und Datenverarbeitung entsprechende Datenquellen zu akquirieren, diese zu erkunden und diese der Analyse zuzuführen [NoSM2020].

Während die Daten entweder direkt in den entsprechenden Systemen verbleiben können, soll zumindest der Zugriff auf diese Daten über ein standardisiertes IT-Werkzeug erfolgen. Zu diesem Zweck wurde der Data Model Canvas entwickelt, welcher mithilfe einer Graph-basierten Oberfläche die angebotenen Systeme, Datentypen und Attribute darstellt. Aus diesen Informationen können anschließend Datenpipelines erstellt werden, welche die eigentlichen Nutzdaten über eine Schnittstelle direkt in den Analysemodulen bereitstellt [EIEG2021], [EIEG2022].

3.2.3. Künstliche Intelligenz im Kontext von PLM und Produktentwicklung

Im Kontext von PLM kann KI vielfältig eingesetzt werden, um aus den vorhandenen Daten neue Informationen zu generieren. So können z.B. mithilfe von Natural Language Processing (NLP) Ähnlichkeiten in Anforderungsspezifikationen gefunden werden, um diese den Ingenieuren mitzuteilen und einen erhöhten manuellen Aufwand zur Reduktion der Daten zu vermeiden. Hierzu werden die Anforderungen in Form von Texten zuerst vorverarbeitet und anschließend mit verschiedenen Wortvektoren analysiert, um Ähnlichkeiten zwischen Sätzen erkennen zu können. Abbildung 3-8 zeigt das dabei gewählte Vorgehen und die eingesetzten KI-Methoden sowie IT-Werkzeuge [BrLL2021].

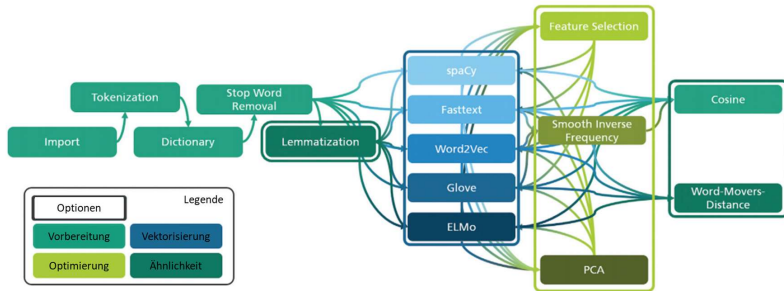


Abbildung 3-8: Auswertung von Ähnlichkeiten in Spezifikationen nach [BrLL2021]

Die Möglichkeiten der Datenanalyse im Bereich von Engineering-Daten zeigen anschaulich [ArMG2016], [ArGM2018], [ArGM2018], [ZhTL2018], [RiDB2019] [Detz2022]. So zeigt sich, dass z.B. in einer gezielten Analyse von Produktänderungen eine verbesserte Ursachenermittlung stattfinden kann. Häufig werden Produktfehler nach ihrer Entdeckung ad-hoc verbessert, ohne die zugrunde liegende Ursache zu ermitteln. Basierend auf einfachen statistischen Analysen (welche Bauteile sind am häufigsten betroffen, welche Fehlermodi tauchen öfters auf), kann anschließend im Detail versucht werden, durch die Analyse eine zugrunde liegende Fehlerursache im Detail zu finden. Allerdings zeigt sich auch, dass ein entscheidender Teil der Datenanalyse die Identifikation entsprechender Datenquellen ist [ArMG2016].

3.2.4. Engineering-Integrationsplattformen

Im Bereich des Engineerings gibt es eine ganze Reihe von Behelfslösungen, spezialisierten Integrationsplattformen, welche Daten aus unterschiedlichen Systemen miteinander vernetzen, ohne hierbei ein ganzheitliches Konzept zu verfolgen, aber auch universelleren Plattformen zu Integration von Daten. Behelfslösungen sind häufig von unternehmensinternen Fachexperten erstellte Hilfsmittel wie Excel-Arbeitsmappen, SharePoint-Listen oder ähnliche eher allgemeine Kollaborationstools, welche für die Engineering-Datenintegration genutzt werden. Insbesondere herstellerneutrale Integrationslösungen haben sich als vorteilhaft erwiesen, da sie ohne Lock-In Effekt Daten miteinander vernetzen können und unterschiedlichste Software anbinden [BiMM2012], [BiMS2017].

Ein mögliches Lösungskonzept im Bereich der Informationsintegration im Bereich MBSE setzt auf das modellierte System als Single Source of Truth. Sollte in diesem Bereich eine Information fehlen, so wird ein Workflow ausgelöst, welcher die für das entsprechende Teilsystem verantwortliche Person ermittelt und dieser einen Auftrag gibt, die benötigten Informationen zur Verfügung zu stellen. Gleichzeitig können die benötigten Informationen auch als Teil des Systems modelliert werden, so dass eine Information auch direkt formal beschrieben vorliegt und automatisiert weiterverarbeitet werden kann. Hierbei liegt der Fokus darauf, dass nicht direkt alles modelliert wird, was vorstellbar ist, sondern iterativ immer nur diese Elemente erstellt

werden, die auch wirklich im Prozess vorliegen müssen. Die Verteilung der Informationsbedarfe wird in diesem Konzept von einem Workflow-Management Tool übernommen [BrTD2016].

Biffel. et al. zeigen mit dem AML-Hub ein Beispiel für eine Integrationsplattform im Bereich der Automatisierungstechnik, welche aufgrund der verschiedenen beteiligten Fachdisziplinen einen erhöhten Bedarf an Datenaustausch hat. AML Hub ist hierbei als eine herstellerneutrale Integrationsplattform auf Basis von Automation ML geeignet, um Daten aus verschiedenen Quellen zu integrieren, abteilungsübergreifende Workflows zu unterstützen und bietet auch eine Unterstützung bei der Identifikation ähnlicher Objekte aus anderen Datenquellen [BiMS2017].

Ein weiteres Beispiel ist die am VPE entwickelte CMBC-Plattform. CMBC steht für Change-Management Backbone & Control, eine Integrationsplattform, welche dem Änderungsmanagement dient und eine Brücke zwischen der Konstruktion mit ihren PLM-zentrierten Daten und Prozessen, sowie der Fertigung und Arbeitsvorbereitung mit ihrer Fokussierung auf ERP-Systeme [Erns2016]. Bisherige Änderungsmanagementprozesse waren sehr sequenziell und fokussiert auf einzelne Abteilungen bzw. waren nicht in der Lage die Auswirkung einer Änderung in Gänze zu erfassen und führten zu hohem manuell Abstimmungsaufwand, hohen Kosten und langen Durchlaufzeiten [Quir2015]. Diese Probleme sollte die Integrationsplattform CMBC durch die Datenbereitstellung und eine vereinheitlichte Oberfläche lösen [Erns2016].

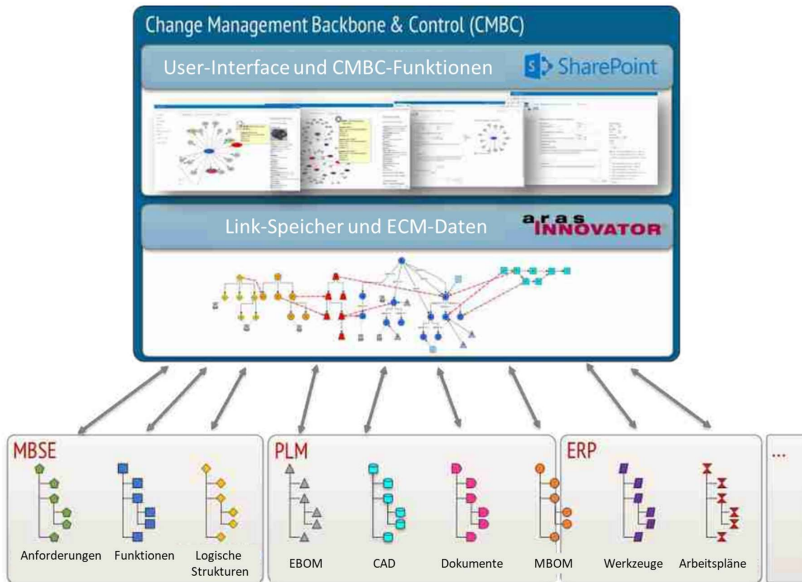


Abbildung 3-9: IT-Architektur des CMBC-Systems nach [Erns2016]

Wie in Abbildung 3-9 gezeigt, besteht die Architektur der CMBC-Plattform aus drei wesentlichen Elementen. Zuerst die Datenquellsystemsicht, in diesem Fall bestehend aus Aras Innovator und SAP, einem Link-Repository, welches ebenfalls in einem Aras Innovator System realisiert wurde und zuletzt einem User-Interface, welches in SharePoint realisiert wurde [Erns2016].

Der Prozess des Änderungsmanagements wurde angelehnt an das Vorgehen der VDA 4965 [VDA2010], [VDA2010] modelliert und entsprechende Oberflächen für die einzelnen Schritte zur Verfügung gestellt. Die Besonderheit an der CMBC-Plattform ist, dass im Prozess von der Änderung betroffene Objekte aus den unterschiedlichen Systemen in einer Graph-Ansicht präsentiert werden, um logische Zusammenhänge auch über Systemgrenzen hinweg sichtbar zu machen und im Prozess zu berücksichtigen [Erns2016].

Kommerziell sind verschiedenste Integrationsplattformen verfügbar, z.B. die Contact Integrationsplattform „Contact Elements“¹⁵, welche aus einer PLM-Software hervorgegangen ist, welche durch die Möglichkeit der Datenintegration aus anderen Quellen erweitert wird. Technisch wird diese Integration insbesondere über standardisierte Schnittstellen aus dem Bereich der Webtechnologie, sowie Mikroservices umgesetzt.

¹⁵ <https://www.contact-software.com/de/produkte/integrationsplattform/>

Einen ähnlichen Ansatz nutzt auch die Plattform OpenPDM¹⁶ der Prostep AG. Hier werden nach Anbinden verschiedener Systeme neue Prozesse und Workflows definiert, welche unterstützt durch Daten aus mehreren Systemen durchgeführt werden sollen. Als typische Anwendungsfälle werden hierbei unter anderem das Erstellen von Prozessplänen an der Schnittstelle zwischen Produktentwicklung und Fertigung, aber auch der bei anschließenden Änderungen notwendige Änderungsprozess selbst identifiziert [Pros2022].

Weitere Plattformen, die Konnektoren für die Integration weiterer Systeme bieten sind unter anderem die Business Integration Suite¹⁷ der Firma Seeburger, das SAP Engineering Control Center¹⁸, eQube MI¹⁹ die Software PDM Connect²⁰ der Firma PDTEC AG.

All den oben kurz genannten Integrationsplattformen ist gemein, dass sie technologisch insbesondere auf Webschnittstellen aufbauen, die eine Konnektivität in die verschiedenen Quellsysteme ermöglicht. Mittels Konnektoren können auch weitere Softwaresysteme angesprochen werden, die über eine standardisierte API verfügen, sondern z.B. direkt mit einem JavaScript, C# oder C++-Programm verbunden werden müssen.

Alle Plattformen haben ebenfalls die Gemeinsamkeit, dass anschließend das Mapping der unterschiedlichen Daten händisch oder (teil)automatisiert vorgenommen werden muss. Eine Hilfestellung, welche Bedeutung die einzelnen Daten haben, wird erst durch die mit dem Mapping betraute Person erstellt. Ebenfalls können nur vorliegende Daten miteinander in Beziehung gesetzt oder semantisch aufgewertet werden, Dashboards dienen dazu diese Daten anschließend den Usern verfügbar zu machen.

3.2.5. Linked Data

Das Konzept der verlinkten Daten entstammt dem Grundgedanken, nach dem auch das Internet aufgebaut ist. Datenobjekte bleiben dort, wo sie sowieso bereits gespeichert sind, und werden von anderen Datenobjekten aus mit identifizierbaren Links versehen, um sie zu erreichen. Hierbei sind zwei zentrale Ansätze zu unterscheiden: Ansätze mit einem zentralistischen Link-Verzeichnis, sowie dezentrale Ansätze, die über kein zentrales Verzeichnis verfügen, sondern ggf. über ein zentrales Regelwerk, welches vorgibt, welche Links erlaubt sind und welche nicht, vgl. Abbildung 3-10 [Reic2022].

¹⁶ <https://www.prostep.com/produkte-und-loesungen/openpdm/openpdm-integrate>

¹⁷ <https://www.seeburger.com/de/plattform/business-integration-suite/>

¹⁸ https://www.cenit.com/de_DE/produkte-loesungen/plm/sap-loesungen/sap-produkte/sap-ectr-engineering-control-center.html

¹⁹ https://www.1eq.com/products_eqube-mi

²⁰ <https://pdtec.de/produkte/pdmconnect/>

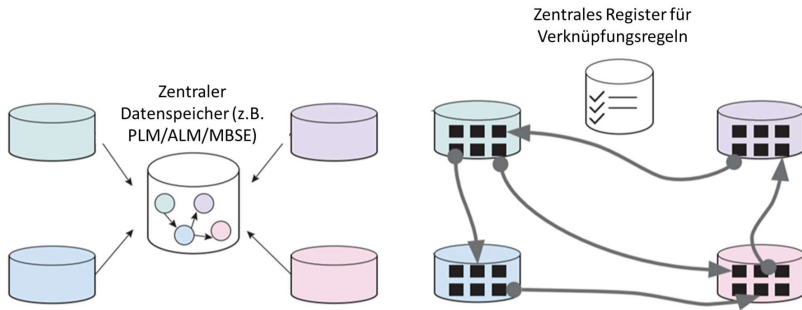


Abbildung 3-10: Zentraler und dezentraler Linked Data Ansatz nach [Reic2022]

Grundlegendes Prinzip von Linked Data sind Tupel, in welchen die Beziehung zwischen zwei strukturierten Objekten abgespeichert sind, entsprechend der Subjekt – Prädikat – Objekt – Logik. Durch die Verknüpfung zwischen verschiedenen Objekten kann Linked Data implizit auch Kontextinformationen speichern. So ist z.B. eine Zuordnung „Würzburg liegt in Deutschland“ auch eine Kontextinformation [GrBa2020]. Beispielhafte Umsetzungen für Linked-Data Ansätze gibt es nicht nur allgemein zur Informationsintegration, sondern auch als Aufwertung eines PLM-Systems, indem eine Semantik zur Beschreibung aller Datenobjekte erstellt und mit den eigentlichen Daten kombiniert wird [KaMK2013].

Genutzt wird diese Technologie unter anderem von der Firma Conweaver in ihrem Produkt Linksphere Knowledge Graph²¹. Linksphere ist zwar eine Engineering Integrationsplattform, aufgrund des technologischen Kerns wird sie in diesem Abschnitt vorgestellt. Dieser technologische Kern besteht aus einem Engineering Knowledge Graph, welcher Assets aus verschiedenen Systemen miteinander verlinkt und so Daten zugänglich macht. Ein Nachteil von Linksphere ist die mangelnde Unterstützung in machen Phasen des Engineering, z.B. in der Innovationsphase [MoEE2022]. Weiterhin eine Verlinkung und Integration von Daten erst nach ihrem Entstehen möglich, was in manchen Anwendungsfällen ein Hindernis darstellt [Kehl2019].

3.2.6. Open Services for Lifecycle Collaboration (OSLC)

OSLC ist ein gemeinsam entwickelter Standard, welcher das Ziel hat, ein Web of Data zu schaffen. Entsprechend dem bereits alltäglichen Internet, in welchem Dokumente untereinander verlinkt sind, sollen auch weitere Daten als Ressourcen beschrieben, miteinander verlinkt und auffindbar sein²². Hierdurch soll eine einfachere Integration föderierter und geteilter Daten entstehen, welche aus unterschiedlichen IT-Werkzeugen in unterschiedlichen Domänen aufgerufen und bearbeitet werden sollen. Die Integration basiert auf der Nutzung von Web-Technologien wie

²¹ <https://www.conweaver.com/technology/de-knowledge-graph>

²² <https://open-services.net/>

REST (Representational State Transfer) und RDF (Resource Description Framework²³). Abbildung 3-11 zeigt die wichtigsten Elemente des OSLC Core. Hierzu gehört die Eigenschaft verschiedene Domänen oder Anwendungsszenarien abzudecken, unter anderem Änderungs- und Konfigurationsmanagement. Der OSLC Core selbst bietet verschiedene Features wie eine Ressourcenvorschau, die Möglichkeit weitere Ressourcen über Links aufzurufen und mit einem gemeinsamen Thesaurus zu klassifizieren. Über Erweiterungen können noch weitere Fähigkeiten und Features mit Ressourcen genutzt werden. Die Linked Data Plattform gewährt den Zugriff auf die verschiedenen Ressourcen.

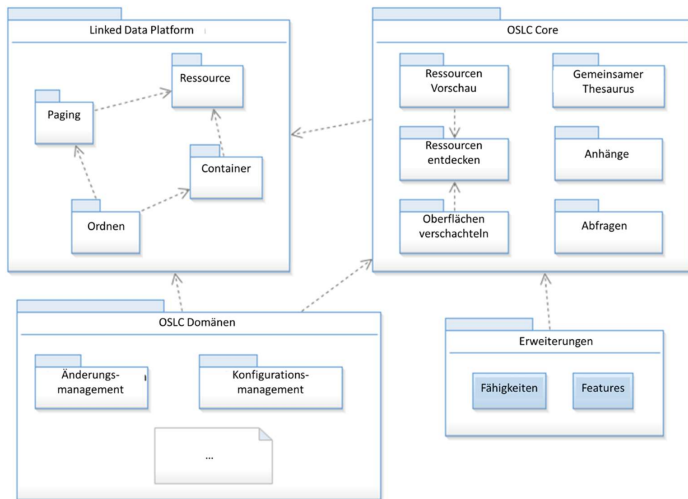


Abbildung 3-11: Überblick über den OSLC-Core 3.0 nach ²³

Der Standard ist offen und kann von jedem implementiert werden, Beispiele hierfür finden sich u.a. im Bereich des Change Managements [BeLD2011], der Integration von Modellierungstools [ElNe2013], [ZhMø2014] oder des Systems Engineering [SaBu2014].

3.2.7. Datenintegration mittels IT-System-Schnittstellen

IT-System-Schnittstellen sind unerlässliche technische Voraussetzungen, um Daten zwischen verschiedenen IT-Systemen automatisiert auszutauschen. Es wird hierbei zwischen direkten und indirekten Schnittstellen unterschieden (vgl. Abbildung 3-12). Weiterhin gibt es neutrale und proprietäre Schnittstellen (vgl. Abbildung 3-13) [EiRZ2014].

²³ <https://docs.oasis-open-projects.org/oslc-op/core/v3.0/os/oslc-core.html>



Abbildung 3-12: Direkte und indirekte Schnittstellen nach [EiRZ2014]

Direkte Schnittstellen erlauben einer Software den direkten Zugriff auf die Daten einer zweiten Software, während sich eine indirekte Schnittstelle einem Austauschformat bedient, in welchem die Daten zum Abruf zur Verfügung gestellt werden und in diesem Format in das nächste System übertragen werden [EiRZ2014].

Eine proprietäre Schnittstelle ist so aufgebaut, dass sie nur in genau der Kombination zwischen spezifischen Softwaresystemen funktioniert, da auf ein geschütztes Format zur Übertragung zurückgegriffen wird, meistens finden sich solche proprietären Schnittstellen in Softwareschnittstellen einer Firma. Neutrale Schnittstellen hingegen verwenden Austauschformate, welche offen und damit auch für andere Softwarehersteller nutzbar sind [EiRZ2014].

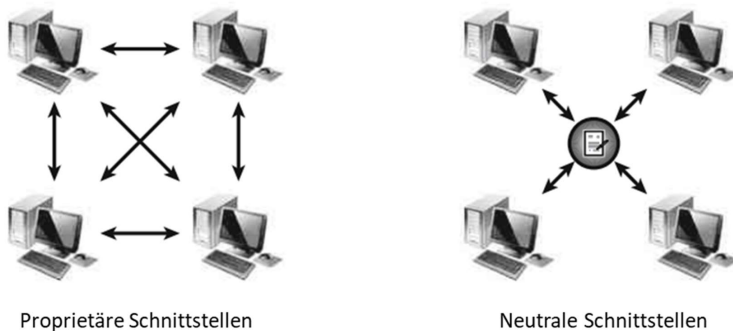


Abbildung 3-13: Proprietäre und neutrale Schnittstellen nach [EiRZ2014]

Gerade im Bereich des Semantic Web spielen Schnittstellen eine immer wichtigere Rolle, da Daten in verschiedenen Systemen vorliegen und automatisiert auf diese zugegriffen werden soll. In einer Auswertung von hunderten von Milliarden Schnittstellen-Aufrufen in der Apigee Cloud wurde gezeigt, dass vor allem offene Schnittstellen (APIs) ein Erfolgsfaktor sind und sich die Anzahl der Apps, die diese Schnittstellen nutzen stark vergrößert, was ganze Ökosysteme beflügelt. Führend sind hierbei Angebote im Bereich Medien, Informationen und Technologien [apig2017].

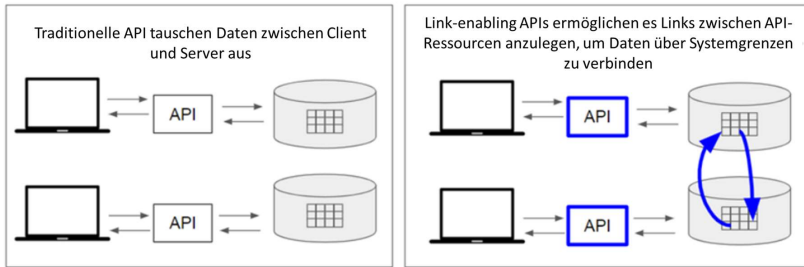


Abbildung 3-14: Traditionelle und Link-Enabling APIs nach [BIPM2022]

Abbildung 3-14 zeigt einen wichtigen Unterschied zwischen traditionellen und Link-Enabling APIs. Letztere sind in der Lage über Links Ressourcen aus verschiedenen API miteinander zu verbinden [BIPM2022].

3.2.8. Erweiterung des PLM für industrielle Produkt-Service Systeme

Während klassische PLM-Systeme und Ansätze von mechanischen oder mechatronischen Produkten sprechen und für diese ausgelegt sind, schlägt Dang [Dang2017] eine methodische Erweiterung für das PLM vor, um auch industrielle Produkt-Service Systeme damit verwalten zu können [Dang2017]. Das Konzept baut auf Vorarbeiten auf, welche im Rahmen eines Transregio-Projekts erarbeitet wurden [AiDA2017].

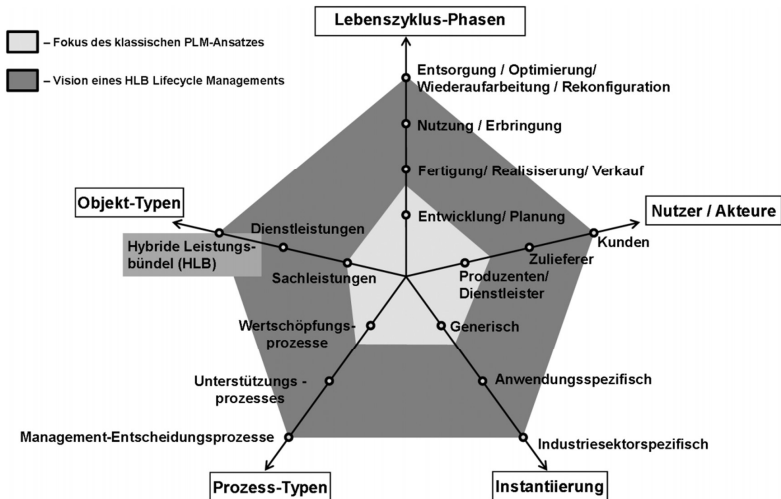


Abbildung 3-15: Dimensionen des IPSS-LM Konzeptes aus [AbMN2008]

Wie in Abbildung 3-15 gezeigt umfasst das IPSS-LM Konzept fünf Dimensionen, in denen der Scope des ursprünglichen PLM-Ansatzes erweitert wird. Dies trägt vor allem der stark gesteigerten Komplexität von IPSS im Vergleich zu traditionellen Produkten Rechnung. So müssen vor allem Daten aus unterschiedlichsten Bereichen des Lebenszyklus verknüpft werden und durch neue Services integriert und zur Datenverarbeitung bereitgestellt werden [Dang2017].

Um die Daten aus unterschiedlichen Quellen miteinander zu verbinden und ein Mapping herzustellen wird eine IPSS-Top Level Ontologie aufgebaut, welche die Daten in verschiedene Klassen unterteilt und diese mithilfe von Axiomen in Beziehung setzt. Diese IPSS Top Level Ontologie wurde anschließend als Datenmodell im PLM-System PTC Windchill implementiert, um möglichst viele Lebenszyklusdaten an einer Stelle zu managen. Abbildung 3-16 zeigt die Systematik, mit der die IPSS-Top Level Ontologie erstellt wurde [Dang2017].

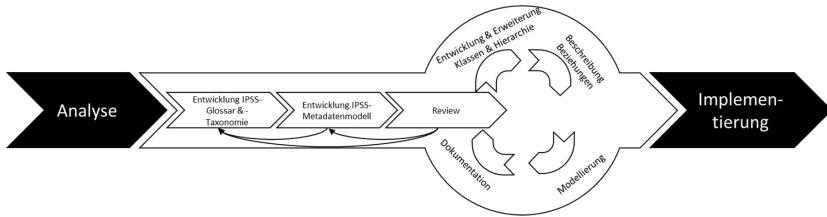


Abbildung 3-16: Systematik zum Aufbau der IPSS Top Level Ontologie nach [Dang2017]

Alle weiteren Objekte, welche nicht in Windchill gemanagt werden können, müssen in einer Graph-Datenbank repräsentiert werden, welche über eine Schnittstelle mit Windchill synchronisiert wird und so den Gesamtüberblick auf den gesamten IPSS-Datenbestand gewährt [Dang2017].

3.2.9. Datenzentrierte Architekturen im PLM-Kontext

Zur besseren Nutzung von Potentialen im Bereich der Semantik und auch um der Situation Rechnung zu tragen, dass verschiedenste IT-Systemen mit entsprechenden Datenbanken und darauf aufbauenden Funktionen bereits vorhanden sind, wird vorgeschlagen, dass die bisherige PLM-IT-Architektur aufgebrochen und verändert wird [BlPM2022].

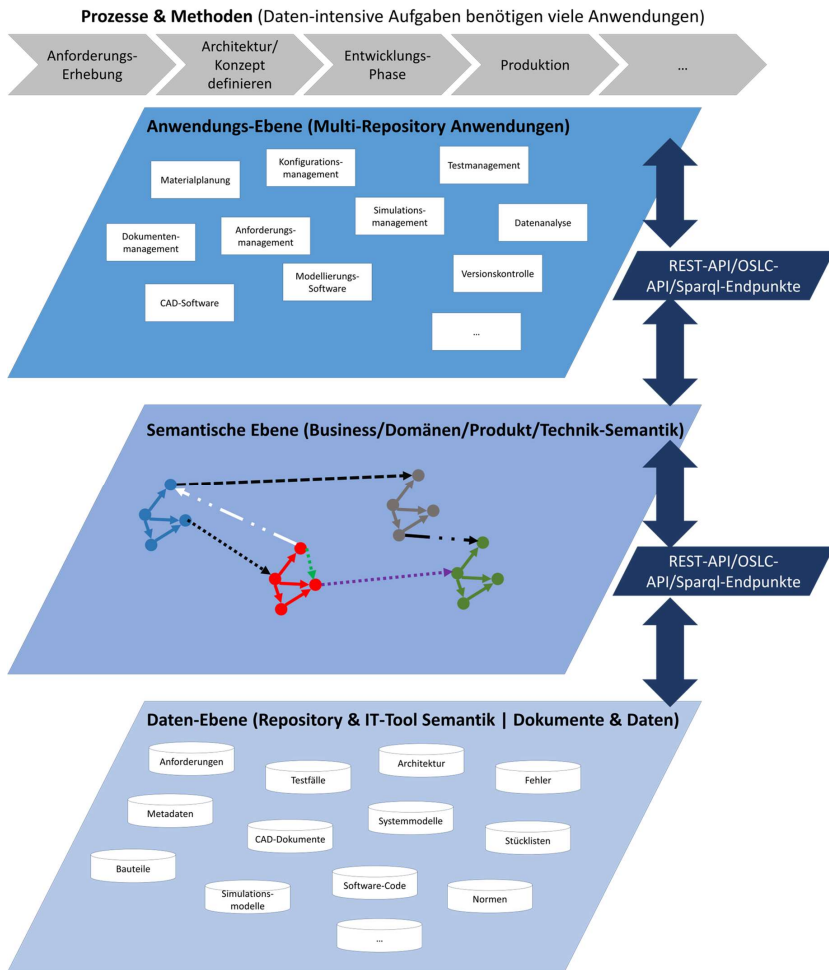


Abbildung 3-17: Datenzentrierte Architektur aus einer Fallstudie nach [BIPM2022]

Die Abbildung 3-17 zeigt eine solche datenzentrierte Architektur, bei der drei Ebenen vorgesehen sind. Zum einen der Data Layer, welcher aus den verschiedensten Datenbanken besteht, darüber ein Semantic Layer, welcher Vernetzungsinformationen über diese Daten legt und Daten miteinander in Beziehung bringt, ohne die Daten direkt zu verändern und zuletzt der Application Layer, auf welchem Microservices angesiedelt sind, welche mit einem Ausschnitt der verfügbaren Daten Services bereitstellen [BIPM2022].

3.2.10. LeWiPro

Das abgeschlossene Projekt LeWiPro hatte das Ziel methodische und IT-Komponenten zu erarbeiten, welche die Entwicklung innovativer Produkte auf Basis von Lebenszykluswissen unterstützt. Zu diesem Zweck wurden unter anderem ein Modell der Wissensrepräsentation von Lebenszykluswissen, eine Methode zur automatisierten Akquise von Wissen, sowie die anwenderfreundliche Wissensbereitstellung untersucht [ScZe2013], [BeZK2014].

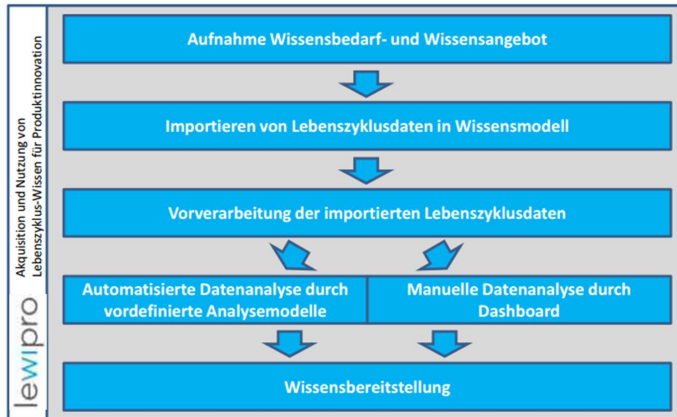


Abbildung 3-18: Methode zur Akquise von Lebenszyklusdaten aus [BeGK2015]

Abbildung 3-18 zeigt die einzelnen Schritte, welche beim Durchführen der LeWiPro-Methodik durchlaufen werden. Im Bereich der Aufnahme von Wissensbedarfen und den Wissensangebot wird keine dynamische Erhebung bei jedem Durchlaufen der Methodik durchgeführt, sondern zu Projektbeginn wurde das Wissen in den Bereichen Service, Instandhaltung und Logistik durch Experteninterviews erfasst und anschließend eine Ontologie erarbeitet, welche dieses Wissen abbildet und dies als Grundlage für eine neue Wissensdatenbank bereitstellt [BeGK2015].

Das Befüllen der Datenbank mit Wissens-elementen erfolgt anschließend durch einen Import von Lebenszyklusdaten in Form von .csv-Daten, welche auch aus anderen Systemen exportiert worden sind. Bei den Daten handelt es sich vor allem um unstrukturierte Texte, welche mithilfe verschiedener Methoden wie Tokenisierung, dem Filtern von Stoppwörtern, dem Ersetzen von ähnlichen Wörtern und nicht zuletzt durch Lemmatisierung und Nutzung der Grundform von Wörtern noch besser auswerten zu können [BeZe2014].

Die so importierten und vorverarbeiteten Daten können anschließend anhand verschiedener vordefinierter Fragestellungen und Verfahren mithilfe der Software RapidMiner analysiert werden. Analyseergebnisse können anschließend zur weiteren Verwertung in einem PLM-System abgespeichert werden [BeGK2015].

3.2.11. ENTIME

Im Projekt ENTIME wurden verschiedene Methoden entwickelt, um semantische Technologien in der Entwurfsphase mechatronischer Systeme einzusetzen [BaAG2011]. Ein Teilziel des Projekts war die Bereitstellung von Lösungselementen im Semantic Web, welche mithilfe von Klassifizierungsmechanismen auch von Ingenieuren gefunden werden können, welche bestimmte Funktionen mithilfe dieser Elemente abbilden wollten. Neben den formalisierten Lösungselementen – welche physischer oder softwaretechnischer Natur sein können – sollen auch Lösungsmuster abgespeichert werden, um interdisziplinäre Lösungsansätze wiederverwendbar zu gestalten und die Lösungselemente weiter zu abstrahieren [GaTS2014].

Ein großes Problem bei der Suche nach Lösungselementen, also funktionierenden physischen oder softwaretechnischen Objekten, welche von der eigenen oder anderen Firmen bereits auf dem Markt angeboten werden, ist das Wissen wie diese aufzufinden sind. Hersteller klassifizieren ihre Produkte unterschiedlich, jeder Hersteller oder Bereitsteller von Lösungselementen tut dies auf anderem Wege und es gibt keine zentralisierte Datenbank. Im Projekt wurde deswegen eine Spezifikation entwickelt, welches Mindestset an Eigenschaften ein Lösungselement benötigt, um ausreichend identifizierbar zu sein. Hierzu gehören Auswahlmerkmale, welche z.B. auf einem Standard wie ECLASS beruhen können, eine Beschreibung des Einsatzkontexts mit Beispielen, ein 3D-Gesamtmodell, Simulationsmodelle, entsprechende Modellparameter und zuletzt Informationen über die Kombinierbarkeit mit weiteren Lösungselementen [GaTS2014].

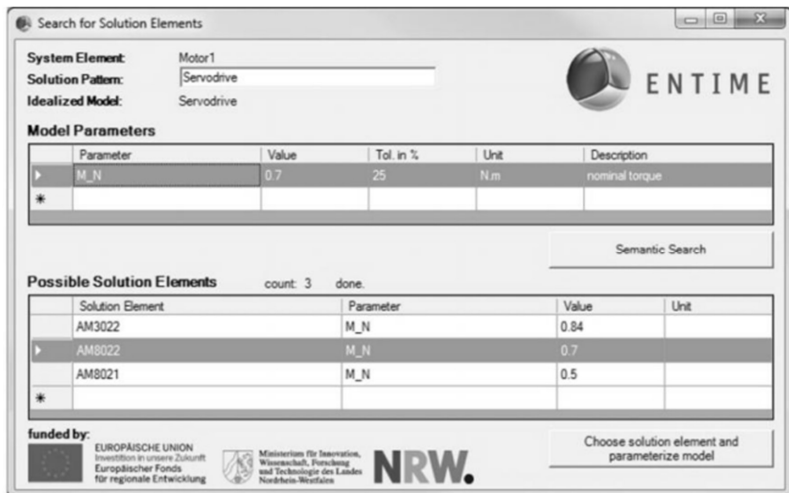


Abbildung 3-19: Semantische Suche nach Lösungselementen aus [TeID2013]

Die semantisch aufbereiteten Informationen werden anschließend über einen Publikationsservice einer Datenbank zugeführt und können so von allen Akteuren in der Entwurfsphase auch gesucht

und gefunden werden. Abbildung 3-19 zeigt die Suchmaske der ENTIME-Suche, mit der Lösungselemente anhand ihrer veröffentlichten Parameter gefunden werden können. Hierbei können die Daten entweder in einem einzigen Server gespeichert werden, oder über auch über eine Vielzahl von kommunizierenden Servern verteilt sein. Die gefundenen Lösungselemente oder Lösungsmuster können anschließend im Entwicklungsprozess genutzt werden und zur Anreicherung des zu entwerfenden Systems dienen. Zur Umsetzung des Semantic Web Ansatzes werden in der Lösung vor allem Open-Source Technologien genutzt wie SPARQL als Abfragesprache, einen RDF-Graphen als Datenspeicher und RDF/XML als standardisierte Syntax. Anfragen werden über HTTP als Protokoll an den Server geschickt [GaTS2014], [OeJT2012].

3.2.12. International Data Spaces

Das Projekt International Data Spaces versucht eine Referenzarchitektur zur Sicherstellung der Datensouveränität in Firmenökosystemen zu schaffen. Die Rolle von Daten im Sinne der Geschäftsmodelle nimmt immer weiter zu. Es stellt sich die Herausforderung sowohl öffentliche als auch firmeninterne Daten miteinander zu vernetzen und darüber Prozesse entlang der Wertschöpfungskette zu versorgen und gewinnbringend zu steuern und dabei andere Partner im Wertschöpfungsnetzwerk ebenfalls mit Daten zu versorgen, ohne die Kontrolle darüber zu verlieren. Hierzu wurde vor allem ein Rollenkonzept erstellt, welches Rollen wie Data Owner, Data Provider und Data User als Kernrollen umfasst, sowie zum Teilen der Daten noch Broker, eine Clearingstelle, einen Identitäts- sowie einen Vokabular-Provider umfasst, welche jeweils einen Service beim Teilen der Daten bereitstellen [OtHW2019]. Die Nutzung eines solchen Frameworks wird von Firmen als strategischer Vorteil gesehen [O'MB2022].

Die IT-Architektur des Projekts umfasst sechs verschiedene Softwarekomponenten, welche im Zusammenspiel zum Teilen und Nutzen von Daten erforderlich sind: Ein Konnektor dient zum Freigeben bzw. zum Transfer der eigentlichen Daten. Eine Usage-Control-Software dient der Überwachung der Prozesse und der Daten. Das Datenmodell ist neben einer konzeptuellen Abbildung auch als Software verfügbar. Ein App-Store, eine Broker-Software und ein Identitätsprovider-Service bilden den funktionellen Kern [OtHW2019].

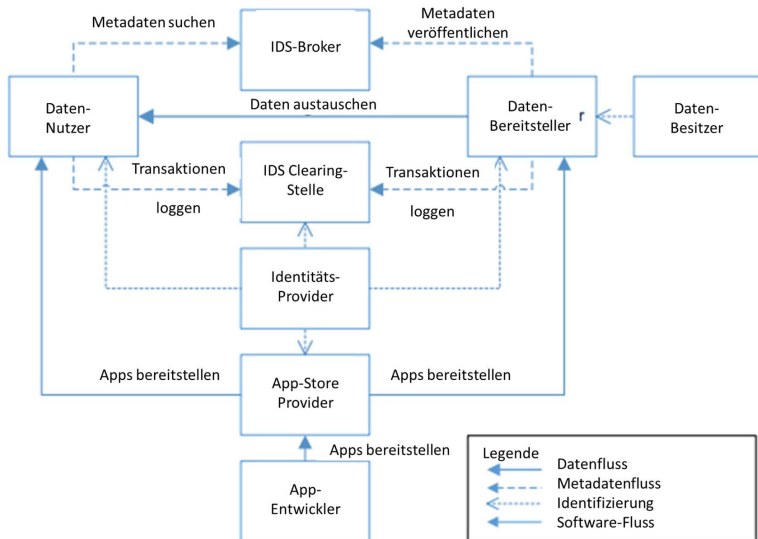


Abbildung 3-20: Rollen und Informationsflüsse im IDS-Ökosystem nach [OtJa2019]

Abbildung 3-20 zeigt, wie die verschiedenen Akteure im International Data Space miteinander kommunizieren und wie sie ihren Rollen gerecht werden. Hierbei werden die Daten bei den Data Providern verteilt gespeichert und jeweils bilateral weitergegeben und genutzt, im Gegensatz zu zentralisierten Datenspeichern wie Data Lakes [OtJa2019], zur Weitergabe der Daten besteht u.a. die Möglichkeit der Nutzung einer Blockchain, welche Vertrauen zwischen den beteiligten Akteuren schaffen kann [PrRU2022].

3.2.13. SP²IDER

Am Lehrstuhl VPE ist in mehreren Iterationen die Vernetzungsplattform SP²IDER (Semantic Product & Process Information, Digitized Engineering Repository) entstanden. Das Ziel der Plattform ist die Vernetzung von Daten und der Aufbau durchgängiger Digitaler Threads über möglichst viele genutzte IT-Systeme hinweg. Hierbei ist es das primäre Ziel, weder die bereits vorhandenen Daten zu kopieren und evtl. inkonsistente Dubletten zu schaffen, noch die Ablösung der bestehenden IT-Systeme und der Einsatz eines zentralen, monolithischen Systems, welches anschließend die Datenspeicherung übernimmt [EiEG2020].

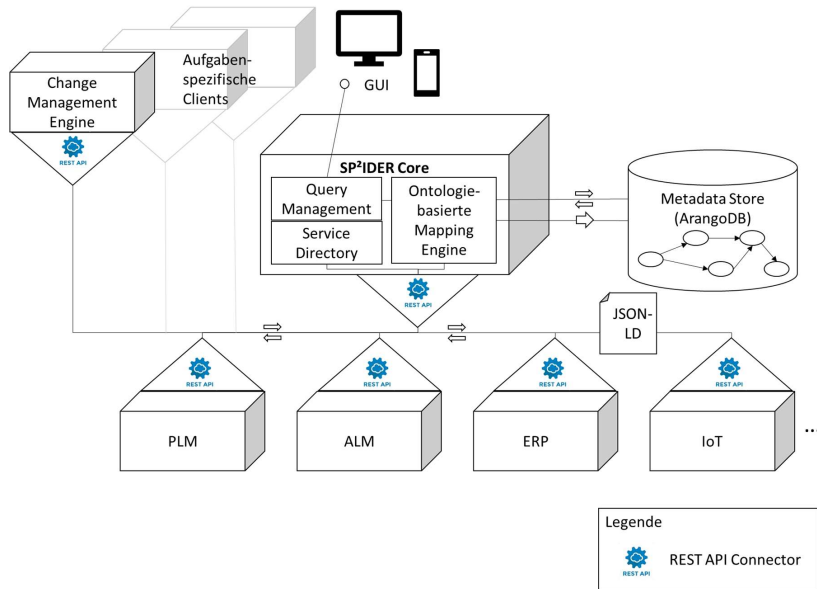


Abbildung 3-21: Die SP²IDER-Architektur nach [EiEG2020]

Abbildung 3-21 zeigt die bestehende Architektur des SP²IDER Systems, welche aus vier Bereichen besteht:

- Datenquellsysteme
- SP²IDER Core
- SP²IDER Services
- Metadata Store

Die Datenquellsysteme sind solche IT-Systeme, welche bereits in der Firma vorhanden sind und der Datenspeicherung dienen, z.B. PLM-Systeme, ERP-Systeme oder spezialisierte Datenbanken in allen anderen Bereichen des Engineerings oder des Produktlebenszyklus. Ein wesentliches Merkmal dieser Ebene ist die große Heterogenität hinsichtlich der IT, der Datenstrukturen, der Schnittstellen und der Einsatzzwecke. Da keine gemeinsamen Standards für Datenaustausch oder Schnittstellen existieren, wird für jedes System ein REST API Connector benötigt, welcher den Datenaustausch mit den weiteren Bereichen der SP²IDER Architektur ermöglicht [EiEG2020].

Zu den Kernkonzepten von SP²IDER gehört unter anderem ein allgemeines Datenmodell, welches aus den Ebenen „IT-System“, „Datentyp“, sowie „Datenobjekt“ besteht. Es wurde bei der Erschaffung möglichst allgemeingültig formuliert, um die Breite der möglicherweise interessanten Datenquellsysteme auch ohne Einschränkungen abbilden zu können [EiEG2020].

Neben dem Datenmodell ist ein weiterer Kernbestandteil des Konzepts ein Prozessmodell, welches daraus ausgerichtet ist, sowohl die Datenquellsysteme als auch deren Datentypen und Datenobjekte in die SP²IDER-Datenbank aufzunehmen. Abbildung 3-22 zeigt einen Ausschnitt dieses Prozessmodells für den Bereich der Datentypen [EiEG2020].

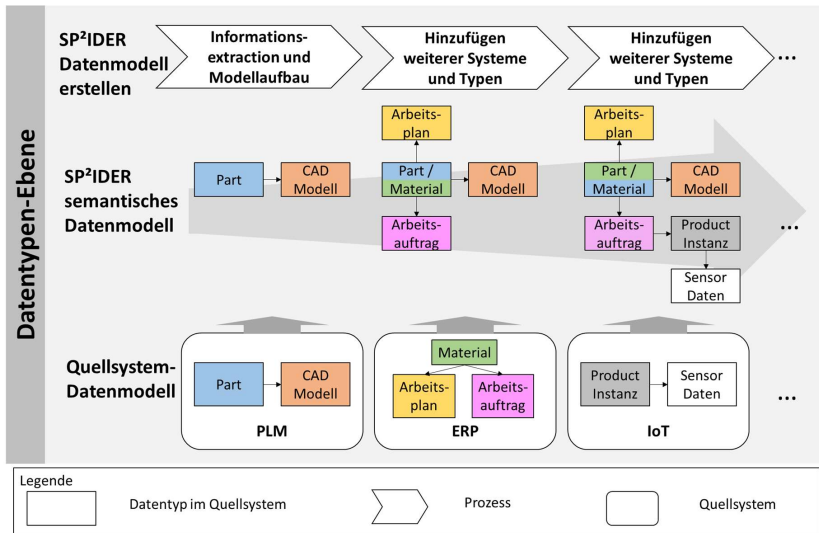


Abbildung 3-22: Ausschnitt des SP²IDER Prozessmodells nach [EiEG2020]

In seiner Konzeption ist SP²IDER zwar in erster Linie als Speicherort für Metadaten konzipiert, gleichzeitig sollen Services aber auch direkt mit diesen Daten interagieren bzw. auf diesen Daten ausgeführt werden. Hierzu gehören ein Information Retrieval-Service, ein Datenaustausch-Service, sowie ein Datenqualitätsservice [GrEE2021].

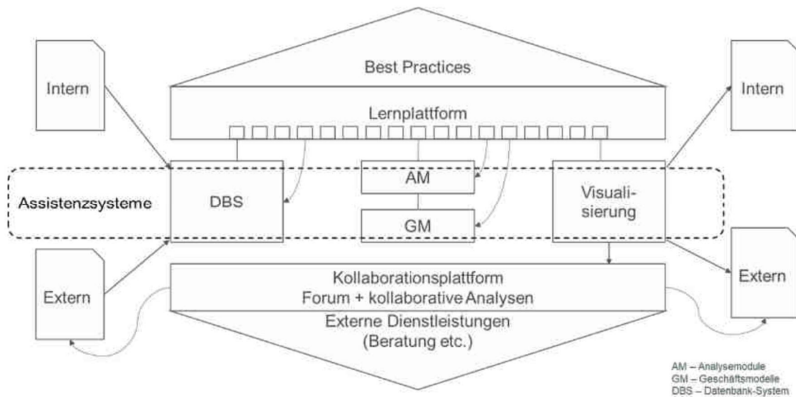


Abbildung 3-23: SP²IDER als Datenbacken-System in Analyse-Prozesse aus [EiGE2020]

Weiterhin kann SP²IDER auch in einem breiteren Kontext und in Interaktion mit IT-ketten eingesetzt werden, wie in Abbildung 3-23: SP²IDER als Datenbacken-System in Analyse-Prozesse aus [EiGE2020] gezeigt. Hier wird SP²IDER als Datenbackend-System im Rahmen einer industriellen Datenanalyse-Plattform eingesetzt, welche eine Vielzahl von Tools vereint, um die Nutzung der Technologie Datenanalyse zu vereinfachen [EiGE2020].

Für die Durchführung von Datenanalysen kann SP²IDER als Datenquelle genutzt werden und über eine REST-Schnittstelle aufbereitete Daten weiterleiten. Ein besonderer Mehrwert entsteht durch ein im Rahmen des Projekts AKKORD entwickeltes Schema, welches die Datenbedarfe spezifiziert und in einem JSON-Objekt verschlüsselt [EiEG2022].

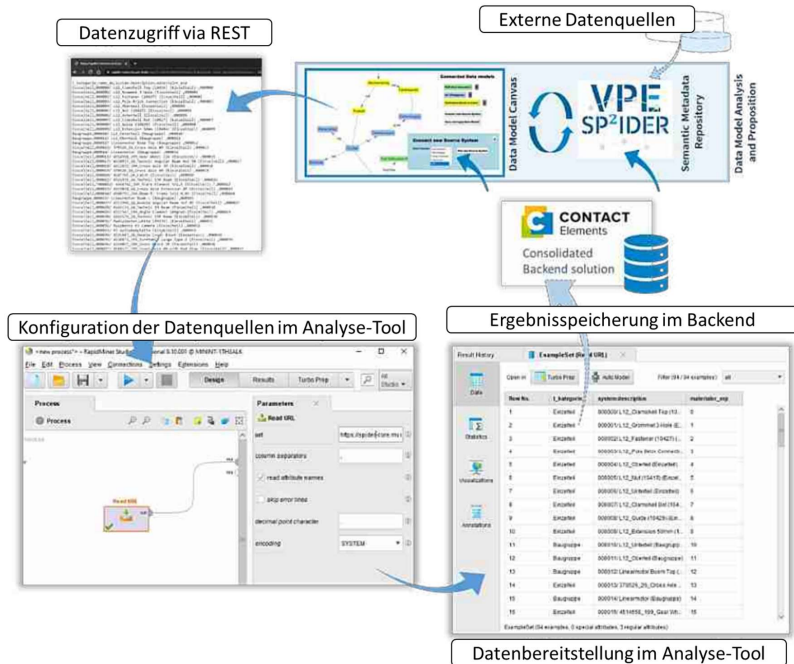


Abbildung 3-24: Datenanalysepipeline basierend auf SP²IDER nach [EIEG2022]

Abbildung 3-24 zeigt hierbei den Ablauf bei der Nutzung der Datenanalysepipeline. Zuerst definiert der Nutzer mithilfe des UI seine Datenanalysebedarfe und speichert das dabei entstandene Modul in Form eines JSON-Objekts. In diesem Objekt ist eine URL verschlüsselt, welche über einen REST-Endpunkt die gewünschten Daten als Liste bereitstellt. Diese Liste kann von einem geeigneten IT-Werkzeugen wie z.B. Rapidminer Studio geladen und die enthaltenen Daten ausgewertet werden. Die Ergebnisse einer solchen Analyse können anschließend wieder in einem Datenbackend gespeichert werden [EIEG2022].

3.2.14. Semantic Data Management im Kontext Smarter Produkte

Abramovici et al. [AbGD2016] zeigen auf, dass für die Entwicklung und insbesondere die Rekonfiguration Smarter Produkte in Zukunft eine breitere Informationsbasis notwendig ist. So werden Komponenteninformationen, Architekturinformationen und Nutzungsdaten der Smarten Produkte benötigt. Komponenteninformationen sind dabei alle Informationen über die Kernkomponenten des Smarten Produkts und seiner Nutzungsszenarien, die Architekturinformationen zeigen die Integration eines Smarten Produkts in ein Gesamtsystem und die sich daraus ergebenden Beeinflussungen. Die Nutzungsinformationen wurden anhand von Daten erhoben, welche bei der Nutzung entstehen, z.B. durch Serviceberichte oder

Aufzeichnungen von Ressourcenverbräuchen. Neben dem Erheben dieser Daten ergibt sich für eine erfolgreiche Entwicklung und Rekonfiguration der Smarten Produkte die Herausforderung der Informationsverknüpfung [AbGD2016].

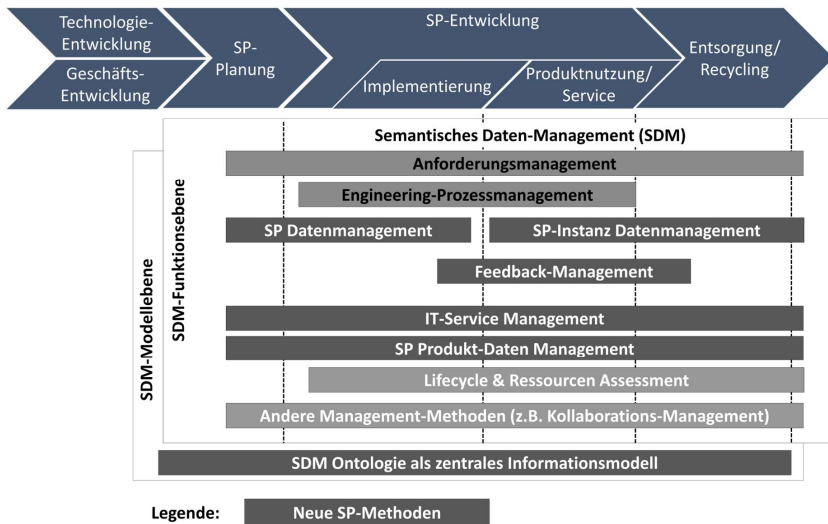


Abbildung 3-25: Semantic Data Management (SDM)-Ansatz nach [AbGD2016]

Abbildung 3-25 zeigt den hier gewählten Lösungsansatz Semantic Data Management. Im Gegensatz zu einem klassischen PLM-Ansatz werden im SDM nicht nur Lebenszyklen von virtuellen, sondern auch von realen Produkten verwaltet. Dies zeigt sich vor allem in den Methoden des SP Instance Data Managements, welches die oben beschriebenen SP-Nutzungsinformationen verwaltet und diese über das Feedback-Management dem SP Data Management, welches die Komponenteninformationen verwaltet, bereitstellt [AbGD2016].

Um diesen SDM-Ansatz zu realisieren, wurde im Kern eine SDM-Ontologie geschaffen, welche als Informationsmodell für die Verknüpfung der unterschiedlichen Daten und der Interoperabilität im Unternehmenskontext fungiert. Innerhalb dieser Ontologie gibt es Konzepte für Smarte Produkte, Smarte Produktinstanzen, Prozesse, Ressourcen sowie Organisationen. Die Umsetzung dieser Ontologie soll auf einer IT-Plattform mit Hilfe einer Graph-Datenbank und Mikroservices stattfinden [AbGD2016].

Nach dem Aufbau eines solchen semantischen Datenmanagements kann es nicht nur zur Verwaltung von Smarten Produkten, sondern auch generell zur Informationsbeschaffung verwendet werden. Hierbei liegt ein Fokus insbesondere auf der Suche in unstrukturierten Dokumenten, da diese in großer Zahl während des Lebenszyklus entstehen und gleichzeitig nur schwer nutzbar gemacht werden können. Um als Informationsquelle zu dienen, ist es umso

wichtiger, dass diese Dokumente mit entsprechenden Metainformationen versehen und in den passenden SDM-Kontext eingebunden werden [AbGG2016].

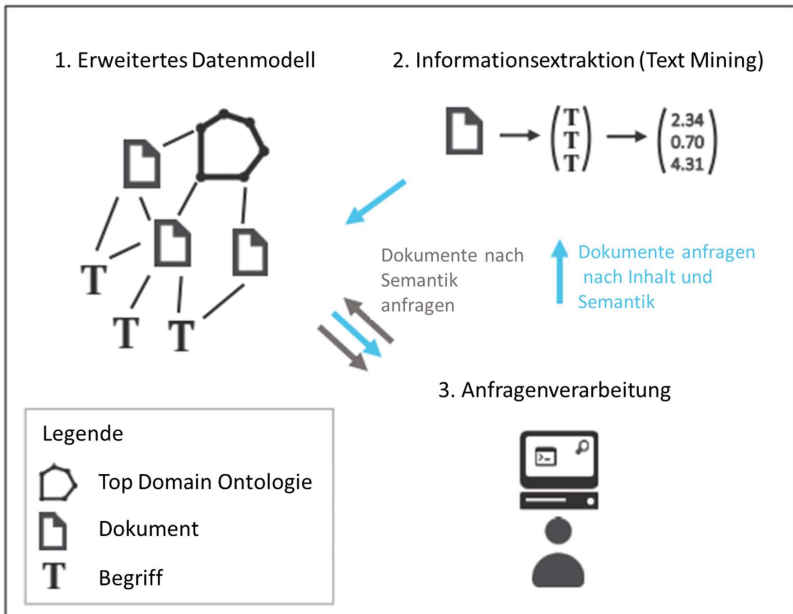


Abbildung 3-26: Komponenten für die Informationsbereitstellung nach [AbGG2016]

Abbildung 3-26 zeigt die drei Hauptkomponenten, welche eine Voraussetzung für die Nutzung als Informationsquelle sind. Nach einer Anpassung des Datenmodells (1) müssen alle Dokumente via Text Mining analysiert und die Einbindung in das Datenmodell vorgenommen werden. Zuletzt gibt es eine Methodik, welche die Suchanfragen des Users so übersetzt, dass alle betroffenen Informationen gefunden werden [AbGG2016].

4. Anforderungen an ein Knowledge Graph basiertes Assistenzsystem

Der Blick auf die heutige (Engineering)-IT-Landschaft in Unternehmen, insbesondere in größeren Konzernen ist geprägt von Heterogenität und einem Mangel an Integrationslösungen, welche den Datenzugriff in verschiedene Systeme ermöglichen, und Workflows unterstützen. Doch mit einer bloßen Bereitstellung von Integration ist es noch nicht getan. Die Mitarbeitenden müssen dazu befähigt werden, sich einen Überblick über die vorhandenen Daten zu verschaffen. Gerade durch Integration steigt die Menge an verfügbaren Daten weiter an, während Mitarbeitende meistens bereits die Möglichkeiten der im Einsatz befindlichen Quellsysteme nicht vollumfänglich kennen.

Bestehende Lösungen der Datenintegration im Bereich des Engineerings beschränken sich vor allem auf das Bereitstellen von Daten. Data Lakes oder Data Warehouses dienen dazu große Mengen Daten in einem IT-Gerüst zu vereinigen, um Zugriffe darauf zu vereinfachen, eine Beschreibung der vorhandenen Daten oder eine tiefergehende Verknüpfung mit Wissen bieten diese Systeme nicht. Eine mögliche Lösung dieser Problematik liegt in der Entwicklung eines Knowledge Graph basierten Assistenzsystems, welches den Nutzer befähigt Wissen über die Unternehmensprozesse mit der realen IT-Landschaft und einem direkten Datenzugriff im Sinne einer Integrationslösung zu kombinieren. In diesem Kapitel werden Anforderungen an eine solche Lösung gesammelt und im Detail beschrieben.

4.1. Ermittlung der Anforderungen

Die Ermittlung der Anforderungen an die Entwicklung eines Knowledge Graph basierten Assistenzsystems beruht auf unterschiedlichen Strategien. Die erste Sammlung von Anforderungen basiert auf einer Literaturrecherche, um Herausforderungen in den betroffenen Bereichen zu sammeln und auch mit bereits bestehenden Lösungsansätzen abzugleichen, aus welchen sich Weiterentwicklungspotential ergibt.

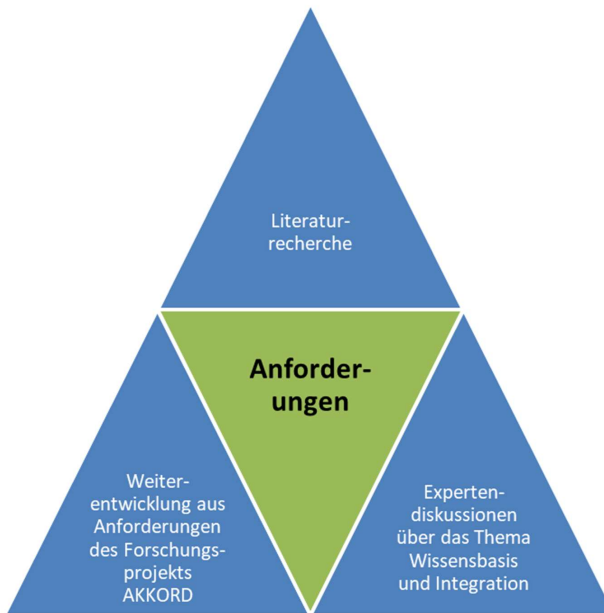


Abbildung 4-1: Strategien zur Anforderungserhebung

Daneben wurde in der Ermittlung von Anforderungen auf die erfassten Bedarfe an eine Datenbackendlösung im Projekt AKKORD zurückgegriffen. Diese wurde in strukturierten Interviews mit Firmenvertretern erstellt [EIGE2020] und im Rahmen dieser Arbeit weiterentwickelt. Dies ist insbesondere deswegen möglich und notwendig, da im Projekt AKKORD der Fokus auf der schnellen Durchführung von Datenanalysen und dem Aufbau einer Datenanalysepipeline lag. Durch die große fachliche Nähe konnte bei der Ermittlung der Anforderungen für das Knowledge Graph basierte Assistenzsystem auch auf diese Arbeit zurückgegriffen werden und durch Weiterentwicklung einige Anforderungen hieraus abgeleitet werden.

Zuletzt wurde das Thema neben der Literaturrecherche und der Weiterentwicklung aus den Anforderungen des AKKORD Projekts auch regelmäßig mit weiteren Experten diskutiert. Diese Diskussionen fanden meistens im Rahmen von persönlichen Gesprächen auf verschiedenen Konferenzen, aber auch intern am Lehrstuhl für Virtuelle Produktentwicklung oder mit Industriepartnern statt. Teilweise wurden diese Gespräche vorbereitet und in Form eines strukturierten Interviews oder anhand von Leitfragen durchgeführt, die Mehrzahl der Gespräche fand allerdings in einem unstrukturierten Rahmen statt, so dass anschließend die wichtigsten Ergebnisse in Form von Notizen oder Gedächtnisprotokollen vorlagen.

Über die drei beschriebenen Strategien hinaus sind auch Anforderungen und Wünsche aus der persönlichen Vorstellung, insbesondere auf Grundlage der geführten Gespräche und ausgewerteter Literatur in den folgenden Listen enthalten.

Abbildung 4-1 zeigt die drei wichtigsten Strategien, welche zur Erfassung der Anforderungen genutzt wurden. Das Ergebnis dieses Anforderungserhebungsprozesses sind die in diesem Kapitel näher beschriebenen Anforderungen, welche sich in mehrere Bereiche aufteilen lassen.

4.2. Grundlegende Anforderungen

Die übergeordneten Anforderungen werden zusammengefasst als grundlegende Anforderungen betrachtet. Sie beschreiben die Lösung auf einer sehr groben Ebene, die sich noch nicht im Detail mit der methodischen oder IT-technischen Umsetzung befasst.

Tabelle 4-1: Grundlegende Anforderungen

| Nr. | Anforderung | Gewichtung |
|--------|---|------------|
| AN-K01 | Die Lösung soll eine Modellierung der gewünschten und benötigten Daten ermöglichen | Muss |
| AN-K02 | Es soll eine Zuordnung zwischen der Modellschicht und der Ist-Darstellung der IT-Quellsysteme vorhanden sein | Muss |
| AN-K03 | Die Modelle sollen sich aus den Engineering-Prozessen ableiten | Soll |
| AN-K04 | Die Lösung soll universell einsetzbar sein, nicht nur für eine bestimmte Art von Prozess, Daten oder Unternehmen eingesetzt werden können | Muss |
| AN-K05 | Es müssen verschiedene Nutzer unterstützt werden, u.a. Ingenieure, die die initiale Vernetzung bereitstellen, aber auch z.B. Domänenexperten, die später mit der vernetzten Lösung arbeiten | Muss |
| AN-K06 | Die Lösung soll in der Lage sein, nicht nur bestehendes Wissen abzubilden, sondern auch neues Wissen zu schaffen | Soll |
| AN-K07 | Modelle müssen veränderbar und erweiterbar sein | Muss |
| AN-K08 | Die Lösung muss so gestaltet sein, dass unterschiedliche Nutzer nur einen Teil des Modells sehen dürfen | Soll |
| AN-K09 | Die Lösung sollte idealerweise auch Datenanalyse-Prozesse wie CRISP-DM oder ASUM-DM unterstützen | Soll |

Kern der Lösung soll die Modellierung von Daten sein (AN-K01), welche anschließend mit einem bereits vorhandenen Metadaten-Repository verbunden werden (AN-K02). Die Modellierung soll dabei nicht ohne eine bereits vorhandene Quelle ablaufen, sondern soll sich aus beschriebenen (Engineering)-Prozessen ableiten (AN-K03).

Eine wichtige Anforderung der Lösung ist die universelle Einsetzbarkeit (AN-K04). Eine eng begrenzte Einsetzbarkeit würde zwangsläufig dazu führen, dass eine neue Lösung nicht zukunftssicher ist und bei einer geringen Änderung der Randbedingungen wahrscheinlich nicht mehr einsetzbar ist.

Auf die Einsetzbarkeit zielt auch Anforderung AN-K05 ab. Die Lösung soll verschiedene Nutzer unterstützen. Eine Fokussierung nur auf eine bestimmte Nutzergruppe, wie z.B. Dateningenieure würde das Erstellen der Modelle zum Selbstzweck bedeuten. Nur Domäneningenieure oder ähnliche Nutzer anzusprechen würde wahrscheinlich die Umsetzung erschweren. Insofern sichert die breite Nutzbarkeit für mehrere Rollen auch die Nutzbarkeit der gesamten Lösung ab.

Die Lösung soll weiterhin in der Lage sein neues Wissen zu schaffen und nicht nur bestehendes Wissen abzubilden (AN-K06). Dies äußert sich darin, dass eine reine Modellierung von Daten noch kein neues Wissen schafft, aber die Verbindung von Datenmodellen mit der Information über deren Speicherorte dies ermöglicht.

Weiterhin müssen sich die Modelle auch an veränderte Umstände angepasst werden können und müssen daher veränderbar sein (AN-K07). Dies ist zum Beispiel dann erforderlich, falls neue IT-Systeme und Daten vernetzt werden müssen, sollten sich die zugrunde liegenden Prozesse verändern oder

Eine zugrunde liegende Motivation für die hier zu entwickelnde Lösung ist die Komplexität der vorhandenen Daten in einem bestehenden Engineering Repository. Dementsprechend ist es erforderlich, dass die Nutzer in dieser Lösung nur solche Dinge sehen, die für sie relevant sind und für die sie die Erlaubnis haben diese Objekte zu sehen. Es kann solche Situationen geben, in denen ein Nutzer zwar Dinge als relevant empfinden, aber es Gründe gibt, die dagegensprechen, dass der Nutzer sie sieht. Dies könnte z.B. bei einer Kollaboration über Firmengrenzen hinweg der Fall sein (AN-K08).

Zuletzt aufgrund der Entstehungsgeschichte dieser Lösung soll sie auch eng an die Prozesse der Datenanalyse angelehnt sein. In diesen Prozessen ist das Verstehen von Daten und das Wissen um die Verfügbarkeit zentraler Bestandteil des Prozessmodells, z.B. in der Phase des Data Understandings. An dieser Stelle soll diese Lösung auch durch ihren Aufbau die Identifikation von Datenquellen unterstützen (AN-K09).

4.3. Bewertung Stand der Technik hinsichtlich der grundlegenden Anforderungen

Im Stand der Technik zeigt sich, dass bereits eine Vielzahl an Methoden und Tools existieren, welche entweder zur Beschreibung von Daten oder zur Nutzung semantisch angereicherter Daten verwendet werden können. Die dort vorgestellten Ansätze sind in ihrem Umfang und ihrer Art sehr unterschiedlich ausgeprägt, weswegen eine Bewertung hinsichtlich ihrer Eignung im Rahmen dieses Konzepts methodisch geplant werden muss. Keiner der vorgestellten Ansätze hat exakt den gleichen Fokus wie das hier angestrebte Konzept, insofern können die möglichen Lösungselemente auch nur eine Teillösung darstellen.

Weiterhin wurde eine Vorauswahl getroffen und nicht jedes einzelne Tool oder jeder Ansatz bewertet, sondern nur die erfolgversprechendsten. Die grundlegenden Technologien zum Beschreiben von Daten wie z.B. Semantiken, Ontologien oder der Dublin Core wurden aufgrund ihrer Einfachheit nicht weiter betrachtet.

Aus der zweiten Kategorie, Ansätze zur Nutzung semantisch angereicherter Daten wurden vier Gruppen von Ansätzen ausgewählt: Integrationsplattformen, Knowledge-Graph-, Datenanalyse- und Mappinglösungen. In jeder Kategorie gibt es verschiedene betrachtete Elemente, hier wurde basierend auf den Rechercheergebnissen des Stands der Technik die jeweils vielversprechendsten Ansätze für eine Bewertung ausgewählt.

Die im Stand der Technik beschriebenen Forschungsansätze zur besseren Nutzung semantisch angereicherter Daten wurden in dieser Bewertung nicht verglichen, da sie –auf einer sehr allgemeinen Ebene – ein ähnliches Ziel verfolgen, aber sich dabei auf andere Methoden und Konzepte fokussieren.

In Bezug auf die erste Anforderung, eine Modellierung der notwendigen Daten, zeigen sich bereits die Unterschiede der Lösungselemente. Bereits bestehende Plattformen bzw. Datenbanklösungen sind in der Lage, eine Datenmodellierung vorzunehmen. Auch im Bereich Knowledge Graphen ist die Datenmodellierung Teil des Konzepts, wobei der aufgeführte Google Knowledge Graph herausfällt, da in ihm nicht durch jede Person Änderungen vorgenommen werden können.

Die zweite Anforderung, die Zuordnung zwischen einer Modellschicht und der Ist-Situation, wie sie in (Engineering)-Datenbanken abgebildet ist zu erschaffen, können anknüpfend an die erste Anforderung vor allem solche Technologien erfüllen, die auch eine Datenbank sind oder entsprechend Daten verwalten. Weiterhin sind die Mapping-Technologien geeignet, um Verknüpfungen abzubilden.

Die Anforderung der Ableitung aus Engineering-Prozessen ist ebenfalls vor allem bei den Technologien erfüllt, die bereits in diesem Umfeld eingesetzt werden, wie z.B. Integrationsplattformen. Ebenfalls gut erfüllen auch Datenanalyse-Lösungen wie die AKKORD-Toolbox diese Anforderung, da sie mit dem Hintergrund der Einsetzbarkeit in der industriellen Datenanalyse entwickelt wurde.

Eine universelle Einsetzbarkeit ist allen Lösungen mehr oder weniger gegeben, da sie entweder allgemein sind oder in ihrer Struktur so viel Veränderungsmöglichkeiten da sind, dass sie auch für andere als den ursprünglichen Einsatzzweck genutzt werden können.

Die Anforderung der Unterstützung verschiedener Nutzer wird von den (IT)-Lösungen erfüllt, welche einen etwas komplexeren Umfang haben und entsprechende Funktionen eingebaut haben. Lösungen auf rein methodischer Ebene sind an dieser Stelle eher agnostisch und sehen keine Nutzer vor, erfüllen diese Anforderung entsprechend nicht.

Die Anforderung der Veränderbarkeit können fast alle Lösungen in Teilen oder in Gänze erfüllen. Vor allem der Google Knowledge Graph ist im Sinne dieser Lösung die Ausnahme. Er ist zwar erweiterbar, aber dies kann nicht durch den Nutzer durchgeführt werden.

Die Steuerung der Sichtbarkeit, so dass verschiedene Nutzer unterschiedliche Elemente sehen und nutzen können erfüllen die meisten Lösungselemente teilweise oder ganz. Schwierig ist diese Anforderung wiederum bei solchen Technologien, die methodisch sind und daher agnostisch gegenüber Nutzern.

| | | AN-K01: Die Lösung soll eine Modellierung der gewünschten und benötigten Daten ermöglichen | AN-K02: Es soll eine Zuordnung zwischen der Modellschicht und der Ist-Darstellung der IT-Quellsysteme vorhanden sein | AN-K03: Die Modelle sollen sich aus den Engineering-Prozessen ableiten | AN-K04: Die Lösung soll universell einsetzbar sein, nicht nur für eine bestimmte Art von Prozess, Daten oder Unternehmen eingesetzt werden können | AN-K05: Es muss verschiedene Nutzer unterstützen, u.a. Ingenieure, die die initiale Vernetzung bereitstellen, aber auch z.B. Domänenexperten, die später mit der vernetzten Lösung arbeiten | AN-K06: Das Konzept soll in der Lage sein, nicht nur bestehendes Wissen abzubilden, sondern auch neues Wissen zu schaffen | AN-K07: Modelle müssen veränderbar und erweiterbar sein | AN-K08: Das Konzept muss so gestaltet sein, dass unterschiedliche Nutzer nur einen Teil des Modells sehen dürfen | AN-K09: Die Lösung sollte idealerweise auch Datenanalyse-Prozesse wie CRISP-DM oder ASUM-DM unterstützen |
|--------------------------|---------------------------|--|--|--|---|---|---|---|--|--|
| Integrationsplattformen | CIMDatabase | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ |
| | Aras Innovator | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ |
| | SP ² IDER | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ |
| Knowledge Graph Lösungen | Graphdatenbanken | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ |
| | Linked Data Ansätze | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ |
| | Google Knowledge Graph | ○ | ○ | ○ | ◐ | ○ | ○ | ○ | ○ | ○ |
| Datenanalyse Lösungen | Rapidminer Studio | ○ | ○ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ |
| | AKKORD-Toolbox | ○ | ○ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ |
| Mapping-Lösungen | manuelle Mapping-Tabellen | ○ | ◐ | ○ | ◐ | ○ | ◐ | ◐ | ○ | ○ |
| | Mapping von Zeichenketten | ○ | ◐ | ○ | ◐ | ○ | ◐ | ◐ | ○ | ○ |
| | Mapping von Wissen | ○ | ◐ | ◐ | ◐ | ◐ | ◐ | ◐ | ○ | ○ |

Legende

nicht erfüllt ○

teilweise erfüllt ◐ ◑ ◒

vollumfänglich erfüllt ●

Abbildung 4-2: Bewertung von Lösungselementen in Bezug auf die Anforderungen

Es zeigt sich, dass es keine Lösung auf dem Markt gibt, bzw. auch die im Stand der Forschung und Technik recherchierten Ansätze nicht in der Lage sind sämtliche grundlegenden Anforderungen an ein Knowledge-Graph basiertes Assistenzsystem zur Datenanalyse-Unterstützung für das Engineering smarter Produkte zu erfüllen. In einzelnen Punkten sind die bereits vorhandenen Lösungselemente allerdings geeignet. Eine Gesamtlösung zur Erfüllung aller Anforderungen kann sich dementsprechend an den evaluierten Technologien und vorhandenen Lösungselementen bedienen.

Daher werden in den weiteren Unterkapiteln werden noch detaillierte Anforderungen in den Bereichen IT-Anforderungen und Methodik erhoben, um die dann entwickelte Lösung daraufhin zu evaluieren. Diese wurden nicht mit dem aktuellen Stand der Technik abgeglichen, da ja bereits die grundlegenden Anforderungen nicht ausreichend von einer Lösung erfüllt werden können, eine detaillierte Bewertung daher keinen Mehrwert erzeugt.

4.4. Informationstechnische Anforderungen

Die Lösung im Rahmen dieser Arbeit soll nicht nur aus einem Konzept bestehen, sondern dieses auch in Form eines Software-Demonstrators umsetzen. Die sich hieraus ergebenden Anforderungen werden in diesem Unterkapitel beschrieben.

Tabelle 4-2: IT-Anforderungen

| Nr. | Anforderung | Gewichtung |
|---------|--|------------|
| AN-IT01 | Die Lösung muss multilingual einsetzbar sein | Soll |
| AN-IT02 | Die Lösung soll benutzerfreundlich sein | Soll |
| AN-IT03 | Daten sollen nicht dupliziert werden, sondern in ihrem IT-Quellsystem belassen werden | Muss |
| AN-IT04 | Das System soll über eine Schnittstelle verfügen, um ggf. weitere Services zu ermöglichen, bzw. eine bidirektionale Integration in andere Systeme zu ermöglichen | Muss |
| AN-IT05 | Das System sollte server- und webbasiert funktionieren, um unternehmens- und standortübergreifend zu arbeiten | Muss |
| AN-IT06 | Das System soll aus Open Source Komponenten aufgebaut sein, um keine Lizenzkosten zu erzeugen | Soll |
| AN-IT07 | Das System soll einen Service bereitstellen, der es ermöglicht einen Wissensgraphen anzulegen und abzuspeichern | Muss |
| AN-IT08 | Das System soll einen Service bereitstellen, welcher Datentypen in unterschiedlichen Quellsystemen inhaltlich zu Wissens-elementen zuordnen kann | Muss |

Die IT-technischen Anforderungen, welche in Tabelle 4-2 aufgelistet werden, dienen dazu eine Umsetzung des Lösungskonzeptes auch in Form eines Software-Demonstrators umzusetzen, welcher das Prinzip demonstriert und die Ergebnisse veranschaulicht, sowie zur Verifikation und Validierung eingesetzt werden kann.

Die ersten beiden Anforderungen können als allgemein im Bereich von Industrie-Software angesehen werden. Um einen möglichst breiten Kreis an Nutzern anzusprechen ist es sinnvoll,

dass die Lösung nicht nur in einer Sprache zur Verfügung steht, sondern idealerweise auf die gewünschte Sprache des Nutzers eingestellt werden kann.

Die Anforderung, dass ein System, welches einen Knowledge Graph in ein Metadaten-Repository integriert keine Kopien der Daten selbst speichern sollte liegt im Konzept begründet. Eine Speicherung von gesamten Datenbeständen stünde im Widerspruch zum Metadaten-Repository und muss verhindert werden.

Das System sollte über eine Schnittstelle verfügen, um weitere Services zu ermöglichen, da mit diesem Konzept eine Menge Daten in Form eines Graphen angelegt werden. Diese Daten können als Grundlage für weitere Services dienen und müssen dementsprechend auch für weitere Anwendungen erreichbar sein.

Das System sollte server- und webbasiert funktionieren. Gerade mit Hinblick auf die Architektur der verschiedenen anzubindenden Datenquellsysteme, welche im Engineering-Bereich mittlerweile sehr stark webbasiert sind, sowie die Anforderungen nach Schnittstellen sind neben der standortübergreifenden und unternehmensübergreifenden Anwendbarkeit Gründe, warum eine Serverstruktur zu bevorzugen ist.

Um eine schnelle und unkomplizierte Umsetzbarkeit des Systems zu gewährleisten, sollte beim Aufbau des Programms vor allem auf Open Source Komponenten zurückgegriffen werden. Solche Komponenten können durch ihre Lizenzen im Internet heruntergeladen und genutzt werden, sei es in Form eines Forschungsdemonstrators oder in Form einer kommerziellen Software.

Als Kern des Konzepts soll ein Knowledge Graph angelegt werden. Dementsprechend muss auch die Software in der Lage sein diesen Knowledge Graph anzulegen und einen Service bereitzustellen, der dies in einem einfachen Prozess auch unterstützt.

Zuletzt soll die Software auch einen Service bereitzustellen, welcher in der Lage ist, Verknüpfungen zwischen dem modellierten Knowledge Graph und den Objekten des Metadaten-Repository zu ziehen.

4.5. Anforderungen an die Methodik

Die integrierte Gesamtlösung, die im Rahmen dieser Arbeit entwickelt werden soll, besteht nicht nur aus einem Konzept und einer IT-Lösung, sondern hat auch einen methodischen Ansatz, welcher eigene Anforderungen erfüllen muss, welche hier im Folgenden beschrieben werden. Methodische Anforderungen sollen insbesondere die grundlegenden Anforderungen aus Tabelle 4-1 genauer beschreiben.

Tabelle 4-3: Anforderungen an die Methodik

| Nr. | Anforderung | Gewichtung |
|--------|--|------------|
| AN-M01 | Die Lösung muss multilingual einsetzbar sein | Soll |

| | | |
|--------|---|------|
| AN-M02 | Die Methodik muss für unterschiedliche Datentypen funktionieren, z.B. für Daten aus einem PLM-System oder aus einer Zeitreihendatenbank | Soll |
| AN-M03 | Die Methodik soll teilautomatisiert sein und Zuordnungen von Datentypen aus der Modellsicht und der IT-Sicht algorithmisch unterstützen | Muss |

Die in Tabelle 4-3 beschriebenen methodischen Anforderungen dienen dazu, das gewünschte Ergebnis der Darstellung von Wissen über Inhalte und Bedeutung von Daten in IT-Systemen abzusichern. Hierbei ist mit Methodik sowohl die Summe an Vorgängen gemeint, welche der Nutzer selbst durchführt, aber auch alle Vorgänge, welche algorithmisch bzw. durch Software unterstützt werden.

Da es sich bei den IT-Systemen nicht ausschließlich um Softwareprodukte aus Deutschland bzw. auf Deutsch handelt, kann nicht davon ausgegangen werden, dass alle entsprechend zu verarbeiteten Informationen auf Deutsch vorliegen. Die Lösung muss daher multilingual einsetzbar sein, insbesondere beim Aufspüren von Ähnlichkeiten zwischen Wissens-elementen und Datentypen in IT-Systemen. Diese Datentypen sind teilweise auch innerhalb eines Systems auch in unterschiedlichen Sprachen angelegt, häufig verwendet wird dabei die englische Sprache (AN-M01).

Ebenfalls soll die Methode hinsichtlich der Datentypen universell sein. Im Unternehmensumfeld sind unterschiedlichste Daten relevant, seien es strukturierte Daten aus der Produktentwicklung, Zeitreihendaten mit Sensorwerten aus Digitalen Zwillingen bzw. der Nutzungsphase oder unstrukturierte Daten in Form von z.B. Serviceberichten. Insofern muss auch eine Lösung, die diese Daten beschreibt in der Lage sein mit entsprechenden Daten zu funktionieren, eine Anbindung sicherstellen, sowie eine Beschreibung der Daten und ihre Einbindung in Produkte und Prozesse (AN-M02).

Angesichts der großen Mengen an Daten und auch an Datentypen in Systemen soll die Methodik nicht komplett manuell Zuordnungen durchführen, sondern soll teilautomatisiert Daten auswerten und eine Übereinstimmung feststellen, um anschließend die Ergebnisse durch den Nutzer verifizieren zu lassen und zur Entscheidungsgrundlage über die Vernetzungsinformation zu machen (AN-M03).

5. Konzept für ein Knowledge Graph basiertes Assistenzsystem

In den vorhergehenden Kapiteln dieser Arbeit wurde die Notwendigkeit gezeigt, ein Assistenzsystem für die Arbeit mit (Engineering)-Daten zu gestalten und zu nutzen. Die bereits existierenden Lösungen können diese Lücke nicht schließen. Es zeigt sich ausgehend von den Anforderungen, welche im vorherigen Kapitel erhoben wurden, dass ein neues Gesamtkonzept einen Beitrag zur Lösung dieses Problems sein kann.

Die Ist-Situation sieht in vielen Unternehmen bereits verschiedene IT-Systeme vor, in welchen Daten abgespeichert sind. Im besten Fall verfügt die Firma bereits über ein Metadaten-Repository, welches Daten aus verschiedenen Quellen aggregiert. Demgegenüber stehen die Prozesse, welche nicht direkt mit den Daten verknüpft sind.

Um diese Verknüpfung herzustellen, gibt es ein Gesamtkonzept aus mehreren Lösungsbausteinen, welches in Abbildung 5-1 dargestellt ist. Durch die Kombination verschiedener Elemente, unter anderem durch eine Erweiterung des vorhandenen Metadaten-Repository, aber auch durch die Bausteine Data Model Canvas und den Ähnlichkeitsalgorithmus. Im Data Model Canvas wird ein Knowledge Graph mit dem Wissensmodell des Unternehmens erstellt und verwaltet, sowie die Ist-Sicht auf die Daten verknüpft. Der Ähnlichkeitsalgorithmus als zweiter Baustein schafft dann eine detaillierte Verknüpfung zwischen den vorhandenen Daten und dem neuen Wissensmodell.

Diese beiden Lösungsbausteine werden so in der IT-Architektur des Unternehmens eingebunden, dass der Data Model Canvas eine Erweiterung des Metadaten-Repository ist und man mit dem neuen Wissensmodell und der Verknüpfung hin zu den bestehenden Systemen arbeiten kann. Der Kern der hier vorgestellten Lösung ist der Knowledge Graph – das Wissensmodell, welcher dann die Verbindung zwischen der IT-Sicht und den unternehmenseigenen Prozessen schafft.

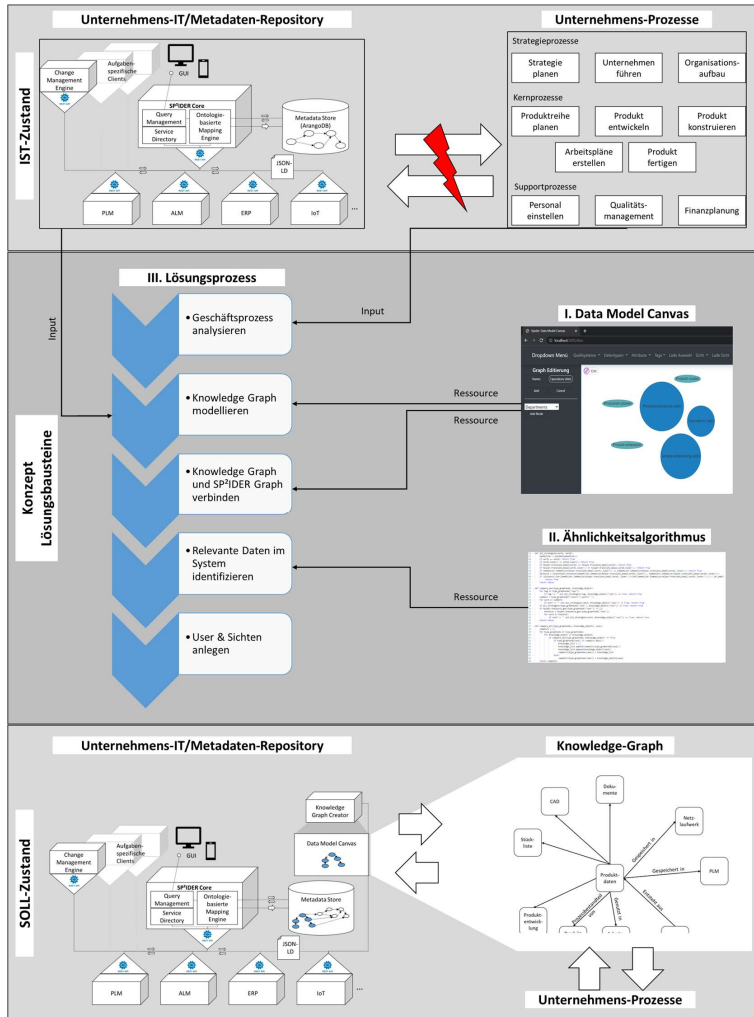


Abbildung 5-1: Grundbausteine des Lösungskonzepts und deren Zusammenhang

Zur Unterstützung der Nutzerfreundlichkeit gibt es ein Sichten-Konzept, welches sicherstellt, dass alle beteiligten Personen und Rollen nur die Informationen des Knowledge-Graphs und der IT-Systeme sehen, die für ihre jeweilige Arbeit relevant sind. Weitere Daten werden ausgeblendet und nicht angezeigt.

5.1. Knowledge Graph basierte Modellierungsumgebung

Als Teil des Konzepts zur Abbildung und Zuordnung der vorhandenen IT-Objekte in allen Datenquellsystemen auf das gewünschte Wissen, welches für Engineering-Prozesse relevant ist, soll ein Knowledge Graph aufgebaut werden. Gemäß der Logik Subjekt – Prädikat – Objekt besteht der Knowledge Graph aus Knoten, welche entweder Subjekt oder Objekt sind und Kanten, welche die Verbindung zwischen den Knoten spezifisch beschreiben. Eine Verbindung kann somit immer als Verknüpfung und Satz ausgedrückt werden.

Im Gegensatz zu universellen Knowledge Graphen, welche versuchen alle theoretisch denkbaren Verknüpfungen und Objekte zu integrieren, ist der hier vorgeschlagene Knowledge Graph eher von kleinem Umfang und besitzt nur ein begrenztes Set an Typen, von deren Art Subjekte und Objekte existieren. Auch die Vielfalt der möglichen Verbindungen wurde bewusst auf einige wenige reduziert, um das Erstellen des Knowledge Graphen handhabbar zu gestalten.

Der Knowledge-Graph bildet so in seiner Summe ein Modell des Wissens um die vorhandenen Daten. Im Sinne Stachowiaks repräsentiert dieses Modell dabei das Original, welches in Form der Prozessbeschreibung vorliegt. Diese Prozessbeschreibung ist ebenfalls ein Modell, welches das Original in Form von Prozessen und Handlungsanweisungen abbildet [Stac1973].

Neben dem bereits erfüllten Abbildungsmerkmal erfüllt der Knowledge-Graph auch die anderen Modellmerkmale der Verkürzung und des Pragmatismus. Eine Verkürzung ist dann gegeben, wenn nicht alle Merkmale des Originals übernommen werden. Das pragmatische Merkmal bedeutet darüber hinaus, dass ein Modell nicht allgemeingültig ist, sondern immer nur in einem bestimmten Zeitraum oder unter weiteren Einschränkungen gültig ist [Stac1973].

Mit dem Ziel eine geeignete Repräsentation für die Unternehmen gespeicherten Daten zu modellieren, stellt sich die Frage, was eine geeignete Ausgangsbasis für diesen Knowledge Graphen ist, bzw. wie dieser erstellt werden soll. Da eine Datenverwaltung und -nutzung nie ein Selbstzweck eines produzierenden Unternehmens ist, dienen sämtliche Daten und Informationen nur als Unterstützung im wertschöpfenden Prozess, welcher im Normalfall das Anbieten eines Produkts oder einer Leistung ist.

Damit dienen die Prozessbeschreibungen als idealer Ausgangspunkt für die Erstellung des Knowledge Graphs. Diese Prozessbeschreibungen, oder alternativ die meist noch kleinteiligeren Workflows, welche so fein beschrieben sind, dass eine Automatisierung des Prozesses möglich ist [Gada2008], können analysiert und anschließend in einen Knowledge Graphen umgewandelt werden. Diese Umwandlung der Information aus einer Prozessbeschreibung hin zum Knowledge Graph sichert die einfachere semantische Nutzbarkeit ab. Während Prozessbeschreibungen in vielerlei Form existieren, z.B. als Grafik, als Text oder im Idealfall auch in einem Standard spezifiziert und in Form von XML-Dateien auch semantisch auswertbar, kann dies nicht als gesichert gelten und wird erst durch die Umwandlung in ein festgelegtes Schema sichergestellt. Weiterhin sind für den Knowledge Graph und die Verknüpfung mit Daten nicht alle in einer

Prozessbeschreibung erfassten Informationen relevant. Dies soll anhand der OMEGA-Notation im Folgenden gezeigt werden.

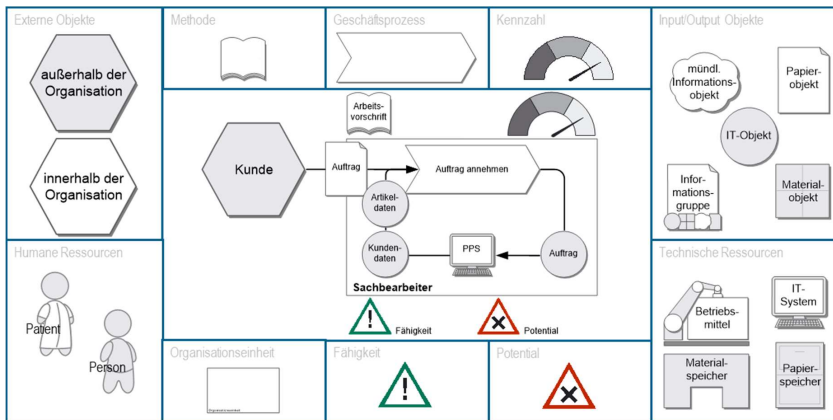


Abbildung 5-2: OMEGA Notationsobjekte

Vergleicht man die Objekte der OMEGA-Notation (vgl. [Fahr1995]) aus Abbildung 5-2 mit den im Folgenden beschriebenen Objekttypen im Knowledge-Graphen, so fällt auf, dass nicht für jedes der oben gezeigten Objekte eine Entsprechung vorliegt, in diesem Sinne ist der zu entwickelnde Knowledge Graph auch nur eine Verkürzung des ursprünglichen Prozessmodells. Seine Gültigkeit ist insbesondere dahingehend eingeschränkt, dass es sich nicht um eine Arbeitsanweisung handelt, so wie das ursprüngliche Prozessmodell an sich, sondern um ein Modell, welches die Bedeutung von in IT-Systemen abgespeicherten Datentypen beschreibt und zumindest eine direkte Verknüpfung zu diesen Daten auch nur dann besteht, sofern dieses System an das Repository angebunden ist

Eine wesentliche Eigenschaft von Graphen, die Möglichkeit Schlüsse zu ziehen aufgrund der bestehenden Verbindungen zwischen Objekten, spielt auch in diesem Konzept eine Rolle. Objekt-Typen im Knowledge Graphen werden mit Daten aus dem IT-System Bestand des Unternehmens verglichen und das Ergebnis dieser Ähnlichkeitsanalyse dient dann der weitergehenden Vernetzung von Objekten im Graph.

Um einen Knowledge-Graphen aus einem Prozessmodell aufzubauen, muss zuerst geklärt werden, welche Elemente im Detail übernommen werden sollen und welche eine Repräsentation im Graphen haben. Auf einer allgemeinen Ebene besteht ein Graph zwar nur aus Knoten und Kanten und ist daher sehr flexibel, es ist aber sinnvoll, bereits konzeptuell die unterschiedlichen Typen, welche dann als Knoten und Kante repräsentiert werden festzulegen, um die Praktikabilität des Graphen sicherzustellen.

Prozessnamen/Use-Cases

Angangsbasis für die Modellierung des Knowledge-Graphen bildet das Prozessmodell, sofern im Unternehmen ein solches existiert. Die Prozessbeschreibung sollte hierbei aus als Knoten im Graph modelliert und mit weiteren Informationen verknüpft werden. Relevant sind hierbei solche Prozesse, die im Engineering stattfinden und auf einer eher detaillierten Ebene beschrieben sind (vgl. Kapitel 2.3.3).

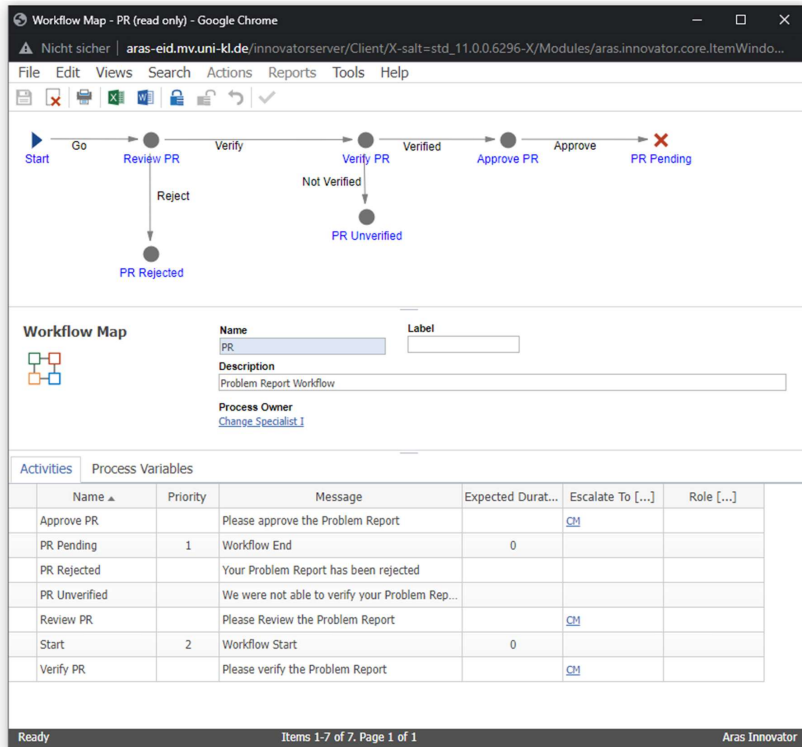


Abbildung 5-3: Beispielprozess im Engineering aus dem IT-System „Aras Innovator“

Auf der Prozess/Workflow-Ebene spielen sich die einzelnen Schritte ab, in denen mit Daten interagiert wird, um z.B. neue CAD-Modelle zu erschaffen oder bereits existierende zu ändern. Insofern sind solche Prozesse direkt mit den weiteren benötigten Typen verknüpft. Abbildung 5-3 zeigt hier beispielhaft die Prozessbeschreibung zum Bearbeiten eines Probleberichts im PLM-System Aras Innovator v11.

Datenobjekte

Datenobjekte im Sinne dieser Methodik sind diejenigen Objekte, welche während des Durchlaufens eines Prozesses oder Prozessschrittes entweder als Input benötigt oder als Output anschließend abgespeichert werden. Ihr Charakter zeichnet sich dadurch aus, dass sie bis in eine gewisse Ebene unterteilt, aber dann eine kleinstmögliche Einheit bilden. Im zu bildenden Knowledge Graphen können Datenobjekte auf verschiedenen Ebenen modelliert und miteinander hierarchisch verknüpft werden, die Objekte müssen sich nicht zwangsläufig auf einer Ebene befinden. So wäre beispielsweise „Produktdaten“ ein Datenobjekt, welches noch weiter untergliedert werden kann in Datenobjekte wie „Stückliste“, „CAD-Modelle“ oder „Anforderungen“.

Datenspeicherorte

Datenspeicherorte bezeichnen die Orte, an denen Datenobjekte gespeichert sein sollen. Dies können entweder analoge Ablageorte sein oder im wahrscheinlicheren Fall ein digitales System. Bei digitalen Systemen ist ein wichtiges Unterscheidungsmerkmal, ob die Speicherorte unstrukturiert sind, z.B. ein Netzlaufwerk, oder ob sie für ein strukturiertes System stehen, in welchem Datenobjekte auch mit Metadaten beschrieben werden und durch das System selbst eine gewisse Struktur erlangen. Beispielhafte Datenspeicherorte wären z.B. „PLM“ oder „ERP“ und können durch genauere Beschreibung spezifiziert werden, z.B. „PLM: Aras Innovator“ oder „PLM CIM Database“.

Abteilungen

Mit Abteilungen sind die Nutzer gemeint, die im Rahmen des betrachteten Prozesses Datenobjekte erstellen, manipulieren oder deren Information verwenden. Aus der Prozessbeschreibung heraus erstellt wäre eine entsprechende Analogie z.B. die Organisationseinheit der OMEGA-Notation oder der Name in einer Swimlane in der BPMN. Die Abgrenzung in diesem Typ ist nicht zwingend an die Abteilungen der Aufbauorganisation eines Unternehmens gebunden, es können auch leichte Abweichungen davon im Knowledge-Graphen eingebaut werden. Diese Unschärfe entsteht u.a. durch den Typen „Rolle“, welcher in diesem Konzept eine andere Bedeutung hat und nicht mit der Abteilung verwechselt werden sollte.

Rollen

Rollen bilden einen Objekttyp, welcher steuert, welche Objekte durch einen Nutzer gesehen werden dürfen und auf welche Objekte ein Nutzer zugreifen darf. Die Rolle steht dabei zwischen dem Nutzer und allen anderen Knoten aus dem Knowledge Graph, bzw. den Knoten aus weiteren verknüpften Metadaten-Repositories. Hier besteht eine n:m Beziehung zu den Nutzern, sprich eine Rolle kann auch durch mehrere, systemseitige Nutzer ausgefüllt werden, bzw. ein Nutzer kann mehrere Rollen zugewiesen bekommen. Die Rolle im Sinne dieses Konzepts ist von den definierten Aufgaben im Prozessablauf getrennt. Sie beschreibt lediglich die Zuordnung der sichtbaren Objekte.

Nutzer/User

Nutzer dienen der Authentifizierung gegenüber dem Knowledge Graphen, indem im Nutzer-Objekt selbst eine Name/Passwort-Kombination abgespeichert wird, die beim Aufruf des Graphen überprüft wird. Im Zusammenspiel mit den bereits definierten Rollen sind Nutzer in der Lage Objekte der verschiedenen Typen aufzurufen und ggf. auch zu verändern.

Attribute der Objekttypen

Zur Beschreibung der Objekttypen gehört nicht nur, dass ein Objekt angelegt wird, sondern es benötigt auch gewisse Attribute, um es zu beschreiben, wieder auffindbar zu machen und generell einen Inhalt zu erzeugen. Tabelle 5-1 zeigt welche Attribute zur näheren Beschreibung der verschiedenen Objekte im Knowledge Graph notwendig sind.

Tabelle 5-1: Attribute der Objekt-Typen

| Bezeichnung | Typ | Erklärung |
|-------------|---------|---|
| Name | String | Sprechender Name zum Erläutern des Datenobjekts |
| Created | Integer | Unix-Datum an dem der Eintrag angelegt wurde |
| Modified | Integer | Unix-Datum, an dem der Eintrag zuletzt bearbeitet wurde |
| Created_by | String | Name des Nutzers, der den Eintrag angelegt hat |

Das hier gewählte Set an Attributen ist bewusst so gering wie möglich gehalten, um eine Grundfunktionalität sicherzustellen, ohne den administrativen Aufwand oder die Komplexität der IT zu sehr zu steigern. Die gewählten Strukturen sind allerdings in der Lage bei Bedarf auch weitere Attribute hinzuzufügen und das Datenmodell der Knowledge-Graph-Knoten auch anzupassen. Dies ist beispielsweise bei Nutzern der Fall, da hier neben dem Namen auch noch ein Passwort gesetzt wird, welches bei Authentifizierungsvorgängen genutzt wird.

5.1.1. Beziehungen zwischen den Objekttypen

Um die einzelnen Objekte dieser Klassen zu einem Knowledge Graphen zu verknüpfen, werden Beziehungen benötigt. Hierbei soll wiederum eine möglichst geringe Anzahl an Beziehungen definiert werden, um den Nutzer nicht zu überfordern und die Methodik einfach nutzbar zu halten. Entsprechend wurden sechs verschiedene Typen von Beziehungen identifiziert, welche ausreichend sind, um den Graphen zu erstellen.

Enthält/Includes

Die „Enthält“-Beziehung dient der Verknüpfung zweier Datenobjekte untereinander. Häufig ist beim Erstellen der Prozessmodellierung und der Ausgestaltung des daraus resultierenden

Knowledge Graphen eine eher allgemeine Beschreibung vorhanden, z.B. „Produktdaten“, welche man dann weiter aufspalten kann und sagen kann, welche weiteren Datenobjekte innerhalb der „Produktdaten“ enthalten sind. Wären z.B. „CAD-Modelle“ und „Anforderungen“ noch detaillierte Ausprägungen des Datenobjekts „Produktdaten“, so würden diese Objekte jeweils mit einer „Enthält“ Beziehung miteinander verknüpft.

Entsteht aus/Made from

Die „Entsteht aus“ Beziehung drückt aus, wenn Datenobjekte als Input für weitere Datenobjekte in späteren Prozessschritten dienen und verknüpft daher zwei Datenobjekte untereinander. Beispielsweise entsteht ein Arbeitsplan unter anderem aus den Produktdaten, so dass diese Beziehung im Graph auch ausgedrückt werden sollte, um Datenobjekte entlang des Produktlebenszyklus bzw. entlang der Lebenszyklen der verschiedenen Datenobjekte zu verfolgen.

Gespeichert in/Saved In

Die „Gespeichert in“ Beziehung verknüpft Datenobjekte und Datenspeicherorte. Sie zeigt, wo die entsprechenden Objekte abgelegt sind bzw. sein sollen, die dann dementsprechend auch eine Repräsentation in einem allgemeinen Metagraph haben, welcher die IT-Systeme der Firma enthält.

Erstellt von/Genutzt von bzw. Made by/Used by

Die „Erstellt von“ bzw. „Genutzt von“ Beziehung verknüpft ein oder mehrere Datenobjekt mit Abteilungen. Es beschreibt entweder, dass ein Datenobjekt in dieser Abteilung erstellt wurde, z.B. „CAD-Modelle“ durch die Entwicklungsabteilung, oder dass Daten durch eine Abteilung genutzt werden, z.B. eben jene „CAD-Modelle“ durch die Arbeitsvorbereitung zur Erstellung eines Arbeitsplans oder eines CNC-Programms.

Prozessbestandteil von/ Part of

Die „Prozessbestandteil von“ Beziehung zwischen Datenobjekten und Use-Cases bzw. Prozessnamen zeigt auf, welche Datenobjekte in einem Prozess erstellt, manipuliert oder verwendet werden. Diese Beziehung dient dazu einen schnellen Überblick über eine größere Anzahl von Objekten zu erhalten, da sie den gesamten Prozess auf einmal zeigt.

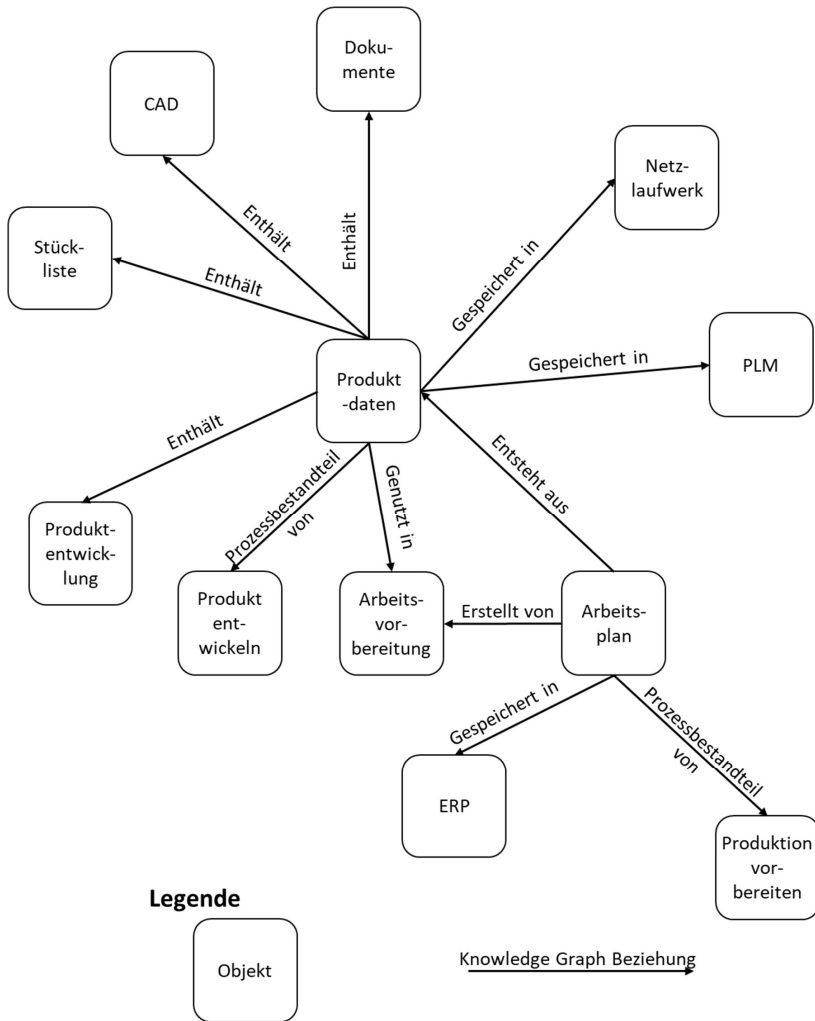


Abbildung 5-4: Enterprise Knowledge Graph-Beispiel des Prozesses „Produkt entwickeln“

Abbildung 2-3 zeigt einen beispielhaften Ausschnitt aus einem Knowledge Graphen, welcher den Prozessschritt „Produkt entwickeln“ auf einer groben Abstraktionsebene enthält und den sich anschließenden Prozess „Arbeitsplan erstellen“ zumindest teilweise modelliert. In diesem Beispiel sieht man Knoten für jedes der in diesem Kapitel vorgeschlagenen Objekttypen, sowie alle vorgeschlagenen Verbindungstypen.

5.1.2. Verbindung des Knowledge Graph mit einem Metadaten-Repository

Der Knowledge Graph steht in diesem Konzept nicht für sich allein, sondern soll gerade durch die Verknüpfung mit einem Metadaten-Repository und den dahinterstehenden und verbundenen IT-Systemen wie PLM-, ERP- oder IoT-Systemen einen Mehrwert schaffen. Dies zeigt sich unter anderem in der „Gespeichert in/Saved in“ Beziehung, welche auf einen Datenspeicherort verweist, welcher idealerweise auch in einem allgemeinen und zugänglichen Metadaten-Repository verwaltet wird.

Prinzipiell könnte diese Verknüpfung mit jedem denkbaren Metadaten-Repository geschehen, welches entsprechend um die hier beschriebenen Datentypen erweiterbar ist. In diesem Konzept übernimmt das am Lehrstuhl VPE entwickelte SP²IDER diese Rolle. Im Folgenden wird deswegen zuerst näher auf die Besonderheiten von SP²IDER im Bereich der Datenspeicherung eingegangen und anschließend geschildert, wie der soeben beschriebene Knowledge Graph mit dem SP²IDER-Graph verknüpft wird.

Das SP²IDER-Datenmodell wurde so geschaffen, dass verschiedenartige Strukturen aus unterschiedlichen Engineering-IT-Systemen abgebildet werden können. Dies umfasst insbesondere strukturierte Daten aus Datenbanken wie z.B. im Bereich PLM oder ERP, aber auch unstrukturierte Daten wie beispielsweise Sensorwerte aus einer IoT-Datenbank oder aus einem Data Warehouse. Ebenfalls ist es möglich gänzlich unstrukturierte Daten aus z.B. Lastenheften oder anderen Spezifikationsdokumenten abzubilden. In all diesen Fällen ist es wichtig, dass es bei den Daten, welche in SP²IDER abgelegt werden, nur um die Metadaten handelt, nicht um die eigentlichen Inhaltsdaten wie z.B. das 3D-CAD-Modell in einer .par-Datei [EiEG2020].

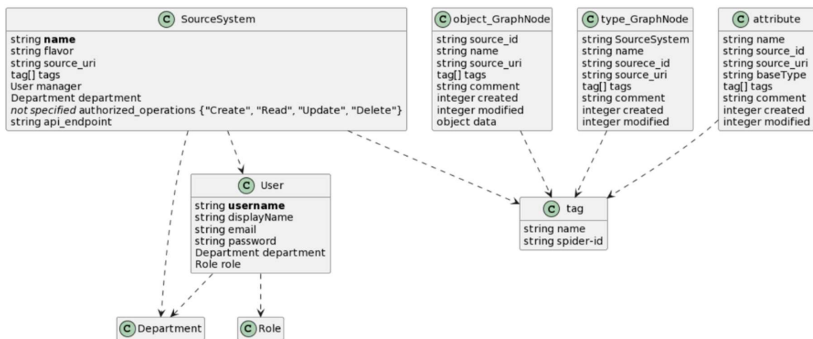


Abbildung 5-5: Ausschnitt aus dem bestehenden SP²IDER Datenmodell nach [EiEG2022]

Abbildung 5-5 zeigt das SP²IDER-Datenmodell, welches im Wesentlichen aus drei Ebenen besteht: Die Systemebene, welche auf die bestehenden IT-Systeme zeigt, aus der Typ-Ebene, welche darstellt, welche Datentypen in einem IT-System gespeichert sind und zuletzt die Objektebene, in welcher gespeichert wird, welche Objekte von einem bestimmten Typenvorhanden sind. Die Ebenen sind untereinander verknüpft [EiEG2020].

Technisch umgesetzt wurde das Datenmodell mittels ArangoDB²⁴. Zum besseren Verständnis sollen hier zuerst einige Konventionen der ArangoDB erläutert werden, da aufgrund der begrifflichen Ähnlichkeit eine große Verwechslungsgefahr besteht.: ArangoDB speichert Daten in Collections. Jede Collection wird dabei als Sammlung von Dokumenten verstanden, welche ein Mindestmaß an Attributen besitzen, um jedes einzelne Dokument zu identifizieren. Da ArangoDB eine Graphdatenbank ist, gibt es entsprechend der üblichen Struktur Knoten und Kanten, wobei jeder Knoten als Dokument in einer „Document-Collection“ abgespeichert wird, jede Kante als Dokument in einer „Edge-Collection“ abgespeichert wird. Jedes Dokument hat als minimales Set die Attribute „_key“, „_id“ und „_rev“, welche es eindeutig identifizieren. Die „_id“ ist dabei eine Zusammenführung aus Collection-Namen und dem „_key“ des Dokuments. Sofern das Dokument eine Kante beschreibt, werden zusätzlich noch ein „_from“ und ein „_to“ Attribut hinzugefügt, welche auf die „_id“ der zu verknüpfenden Dokumente zeigen. Weitere Attribute zur Beschreibung der Knoten und Kanten können abgespeichert werden, müssen aber nicht.

Für die Nutzung als SP²IDER Metadastore wurden für jede Collection, in welcher Knoten oder Kanten gespeichert werden, zusätzliche Attribute angelegt, welche im Folgenden näher beschrieben werden. Hierbei sind nicht alle Collections identisch relevant. Von besonderem Interesse für die Umsetzung des hier beschriebenen Konzepts sind die Datentypen „SourceSystem“, „type_GraphNode“, „attribute“, „Object_GraphNode“, sowie „User [EiEG2022].

SourceSystem

In der Collection SourceSystem werden Informationen über alle an SP²IDER verknüpften Quellsysteme gespeichert. Neben den oben beschriebenen Attributen zur Identifikation werden weiterhin ein Name des jeweiligen Systems, die Art des Systems, also z.B. Aras Innovator v12 oder Contact Elements, die URI, unter welcher das System erreichbar ist und der API-Endpunkt des SP²IDER Connectors für dieses System als Attribute vorgesehen. Weitere Attribute wie Manager und Department sind bereits vorgesehen, werden aber aktuell nicht genutzt und können später für eine detaillierte Rechteverwaltung verwendet werden. Daneben gibt es noch tags, mit denen das angelegte Datenquellsystem weiter beschrieben und verschlagwortet werden kann. Abbildung 5-6 zeigt die gesamte Beschreibung des Datenmodells aus der API-Dokumentation inkl. der zugrundeliegenden Typen der einzelnen Attribute.

²⁴ <https://www.arangodb.com/>

```
SourceSystem -  
  
name  
String Name of the source System  
flavor (optional)  
String Type of the source system, e.g. Aras, CII  
source_uri (optional)  
String URI of the Source System  
tags (optional)  
array\[tag\]  
manager (optional)  
User  
department (optional)  
Object  
authorized_operations (optional)  
String Grants access to various operations  
    Enum:  
        Create  
        Read  
        Update  
        Delete  
api_endpoint (optional)  
String URI of the Source System Connector
```

Abbildung 5-6: Datenmodell der SP²IDER SourceSystem Collection

Die Zugangsdaten in das einzelne Quellsystem hinein werden dabei nicht im SP²IDER Metadastore abgelegt, sondern verbleiben im entsprechenden Connector und werden dort dann zur Kommunikation in das System hinein oder aus dem System heraus eingesetzt.

Type_Graphnode

In der Collection type_graphnode werden Informationen zu allen Datentypen aus allen verbundenen Quellsystemen gespeichert. Neben den Attributen zur Identifikation werden weiterhin der Name des Datentyps, eine ID aus dem Quellsystem – sofern vorhanden, bzw. eine URI aus dem Quellsystem – sofern vorhanden, gespeichert. Über die Attribute „tags“ und „comment“ kann direkt in der SP²IDER-Ebene eine nähere Beschreibung des Datentyps erfolgen. Zwei Zeitstempel „created“ und „modified“ geben an, wann der type_GraphNode erstmalig in SP²IDER angelegt und wann letztmalig angepasst wurde. Das Attribut source_system ist aktuell noch im Datenmodell vorhanden, wird allerdings nicht mehr genutzt. Über eine Kante sind die type_GraphNodes auch immer direkt einem SourceSystem zugeordnet und stehen nicht allein für sich.


```
_id: type_graphnodes/2069391
_rev: _egpE7Nu---
_key: 2069391

Code ▾

1 - {
2   "name": "CAD",
3   "source_system": "aras",
4   "source_id": "CCF205347C814DD1AF056875E0A880AC",
5   "source_uri": "http://aras.mv.uni-kl.de/innovatorserver/server
6     /odata/itemtype('CCF205347C814DD1AF056875E0A880AC')",
7   "tags": [],
8   "comment": ""
}
```

Abbildung 5-7: Beispiel type_GraphNode im SP²IDER Metadastore

Ein Beispiel für ein type_graphnode zeigt Abbildung 5-7. Hier zu sehen ist die Abbildung des Aras-Itemtype „CAD“ aus einem System, welches auf dem Server <http://aras.mv.uni-kl.de> gehostet ist. Mithilfe der ID oder der URI kann direkt im Quellsystem auf das Objekt zugegriffen werden, sofern man über Zugangsdaten verfügt.

Attribute

In der Collection attribute wird gespeichert, welche Attribute in den Quellsystemen zu den einzelnen Dateitypen gespeichert sind. Neben den Informationen zur Identifikation in ArangoDB werden zusätzlich noch der Name, ein Basis-Datentyp wie z.B. „String“ oder „Integer“, eine ID und URI aus dem Datenquellsystem, sowie Tags, Kommentare, das Erstellungs- und das letzte Änderungsdatum.

| Content |
|---|
| <code>{"_from": "type_graphnodes/2069391", "_id": "AttributesFromType/2076517", "_key": "2076517", "_rev": "_eHyDdfq---", "_to": "attributes/2076515"}</code> |
| <code>{"_from": "type_graphnodes/2069391", "_id": "AttributesFromType/2076523", "_key": "2076523", "_rev": "_eHyDd1S---", "_to": "attributes/2076521"}</code> |
| <code>{"_from": "type_graphnodes/2069391", "_id": "AttributesFromType/2076529", "_key": "2076529", "_rev": "_eHyDd1K---", "_to": "attributes/2076527"}</code> |
| <code>{"_from": "type_graphnodes/2069391", "_id": "AttributesFromType/2076535", "_key": "2076535", "_rev": "_eHyDdo0---", "_to": "attributes/2076533"}</code> |
| <code>{"_from": "type_graphnodes/2069391", "_id": "AttributesFromType/2076541", "_key": "2076541", "_rev": "_eHyDdr0---", "_to": "attributes/2076539"}</code> |
| <code>{"_from": "type_graphnodes/2069391", "_id": "AttributesFromType/2076547", "_key": "2076547", "_rev": "_eHyDduK---", "_to": "attributes/2076545"}</code> |
| <code>{"_from": "type_graphnodes/2069391", "_id": "AttributesFromType/2076553", "_key": "2076553", "_rev": "_eHyDdu0---", "_to": "attributes/2076551"}</code> |
| <code>{"_from": "type_graphnodes/2069391", "_id": "AttributesFromType/2076559", "_key": "2076559", "_rev": "_eHyDd0e---", "_to": "attributes/2076557"}</code> |

Abbildung 5-8: Kanten, die einen type_GraphNode mit Attributen verknüpfen

Abbildung 5-8 zeigt einen Ausschnitt der Kanten, welche den beispielhaften type_GraphNode „CAD“ mit all seinen Attributen verknüpfen. Insgesamt sind dies in diesem Fall 50 Attribute, welche im Quellsystem Aras den „Properties“ des „Itemtype CAD“ entsprechen.

Die drei Collections, welche im vorhergegangenen Abschnitt beschrieben wurden, sind ausreichend, um das Datenmodell eines beliebigen Datenquellsystems vollständig zu beschreiben. Bei klassischen und weitverbreiteten relationalen Datenbanken [Harr2016] wie SQL-Datenbanken oder Software, die darauf basiert, wie z.B. Aras²⁵ oder CIM Database²⁶, bilden die type_GraphNodes hier die einzelnen Tabellen ab, die Attribute sind die Spalten der Tabellen.

Object_GraphNode

In der Object_GraphNode Collection werden Informationen über einzelne Datenobjekte aus einem IT-System gespeichert. Hier findet sich z.B. eine Repräsentation eines jeden Bauteils und einer jeder Stückliste aus einem PLM-System, eines jeden Arbeitsplans aus einem ERP-System oder jeden Assets oder Sensorwert aus einem IoT-System. Aufgrund der Vielzahl von Objekten und ihrer Menge wird in dieser Collection vermutlich die größte Datenmenge abgespeichert werden, sobald das System in einem Live-Betrieb getestet oder eingesetzt wird. Wie auch bei den anderen Collections werden nicht die eigentlich Inhaltsdaten in SP²IDER abgespeichert, sondern lediglich die Metadaten. Bezogen auf ein Bauteil in einer Stückliste würde

User

Die User-Objekte dienen in erster Linie der Authentifizierung gegenüber dem System. Gespeichert sind ein Login Name und ein Passwort, welches zum Einloggen genutzt werden können, dabei

²⁵ <http://aras.com>

²⁶ <https://www.contact-software.com/de/produkte/plm-software-product-lifecycle-management>

einen Token generieren, der bei den Abrufen der anderen Objekte mitgesendet werden muss, um eine Antwort zu erhalten.

Neue Beziehungen zwischen den Objekttypen

Nachdem im vorhergehenden Kapitel die bestehenden Objekttypen des Knowledge Graphen beschrieben wurden und in diesem Kapitel die wichtigsten bereits bestehenden SP²IDER-Objekttypen beschrieben wurden, müssen neue Verbindungen zwischen beiden Objekttypen definiert werden, um auch zwischen Knowledge Graph und SP²IDER-Graph hin und her zu navigieren und die Vorteile aus der Verknüpfungen zu ziehen. Dies ist insbesondere der Wechsel von der abstrakten Modellebene hin zur konkreten Datenebene. In der Datenebene werden dann bestehende Verbindungen aus dem SP²IDER-Graph genutzt.

Ist eine/IsA

Die „Ist eine“ bzw. „IsA“-Beziehung dient der Verknüpfung der Datenspeicherorte im Knowledge Graphen und der vorhandenen Systeme im bestehenden SP²IDER-Graphen. Beispielsweise wäre dort der Knoten „Aras Innovator v12“ und im Knowledge Graphen wäre „PLM“ als DataSource/Datenquelle vorhanden, dann könnten diese beiden Knoten mit der „Ist eine“/„IsA“ Verbindung verknüpft werden.

Entspricht/Equal_to

Die Entspricht-Beziehung dient dazu, Type_GraphNodes, also die Datentypen des SP²IDER-Graphs den DataObjects des Knowledge-Graphen zuzuordnen. Diese Beziehung ist die zentrale Neuerung des gesamten Konzepts und dient der Erklärung und Sichtbarmachung des Datenbestands. Anstatt innerhalb mehrerer Systeme aus hunderten mehr oder weniger kryptischen Namen die entsprechenden Daten zu suchen, kann über diese Beziehung die Suche im stark vereinfachten und abstraktem Knowledge Graphen durchgeführt und über die Entspricht-Beziehung dann in die reale IT abgesprungen werden.

Sieht/Sees

Die Sieht-Beziehung dient dem Erstellen von Ausschnitten des Graphen. Die Beziehung dient der Komplexitätsreduzierung und wurde eingeführt, um ein Sichtenkonzept auf den Knowledge- und den SP²IDER-Graphen umzusetzen. Durch eine Auswertung der verbundenen Knoten kann die Sichtbarkeit des Gesamtmodells auf einzelne Elemente reduziert werden, um unterschiedlichen Nutzern jeweils nur ein Teilmodell anzuzeigen. Dies dient zum einen der Reduktion der Komplexität, aber auch als Ansatz eines Rollen- und Rechtekonzepts. Eine weitere Beschreibung hierzu findet in Kapitel 5.6 statt.

Hat eine/HasA

Die „Hat eine“/„HasA“-Verknüpfung verbindet User mit Rollen. Sie dient dazu, einem User mehrere Rollen zuzuweisen, zwischen denen gewechselt werden kann. Durch diese Beziehung und Trennung der User von ihren Rollen können letztere mehrfach verwendet werden. Beispielsweise

kann so ein Rollenprofil „Engineering“ und ein weiteres „Produktion“ erstellt werden. Einzelne User sehen dann nur eines dieser Profile, manche aber auch beide Profile.

Tabelle 5-2: Verbindungen zwischen Objekt-Typen beider-Graphen

| | DataObjects | DataSources | Processes | Departments | systems | role | user | type_graphnode |
|----------------|--------------------|-------------|-----------|-----------------|---------|------|------|----------------|
| DataObjects | includes,made_from | saved_in | part_of | used_by,made_by | | | | |
| DataSources | | | | | | | | |
| Processes | | | | | | | | |
| Departments | | | | | | | | |
| systems | | IsA | | | | | | |
| type_graphnode | equal_to | | | | | | | |
| role | sees | sees | sees | sees | sees | | | sees |
| user | | | | | | hasA | | |

Tabelle 5-2 zeigt eine Übersicht über alle Verbindungen, welche zwischen den einzelnen Elementen des Knowledge- sowie des SP²IDER-Graphen vorgesehen sind. Die Verbindungen sind alle gerichtet und müssen in dieser Tabelle von der ersten Spalte nach rechts und dann nach oben gelesen werden. Beispielsweise: „DataObjects part_of Processes“. Auf diese Art und Weise erfüllen alle Verbindungen die Anforderungen an RDF-Tripel und können als solche ausgedrückt werden.

5.2. Knowledge Graph basierte Lösungsarchitektur

Durch die Erweiterung des SP²IDER-Konzepts und die neue Verwaltung von Wissensmodellen ergibt sich die Frage, wie diese Verwaltung genau aussehen kann. Auch wenn eine nahtlose Einbeziehung von Wissensmodellen in die bisherige Struktur möglich wäre, z.B. indem eine Art „Wissensquellsystem“ geschaffen und die dort verwalteten Elemente dann auch im SP²IDER-Graph als type_GraphNode und object_GraphNode repräsentiert werden, wird diese Möglichkeit nicht in Betracht gezogen. Zum einen wäre die Art der Speicherung so ähnlich dem SP²IDER Core und SP²IDER Metadatenstore, dass eine Dopplung erschaffen würde, zum anderen soll aufgrund der integralen Bedeutung der Wissensmodellen eine neue Architekturebene direkt in SP²IDER geschaffen werden, so dass neben Systemen, Typen und Objekten nun auch Wissen verwaltet wird.

5.2.1. Neue Ebene im SP²IDER Prozess- und Datenmodell

Die momentane Ausbaustufe der semantischen Vernetzungsplattform SP²IDER bildet lediglich einen Stand ab, der jedes Objekt in einem Datenquellsystem in einem Metadatenobjekt in SP²IDER repräsentiert. Das heißt es gibt beispielsweise das Systemobjekt „Aras Innovator“, in welchem die Typ-Objekte „Part“ und „CAD“ gespeichert sind und die Objekte „0815 Bauteil A“ und „0816 Bauteil B“ als Stücklistenpositionen. Abbildung 5-9 zeigt die Typ- und Objekt-Ebene.

Welcher Art diese Daten sind, welche Bedeutung sie haben und in welchem Zusammenhang sie stehen kann nur indirekt durch vorhandenes Wissen des Benutzers erschlossen werden. Zusätzlich sind hunderte bis tausende Typen sowie zehntausende Objekte repräsentiert, welche für Nutzer im Engineering keinerlei Relevanz haben, da sie eher technischer Natur sind und Informationen aus den Quellsystemen zeigen, die nicht benötigt werden.

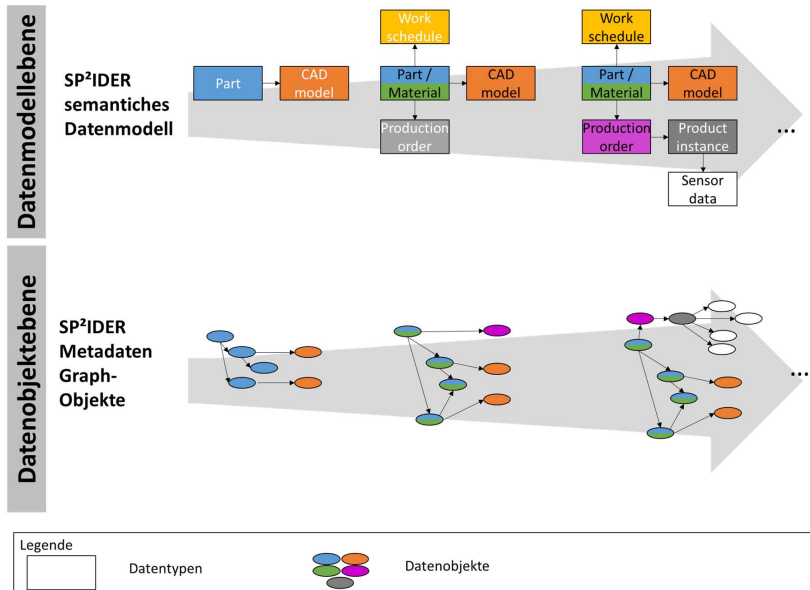


Abbildung 5-9: SP²IDER Datenmodell mit verschiedenen Ebenen nach [EIEG2020]

Die neue Ebene, welche über den bisherigen Ebenen in SP²IDER angesiedelt ist, besteht aus dem beschriebenen Knowledge Graphen. Er dient dazu, eine Übersicht zu schaffen, welche Daten überhaupt im Unternehmen verwaltet werden mit einem Fokus auf Relevanz dahingehend, dass die Daten auch in den Prozessen des Unternehmens erstellt oder genutzt werden. Der Knowledge-Graph kann so als Einstiegspunkt zur Identifikation der relevanten Daten und zum Filtern der angezeigten Datenmenge werden. Um dieser Wichtigkeit Rechnung zu tragen, soll die Erstellung des Knowledge-Graphen auch nicht „nur“ ein Service in der bestehenden SP²IDER-Architektur sein, sondern wie in Abbildung 5-10 gezeigt, eine herausragende Stellung bekommen und als fünfte SP²IDER-Architekturkomponente betrachtet werden, so dass die vorgeschlagene SP²IDER-Architektur dann folgende ist:

- SP²IDER Core
- SP²IDER Metadastore
- SP²IDER Services
- SP²IDER Knowledge Graph Creator

- Datenquellsysteme

Der Knowledge Graph Creator dient in dieser Architektur als User Interface, sowohl des Knowledge Graph, aber auch des SP²IDER Graph. Als Grundlage für diesen neuen Bestandteil der SP²IDER-Architektur dient der Webservice des Data Model Canvas, welcher im Projekt AKKORD entwickelt wurde.

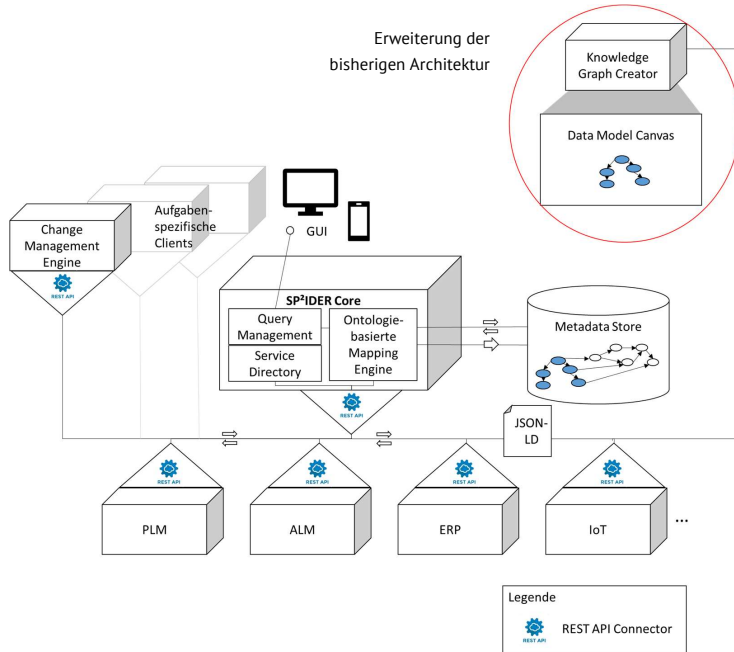


Abbildung 5-10: Erweiterte SP²IDER-Architektur zur Modellierung des Knowledge Graph

Neben der reinen Funktionalität des graphischen Modellierens des Knowledge Graphen ist in diesem neuen SP²IDER-Architekturelement auch Kommunikationsfähigkeit entscheidend. Hier wird ebenfalls auf die bereits existierende REST API gesetzt, welche bereits im SP²IDER Core und den SP²IDER-Konnektoren zur Kommunikation mit den Datenquellsystemen eingesetzt wird. Die notwendigen Erweiterungen der REST API Spezifikation finden im SP²IDER Core statt.

5.2.2. Erweiterung des SP²IDER Core

Entsprechend des Datenmodells des SP²IDER Knowledge Graphen muss auch der SP²IDER Core erweitert werden, um die benötigten Daten über die API in den Core hinein und damit den Metadata Store, bzw. den Knowledge Graph Creator zu übertragen. Hierzu muss für jede neue Ressource bzw. Objekttyp Code entwickelt werden, um Objekte anzulegen (Create), abzurufen (Read), zu verändern (update) bzw. zu löschen (delete). Je nachdem ob bereits ein Objekt existiert

oder nicht, unterscheidet sich der Aufruf eines API-Endpunktes und es gibt pro Objekt zwei API-Endpunkte für die Funktionsaufrufe:

Neben den Endpunkten für das Anlegen, Abrufen oder Verändern von Objekten gibt es auch neue Endpunkte, um die Verknüpfungen zwischen zwei Objekten als neue Kante im Graph zu ziehen. Dies gilt sowohl für die Kanten zwischen den einzelnen Objekten des Knowledge-Graphen als auch für Verbindungen zwischen diesen Objekten und den originären SP²IDER-Objekten, so wie sie im grundlegenden Konzept beschrieben sind.

```

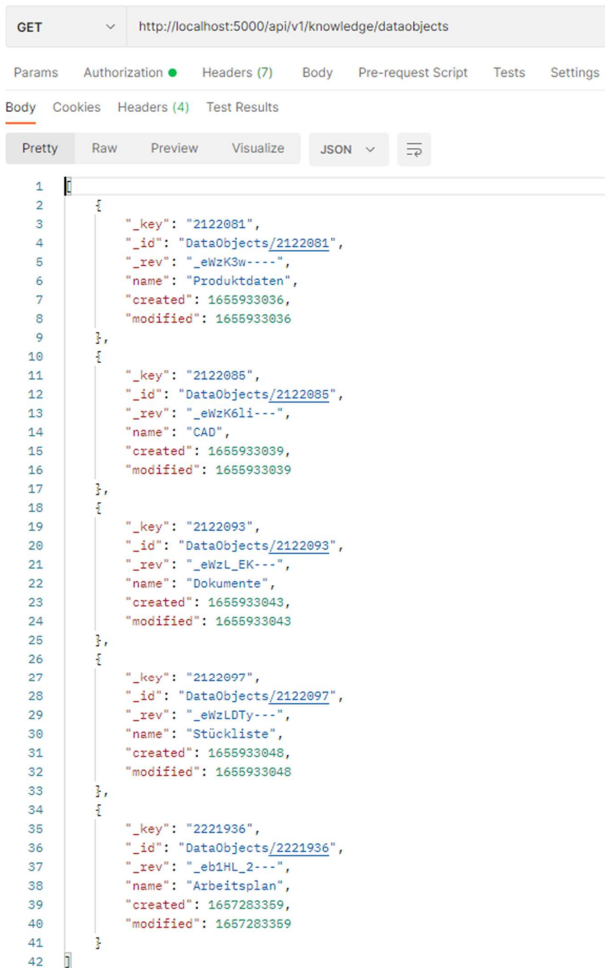
api.add_resource(AuthenticationAPI, '/login')
api.add_resource(ObjectsIDAPI, '/api/v1/objects/<collection>/<key>')
api.add_resource(ObjectsAPI, '/api/v1/objects')
api.add_resource(ServiceAPI, '/api/v1/services')
api.add_resource(ServiceIDAPI, '/api/v1/services/<collection>/<key>')
api.add_resource(SystemsAPI, '/api/v1/systems')
api.add_resource(SystemsIDAPI, '/api/v1/systems/<collection>/<key>')
api.add_resource(TypesAPI, '/api/v1/types')
api.add_resource(TypesIDAPI, '/api/v1/types/<collection>/<key>')
api.add_resource(TypeAttributesAPI, '/api/v1/types/<collection>/<key>/attributes')
api.add_resource(GenericEdgeAPI, '/api/v1/edges')
api.add_resource(TypeEdgesAPI, '/api/v1/types/connections')
api.add_resource(TypeConnectionAPI, '/api/v1/types/<from_collection>/<from_key>/to/<to_collection>/<to_key>')
api.add_resource(DataObjectAPI, '/api/v1/knowledge/dataobjects')
api.add_resource(DataObjectIDAPI, '/api/v1/knowledge/dataobjects/<collection>/<key>')
api.add_resource(ProcessAPI, '/api/v1/knowledge/processes')
api.add_resource(ProcessIDAPI, '/api/v1/knowledge/processes/<collection>/<key>')
api.add_resource(DataSourceAPI, '/api/v1/knowledge/datasources')
api.add_resource(DataSourceIDAPI, '/api/v1/knowledge/datasources/<collection>/<key>')
api.add_resource(DepartmentAPI, '/api/v1/knowledge/departments')
api.add_resource(DepartmentIDAPI, '/api/v1/knowledge/departments/<collection>/<key>')
api.add_resource(RolesAPI, '/api/v1/knowledge/roles')
api.add_resource(RolesIDAPI, '/api/v1/knowledge/<collection>/<key>')
api.add_resource(IncludesAPI, '/api/v1/knowledge/<from_collection>/<from_key>/includes/<to_collection>/<to_key>')
api.add_resource(MadeFromAPI, '/api/v1/knowledge/<from_collection>/<from_key>/madefrom/<to_collection>/<to_key>')
api.add_resource(SavedInAPI, '/api/v1/knowledge/<from_collection>/<from_key>/savedin/<to_collection>/<to_key>')
api.add_resource(PartOfAPI, '/api/v1/knowledge/<from_collection>/<from_key>/partof/<to_collection>/<to_key>')
api.add_resource(UsedByAPI, '/api/v1/knowledge/<from_collection>/<from_key>/usedby/<to_collection>/<to_key>')
api.add_resource(MadeByAPI, '/api/v1/knowledge/<from_collection>/<from_key>/madeby/<to_collection>/<to_key>')
api.add_resource(ISAAPI, '/api/v1/knowledge/<from_collection>/<from_key>/isa/<to_collection>/<to_key>')
api.add_resource(KnowledgeEdgeAPI, '/api/v1/knowledge/edges/<col>/<key>')
api.add_resource(HasAAPI, '/api/v1/knowledge/<collection>/<key>/hasa')
api.add_resource(KnowledgeGraphAPI, '/api/v1/knowledge/getall')
api.add_resource(EqualToAPI, '/api/v1/knowledge/<from_collection>/<from_key>/equalto/<to_collection>/<to_key>')
api.add_resource(UserAPI, '/api/v1/knowledge/users')
api.add_resource(UserIDAPI, '/api/v1/knowledge/users/<collection>/<key>')
api.add_resource(ViewAPI, '/api/v1/knowledge/role/view/<col>/<key>')

```

Abbildung 5-11: API-Endpunkte des erweiterten SP²IDER-Core

Abbildung 5-11 zeigt einen Ausschnitt der programmierten API-Endpunkte im Quellcode des SP²IDER Core und welche Funktionen bei einem Request aufgerufen werden. Die ursprüngliche SP²IDER-API wird immer aufgerufen durch einen Endpunkt mit der Syntax „<Server>/api/v1/...“, bspw. als konkreter Aufruf in einer lokalen Maschine „http://localhost:5000/api/v1/systems“. Bei einem Aufruf dieser URL mit der Methode „http-get“ bekommt der Nutzer eine Liste mit allen in SP²IDER gespeicherten Systemen zurück. Zur Unterscheidung zwischen den bisherigen SP²IDER-Funktionen und den neu hinzugefügten Funktionen wurden für die API-Aufrufe mit Bezug auf den Knowledge-Graphen immer Endpunkte mit der Syntax „<Server>/api/v1/knowledge/...“ gestaltet.

Das Ergebnis einer Beispielabfrage des Endpunktes für die DataObjects mithilfe des Programms Postman²⁷ zeigt Abbildung 5-12.



```
1 GET http://localhost:5000/api/v1/knowledge/dataobjects
2
3 Params Authorization Headers (7) Body Pre-request Script Tests Settings
4
5 Body Cookies Headers (4) Test Results
6
7 Pretty Raw Preview Visualize JSON
8
9
10 {
11   "_key": "2122081",
12   "_id": "DataObjects/2122081",
13   "_rev": "_ewzK3w---",
14   "name": "Produktdaten",
15   "created": 1655933036,
16   "modified": 1655933036
17 },
18 {
19   "_key": "2122085",
20   "_id": "DataObjects/2122085",
21   "_rev": "_ewzK6li---",
22   "name": "CAD",
23   "created": 1655933039,
24   "modified": 1655933039
25 },
26 {
27   "_key": "2122093",
28   "_id": "DataObjects/2122093",
29   "_rev": "_ewzL_EK--",
30   "name": "Dokumente",
31   "created": 1655933043,
32   "modified": 1655933043
33 },
34 {
35   "_key": "2122097",
36   "_id": "DataObjects/2122097",
37   "_rev": "_ewzLDTy--",
38   "name": "Stückliste",
39   "created": 1655933048,
40   "modified": 1655933048
41 },
42 {
43   "_key": "2221936",
44   "_id": "DataObjects/2221936",
45   "_rev": "_eb1HL_2--",
46   "name": "Arbeitsplan",
47   "created": 1657283359,
48   "modified": 1657283359
49 }
50 }
```

Abbildung 5-12: Beispielabfrage der DataObjects im Knowledge-Graphen

Mit einem Request, der einen API-Endpunkt aufruft, wird im Core geprüft. Ob eine Authentifizierung vorliegt und falls ja, dann wird geprüft, um welche HTTP-Methode es sich

²⁷ <https://www.postman.com/>

handelt. Bei einem „GET“-Request werden die vorhandenen Objekte zur Abfrage ausgegeben, ein „POST“-Request erstellt ein neues Objekt der Kategorie, ein „PATCH“-Request verändert das Objekt, ein „DELETE“-Request löscht es. Je nach Art der Methode müssen noch weitere Informationen bereitgestellt werden, wie z.B. die ID des zu verändernden oder zu löschenden Objekts, sowie im Fall eines „Patch“-Requests noch die Informationen, die anschließend gespeichert werden sollen.

5.2.3. Erweiterung der SP²IDER-Konnektoren

SP²IDER-Konnektoren haben Funktionalität, um Daten zwischen einem beliebigen System und dem SP²IDER Core auszutauschen. Eine Erweiterung der SP²IDER-Konnektoren wäre ein denkbarer Lösungsansatz, sofern der Knowledge Graph Creator so gestaltet würde, als wäre er ein weiteres Datenquellsystem. Auch wenn diese Lösungsmöglichkeit verworfen wurde (vgl. Kapitel 5.2), so soll hier ein mögliches Szenario geschildert werden.

In diesem Fall wäre der Knowledge Graph ein weiteres Datenquellsystem, welches als Datenmodell ebenfalls einen Graphen beinhalten würde. Entsprechend des regulären SP²IDER-Datenmodells würden alle Dinge in diesem Datenquellsystem entweder einem Datentypen oder einem Datenobjekt entsprechen. Die vorgeschlagenen Typen „DataObjects“, „DataSources“, „Processes“, „Roles“ und „Users“ würden jeweils als ein Type_GraphNode angelegt, sämtliche Objekte dieser Typen, z.B. die einzelnen DataObjects würden als Object_GraphNode angelegt. Es wäre zwar im Einklang mit den Regeln des SP²IDER-Datenmodells, zwischen zwei Object_GraphNodes aus unterschiedlichen Systemen eine Verbindung zu ziehen, allerdings würden diese semantisch exakt gleichbedeutend sein. Ein Wissenselement hätte für SP²IDER die gleiche Bedeutung wie z.B. ein CAD-Modell oder eine Stücklistenposition.

Ebenfalls würde die Synergie zwischen dem bereits im SP²IDER-Datenmodell vorgesehenen Nutzer und dem im Knowledge Graph vorgesehenen Nutzer nicht zustande kommen, da alle Nutzer in beiden Systemen angelegt werden müssen und eine Verknüpfung zwischen beiden erst hergestellt werden müsste. Weiterhin ist der entscheidende Grund, warum dieses Vorgehen nicht gewählt wurde, die fehlende Bedeutung der unterschiedlichen Kanten. Im regulären SP²IDER-Datenmodell entsprechen Kanten entweder der Bedeutung „Objekt ist vom Typ“ oder „Objekt steht in Verbindung mit Objekt“. Diese rudimentäre Bedeutung ist nicht ausreichend, um das Potential eines Knowledge Graphen zu nutzen.

Alternativ könnte auch der Knowledge Graph Creator sämtliche Informationen aus dem SP²IDER-Graph abrufen und diese alle spiegeln, um den Knowledge Graph mit dem SP²IDER-Graph kombinieren zu können. Dies würde eine Weiternutzung der bestehenden Struktur ermöglichen und alle zusätzlichen Informationen in diesem zusätzlichen Service bündeln. Nachteilig ist allerdings die immer noch bestehende Parallelstruktur durch den Aufbau eines zweiten Graphen und die Tatsache, dass anderen Services die Informationen nicht zur Verfügung stehen. Insgesamt betrachtet wird daher der Ansatz eines zusätzlichen „Wissensquellsystems“ nicht weiterverfolgt,

sondern die bereits angesprochene Neugestaltung der gesamten SP²IDER-Architektur vorgeschlagen.

5.3. Prozessmodellierung zur Nutzung des Systems

Das hier vorgestellte Konzept basiert sehr stark auf den Unternehmensprozessen als Ausgangsbasis der benötigten Information. Größere Unternehmen sind fast durchgehend nach DIN EN ISO 9001 [DIN 2015] zertifiziert und verfügen dementsprechend über alle Daten, Schnittstellen und Speicherorte in ihren Prozessen beschrieben haben. Häufig ist dem allerdings nicht so und die beschriebenen Prozesse weichen von der Realität ab. Ein Beispiel dafür beschreibt u.a. [HeTE2019]. Insofern bietet das hier vorgestellte Konzept Handlungsspielräume in beide Richtungen: Die Prozessbeschreibungen dienen als Ausgangsbasis für den Knowledge Graphen und werden mit bestehenden IT-Werkzeugen verknüpft, so dass die IT-Sicht auch die Prozesse abbilden soll. Ist dies nicht der Fall und die Zuordnung kann nicht durchgeführt werden, so sind zwei Handlungsrichtungen denkbar: Es kann sein, dass die entsprechenden Datentypen nur schwer zu identifizieren sind, z.B. aufgrund kryptischer Namen und eine manuelle Nacharbeit notwendig ist, oder aber die Prozesse bilden nicht die Realität ab. In diesem Fall kann eine festgestellte Diskrepanz auch als Anlass genommen werden, um die Prozessbeschreibungen zu überarbeiten und an die Realität anzupassen. Der Prozess kann weiterhin auch iterativ stattfinden, so dass Anpassungen in beide Richtungen stattfinden und anschließend sowohl das Engineering-Repository und der verknüpfende Knowledge Graph als auch die Prozesse gepflegt sind und die Realität abbilden.

Neben den Unternehmensprozessen als Datenquelle für den Knowledge-Graph spielen Prozesse auch eine wichtige Rolle in der Beschreibung der Aktivitäten der Beteiligten, welche den Knowledge Graph modellieren, an der Verknüpfung der Datenbestände mitwirken oder letztlich die Daten nutzen. Aufgrund des Umfangs der Konzepterweiterung des SP²IDER-Metadaten-Repository wird hier ein umfassendes Prozessmodell beschrieben. Das Prozessmodell soll die Nutzer durch das neuartig erweiterte Konzept führen und seine Nutzung erleichtern.

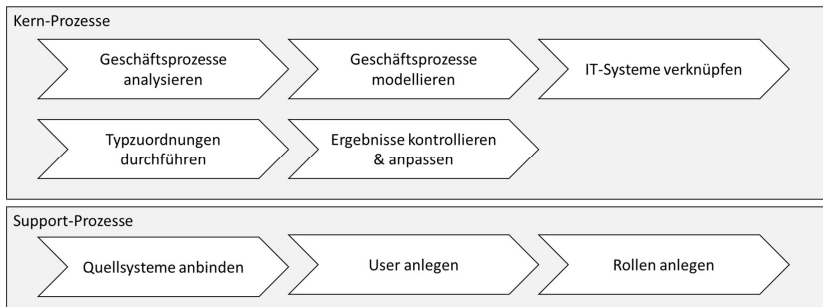


Abbildung 5-13: Prozessmodell der Erweiterung des Metadaten-Repository

Abbildung 5-13 zeigt eine Übersicht über die Prozesse, welche im Zentrum des neuen Konzepts stehen. Die Kern-Prozesse bilden hierbei eine Prozesskette, welche nacheinander abläuft, allerdings auch iterativ und wiederholt durchgeführt werden können. Die Support-Prozesse sind unterstützend und können, müssen aber nicht durchgeführt werden. Sie erweitern das Konzept insbesondere durch das Rollen und Rechtemanagement und sichern so die Nutzbarkeit ab.

5.3.1. Geschäftsprozesse analysieren

Der erste Prozessschritt ist die Analyse einer Unternehmensprozesslandkarte bzw. der niedergeschriebenen Unternehmensprozesse, um herauszufinden welche Objekte relevant sind, wo diese Objekte gespeichert sind und wie diese Objekte miteinander in Beziehung stehen, um eine ganzheitliche Sicht auf Daten zu ermöglichen. Genauso hat dieser Prozessschritt das Ziel, dass unwichtige und irrelevante (weil nicht in Prozessen verwendete) Informationen außen vor bleiben und nicht das Gesamtbild stören und eine unnötige Komplexität erzeugen.

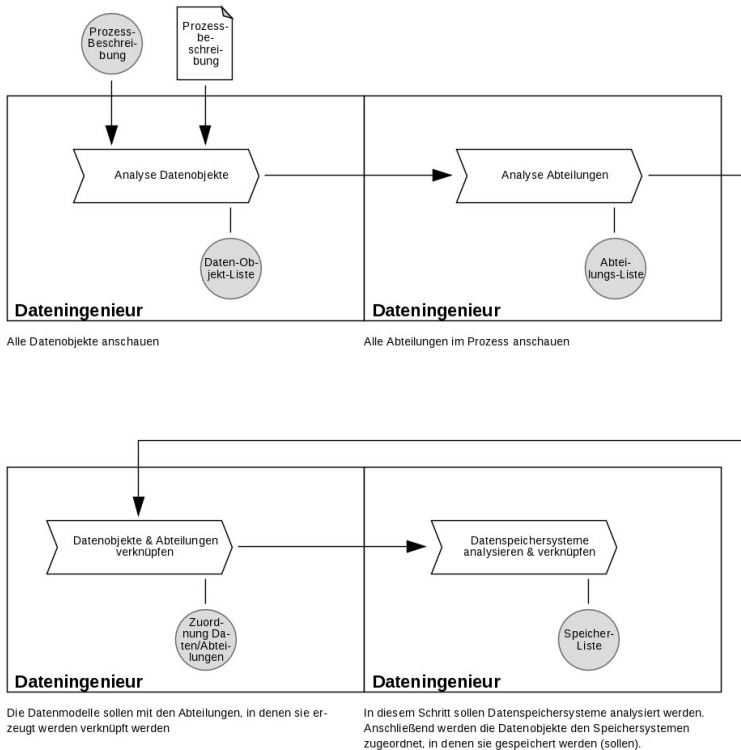


Abbildung 5-14: Prozessmodelle analysieren

Je nach Ausmaß der Prozessorganisation in einer Firma ist der Ablauf dieses Prozessschrittes in verschiedenen Automatisierungsstufen denkbar. Auf der niedrigsten Stufe, sofern keinerlei Prozessdokumentation vorliegt, erscheint der Prozessschritt mühsam und muss manuell durch eine Abstraktion der nicht modellierten Prozesse vorgenommen werden. Da die gesamten Rahmenbedingungen des Konzepts allerdings eher auf größere Unternehmen abzielen, z.B. das Vorhandensein großer Datenmengen in mehreren Datenquellsystemen, Nutzung von mehreren großen Datenbanken wie PLM, ERP, IoT-Datenbanken und auch das Vorhandensein komplexer Prozesse, die nicht mehr von einzelnen Mitarbeitern komplett gesteuert sind, kann davon ausgegangen werden, dass auch eine Zertifizierung bspw. nach DIN EN ISO 9001 [DIN 2015] vorliegt, was bedeutet, dass die Prozesse bereits modelliert sind.

In diesem Fall können die Prozessmodelle als Ausgangsbasis der Analyse genutzt werden und die entsprechenden Objekte abgeglichen werden. Sollte perspektivisch auch noch ein automatisierter Übersetzungsservice für die Prozessmodelle verfügbar sein, so kann der Prozess der Prozessmodellanalyse noch weiter vereinfacht werden und sich auf eine Ergebniskontrolle einer

automatisierten Analyse kontrollieren. Ein weiterer Vorteil einer solchen Automatisierung wäre eine periodische Kontrolle auf Änderungen, sollten z.B. neue Objekte in den Modellen erkannt werden. Ebenfalls kann eine solche Automatisierungslösung auch direkt im nächsten Prozess genutzt werden und dort bereits einen automatisiert erstellten Vorschlag generieren. Durchgeführt wird der Prozess von einem Dateningenieur, welcher sich mit der Modellierung von Graphen und Daten auskennt (vgl. Kapitel 5.5.1.).

5.3.2. Knowledge Graph modellieren

Aus den im vorherigen Schritt erarbeiteten Listen mit Datenobjekten, Abteilungen, Speicherorten kann im zweiten Schritt der Knowledge Graph modelliert werden. Hierzu wird im SP²IDER Knowledge Creator Service schrittweise der Graph aufgebaut, indem die entsprechenden Listen in Objekte umgewandelt und als Knoten angelegt werden. Anschließend werden die Verbindungen in Form der Kanten gezogen und alle Informationen im Metadata-Store permanent gespeichert.

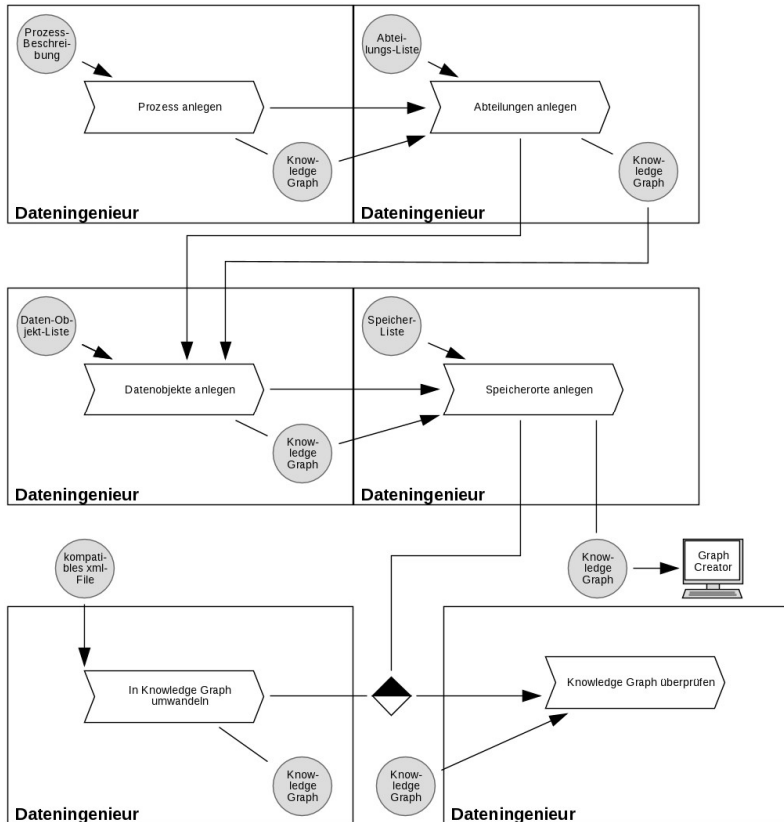


Abbildung 5-15: Knowledge Graph modellieren

Je nachdem ob die Umwandlung des Prozessmodells in den Knowledge Graph automatisiert erstellt wird oder nicht, teilt sich der Prozess in zwei Stränge auf, die allerdings am Ende wiederum zum gleichen Endzustand führen, wie Abbildung 5-15 zeigt. Während die automatisierte Erstellung bei der Auswahl der Reihenfolge keine Rolle spielt, kann der Nutzer in der manuellen Erstellung des Knowledge Graph entweder die hier vorgeschlagene Reihenfolge einhalten, oder diese verändern.

Die hier vorgeschlagene Reihenfolge (Prozesse → Abteilungen → Datenobjekte → Datenspeicherorte) ermöglicht eine Modellierung, deren Objekte tendenziell immer detaillierter und kleinteiliger werden. Die Prozesse als Ausgangsbasis bilden dabei ein Gerüst, das sich ähnlich wie die beteiligten Abteilungen durch den Produktlebenszyklus zieht. Die Datenobjekte sind hierbei zum einen auch Ankerpunkte zwischen den Prozessen, sofern sie als Output von Prozess A und als Input von Prozess B dienen, aber können darüber hinaus noch durch weitere

Datenobjekte, welche nur innerhalb eines Prozesses eine Rolle spielen kleinteiliger werden. Die Speicherorte als letztes Modellierungselement sind dann wiederum eine Art Verbindungselement für die Datenobjekte und die Prozesse.

5.3.3. Knowledge Graph und SP²IDER-Graph verknüpfen

Der aufgebaute Graph kann nun schrittweise genutzt werden, um relevante Daten in den unterschiedlichen Systemen zu identifizieren und zuzuordnen. Hierzu wird im ersten Schritt der Knowledge Graph bestehend aus Datenobjekten, Abteilungen und den Prozessschritten eingblendet. Anschließend können alle angebundenen IT-Systeme eingblendet werden.

Zur weiteren Bearbeitung und Verknüpfung muss auf Seiten des SP²IDER-Experten auch ein umfangreiches Wissen vorliegen, wie die vorhandenen IT-Systeme zu klassifizieren sind, spricht er muss auch wissen, ob es sich bei „Aras Innovator v12“ um ein PLM-System handelt oder ob dies etwas anderes ist. Sind alle Systeme bekannt, können „IsA“-Verbindungen zwischen den theoretischen Datenquellen aus dem Knowledge Graph und den realen Systemen aus dem SP²IDER-Graph gezogen werden, wie in Abbildung 5-16 gezeigt. Hierbei werden idealerweise so viele Verbindungen gezogen, wie es abstrahierte Datenquellen und reale Systeme gibt. Es kann allerdings auch sein, dass nicht alle Systeme an den SP²IDER-Graph angebunden sind, weswegen weniger Verbindungen gezogen werden können.

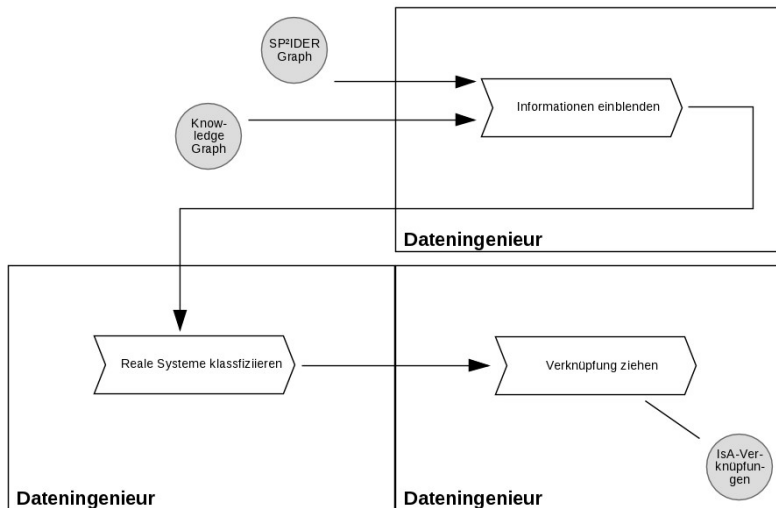


Abbildung 5-16: Knowledge- und SP²IDER-Graph verknüpfen

Entscheidendes Ergebnis dieser Verknüpfung ist die Zuordnung der Systeme aus den Knowledge Graph mit den realen IT-Systemen, wie sie im Unternehmen eingesetzt werden. Beispielsweise wird eine Verbindung zwischen dem Objekt „PLM“ aus dem Knowledge Graph und dem IT-System „Contact CIMDatabase“ oder „Aras Innovator v12“ gezogen. Hierbei ist eine Zuordnung auf einer

n:m Basis denkbar, wenngleich eine 1:1 Zuordnung wahrscheinlicher ist, da Datentypen häufig auch nur in einem System gespeichert werden. Sollte an dieser Stelle des Prozesses festgestellt werden, dass die realen Daten nicht in den vorher bereits im Knowledge Graph abstrahierten Systemen abgespeichert sind, so muss im Prozess zur Modellierung des Graph zurückgesprungen werden und der entsprechende Prozessschritt wiederholt werden, bis der Graph die reale Situation aufzeigt.

5.3.4. Relevante Daten im System identifizieren und verbinden

Sobald die Verbindungen zwischen den theoretischen Datenquellen aus dem Knowledge Graph und den realen IT-Systemen aus dem SP²IDER-Graph gezogen sind, können diese nacheinander ausgewertet werden, um in den realen Systemen die Objekte zu finden, welche im Knowledge Graph repräsentiert sind. Diese Auswertung übernimmt ein Algorithmus, welcher in Kapitel 5.4.2 beschrieben ist.

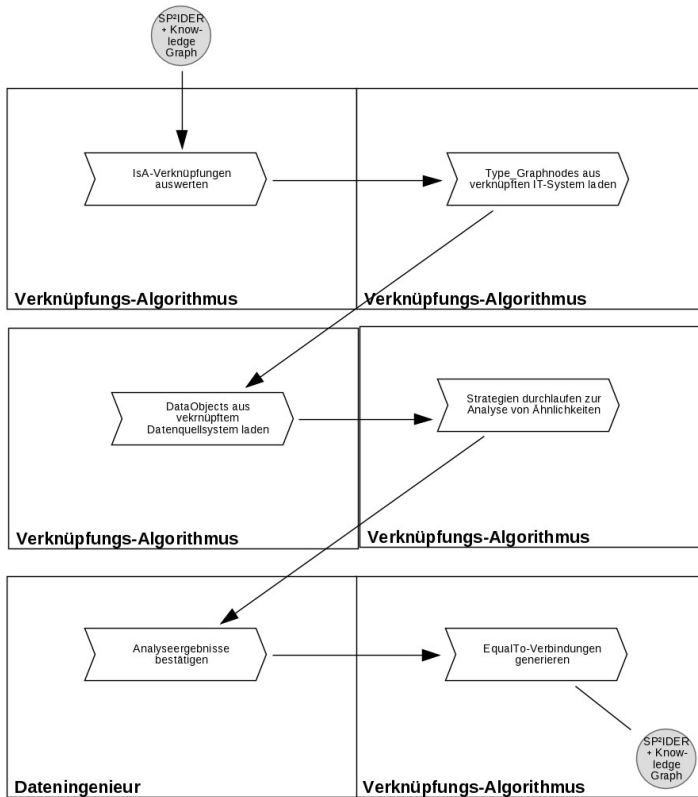


Abbildung 5-17: Relevante Daten im System identifizieren und verbinden

Abbildung 5-17 zeigt die einzelnen Schritte des Prozesses. Der Algorithmus nutzt die gezogenen „IsA“-Verbindungen und lädt sich über die API die damit verknüpften Systeme und in einem zweiten Schritt die Type_GraphNodes, sowie die DataObjects, welche im Knowledge Graph angelegt sind und mit einem DataSource-Objekt verbunden wurden. Anschließend werden sämtliche Strategien zur Ähnlichkeitssuche umgesetzt und das Ergebnis als Vorschlag dem SP²IDER-Experten präsentiert, welcher nun für jede Verbindung entscheiden muss, ob der Algorithmus hier einen validen Vorschlag für eine Ähnlichkeit macht, oder ob die Entscheidung falsch war. Anschließend werden für alle korrekt identifizierte Ähnlichkeiten „EqualTo“ Verbindungen angelegt, welche direkt DataObjects mit Type_GraphNodes verknüpfen.

5.3.5. User anlegen

Ein Teil des Konzepts umfasst auch ein neues Rollen und Rechtekonzept, welches es ermöglicht anderen Nutzern die tägliche Arbeit mit aufgewerteten Metadaten-Repository zu erlauben und

gleichzeitig ein möglichst kleinteiliges Rechtemanagement bezüglich der Sichtbarkeit zu ermöglichen.

Die User-Objekte dienen hierbei IT-technisch als das Datenobjekt, welches den Login im System ermöglicht, sprich das Checken des Passworts gegen das im System hinterlegte Passwort durchführen.

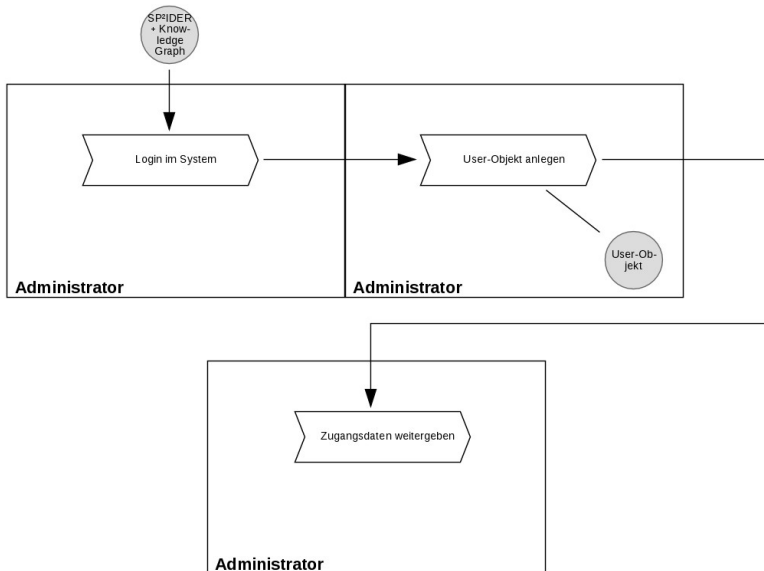


Abbildung 5-18: User im System anlegen

Abbildung 5-18 zeigt den entsprechenden Prozess zum Anlegen des Users. Beim Anlegen des Users müssen entsprechende Zugangsdaten angelegt werden wie z.B. der Login-Name und ein Passwort für die Authentifizierung, welche anschließend auch dem entsprechenden Nutzer zugänglich gemacht werden müssen. Momentan ist dieser Prozess manuell, perspektivisch müsste das IT-System so ertüchtigt werden, dass die Zugangsdaten direkt im System zufällig generiert werden und dem Nutzer per Mail zugesendet werden. Im Rahmen des Demonstrators wurden solche aufwendigen Prozesse rund um die User-Authentifizierung nicht verwirklicht, da sie funktional nicht im Kern des Konzepts angesiedelt sind.

5.3.6. Sichten anlegen

Als letzter Schritt wird vom Dateningenieur jede Sicht definiert, mit der im weiteren Verlauf Nutzer auf das Datenmodell des Knowledge Graph und des SP²IDER-Graph schauen und mit diesen Daten arbeiten können. Dieser Prozess lässt sich iterativ ausführen und kann jederzeit erneut durchgeführt werden, um weitere Sichten anzulegen.

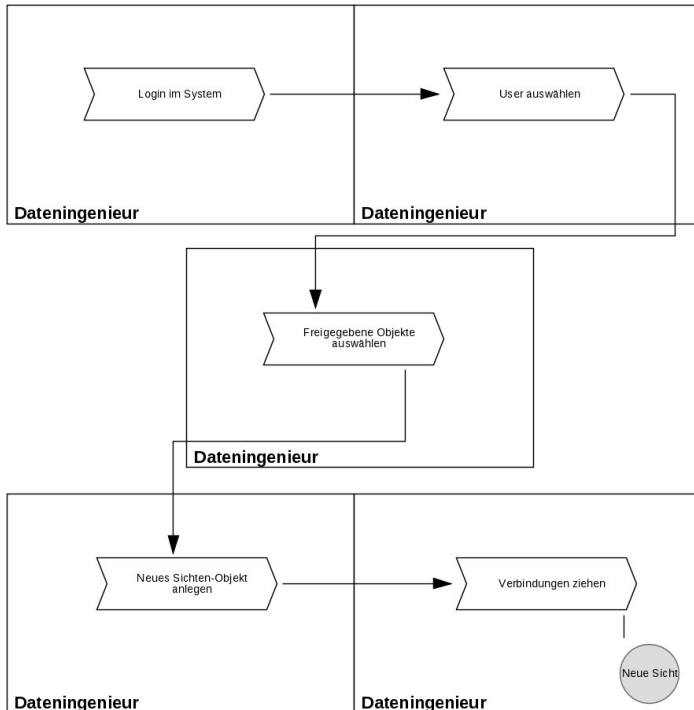


Abbildung 5-19: Neue Sicht anlegen

Abbildung 5-19 zeigt die Prozessschritte zum Anlegen einer neuen Sicht. Hierbei ist vor allem die Identifikation der Objekte wichtig, welche dann später von dieser Rolle bzw. den Nutzern, die diese Rolle ausfüllen, zu sehen sind. Auch wenn dieser Prozess sehr technisch klingt, wird er allerdings durch den Dateningenieur durchgeführt, da hierzu das Wissen über die Datenquellen und die damit verbundenen Datentypen im Vordergrund steht. Um diesen Prozess sinnvoll durchlaufen zu können, ist auch eine Absprache mit Domänenexperten sinnvoll, welche die zusätzlichen Datenzugriffsbedürfnisse der Nutzer einschätzen und weitergeben können.

5.3.7. Automatisierte Erzeugung des Knowledge Graphen

Idealerweise lässt sich der Knowledge Graph auch komplett automatisiert erstellen, sofern entsprechende Prozessmodelle im Unternehmen vorliegen, welche von einem Übersetzer ausgelesen werden und in die Graph-Notation überführt werden können. Dies ist z.B. der Fall bei

der Verwendung des IT-Werkzeugs bpmn.io²⁸, welches dank der standardisierten OMG-Modelle²⁹ der BPMN (vgl. [WeWG2015]) übersetzt werden kann, da die Strukturen klar ersichtlich sind. Abbildung 5-20 zeigt die Struktur einer Prozessmodellierung. Man sieht die klaren Zuordnungen der Aufgaben zu einem Prozess, indem ein Prozesselement die einzelnen Aufgaben klar umschließt durch die <process> </process>-Tags. Die Zuordnung von Identitäten zu einem Prozess geschieht direkt im Element durch das Attribut „processRef=“, welches auf eine Prozess-ID verweist. Die Zuordnung eines Datenspeichers zu einer Aufgabe geschieht wiederum mit Tags, indem ein <targetRef> </targetRef>-Tag die ID des Datenspeichers umschließt. Im Rahmen dieser Arbeit wurde kein automatisierter Übersetzer realisiert, aber die grundlegende Funktionsweise ist nicht anders als die der bereits bestehenden SP²IDER-Konnektoren, welche für verschiedene PLM- und IoT-Systeme prototypisch bereits umgesetzt wurde.

```

<?xml version="1.0" encoding="UTF-8" ?>
<definitions xmlns="http://www.omg.org/spec/BPMN/20100524/MODEL" xmlns:bpmndi="http://www.omg.org/spec/BPMN/
<collaboration id="Collaboration_0zhgqt4">
  <participant id="Participant_12qbs86" name="Prozessbeauftragter" processRef="Process_1" />
  <participant id="Participant_0dif6v8" name="Testcoenzer" processRef="Process_0ajpdgt" />
</collaboration>
<process id="Process_1" isExecutable="false">
  <startEvent id="StartEvent_ly45yut" name="hunger noticed">
    <outgoing>SequenceFlow_0h2ix7r</outgoing>
  </startEvent>
  <task id="Task_lhcentk" name="choose recipe">
    <incoming>SequenceFlow_0h2ix7r</incoming>
    <outgoing>SequenceFlow_0wnb4ke</outgoing>
    <dataOutputAssociation id="DataOutputAssociation_036j9lg">
      <targetRef>DataStoreReference_0ks8hss</targetRef>
    </dataOutputAssociation>
  </task>
  <exclusiveGateway id="ExclusiveGateway_15hulpt" name="desired dish?">
    <incoming>SequenceFlow_0wnb4ke</incoming>
  </exclusiveGateway>
  <dataStoreReference id="DataStoreReference_0ks8hss" name="PLM" />
  <sequenceFlow id="SequenceFlow_0wnb4ke" sourceRef="Task_lhcentk" targetRef="ExclusiveGateway_15hulpt" />
  <sequenceFlow id="SequenceFlow_0h2ix7r" sourceRef="StartEvent_ly45yut" targetRef="Task_lhcentk" />
</process>
<process id="Process_0ajpdgt">
  <startEvent id="Event_0a3az7c" name="Testauftrag">
    <outgoing>Flow_0abq8zw</outgoing>
  </startEvent>
  <sequenceFlow id="Flow_0abq8zw" sourceRef="Event_0a3az7c" targetRef="Activity_lsqhwzk" />
  <endEvent id="Event_085xoui" name="Test abgeschlossen">
    <incoming>Flow_0t7h840</incoming>
  </endEvent>
  <task id="Activity_lsqhwzk" name="Test durchführen">
    <incoming>Flow_0abq8zw</incoming>
    <outgoing>Flow_0t7h840</outgoing>
  </task>
  <sequenceFlow id="Flow_0t7h840" sourceRef="Activity_lsqhwzk" targetRef="Event_085xoui" />
</process>

```

Abbildung 5-20: BPMN-Prozessdefinition erstellt mit <http://bpmn.io>

Ein Faktor, der bei der automatisierten Erstellung eines Knowledge Graphen aus der Prozessbeschreibung heraus bedacht werden muss, ist die Qualität der zugrunde liegenden Prozessmodellierung. Sollte diese nicht gut genug sein und die gesuchten Daten nicht ausreichend darstellen, dann muss der Graph anschließend manuell angepasst werden, entsprechende Schritte sind im Prozessmodell der Methodik vorgesehen. Insofern ist auch bei der automatisierten Erstellung eines Knowledge-Graph die manuelle Bearbeitung ein relevanter Faktor und muss als Korrekturmaßnahme zur Verfügung stehen.

²⁸ <http://bpmn.io>

²⁹ <https://www.omg.org/spec/BPMN/>

Ein weiterer relevanter Faktor ist dabei die Lesbarkeit der Prozessbeschreibung. Das oben angerissene Beispiel zeigt einen auf bpmn.io modellierten Prozess, welcher entsprechend der OMG-Spezifikation für BPMN abgespeichert wurde. Andere IT-Werkzeuge können bei der Speicherung eine andere Syntax verwenden, welche jeweils darauf überprüft werden muss, ob eine automatisierte Übersetzung möglich ist. Das im Rahmen dieser Arbeit zur Erstellung der Prozessschabilder genutzte IT-Werkzeug der Firma BOC Group³⁰ verfügt z.B. über eine API³¹, mit der man auf Modelle zugreifen kann. Dies vereinfacht eine automatisierte Anbindung oder könnte sogar über einen Abfrageservice entsprechend auch bei Prozessänderungen die Veränderungen in den Knowledge Graph übernehmen und diesen so steht auf dem aktuellen Stand halten.

5.4. Methodik für die Datenverarbeitung im Kontext des Systems

Im Gegensatz zum Prozess, welcher beschreibt „was“ getan werden muss, um ein bestimmtes Ziel zu erreichen, dient die Methode dazu das „wie“ genauer zu beschreiben. Prozess und Methode greifen dabei Hand in Hand und können insbesondere durch IT-Tools unterstützt werden, welche die definierten Prozesse und Methoden kombinieren. Die folgenden Unterkapitel beschreiben den methodischen Ansatz, der gewählt wird, um Ähnlichkeiten zwischen der Ebene des Knowledge Graphen und den Objekten des SP²IDER-Graphen zu finden. Die Methodik ist dabei teilweise angelegt an den Mapping-Mechanismus von SP²IDER, welcher im Rahmen des Forschungsprojekts AKKORD entwickelt wurde.

5.4.1. Mappingvorgänge im SP²IDER-Graph

Auch im reinen Metadaten-Repository SP²IDER sind Mapping-Vorgänge ein wichtiger Teil des Konzepts. Diese sollen in der Mapping-Engine in mehreren Abstufungen möglich sein, ausgehend von einem manuellen Mapping bis hin zu einem System, welches durch KI unterstützt wird. Abbildung 5-21 zeigt die Einbindung der Mapping-Engine als Teil des SP²IDER-Core und in verschiedenen Ausprägungen [EiEG2020]. Das Mapping ist als solches sowohl auf der Typen- als auch auf der Objektebene durchführbar [EiEG2021].

³⁰ <https://www.boc-group.com/de/>

³¹ <https://developer.boc-group.com/adoxx/en/API-introduction/>

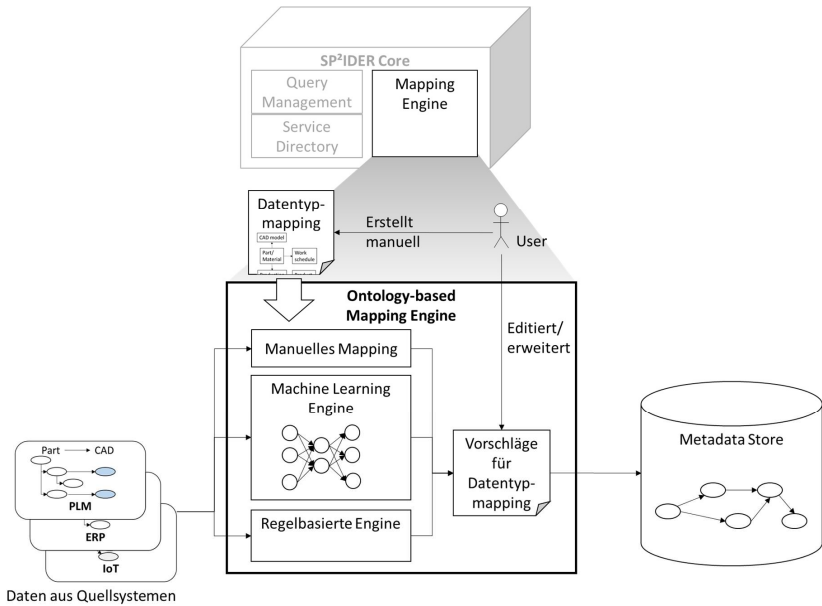


Abbildung 5-21: Mapping-Mechanismen als Teil des SP²IDER-Konzepts nach [EiEG2020]

Zur Durchführung werden im ersten Schritt die Datenmodelle der Datenquellsysteme durch den spezifischen Konnektor ausgelesen und an den SP²IDER Core übermittelt. Hierbei findet eine strukturelle Übersetzung der Datenschemata in die oben beschriebenen Strukturen statt, so dass jeder Datentyp als `type_GraphNode` in SP²IDER vorliegt. Sobald mindestens ein zweites System in SP²IDER eingelesen wurde, kann ein Mapping zwischen Datentypen stattfinden. Mapping bedeutet in diesem Fall, dass eine „entspricht“ Beziehung zwischen zwei `type_GraphNodes`, respektive zwischen zwei `object_GraphNodes` auf der Objekt-Ebene gezogen wird [EiEG2021].

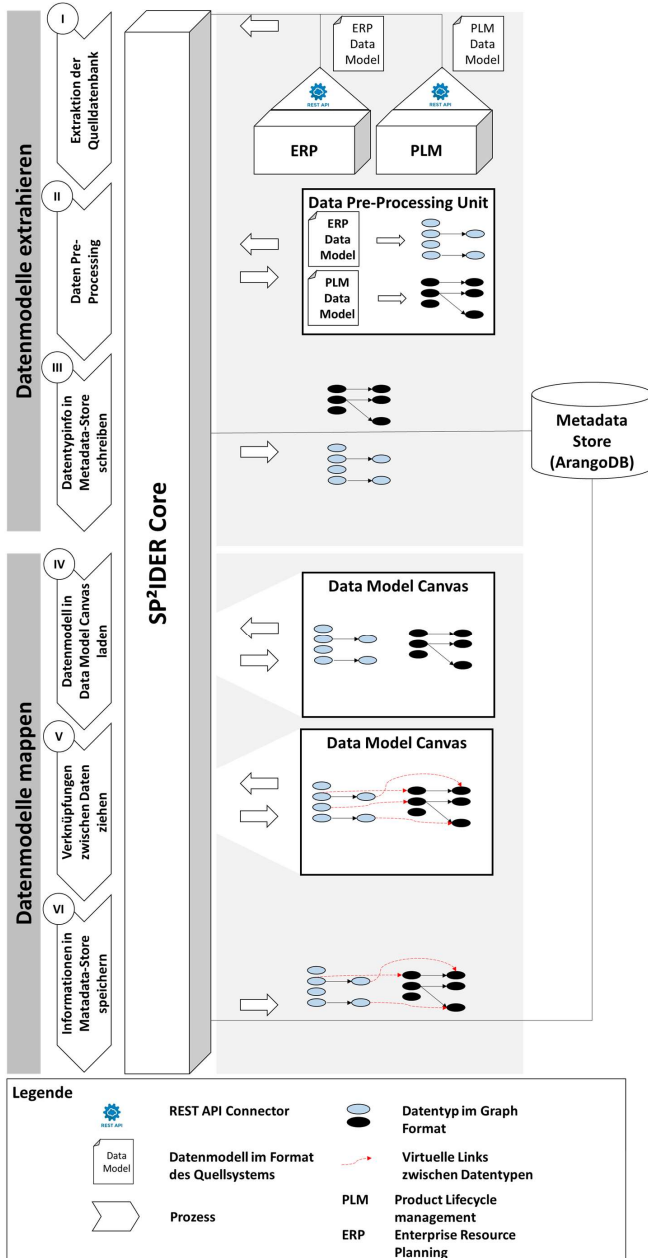


Abbildung 5-22: Beschreibung des SP²IDER Mapping Ansatzes nach [EiEG2021]

Abbildung 5-22 zeigt den Prozess, der vor, während und nach des Mappings durchlaufen wird. Im Sinne der mehrstufigen Mapping-Engine wird hier nur der manuelle Mapping-Prozess mithilfe des Data Model Canvas (DMC) geschildert [EiEG2021]. Abbildung 5-23: Erster Prototyp des Data Model Canvas aus [EiEG2021] zeigt dabei einen ersten Prototypen des DMC, in welchem Objekte aus unterschiedlichen Systemen durch das Ziehen einer Kante, was mit verschiedenen visjs-Bedienelementen realisiert wurde, miteinander vernetzt wurden [EiEG2021].

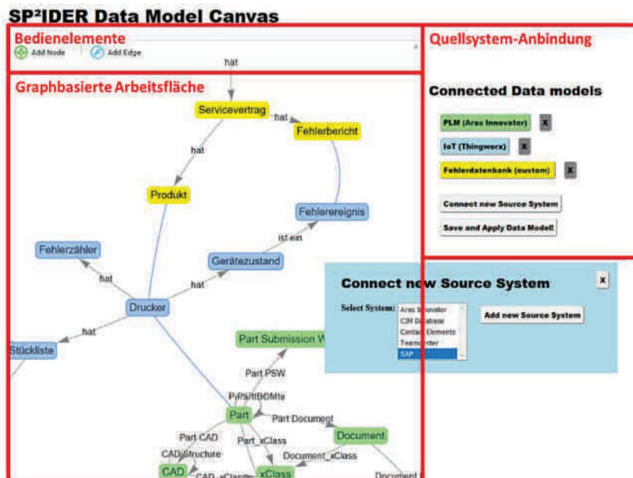


Abbildung 5-23: Erster Prototyp des Data Model Canvas aus [EiEG2021]

Aufbauend auf der Vernetzungsmethodik von SP²IDER durch das Ziehen neuer Kanten zwischen zwei Objekten soll nun auch eine Vernetzung zwischen Objekten des Knowledge Graph und des SP²IDER-Graph durchgeführt werden. Auf der Systemebene kann dies auch ohne große Komforteinbußen oder Zeitaufwand manuell durchgeführt werden. Hierzu werden Knoten wie z.B. „PLM“ und „Ara v12“, sprich die abstrakte Repräsentation eines Objekts im Knowledge Graph und seine IT Entsprechung miteinander markiert. Wie dies im Detail geschieht, erläutert das folgende Kapitel 5.4.2.

5.4.2. Assistiertes Finden von entsprechenden Datentypen

Ein wichtiger Schritt innerhalb der Methodik ist das automatisierte Verknüpfen von Elementen aus dem Knowledge Graph mit Datentypen, welche bereits im SP²IDER-Graph abgebildet sind. Dies dient dazu den Nutzer zu unterstützen und eine manuelle Suche innerhalb aller Datentypen zu vermeiden, da in modernen IT-Systemen häufig eine enorme Anzahl von Datentypen verwaltet wird. Die meisten dieser Typen haben keine Relevanz im Sinne eines Engineering-Repository und

dienen nur zur Sicherstellung von technischen Funktionalitäten. So sind z.B. in einem Aras Innovator v12 bereits über 460 Datentypen, ohne die Verknüpfungstypen hinzuzuzählen. Ein Contact CIM Database enthält sogar mehr als 860 Datentypen. Die genaue Anzahl kann abweichen, da es sich bei den hier genannten Zahlen um neu installierte und nicht weiter angepasste Systeme handelt, ist eine Abweichung insbesondere zu noch höheren Zahlen wahrscheinlich.

Basis des automatisierten Abgleichs ist die Verknüpfung eines Quellsystems im SP²IDER-Graph mit einer Datenquelle oder „Data Source“ im Knowledge Graph. Durch den SP²IDER-Core können alle Datentypen des Quellsystems und alle Datenobjekte der Datenquelle aus dem Knowledge Graph abgerufen und anschließend automatisiert ausgewertet werden.

Die so gespeicherten Daten werden mit all ihren Eigenschaften in zwei Listen abgespeichert und paarweise verglichen. So ist sichergestellt, dass jedes Objekt auch durchlaufen wird und falls eine der dabei ausgewerteten Strategien einen Treffer meldet, der Nutzer diesen dann auch bestätigen kann. Tabelle 5-3 zeigt eine Übersicht aller Objekte, welche mit den verschiedenen Strategien miteinander verglichen werden. Die verschiedenen Objekte liegen dabei alle als String vor.

Tabelle 5-3: Vergleichsobjekte zum Finden von Datentypen

| Vergleichsobjekt „type_Graphnode“ | Vergleichsobjekt „Data Object“ |
|-----------------------------------|--------------------------------|
| Name | Name |
| Einzelne Tags | Name |
| Wörter im Kommentarfeld | Name |
| Synonyme des Namens | Name |

Zusammengefasst stehen zum Vergleich der Objekte aus beiden Graphen folgende Methoden und Strategien zur Verfügung, welche alle in einer Python-Funktion zusammengefasst sind. Ausgewählt wurden die folgenden Strategien, welche im Folgenden beschrieben werden:

1. Übereinstimmung der Strings
2. Übereinstimmung der Strings nachdem alle Zeichen klein geschrieben sind
3. Übereinstimmung der Strings nach Übersetzung ins Englische
4. Übereinstimmung der Strings nach Übersetzung der klein geschriebenen Wörter ins Englische
5. Übereinstimmung der Strings nach Übersetzung ins Englische und Lemmatisierung
6. Auswertung der Levenshtein-Distanz nach Übersetzung ins Englische und Lemmatisierung

Der entscheidende Code, der alle Methoden umfasst ist in Abbildung 5-24 dargestellt. Durch die kompakte Arbeitsweise der Funktion und der prozeduralen Struktur des Codes können auch mit kleinen Veränderungen leicht weitere Strategien implementiert werden.

```

12 def all_strategies(word1, word2):
13     lemmatizer = WordNetLemmatizer()
14     if word1 == word2: return True
15     if word1.lower() == word2.lower(): return True
16     if helper.translate_deepl(word1) == helper.translate_deepl(word2): return True
17     if helper.translate_deepl(word1.lower()) == helper.translate_deepl(word2.lower()): return True
18     if lemmatizer.lemmatize(helper.translate_deepl(word1).lower()) == lemmatizer.lemmatize(helper.translate_deepl(word2).lower()): return True
19     distance = levenshtein_distance(lemmatizer.lemmatize(helper.translate_deepl(word1).lower()), lemmatizer.lemmatize(helper.translate_deepl(word2).lower()))
20     if distance/len(lemmatizer.lemmatize(helper.translate_deepl(word1).lower()))<len(lemmatizer.lemmatize(helper.translate_deepl(word2).lower())) < 1ev_empf:
21         return True
22     return False
23
24 def compare_new(type_graphnode, knowledge_object):
25     for tag in type_graphnode["tags"]:
26         if tag != "" and all_strategies(tag, knowledge_object["name"]) == True: return True
27     comment = type_graphnode["comment"].split(" ")
28     for word in comment:
29         if word != "" and all_strategies(word, knowledge_object["name"]) == True: return True
30     if all_strategies(type_graphnode["name"], knowledge_object["name"]) == True: return True
31     if helper.thesaurus_get(type_graphnode["name"]) != []:
32         theslist = helper.thesaurus_get(type_graphnode["name"])
33         for word in theslist:
34             if word != "" and all_strategies(word, knowledge_object["name"]) == True: return True
35     return False
36
37 def compare_all(type_graphnodes, knowledge_objects, case):
38     compdict = {}
39     for type_graphnode in type_graphnodes:
40         for knowledge_object in knowledge_objects:
41             if compare_new(type_graphnode, knowledge_object) == True:
42                 if type_graphnode["case"] in compdict.keys():
43                     knowledge_list = []
44                     knowledge_list.append(compdict[type_graphnode["case"]])
45                     knowledge_list.append(knowledge_object["case"])
46                     compdict[type_graphnode["case"]] = knowledge_list
47             else:
48                 compdict[type_graphnode["case"]] = knowledge_object["case"]
49     return compdict

```

Abbildung 5-24: Code zur Überprüfung der Ähnlichkeit von Strings

Die naheliegendste Methode ist der direkte Vergleich der zwei Objekte. So wäre „Part“ == „Part“ als Ergebnis True und damit eine Übereinstimmung. Das Problem bei dem direkten Vergleich ist, dass eine wirkliche Zeichengleichheit vorliegen muss. Sobald auch nur der geringste Unterschied vorhanden ist, wird kein Treffer mehr erkannt. Dies ist z.B. relevant, falls ein Leerzeichen doppelt gesetzt wurde oder fehlt, falls ein Wort in einem System groß- und im anderen kleingeschrieben wurde. Aus diesem Grund wird als nächste Strategie der Vergleich von Strings nach Überführung in die Kleinschreibung gewählt.

Der Vergleich von Strings, die Unterschiede in der Groß- und Kleinschreibung aufweisen kann durch ein Umwandeln in komplett klein geschriebene Strings durchgeführt werden. „Part“.lower() == „part“.lower() ist in dieser Strategie auch True und damit eine Übereinstimmung. Dies kann vor allem vorteilhaft sein, sollten Namen aus mehreren Bestandteilen bestehen und sich in Groß/Kleinschreibung unterscheiden. Das bereits vorher bestehende Problem bei einer Ungleichheit in der Nutzung von Leerzeichen oder auch Satzzeichen bleibt auch in dieser Strategie bestehen.

Als nächstes Strategiepaar werden eine Übersetzung bzw. eine Übersetzung in Kombination mit Kleinschreibung verglichen. Beispielsweise würde „Document“ und „Dokument“ als Übereinstimmung erkannt und als Antwort auf die Frage True zurückgeliefert. Hierzu wurde über den Webdienst deepl³² eine API gewählt, an welche Wörter in beliebiger Sprache gesendet werden können, um anschließend eine Übersetzung in die englische Sprache zurückzuliefern. Stichproben haben gezeigt, dass die Übersetzung in den meisten Fällen gut funktioniert, wengleich manche Zuordnungen auch falsch sind und teilweise die Wörter einer anderen

³² <https://www.deepl.com/translator>

Ausgangssprache zugeordnet werden. Allerdings ist die Fehlerquote an dieser Stelle so gering, dass diese toleriert werden kann. Aufgrund der begrenzten Zeichenzahl, welche mit der kostenlosen deepl-API übersetzt werden können, werden die Übersetzungsergebnisse gleichzeitig auch in einem JSON-Dictionary abgespeichert, auf das für weitere Übersetzungen zugegriffen werden kann.

Als letztes Strategiepaa dient eine gleichzeitige Anwendung von Kleinschreibung, Übersetzung, Lemmatisierung und der Levenshtein-Distanz. Durch Kleinschreibung und Übersetzung wird eine möglichst ähnliche Ausgangsbasis für die zu vergleichenden Wortpaare herbeigeführt. Anschließend findet mittels Lemmatisierung die Zurückführung der Wörter auf ihre Grundform statt. Dies geschieht algorithmisch durch die Nutzung einer Python-Bibliothek statt. Hierzu wird die Bibliothek nltk³³ – natural language toolkit – gewählt. Durch die Lemmatisierung wird z.B. „parts“ auf seine Grundform „part“ zurückgeführt. Anschließend werden nicht diese Grundformen der Wörter verglichen, sondern direkt ein Schritt weiter gegangen und mithilfe der Levenshtein-Distanz die Abweichung der Strings festgestellt. Hierbei wird zeichenweise verglichen und für jede Abweichung die Levenshtein-Distanz um eins erhöht. Dies soll hier beispielsweise gezeigt werden:

P A R T S

B A U T E I L E

1 0 1 0 1 1 1 1 = 6 // 1 falls Ungleichheit

Um aus dem String „PARTS“ den String „BAUTEILE“ zu formen sind sechs Ersetzungen bzw. Ergänzungen notwendig, die Levenshtein-Distanz entsprechend sechs. Die Levenshtein-Distanz allein ist erstmal noch nicht aussagekräftig, bei einem sehr langen String oder Wort kann auch eine höhere absolute Zahl noch eine Ähnlichkeit bedeuten, während es bei einem kurzen Wort keine Ähnlichkeit gibt. Gleichzeitig ist ein weiteres Problem der Vergleich von unterschiedlich langen Wörtern. Welches der Wörter wird hierbei als Ausgangspunkt für einen Faktor genommen. Um diesem Problem zu begegnen, wird beim Vergleich die Levenshtein-Distanz durch die durchschnittliche Länge beider Strings geteilt und zusätzlich noch ein Schwellenwert definiert, unter dem eine Ähnlichkeit festgestellt wird. Die genaue Größe der Schwelle ist dabei keine zwingende Größe, sondern kann individuell angepasst werden. Als ein pragmatisches Maß wird ein Wert von 0,3 vorgeschlagen und im Programm hinterlegt. In einer späteren Umsetzung ist es denkbar, dass diese Schwelle auch direkt über eine veränderbare Einstellungsmöglichkeit angepasst werden kann. Ein Anlass für eine Anpassung wären reale Tests, bei denen entweder zu viele vermeintliche Übereinstimmungen gefunden werden, die allerdings vom Nutzer nicht bestätigt werden, oder zu wenige Vorschläge durch den Algorithmus. Je nach Situation würde die Schwelle entweder nach oben oder unten angepasst und ein neuer Durchlauf des Algorithmus gestartet.

³³ <https://www.nltk.org/>

Nach dem Durchlaufen der verschiedenen Analyseschritte für die Wortpaare aus Tags, Kommentaren und Namen auf der einen Seite und den Datenobjekten auf der anderen Seite gibt es für jedes Paar entweder eine Übereinstimmung oder nicht. Im Falle einer Übereinstimmung werden die beiden Objekte des Paares dem Nutzer präsentiert und er kann entscheiden, ob er dieses Ergebnis des Algorithmus übernehmen will, in diesem Fall wird eine dauerhafte Verbindung zwischen den beiden Knoten gezogen, oder ob der Algorithmus eine falsche Entscheidung getroffen hat und diese Entscheidung nicht übernommen werden soll, sondern verworfen wird.

5.5. Rollenkonzept für die industrielle Einführung

Durch die Integration des Knowledge Graph in das SP²IDER-Metadaten-Repository und die in den vorherigen Kapiteln beschriebenen zusätzlichen Funktionen und Prozessschritte steigt die Komplexität des IT-Systems stark an. Dem soll durch ein neues Rollenkonzept Rechnung getragen werden, welches die funktionalen Aspekte der Konzepterweiterung abdeckt. Alle Rollen, welche für ihre Aufgaben besondere Funktionen benötigen, sind hier abgedeckt. Im Gegenzug dazu steht das Sichtenkonzept, welches innerhalb der Rolle „allgemeiner Nutzer“ die Sichtbarkeit von Objekten steuert.

5.5.1. Dateningenieur/experte

Der Dateningenieur ist die Rolle, die vorwiegend mit der Abwicklung der Prozesse in diesem Konzept betraut ist. Dies umfasst insbesondere sämtliche Kernprozesse, welche in Kapitel 5.3 beschrieben sind. Er wertet die vorhandenen Prozessbeschreibungen aus, erstellt darauf basierend den Knowledge Graph und sorgt für eine regelmäßige Aktualisierung dessen. Weiterhin stößt der Datenexperte auch die automatisierten Verknüpfungsprozesse an, evaluiert die Ergebnisse und stellt so die Nutzbarkeit des Knowledge Graph basierten Assistenzsystems für alle beteiligten Nutzer sicher.

Weiterhin ist eine wichtige Aufgabe des Datenexperten die Definition der verschiedenen Rollen und das Zuweisen von Objekten zu einer Sicht, die einer Rolle zugewiesen wird. Der Datenexperte legt dadurch fest, welche Daten beispielsweise für einen Konstrukteur oder einen Disponenten in der Arbeitsvorbereitung relevant sind.

5.5.2. Administrator

Der Administrator ist im IT-System eine rein technisch/funktionale Rolle zur Sicherstellung der grundlegenden Funktionalitäten. Hierzu gehört es, dass aufgrund seiner umfassenden Rechte im System, er die weiteren Nutzer anlegen kann und den Nutzern bezogen auf das Metadata Repository- oder den Knowledge Graphen funktionale Rollen wie den Domäneningenieur oder die Rolle des Datenexperten/Dateningenieurs zuweisen kann.

5.5.3. Domäneningenieur

Domäneningenieure sind Nutzer, die sich vor allem mit Daten aus einem Bereich befassen und diese nutzen, um ihrer festgelegten Arbeit nachzugehen. Beispiele wären z.B. ein Konstrukteur in der Produktentwicklung oder ein Disponent in der Arbeitsvorbereitung. Domäneningenieure haben durch ihr Stellenprofil bereits Zugang zu vielen notwendigen Daten in einem oder mehreren IT-Systemen. In der Praxis zeigt sich allerdings oftmals, dass notwendige Daten doch nicht vollständig bekannt oder erreichbar sind, weswegen sie durch das Knowledge Graph basierte Assistenzsystem die Möglichkeit erlangen, weitere Daten in den IT-Systemen zu finden, um anschließend einen Zugriff freigeschaltet zu bekommen.

5.5.4. allgemeiner Nutzer

Weitere Nutzer können unbegrenzt im System angelegt werden. Sie haben allerdings weniger eine vorher fest definierte Rolle, welche auch Funktionalitäten hinsichtlich des Knowledge Graphen umfasst, als mehr die Funktion des Authentifizierens um anschließend über das Sichtenkonzept bestimmte Ausschnitte des Modells anzuzeigen. Der allgemeine Nutzer ist eine übergeordnete Klasse, welche auch Administratoren, Dateningenieure und Domänennutzer in sich vereint.

5.6. Sichten auf den Knowledge- und SP²IDER-Graph als Berechtigungsinstrument

Ein wichtiges Ziel des Konzepts ist die bessere Nutzbarkeit eines Metadata-Repositories, konkret im Prototypen-Demonstrator des SP²IDER-Graphen. Abbildung 5-25 zeigt wie unübersichtlich eine Ansicht des SP²IDER-Graphen werden kann. In diesem Fall handelt es sich lediglich um alle Type_Graphnodes, welche dem System Aras Innovator v12 zugeordnet sind. Vor allem durch die vielen Verbindungen und die systemseitig angelegten Datentypen wird die Menge der angezeigten Daten sehr groß.

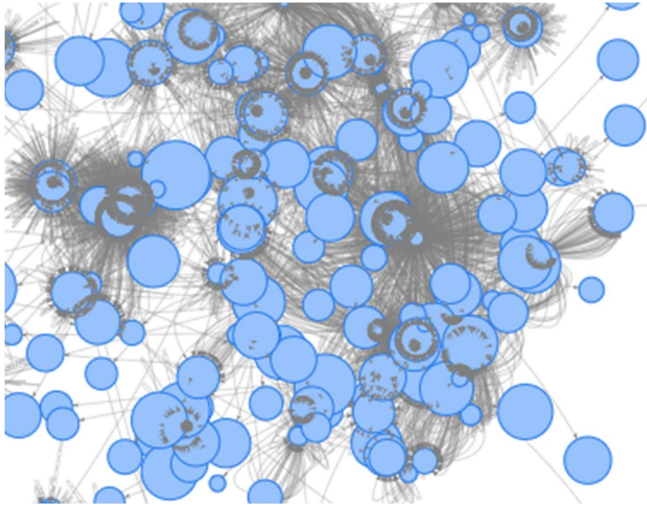


Abbildung 5-25: Illustration der Komplexität in einem Metadaten-Repository

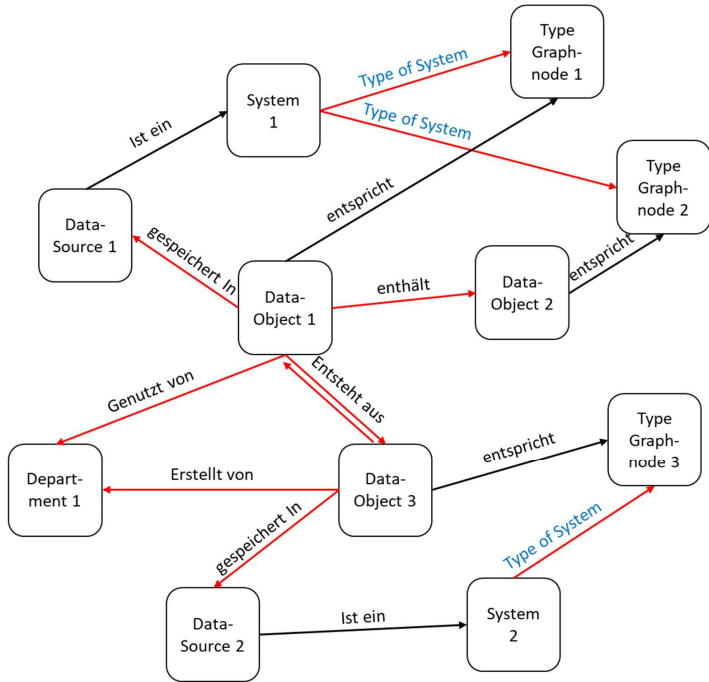
Ebenfalls ist die Performance der Ansicht nicht nutzerfreundlich und hat Ladezeiten im Bereich mehrerer Minuten zur Folge. Auch wenn dies durch die Wahl einer schnelleren Graph-Bibliothek in JavaScript voraussichtlich positiv beeinflusst werden könnte, so bleibt doch das Grundproblem der fehlenden Übersichtlichkeit bestehen, wenn die Sicht auf dem Graphen nicht eingeschränkt wird. Dieses Problem wird mit dem Knowledge Graph basierten Assistenzsystem erstmal noch größer, da das Konzept davon lebt, dass viele verschiedene IT-Systeme angebunden werden, sowie der Knowledge Graph als zusätzliche Instanz auch abgebildet werden muss.

Als Lösungsansatz für dieses Problem dient ein Rollen- und Sichtenkonzept, welches die Sichtbarkeit aller Objekte im Knowledge Graph, aber vor allem die Sichtbarkeit der IT-Objekte aus dem Metadaten-Repository steuert. Unterschiedliche Rollen haben unterschiedliche Zugriffsrechte auf die Objekte und sehen entsprechend nur für sie freigegebene Dinge. Dieser Ansatz orientiert sich an der Idee unterschiedlicher Sichten auf Engineering Objects (EO) [FaDa2007], welcher von Mogo Nem [Mogo2011] entwickelt wurde. Die zentrale Idee ist es, dass Engineering Objekte nur in einem speziellen Kontext und zu einem speziellen Zeitpunkt sinnvoll betrachtet werden können. Dementsprechend werden sie gebündelt und die zu einem speziellen Zeitpunkt sinnvolle Sicht fixiert [Mogo2011].

Das Anlegen der entsprechenden Sichten und Rollen, sowie User ist in Kapitel 5.3.5 sowie 5.3.6 beschrieben. Nach einer anschließenden Zuweisung einer Rolle zu einem User stellt sich nach dem Login folgende Situation dar: Der Nutzer hat die Möglichkeit aus allen ihm zugewiesenen Rollen diejenige auszuwählen, die er gerade als Sichtweise auf den Knowledge Graph und das Metadaten-Repository benötigt. Diese Auswahl einer Rolle bzw. Sicht nach dem Login soll wechselbar sein, um bei Bedarf weitere Typen oder Objekte sehen zu können.

Die Verknüpfungen zwischen der Rolle und den weiteren Elementen des Knowledge-Graph und den Elementen des Metadaten-Repository werden über „sieht“ bzw. „sees“-Verbindungen realisiert. Die Schwierigkeit liegt darin begründet, dass in einem Knowledge oder Metadata Graph letztlich alle Objekte miteinander verbunden sind, sofern sie nicht eine fachlich und systemseitig vollkommen isolierte Insel bilden. Eine reine Beschränkung auf eine Anzahl von Verbindungen x , welche maximal ausgewertet werden dürfen, ist zu allgemein und kann zu Objekten führen, welche nicht sichtbar sein sollte. Aus diesem Grund wurde definiert, dass manche Verbindungen in eine oder beide Richtungen nicht ausgewertet werden dürfen, um nicht unbeabsichtigt zu umfassenderen Graph-Objekten zu gelangen, wie z.B. einem gesamten IT-System.

Zeigt eine Rolle direkt auf einen Type_GraphNode oder ein DataObject, dann darf der User den zugeordneten Type_GraphNode sehen. Die Verbindung ist entweder direkt oder wird über die Equal_To-Verbindung gezogen. Zeigt eine Rolle auf eine DataSource bzw. ein System, dann werden alle hier zugeordneten Type_GraphNodes durch Auswertung der Verbindung „IsA“ zwischen System und DataSource und anschließend durch Auswertung der „TypeOfSystem“ innerhalb des SP²IDER-Graphen angezeigt. Wenn die Rolle auf eine Abteilung/Department zeigt, dann, und nur dann, werden die „Used By“ und „Made by“ Verbindungen ausgewertet und über die dann verbundenen DataObjects entsprechende Type_Graphnodes angezeigt. Die Beziehungen „Used By“ und „Made By“ werden in allen anderen Fällen nicht ausgewertet, da sie sonst die Wirkung haben, DataObjects aus weiteren Systemen und Abteilungen anzuzeigen. Zeigt eine Rolle auf einen Prozess, dann wird die Verbindung „Part of“ ausgewertet und das entsprechende DataObject bzw. Type_Graphnode angezeigt.



Legende



Knowledge Graph Beziehung, darf ausgewertet werden

Knowledge Graph Beziehung, darf nicht ausgewertet werden

Metadaten-Beziehung, darf nicht ausgewertet werden

Abbildung 5-26: Nicht ausgewertete Verbindungen im Rollen und Sichtenkonzept

Abbildung 5-26 zeigt die Verbindungen, welche nicht ausgewertet werden dürfen, um nicht wesentlich mehr Objekte sichtbar zu machen, als unbedingt notwendig und beabsichtigt. Bei den meisten Verbindungen ist vor allem eine Richtung betroffen, da eine Auswertung des Graphs in diese Richtung den Graphen bedeutend vergrößern würde. Beispielsweise durch das Auswerten einer „saved in“-Verbindung würden anschließend das ganze System ausgewählt sein, auch wenn die Rolle nur einzelne DataObjects sehen sollte. Auch werden die Verbindungen „made from“, „used by“ und „made by“ nicht ausgewertet, da sie die geeignet sind, Verbindungen quer durch den ganzen Produktlebenszyklus zu ziehen. So könnte über mehrere „made from“ Beziehungen ausgehend von einer Anforderung dann auch ein IoT-Device sichtbar sein. Dies widerspricht

sowohl dem Konzept der Rollen als auch der Grundidee, durch Ausschnitte die Komplexität in den Daten beherrschbar zu gestalten.

6. Verifikation und Validierung

Das im vorherigen Kapitel vorgestellte Konzept soll nicht nur beschrieben, sondern in Teilen auch verifiziert und validiert werden. Die Verifikation findet hierbei insbesondere auf Ebene der Software des Demonstrators statt, welcher das Konzept direkt benutzbar umsetzt. Als Testfall dient dabei der Use-Case des interdisziplinären Engineerings und der Produktionsplanung, welcher sich über die Produktentwicklung bis hin in die Arbeitsvorbereitung zieht und verschiedene IT-Systeme umfasst. Anhand von Screenshots wird gezeigt, wie entlang dieses Testfalls Daten miteinander integriert werden, um anschließend den Knowledge Graph nutzen zu können. Darüber hinaus fand in Ansätzen eine Validierung durch Interviews mit Forschenden statt, welchen die Idee präsentiert wurde, um die Sinnhaftigkeit der Anforderungen und des Konzepts zu diskutieren.

6.1. Anwendungsszenario

Das Beispiel sieht einen Ingenieur vor, welcher in der Produktentwicklung im Zuge einer Produkt Rekonfiguration eine Überarbeitung vornehmen soll. Hierbei sollen neben den bestehenden Anforderungen insbesondere auch Gesichtspunkte der Fertigung berücksichtigt werden, um die Auswirkung der Änderung in diesem Bereich möglichst zu optimieren. Hierzu zählt eine Weiterverwendung bestehender Fertigungshilfsmittel, eine schnelle Benachrichtigung, welche CNC-Programme und Arbeitspläne angepasst werden müssen, eine Berücksichtigung wie viele Bauteile noch im Lager vorhanden sind und wiedergenutzt werden können. Ebenfalls sollen Fehler und Rückmeldungen aus der Nutzungsphase des Produkts in die Produktentwicklung integriert werden.

Die benötigten Daten erstrecken sich über verschiedene Datensilos aus unterschiedlichen Abteilungen, welche einem Konstrukteur nicht in Gänze bekannt sind, bzw. auf die er keinen Zugriff hat. Insbesondere betroffen sind in diesem Anwendungsbeispiel ein PLM-System zur Speicherung der Daten aus der Produktentwicklung, ein ERP-System, in welchem die Daten aus der Produktion und Arbeitsvorbereitung gespeichert sind. Die Nutzungs- und Fehlerdaten der Produkte sind in einem IoT-System hinterlegt.

Um den Konstruktionsprozess zu vereinfachen ist es sinnvoll, dass der Entwicklungsingenieur über all diese Datenquellen Bescheid weiß und auf die entsprechenden Daten zugreifen kann. Hierzu muss vorher von der Engineering IT entsprechende Vorarbeit geleistet werden.

Grundlegend muss die Softwarelösung aus dieser Arbeit in der Firma implementiert und eingerichtet werden. Hierzu gehört sowohl das Metadaten-Repository SP²IDER und seine Verknüpfung mit den bereits vorhandenen IT-Systemen als auch der Data Model Canvas als Nutzeroberfläche zur Interaktion mit den Daten aus SP²IDER und dem Knowledge Graph.

Anschließend müssen die Unternehmensprozesse analysiert und in die Form des Knowledge Graph übertragen werden. Anschließend können die bestehenden IT-Systeme mit den abstrakten IT-System-Knoten aus dem Knowledge Graph verbunden werden und der Ähnlichkeitsalgorithmus gestartet werden. Dieser wertet Ähnlichkeiten zwischen den Datentypen der IT-Systemen und den Datenobjekten aus dem Knowledge Graph aus und schlägt dem Nutzer vor Verknüpfungen zu ziehen. Anschließend können auch manuell weitere Verknüpfungen zwischen Knowledge-Graph und IT-Systemen bzw. Daten gezogen werden. Das Ergebnis dieses Prozesses ist ein vollintegriertes System, welches dann auch vom oben erwähnten Konstrukteur genutzt werden kann, um bei der Rekonfiguration eines Produkts alle Informationsartefakte zu finden. Die Zugriffsrechte darauf müssen ggf. noch bei der IT beantragt und gewährt werden.

Aus pragmatischen Gründen wird der Fokus des Anwendungsszenarios insbesondere auf die Arbeit mit dem Data Model Canvas, die dort beinhaltete Modellierung des Knowledge-Graph, sowie die Verknüpfung mit den Ist-Daten gelegt. Die vorbereitenden Schritte, wie die Verbindung zu Datenquellsystemen wird im Fokus dieses Anwendungsszenario nicht vertieft, da es bereits u.a. in [EiEG2020] und [EiEG2021] beschrieben wurde.

6.1.1. Modellierung des Knowledge Graphen

Wenn klar ist, welche Elemente im Knowledge Graph enthalten sein sollen und alle Vorarbeiten abgeschlossen sind, dann kann die Modellierung beginnen. Hierzu wird der Data Model Canvas aufgerufen, welcher im ersten Moment eine leere Oberfläche anzeigt – vgl. Abbildung 6-1. Auf der linken Seite können neue Elemente hinzugefügt werden, im mittleren Bereich werden alle Elemente angezeigt, entweder aus dem Knowledge Graph, aus IT-Quellsystemen oder aus beiden Bereichen. Auf der rechten Seite gibt es einige Menüoptionen wie z.B. die Systemauswahl oder das Anzeigen von bereits angelegten Wissensselementen. Weiterhin gibt es noch andere Menüelemente, welche allerdings in diesem Anwendungsszenario nicht benötigt werden.

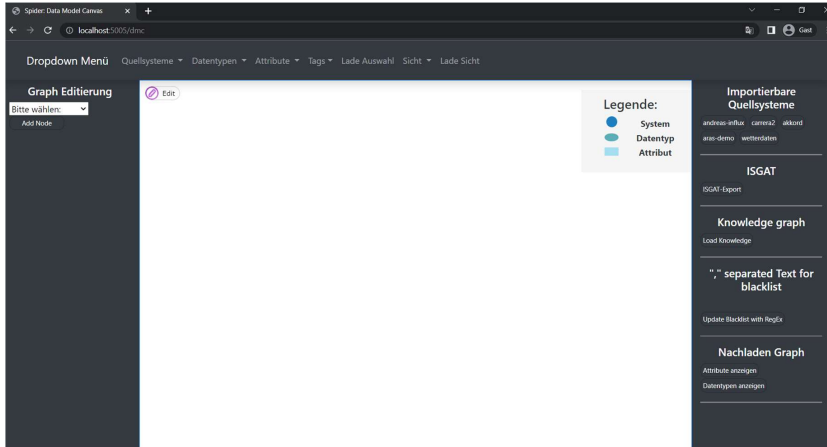


Abbildung 6-1: Template des DMC zum Beginn der Modellierung

Der erste Schritt, wie in Kapitel 5.3.2 beschrieben ist die Modellierung der einzelnen Prozessschritte. In diesem Fall sind dies gemäß der Beschreibung des Anwendungsfalls „Produkt entwickeln“, „Produktion planen“, sowie „Produkt nutzen“. Die modellierten Prozesse werden im Data Model Canvas in Form von Ellipsen angezeigt und mit einer eigenen Farbe dargestellt. Abbildung 6-2 zeigt eine Detailansicht der modellierten Elemente, auf der linken Seite ist die letzte Eingabe, der Prozess mit dem Namen „Produkt nutzen“ noch sichtbar. Durch ein Klicken auf den Button „Add“ wird direkt das Objekt angelegt und in der Datenbank gespeichert.

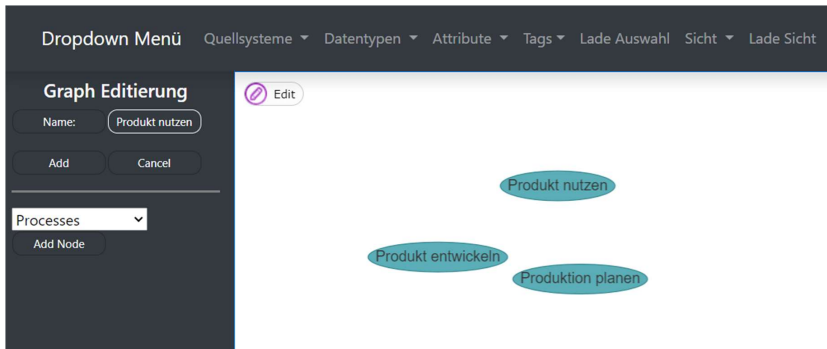


Abbildung 6-2: Detailansicht DMC mit modellierten Prozessen

Der nächste Schritt nach dem Modellieren der Prozesse ist das Anlegen der einzelnen Abteilungen, in welchen diese Prozesse ablaufen. In diesem Szenario werden dafür die Abteilungen Produktentwicklung, Arbeitsvorbereitung und Operations angelegt. Um keine Verwechslungen zwischen Prozessen und Abteilungen entstehen zu lassen, werden hinter die

Namen noch (Abt) als Kürzel hinzugefügt. Um Verwechslungen weiter vorzubeugen wird eine andere Farbe und Form als für die oben angelegten Prozesse verwendet. Abbildung 6-3 zeigt den Data Model Canvas mit den modellierten Abteilungen.

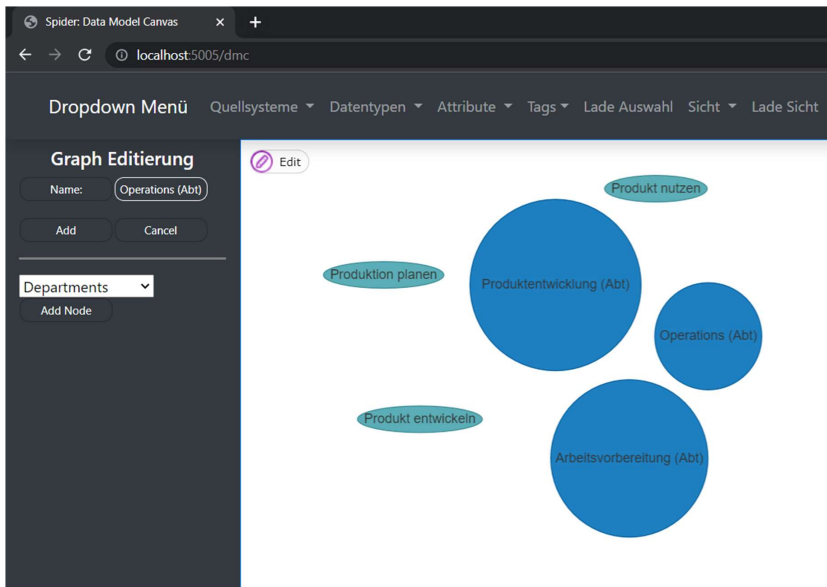


Abbildung 6-3: Detailansicht DMC mit Prozessen und modellierten Abteilungen

Der nächste Schritt ist das Modellieren der Datenobjekte, welche in diesem Prozess eine Rolle spielen. Im oben beschriebenen Anwendungsszenario sind dies insbesondere Produktdaten, Stücklistendaten, Prozesspläne, Arbeitspläne, Messmittel und Werkzeuge, sowie Nutzungsdaten. Abbildung 6-4 zeigt den Stand des Knowledge Graph nach diesem Modellierungsschritt.

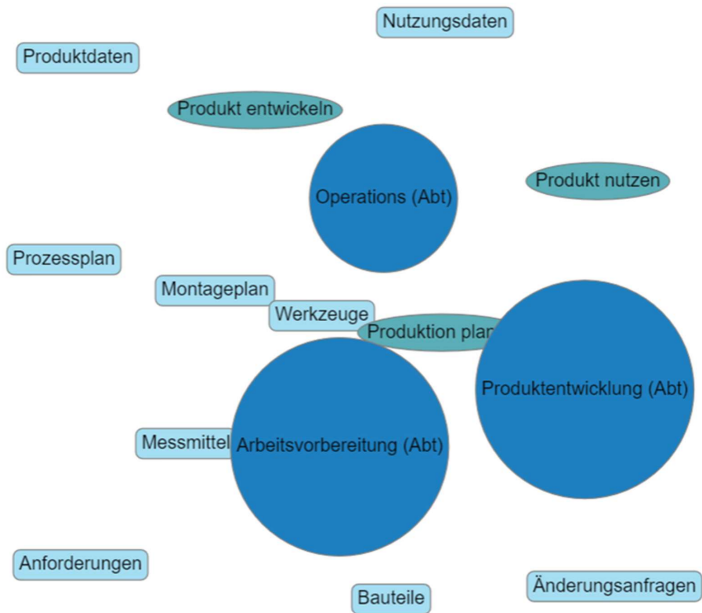


Abbildung 6-5: Knowledge Graphen mit allen modellierten Elementen

Trotzdem ist die Modellierung des Knowledge Graphen noch nicht abgeschlossen. Um ein sinnvolles Arbeiten zu ermöglichen, müssen noch alle Verbindungen zwischen diesen Objekten generiert werden. Dies kann auch direkt im Data Model Canvas durchgeführt werden, mithilfe der Editierfunktion in der oberen linken Ecke des Data Model Canvas. Nach einem Klick auf Edit öffnet sich ein Menü, welches das Hinzufügen von Knoten und Kanten erlaubt (vgl. Abbildung 6-6)

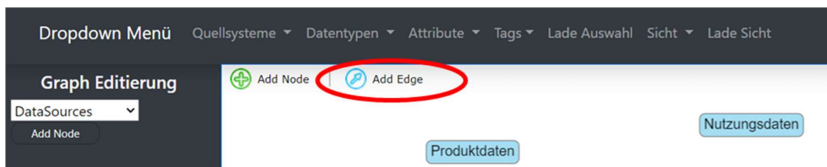


Abbildung 6-6: Editierfunktion für Kanten im Data Model Canvas

Die Funktion Kanten hinzufügen hat kein Auswahlmenü, um die entsprechende Kante auszuwählen, diese wird definiert über die Objekte, die durch sie verbunden werden. Hierbei gibt es drei Fälle: Eine Kante kann eindeutig sein, z.B. zwischen einem Datenobjekt und einem Datenspeichersystem. Hierbei handelt es sich immer um eine Kante mit der Eigenschaft „saved in“ bzw. „gespeichert in“. Diese Kante wird sofort gezogen und im System hinterlegt. Sollten zwei

Objekte miteinander verbunden werden, zwischen denen keine Verbindung definiert wurde, dann wird der User darauf hingewiesen, dass die entsprechende Kante nicht möglich ist und es wird keine Verbindung angelegt. Zuletzt kann eine Verbindung gezogen werden, bei der mehrere Kanten möglich sind. In diesem Fall wird der User über ein Eingabefenster über die Möglichkeiten informiert und kann anschließend durch die Eingabe einer Zahl die entsprechende Kante auswählen und anlegen lassen. Abbildung 6-7 zeigt dieses Eingabefenster am Beispiel einer Verbindung zwischen Datenobjekt und Abteilung an.

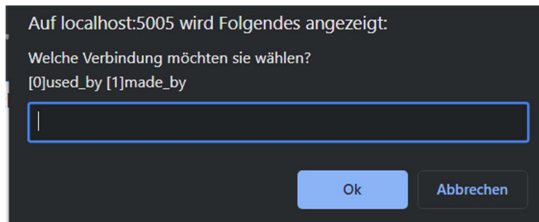


Abbildung 6-7: Benutzereingabe bei mehreren Kanten-Optionen

Der letzte Schritt, das Anlegen der Kanten kann auf verschiedenen Wegen systematisch durchgeführt werden. Entweder der User legt zuerst alle Kanten zwischen Objekttyp A und Objekttyp B an und arbeitet eine Kategorie Kanten nach der anderen ab.

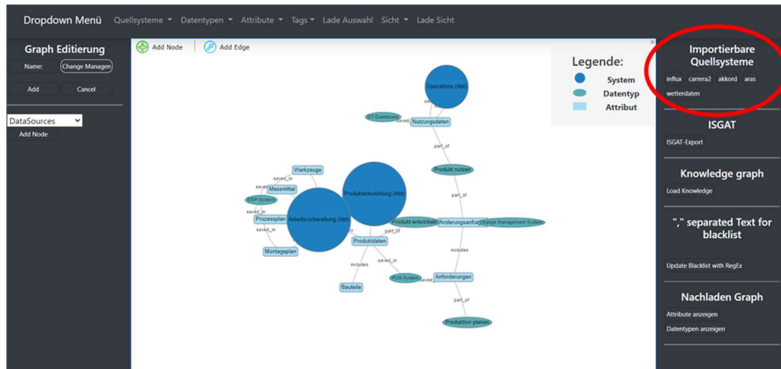


Abbildung 6-9: Importierbare Quellsysteme im rechten Menü des DMC.

Abbildung 6-9 zeigt die verfügbaren Quellsysteme im rechten Auswahlmenü des Data Model Canvas an. Durch den Klick auf ein System wird der entsprechende Knoten in den Data Model Canvas geladen, dort mit einigen wichtigen enthaltenen Datentypen dargestellt und können anschließend mit den Datenspeicherort-Knoten des Knowledge Graphen verbunden werden.



Abbildung 6-10: Aras als verknüpftes PLM & Change Management System

Abbildung 6-10 zeigt wie das IT-System Aras vom User als Change-Management System sowie als PLM-System ausgewählt und verbunden wurde. Diese Verbindung wird in einem nächsten Schritt durch einen Algorithmus ausgewertet und die Datenobjekte des Knowledge Graphen werden in den realen Datenobjekten des IT-Systems Aras gesucht bzw. eine Ähnlichkeitsuche findet.

Der Algorithmus muss momentan noch manuell gestartet werden. Hierzu wird ein spezielles Datenspeichersystem, wie z.B. das oben ausgewählte „aras-demo“-System mit allen Datenobjekten aus dem Knowledge Graph verglichen. Als Ergebnis dieser Vergleiche gibt es mögliche Treffer. Abbildung 6-11 zeigt in einem Python Dictionary die ermittelten Treffer als Wertepaar. Ein „type_graphnode“ ist dabei ein Objekt aus dem bestehenden IT-System, in diesem Fall Aras, ein „DataObject“ ein Objekt aus dem neu angelegten Knowledge Graph.

```
{'type_graphnodes/3035275': 'DataObjects/3254422', 'type_graphnodes/3035285': 'DataObjects/3254422', 'type_graphnodes/3035355': 'DataObjects/3254430', 'type_graphnodes/3035917': ['DataObjects/3254422', 'DataObjects/3254430'], 'type_graphnodes/3036357': 'DataObjects/3254422', 'type_graphnodes/3036547': 'DataObjects/3254422', 'type_graphnodes/3037277': 'DataObjects/3254422', 'type_graphnodes/3037651': 'DataObjects/3254458', 'type_graphnodes/3038331': 'DataObjects/3254470', 'type_graphnodes/3038651': 'DataObjects/3254410', 'type_graphnodes/3038671': 'DataObjects/3254410', 'type_graphnodes/3038701': ['DataObjects/3254410', 'DataObjects/3254458'], 'type_graphnodes/3038711': ['DataObjects/3254410', 'DataObjects/3254458'], 'type_graphnodes/3038721': ['DataObjects/3254410', 'DataObjects/3254458'], 'type_graphnodes/3039545': 'DataObjects/3254430'}
```

Abbildung 6-11: Liste aller möglichen Übereinstimmungen

Nach dem Durchlaufen aller Strategien und der Ausgabe der möglichen Paare liegt es am User, diese Vorschläge zu bewerten und zu bestätigen, damit eine Verbindung zwischen diesen Objekten angelegt wird. Diese manuelle Überprüfung wurde zusätzlich gewählt, da die Vorschläge des Algorithmus zwar Ähnlichkeiten finden, allerdings nur aufgrund dieser Information keine sichere Aussage getroffen werden kann, sondern eine Kenntnis über die wahre Bedeutung fehlt.

```
{'_key': '3035285', '_id': 'type_graphnodes/3035285', '_rev': 'ezPyfaa---', 'name': 'CAD Part', 'source_system': 'aras-demo', 'source_id': '7B6A83878B184D86A09472D856219D81', 'source_uri': 'http://aras.mv.uni-kl.de/innovatorserver/server/odata/itemtype('7B6A83878B184D86A09472D856219D81')', 'tags': [], 'comment': 'Verknüpfung zu Dokumenten'}  
{'_key': '3254422', '_id': 'DataObjects/3254422', '_rev': 'gRw1dda---', 'name': 'Bauteile', 'created': '1688940805', 'modified': '1688940805'}  
Sollen diese Objekte verknüpft werden? --> y
```

Abbildung 6-12: Manuelle Bestätigung der ermittelten Vorschläge

Abbildung 6-12 zeigt das Konsolenfenster, welches um eine manuelle Bestätigung der Vorschläge bittet. In diesem Fall wurde eine Ähnlichkeit der Objekte „CAD-Part“ aus dem System „aras-demo“ und dem Objekt „Bauteile“ des Knowledge Graph festgestellt.

```
"GET /api/v1/types/type_graphnodes/3035285 HTTP/1.1" 200 -  
"GET /api/v1/knowledge/dataobjects/DataObjects/3254422 HTTP/1.1" 200 -  
"POST /api/v1/knowledge/type_graphnodes/3035285/equalto/DataObjects/3254422 HTTP/1.1" 200 -
```

Abbildung 6-13: Anlegen der Verbindung zwischen den Objekten

Sollte eine Ähnlichkeit festgestellt werden, wird mithilfe eines „POST“ Requests anschließend im SP²IDER Core auch eine entsprechende Verbindung angelegt, wie in Abbildung 6-13 gezeigt. Als Endergebnis, nachdem sämtliche Vorschläge durchlaufen sind, alle Verbindungen gezogen wurden, ist der vernetzte SP²IDER und Knowledge Graph abgerufen und kann für weitere Engineering-Tasks verwendet werden.

6.2. Verifikation

Die Verifikation des Konzepts findet zweistufig statt. Bezüglich der grundlegenden und methodischen Anforderungen wird das im vorherigen Kapitel vorgestellte Anwendungsszenario betrachtet, für die IT-technischen Anforderungen werden nochmal spezielle Szenarien mit dem Softwaredemonstrator zur Überprüfung der Anforderungen erstellt. Sofern sich nach diesem Test das gewünschte Endergebnis einstellt, ist der Test erfolgreich und die Verifikation in diesem Punkt abgeschlossen.

6.2.1. Verifikation der grundlegenden und methodischen Anforderungen

Aus dem Anwendungsszenario heraus können bereits viele Schlüsse in Bezug auf die Verifikation des Konzepts gezogen werden. Tabelle 6-1 zeigt die zusammengefassten grundlegenden und methodischen Anforderungen aus Kapitel 4.

Tabelle 6-1: Grundlegende und methodische Anforderungen

| Nr. | Anforderung | Gewichtung |
|--------|---|------------|
| AN-K01 | Die Lösung soll eine Modellierung der gewünschten und benötigten Daten ermöglichen | Muss |
| AN-K02 | Es soll eine Zuordnung zwischen der Modellschicht und der Ist-Darstellung der IT-Quellsysteme vorhanden sein | Muss |
| AN-K03 | Die Modelle sollen sich aus den Engineering-Prozessen ableiten | Soll |
| AN-K04 | Die Lösung soll universell einsetzbar sein, nicht nur für eine bestimmte Art von Prozess, Daten oder Unternehmen eingesetzt werden können | Muss |
| AN-K05 | Es müssen verschiedene Nutzer unterstützt werden, u.a. Ingenieure, die die initiale Vernetzung bereitstellen, aber auch z.B. Domänenexperten, die später mit der vernetzten Lösung arbeiten | Muss |
| AN-K06 | Die Lösung soll in der Lage sein, nicht nur bestehendes Wissen abzubilden, sondern auch neues Wissen zu schaffen | Soll |
| AN-K07 | Modelle müssen veränderbar und erweiterbar sein | Muss |
| AN-K08 | Die Lösung muss so gestaltet sein, dass unterschiedliche Nutzer nur einen Teil des Modells sehen dürfen | Soll |
| AN-K09 | Die Lösung sollte idealerweise auch Datenanalyse-Prozesse wie CRISP-DM oder ASUM-DM unterstützen | Soll |
| AN-M01 | Die Lösung muss multilingual einsetzbar sein | Soll |

| | | |
|--------|---|------|
| AN-M02 | Die Methodik muss für unterschiedliche Datentypen funktionieren, z.B. für Daten aus einem PLM-System oder aus einer Zeitreihendatenbank | Soll |
| AN-M03 | Die Methodik soll teilautomatisiert sein und Zuordnungen von Datentypen aus der Modellsicht und der IT-Sicht algorithmisch unterstützen | Muss |

Das Anwendungsszenario basiert auf einem Konstruktionsprozess in einem Unternehmen, welcher zuerst als Knowledge Graph modelliert wird, um anschließend mit IT-Systemen verknüpft zu werden, um einen direkten Datenzugriff zu gewähren. Durch das erfolgreiche Durchlaufen des Szenarios können die Anforderungen AN-K01, AN-K02 und AN-K03 erfüllt werden. Weiterhin zeigt das Anwendungsszenario bereits, dass durch das iterative Durchlaufen auch die Anforderung AN-K07 der Veränderbar- und Erweiterbarkeit erfüllt ist. Ebenfalls zeigt sich, dass das System die Zuordnung von Datentypen aus Modell- und IT-Sicht algorithmisch unterstützt, weswegen AN-M03 als erfüllt gelten kann.

Bezogen auf Anforderung AN-K04, die universelle Einsetzbarkeit, so ist dies in einem Anwendungsszenario naturgemäß nicht zeigbar, mit Verweis auf die Definition des Knowledge Graph in Kapitel 5.1 zeigt sich allerdings, dass die eingeführten Knoten und Kanten so allgemein gewählt sind, dass sie nicht auf bestimmte Prozesse, Daten oder Unternehmen festgelegt sind. In Bezug auf das verwendete Metadaten-Repository SP²IDER kann mit Verweis auf Kapitel 3.2.13 das gleiche gesagt werden. Methodisch gilt dies auch für die Anforderung AN-M02, die Nutzbarkeit mit verschiedenen Datentypen aus unterschiedlichen IT-Systemen.

In Bezug auf AN-K05, die Hilfestellung zum Auffinden von Daten durch Domäneningenieure und AN-K08 die Einschränkung der sichtbaren Elemente zeigt Kapitel 5.5 das Rollen- und Sichtenkonzept, welches eine Einschränkung der sichtbaren Elemente vorsieht und so gerade auch Domäneningenieure in die Lage versetzt schnell in einem begrenzten Bereich die vernetzten Daten zu sehen und durch das Wissen des Vorhandenseins der Daten in ihrer weiteren Arbeit profitieren können. Dies zeigt auch die Erfüllung der Anforderung AN-K06, dass nicht nur Wissen dargestellt, sondern auch erschaffen werden soll. Dies geschieht insbesondere durch das Vernetzen der Daten, so dass nun – wie in Kapitel 2.7 beschrieben – die organisationale Wissensbasis um eben genau diese Vernetzungsinformationen angewachsen ist.

Bezüglich der Anforderung AN-K09, der Anbindung von Datenanalyse-Prozessen ist in dieser Arbeit kein besonderer Fokus gelegt worden. Die entsprechende Anbindung wurde im Rahmen des Projekts AKKORD entwickelt und stellt über die Einbindung von „ISGAT“-Modulen eine Möglichkeit zur Nutzung verteilter Datenquellen durch die Software Rapidminer zur Verfügung. Weitere Möglichkeiten zur Einbindung eines Metadaten-Repository in die Datenanalyse sind in [EiEG2022] beschrieben, weswegen diese Anforderung ebenfalls als erfüllt gelten kann.

Nicht explizit umgesetzt, aber methodisch leicht umsetzbar ist eine multilinguale Unterstützung von Nutzern, wie in AN-M01 gefordert ist. Eine Umsetzung würde lediglich eine Übersetzung der Eingabemasken des Software-Demonstrators erfordern.

6.2.2. Verifikation der IT-technischen Anforderungen

Tabelle 6-2: IT-Anforderungen

| Nr. | Anforderung | Gewichtung |
|---------|--|------------|
| AN-IT01 | Die Lösung muss multilingual einsetzbar sein | Soll |
| AN-IT02 | Die Lösung soll benutzerfreundlich sein | Soll |
| AN-IT03 | Daten sollen nicht dupliziert werden, sondern in ihrem IT-Quellsystem belassen werden | Muss |
| AN-IT04 | Das System soll über eine Schnittstelle verfügen, um ggf. weitere Services zu ermöglichen, bzw. eine bidirektionale Integration in andere Systeme zu ermöglichen | Muss |
| AN-IT05 | Das System sollte server- und webbasiert funktionieren, um unternehmens- und standortübergreifend arbeiten | Muss |
| AN-IT06 | Das System soll aus Open Source Komponenten aufgebaut sein, um keine Lizenzkosten zu erzeugen | Soll |
| AN-IT07 | Das System soll einen Service bereitstellen, der es ermöglicht einen Wissensgraphen anzulegen und abzuspeichern | Muss |
| AN-IT08 | Das System soll einen Service bereitstellen, welcher Datentypen in unterschiedlichen Quellsystemen inhaltlich zu Wissensselementen zuordnen kann | Muss |

Die Verifikation findet statt mit Szenarien, welche sich direkt aus den in Kapitel 4.3 beschriebenen Anforderungen ergeben. Die Einschränkung auf die IT-Technischen Anforderungen wurden getroffen, da sie direkt auch im Software-Code implementiert sein sollten. Sie sind auch in Tabelle 6-2 dargestellt. Weiterhin wird die Verifikation insbesondere auf die Muss-Anforderungen fokussiert, um sicherzustellen, dass diese auch im System umgesetzt wurden.

6.2.2.1. Szenario AN-IT03

Anforderung laut Liste: Daten sollen nicht dupliziert werden, sondern in ihren IT-Quellsystemen belassen werden.

Ausgangssituation im Test: Metadata-Store ist leer, Quellsystem-Konnektor ist angebunden und eingerichtet.

Szenario: Nach dem Verbinden eines Quellsystems, dessen Daten mit dem Knowledge-Graphen verbunden werden sollen, dürfen keine kompletten inhaltlichen Datensätze im System vorhanden sein, sondern nur beschreibende Metadaten

Ist-Situation nach Durchführen des Tests: Nach dem Upload der entsprechenden Daten aus einem Quellsystem mit einem Konnektor, welcher das SP²IDER-Datenschema als Ausgabe erzwingt, ist für jeden Datentyp ein Knoten im Metadata-Store angelegt, welcher beispielhaft in Abbildung 6-14 dargestellt ist.

```
_id: type_graphnodes/31418637
_rev: _e-TW4e----
_key: 31418637

Code ▾
1 {
2   "name": "Properties",
3   "source_system": "carrera",
4   "source_id": "https://akkord-iot.mv.uni-kl.de:8443/api/v1/collection/entity/cdb_prop_desc",
5   "source_uri": "https://akkord-iot.mv.uni-kl.de:8443/api/v1/collection/entity/cdb_prop_desc",
6   "tags": [
7     "blacklist"
8   ],
9   "comment": ""
10 }
```

Abbildung 6-14: Ist-Situation nach dem Upload eines Datentyps aus einem Quellsystem

Wie man in Abbildung 6-14 sieht, werden im Metadata-Store lediglich die Daten abgespeichert und angezeigt, welche im SP²IDER-Datenschema vorgesehen sind. Neben dem Namen ist hier vor allem auch ein Verweis auf das Quellsystem entscheidend, die eigentlichen Daten verbleiben dort.

Ergebnis: Szenario wurde erfolgreich durchlaufen und die Anforderung erfüllt.

6.2.2.2. Szenario AN-IT04

Anforderung laut Liste: Das System soll über eine Schnittstelle verfügen, um ggf. weitere Services zu ermöglichen, bzw. eine bidirektionale Integration in andere Systeme zu ermöglichen

Ausgangssituation im Test: Es ist kein Knowledge-Graph erstellt worden, der Metadata-Store ist leer, der gewünschte Graph ist aber bereits theoretisch erdacht.

Szenario: Es sollen die entsprechenden Schnittstellen getestet werden, indem zuerst ein Aufruf zum Erzeugen jedes einzelnen Objects (DataObject, DataSource, Process, Department) an den Server gesendet werden, um anschließend zu überprüfen, ob der Server bei einem Aufruf aller bestehenden Objekte diese dann auch wieder ausgibt. Ebenso kann überprüft werden, ob die Daten auch im Metadata-Store persistent gespeichert sind und nicht nur aktiv im System oder im Arbeitsspeicher vorgehalten werden.

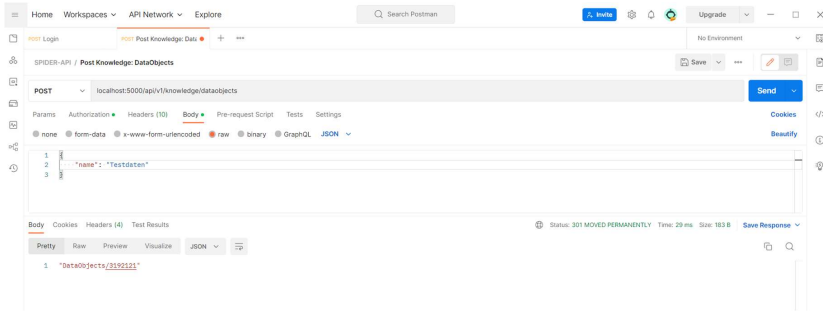


Abbildung 6-15: POST-Aufruf zum Erzeugen eines DataObjects

Abbildung 6-15 zeigt den Aufruf eines entsprechenden Endpunkts mit einem POST-Aufruf zur Erzeugung eines Objekts. Hier werden im Body des Aufrufs Parameter mitgesendet, in diesem Fall der Name „Testdaten“, welcher ein Datenobjekt beschreibt. Als Ergebnis des Aufrufs steht im unteren Feld der Rückgabewert des Servers in Form eines Strings „DataObjects/3192121“, welcher die ID im Metadata-Store ausdrückt.

Ist-Situation nach Durchführung des Tests: Auch wenn bereits nach dem Durchführen des POST-Aufrufs das Ergebnis des Servers in Form einer ID zeigt, dass man Daten via einer Schnittstelle in das System und damit in den SP²IDER Core und den SP²IDER Metadata-Store speichern kann, muss noch gezeigt werden, ob die Daten auch von außen mit einer Schnittstelle wieder abgerufen werden können. Hierzu werden nun nacheinander GET-Aufrufe an das System gesendet, welche sämtliche Objekte der Typen DataObject, DataSource, Process und Department ausgeben. Die Ergebnisse eines dieser Aufrufe der Ressource DataObject zeigt Abbildung 6-16. Hier kann man sehen, dass mehrere Objekte ausgegeben werden, unter anderem das in diesem Szenario erstellte Objekt mit dem Namen Testdaten und der ID „DataObjects/3192121“.

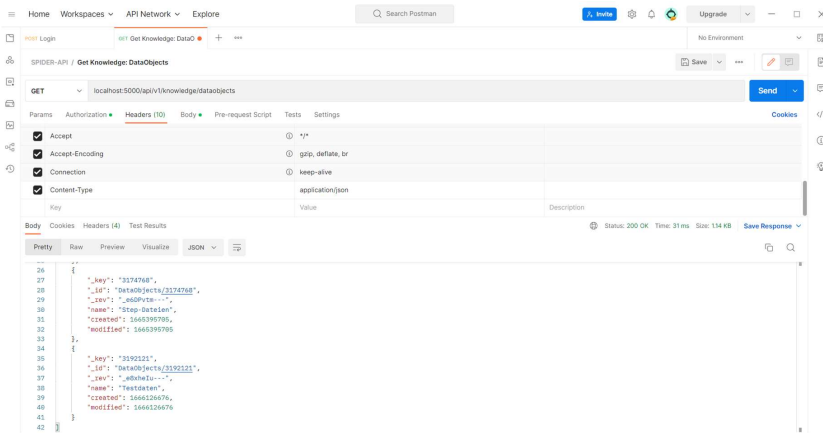


Abbildung 6-16: GET-Aufruf zum Abrufen aller DataObjects

Es ist ebenfalls ersichtlich, dass nicht nur das gerade erzeugte Objekt zurückgegeben wurde, sondern sämtliche bereits angelegten DataObjects, wodurch sichergestellt werden kann, dass hier nicht nur ein Objekt aus dem Arbeitsspeicher ausgegeben wurde, sondern ein Zugriff auf den Metadata-Store erfolgt ist und die Objekte auch dort gespeichert wurden.

Ergebnis: Szenario wurde erfolgreich durchlaufen und die Anforderung erfüllt.

6.2.2.3. Szenario AN-IT05

Anforderung laut Liste: Das System sollte server- und webbasiert funktionieren, um unternehmens- und standortübergreifend arbeiten

Ausgangssituation im Test: Im System sind bereits Daten hinterlegt und sollen nun aus dem Internet abgerufen werden.

Szenario: Es sollen der Metadata-Store und der DMC im Browser über eine öffentliche Adresse aufgerufen werden, um zu zeigen, dass sie webbasiert funktionieren und unternehmens- bzw. standortübergreifend erreichbar wären.

Ist-Situation nach Durchführung des Tests: Da die Programmierung der entsprechenden Services komplett webbasiert erfolgt ist, konnten die entsprechenden Programme auch auf einem Linux-Server öffentlich bereitgestellt werden.

Ergebnis: Szenario wurde erfolgreich durchlaufen und die Anforderung erfüllt.

6.2.2.4. Szenario AN-IT06

Anforderung laut Liste: Das System soll aus Open Source Komponenten aufgebaut sein, um keine Lizenzkosten zu erzeugen

Ausgangssituation im Test: Die Programmierung der einzelnen Services ist abgeschlossen und besteht sowohl aus selbst geschriebenem Code als auch aus externen Python-Paketen, welche von dritter Seite bereitgestellt sind.

Szenario: Es soll die Lizenz für die verschiedenen von extern bereitgestellten Programmen untersucht werden. Hierzu wird einerseits die Lizenz des Metadata-Store in Form einer ArangoDB untersucht, aber auch mit dem Python-Paket pip-licenses untersucht, welche Lizenzen direkt im Code vorliegen.

Ist-Situation nach Durchführung des Tests: Die Lizenz von ArangoDB kann direkt im GitHub-Repository³⁴ abgerufen werden. Es handelt sich um eine Apache License 2.0, welche als Open Source Lizenz gilt.

```
(venv) root@MININT-1THBALK:/mnt/c/Users/elden/repos/spiderserv-modellierung# pip-licenses
Name          Version      License
Flask         2.0.3        BSD License
Flask-SQLAlchemy 2.5.1        BSD License
dinjaja       3.0.3        BSD License
Levenshtein   0.18.1       GNU General Public License v2 or later (GPLv2+)
MarkupSafe    2.1.1        BSD License
SQLAlchemy    1.4.22       MIT License
Werkzeug      2.0.3        BSD License
certifi       2021.10.8    Mozilla Public License 2.0 (MPL 2.0)
charset-normalizer 2.0.12       MIT License
click         8.0.4        BSD License
greenlet      1.1.2        MIT License
idna          3.3          BSD License
itsdangerous  2.1.1        BSD License
jarowinkler   1.0.2        MIT License
joblib        1.1.0        BSD License
nlTK          3.7          Apache Software License
pkg-resources 0.0.0        UNKNOWN
rapidfuzz     2.0.7        MIT License
regex         2022.3.15    Apache Software License
requests      2.27.1       Apache Software License
tqdm          4.63.0       MIT License; Mozilla Public License 2.0 (MPL 2.0)
urllib3       1.26.9       MIT License
```

Abbildung 6-17: Lizenzen der Python-Pakete im Projekt

Das Überprüfen der Pakete in den beiden maßgeblich erarbeiteten Programme „spiderserv-modellierung“ (Abbildung 6-17), sowie „spider_django“ (Abbildung 6-18) ergibt, dass verschiedene Lizenzformen in den eingebundenen Paketen vorhanden sind. Hierzu zählen die BSD-Lizenz, die GNU General Public License v2, die MIT Lizenz, die Mozilla Public License 2.0 und die Apache Software License. Bei all diesen Lizenzen handelt es sich um Open Source Lizenzen, welche eine Nutzung ermöglichen. Kritisch zu betrachten ist hierbei die Nutzung des Pakets Levenshtein unter der GNU General Public License v2 steht. Diese Lizenz erzwingt den Nutzer die darauf aufbauenden Codes auch unter eine GNU-Lizenz zu stellen, sprich als Open Source zur Verfügung zu stellen.

³⁴ ArangoDB-Lizenz: <https://github.com/arangodb/arangodb/blob/devel/LICENSE>

```
(venv) root@MININT-1TH3ALK:/mnt/c/Users/eiden/repos/spider_django# pip-licenses
Name          Version      License
Deprecated   1.2.13      MIT License
Flask         2.0.2       BSD License
Flask-RESTful 0.3.9       BSD License
Flask-SQLAlchemy 2.5.1      BSD License
Jinja2        3.0.3       BSD License
MarkupSafe    2.0.1       BSD License
SQLAlchemy    1.4.29      MIT License
Werkzeug      2.0.2       BSD License
amqp          5.0.9       BSD License
arango601    0.0.1       BSD License
billiard      3.6.4.0     BSD License
celery        5.2.3       BSD License
certifi       2021.10.8   Mozilla Public License 2.0 (MPL 2.0)
charset-normalizer 2.0.9      MIT License
click         8.0.3       BSD License
click-didyoumean 0.3.0      MIT License
click-plugins 1.1.1       BSD License
click-repl    0.2.0       MIT
colorama      0.4.4       BSD License
greenlet      1.1.2       MIT License
idna          3.3         BSD License
itsdangerous  2.0.1       BSD License
kombu         5.2.3       BSD License
packaging     21.3        Apache Software License; BSD License
pkg-resources 0.0.0       UNKNOWN
prompt-toolkit 3.0.24      BSD License
pyparsing     3.0.6       MIT License
pytz          2021.3      MIT License
redis         4.1.0       MIT License
requests      2.27.0      Apache Software License
six           1.16.0      MIT License
urllib3       1.26.7      MIT License
vine          5.0.0       BSD License
wcwidth      0.2.5       MIT License
wrapt         1.13.3      BSD License
(venv) root@MININT-1TH3ALK:/mnt/c/Users/eiden/repos/spider_django#
```

Abbildung 6-18: Lizenzen der eingebundenen Python-Pakete im Projekt „spider_django“

Ergebnis: Szenario wurde erfolgreich durchlaufen und die Anforderung erfüllt.

6.2.2.5. Szenario AN-IT07

Anforderung laut Liste: Das System soll einen Service bereitstellen, der es ermöglicht einen Wissensgraphen anzulegen und abzuspeichern

Ausgangssituation im Test: Es wurde ein einfaches Beispiel gewählt, welches grob und mit einigen wenigen Objekten das grobe Szenario der Produktentwicklung beschreibt. Aus Anforderungen werden Produktdaten erstellt, diese in einem PLM-System gespeichert und von der Arbeitsvorbereitung in einen Arbeitsplan umgewandelt, um in einem ERP-System gespeichert zu werden.

Szenario: Das oben beschriebene Szenario liegt entsprechend der Prozessbeschreibung in Kapitel 5.3.1 in Form von Listen aller Objekte vor. Nun sollen während des Tests im dafür vorgesehenen Softwaretool in mehreren Schritten zuerst die DataObjects, anschließend DataSources, Processes und Departments modelliert werden, indem für jedes Objekt ein Knoten angelegt wird und der entsprechende Name eingetragen wird. Anschließend findet eine Vernetzung der Knoten statt, so wie es sich aus den zugrunde liegenden Prozessbeschreibungen und den vorhandenen Objekten ergibt.

Ist-Situation nach Durchführung des Tests: Wie in Abbildung 6-19 zu sehen ist, wurde in der Weboberfläche des Data Model Canvas ein Knowledge Graph erstellt, welcher vier verschiedene Arten von Knoten anzeigt. Durch die Speicherung in den unterschiedlichen Objektformen wird durch die Programmierung des DMC jeweils eine unterschiedliche Form und Farbe zur einfacheren Identifizierung und Klassifizierung durch den Nutzer gewählt. Neben den Knoten aus

den verschiedenen Objekten zeigt sich, dass diese durch Kanten verbunden werden, welche alle mit einer Beschriftung versehen sind, welche die Bedeutung der Vernetzung ausdrückt. So besteht zwischen Arbeitsplan und Produktdaten die Beziehung „made_from“, was sich als Tripel lesen lässt: Arbeitsplan made from Produktdaten.

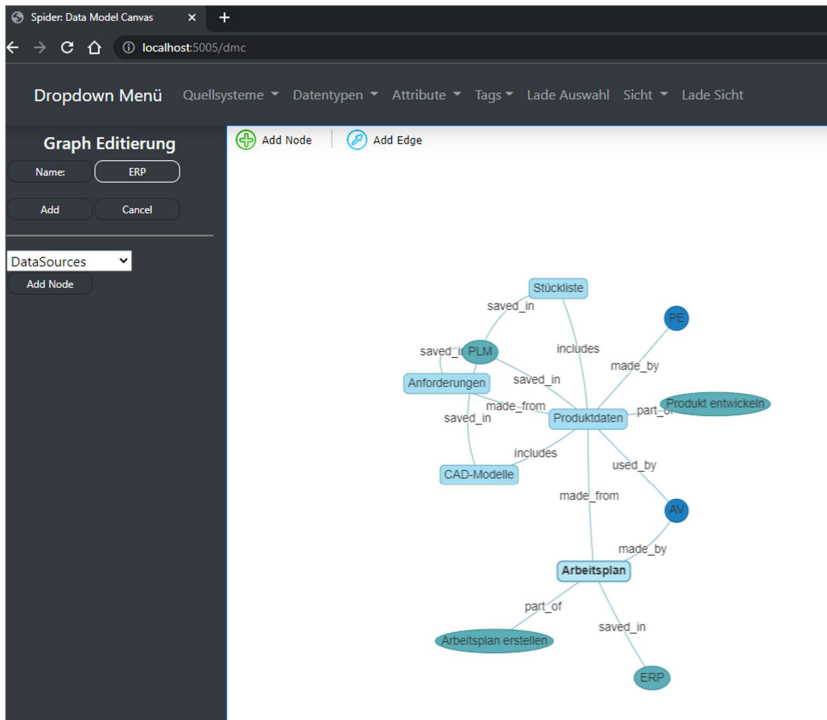


Abbildung 6-19: Knowledge Graph im Data Model Canvas als Service

Weiterhin wird auf der linken Seite die Oberfläche mit den Steuerelementen zum Hinzufügen weiterer Knoten angezeigt. Im Dropdown-Menü lässt sich einstellen, von welchem Typ ein weiteres Objekt hinzugefügt werden soll. Obendrüber sieht man noch das zuletzt angelegt Objekt, welches der „ERP“-Knoten ist. Durch ein Drücken auf „Add“ würde nun ein weiterer Knoten angelegt.

Ergebnis: Szenario wurde erfolgreich durchlaufen und die Anforderung erfüllt.

6.2.2.6. Szenario AN-IT08

Anforderung laut Liste: Das System soll einen Service bereitstellen, welcher Datentypen in unterschiedlichen Quellsystemen inhaltlich zu Wissens-elementen zuordnen kann

Ausgangssituation im Test: Direkt aufbauend auf dem vorhergehenden Szenario soll nun überprüft werden, wie eine Zuordnung zwischen den vorhandenen Type_Graphnodes aus einem Quellsystem und den DataObjects aus dem Knowledge Graph programmatisch unterstützt werden kann. Hierzu ist neben dem modellierten Knowledge Graph noch eine Verbindung zu mindestens einem Quellsystem notwendig, welches durch den entsprechenden Konnektor sein Datenmodell ebenfalls in den Metadata-Store überführen muss.

Szenario: Entsprechend dem modellierten Knowledge-Graph, welcher vor allem einen Ausschnitt aus der Produktentwicklung zeigt und als modellierte Datenquellen ein PLM und ein ERP vorsieht, sollen die enthaltenen DataObjects in einem verbundenen Aras Innovator PLM gesucht werden. Hierzu entsprechend Kapitel 5.4.2 verschiedene Strategien angewendet, welche nacheinander durchlaufen werden, um Übereinstimmungen zu finden.

```
{'type_graphnodes/3035265': 'DataObjects/3202136', 'type_graphnodes/3035275': 'DataObjects/3202136', 'type_graphnodes/3035295': 'DataObjects/3202136', 'type_graphnodes/3035355': 'DataObjects/3202279', 'type_graphnodes/3035917': 'DataObjects/3202279', 'type_graphnodes/3037821': 'DataObjects/3202144', 'type_graphnodes/3037841': 'DataObjects/3202144', 'type_graphnodes/3038651': 'DataObjects/3202116', 'type_graphnodes/3038671': 'DataObjects/3202116', 'type_graphnodes/3038701': 'DataObjects/3202116', 'type_graphnodes/3038711': 'DataObjects/3202116', 'type_graphnodes/3038721': 'DataObjects/3202116', 'type_graphnodes/3039545': 'DataObjects/3202279'}
```

Abbildung 6-20: Paare aus DataObject und Type_Graphnode als JSON-Dictionary

Sollte der Algorithmus hier ein Paar aus DataObject und Type_Graphnode gefunden haben, welches ein potenzieller Match ist, so wird dies dem Nutzer zur Bestätigung vorgeschlagen. Abbildung 6-20 zeigt hierbei sämtliche Vorschläge, welche der Algorithmus anschließend nacheinander abfragt und nach einer Bestätigung durch den Nutzer eine entsprechende Verbindung zieht, wie Abbildung 6-21 zeigt. Hier wurde der erste Vorschlag vom Nutzer bestätigt, woraufhin eine „EqualTo“ Kante erstellt wurde. Der zweite Vorschlag wurde noch nicht beantwortet und kann noch bestätigt oder abgelehnt werden.

```
{'key': '3035295', 'id': 'type_graphnodes/3035295', '_rev': 'e2PyfFee---', 'name': 'CADFiles', 'source_system': 'aras-demo', 'source_id': '6093CB60290141C1998C0833FEAA1A8C0', 'source_url': 'http://aras.mv.uni-kl.de/innovatorserver/server/odata/itetype('6093CB60290141C1998C0833FEAA1A8C0')', 'tags': [], 'comment': 'CAD-Dateien'}
{'key': '3202136', 'id': 'DataObjects/3202136', '_rev': 'e9GfFZG---', 'name': 'CAD-Modelle', 'created': 1666215413, 'modified': 1666215413}
Sollen diese Objekte verknüpft werden? --> y
"EqualTo/3204859"
{'key': '3035355', 'id': 'type_graphnodes/3035355', '_rev': 'e2PyfZi---', 'name': 'CE_Document', 'source_system': 'aras-demo', 'source_id': '8CB6CD331DF044808B59C931A84110F4', 'source_url': 'http://aras.mv.uni-kl.de/innovatorserver/server/odata/itetype('8CB6CD331DF044808B59C931A84110F4')', 'tags': [], 'acklist': {}, 'comment': ''}
{'key': '3202279', 'id': 'DataObjects/3202279', '_rev': 'e9GuqxI---', 'name': 'Anforderungen', 'created': 1666215621, 'modified': 1666215621}
Sollen diese Objekte verknüpft werden? --> y
```

Abbildung 6-21: Abfrage zur Bestätigung des Datenmappings

Ist-Situation nach Durchführung des Tests: Nach Bestätigen dieses einen Vorschlags durch den Nutzer, wie im obigen Szenario beschrieben, wird beim erneuten Laden des Knowledge Graphs nun auch eine weitere Kante und der damit verbundene Type_Graphnode Knoten angezeigt.

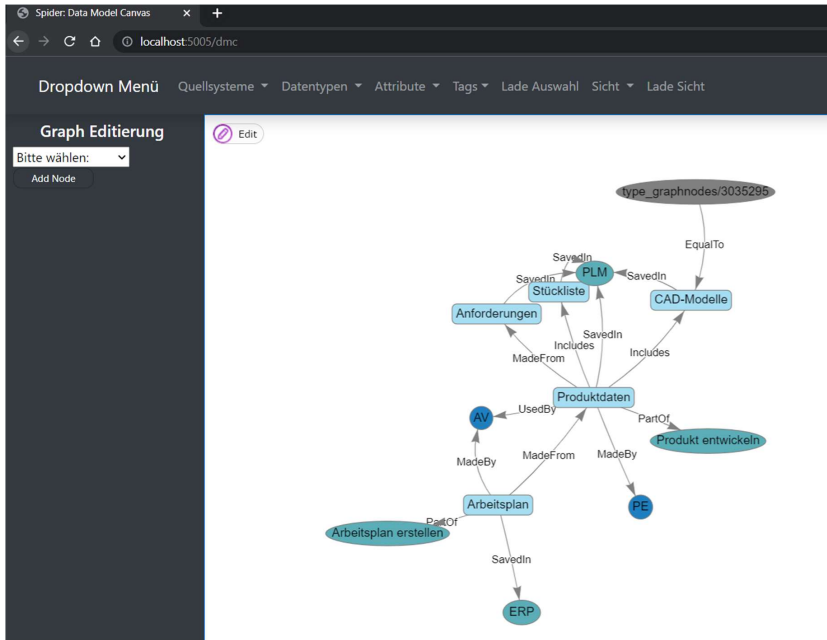


Abbildung 6-22: Knowledge Graph mit Verbindung zu Datentypen aus einem IT-System

Ergebnis: Szenario wurde erfolgreich durchlaufen und die Anforderung erfüllt.

6.2.3. Ergebnis der Verifikation

Im Rahmen der Verifikation wurde überprüft, inwieweit die Erfüllung der Anforderungen auch durch nachprüfbare Szenarien sichergestellt werden kann. Sofern dies nicht möglich war, wurde auf die Konzeptbeschreibung zurückgegriffen, um eine Verifikation durchzuführen.

Sowohl das Anwendungsszenario in Kapitel 6.1, als auch die IT-Szenarien in Kapitel 6.2.2 zeigen, dass alle Anforderungen an das Konzept für ein Knowledge-Graph basiertes Assistenzsystem zur Unterstützung der Datenanalyse für das Engineering Smarter Produkte vollumfänglich erfüllt sind und die Verifikation erfolgreich abgeschlossen ist.

Die folgende Tabelle zeigt nochmal übersichtlich den Grad der Erfüllung der Anforderungen.

Tabelle 6-3: Grad der Erfüllung der Anforderungen

| Nr. | Anforderung | Gewichtung | Erfüllungsgrad |
|---------|--|------------|----------------|
| AN-IT01 | Die Lösung muss multilingual einsetzbar sein | Soll | ● |
| AN-IT02 | Die Lösung soll benutzerfreundlich sein | Soll | ● |
| AN-IT03 | Daten sollen nicht dupliziert werden, sondern in ihrem IT-Quellsystem belassen werden | Muss | ● |
| AN-IT04 | Das System soll über eine Schnittstelle verfügen, um ggf. weitere Services zu ermöglichen, bzw. eine bidirektionale Integration in andere Systeme zu ermöglichen | Muss | ● |
| AN-IT05 | Das System sollte server- und webbasiert funktionieren, um unternehmens- und standortübergreifend arbeiten | Muss | ● |
| AN-IT06 | Das System soll aus Open Source Komponenten aufgebaut sein, um keine Lizenzkosten zu erzeugen | Soll | ● |
| AN-IT07 | Das System soll einen Service bereitstellen, der es ermöglicht einen Wissensgraphen anzulegen und abzuspeichern | Muss | ● |
| AN-IT08 | Das System soll einen Service bereitstellen, welcher Datentypen in unterschiedlichen Quellsystemen inhaltlich zu Wissens-elementen zuordnen kann | Muss | ● |

| Legende | |
|------------------------|-------|
| nicht erfüllt | ○ |
| teilweise erfüllt | ◐ ◑ ◒ |
| vollumfänglich erfüllt | ● |

6.3. Validierung

Im Gegensatz zur Verifikation im Sinne der Erfüllung von Anforderungen soll in der Validierung festgestellt werden, ob das gesamte Konzept tragfähig ist und so gebaut wird, dass auch Probleme der möglichen Nutzer gelöst werden. Hierzu wurde die Validierung insbesondere auf

einer konzeptuellen Ebene gewählt. Um sicher zu stellen, dass auch eine breite Anwendbarkeit des Konzepts im Einsatz gewährleistet wird, wurde das Konzept verschiedenen Forschern präsentiert, welche durch ihre Mitarbeit mit verschiedensten Industriepartnern in verschiedenen Projekten eine diverse Sicht auf das Konzept liefern können.

Methodisch wurde die Validierung in Form von Interviews geführt, welche daraus bestanden, zuerst das Konzept vorzustellen und anschließend die verschiedenen Anforderungen vorzustellen, welche das Konzept erfüllen soll. Anschließend werden die gegebenen Antworten niedergeschrieben und daraus entsprechende Schlüsse gezogen.

Tabelle 6-4: Konzeptuelle Anforderungen

| Nr. | Anforderung | Gewichtung |
|--------|---|------------|
| AN-K01 | Die Lösung soll eine Modellierung der gewünschten und benötigten Daten ermöglichen | Muss |
| AN-K02 | Es soll eine Zuordnung zwischen der Modellschicht und der Ist-Darstellung der IT-Quellsysteme vorhanden sein | Muss |
| AN-K03 | Die Modelle sollen sich aus den Engineering-Prozessen ableiten | Soll |
| AN-K04 | Die Lösung soll universell einsetzbar sein, nicht nur für eine bestimmte Art von Prozess, Daten oder Unternehmen eingesetzt werden können | Muss |
| AN-K05 | Es muss verschiedene Nutzer unterstützen, u.a. Ingenieure, die die initiale Vernetzung bereitstellen, aber auch z.B. Domänenexperten, die später mit der vernetzten Lösung arbeiten | Muss |
| AN-K06 | Das Konzept soll in der Lage sein, nicht nur bestehendes Wissen abzubilden, sondern auch neues Wissen zu schaffen | Soll |
| AN-K07 | Modelle müssen veränderbar und erweiterbar sein | Muss |
| AN-K08 | Das Konzept muss so gestaltet sein, dass unterschiedliche Nutzer nur einen Teil des Modells sehen dürfen | Soll |
| AN-K09 | Die Lösung sollte idealerweise auch Datenanalyse-Prozesse wie CRISP-DM oder ASUM-DM unterstützen | Soll |

In der Validierung mittels Interviews wurden insbesondere die grundlegenden konzeptuellen Anforderungen an eine Lösung in den Fokus gesetzt, so wie sie in Tabelle 6-4 dargestellt werden. Die Auswahl dieser Anforderungen liegt darin begründet, dass sich die weiteren Anforderungen hieraus ableiten und somit bei einer Validierung nachrangig sind. Sollte bereits ein konzeptueller

Fehler auf Basis der grundlegenden Anforderungen vorliegen, so müssen in der Folge auch die IT-technischen und methodischen Anforderungen überarbeitet werden.

Es wurden insgesamt neun Interviews zur Validierung des Konzepts durchgeführt. Im Folgenden wird zuerst der Ablauf des Interviews erläutert, anschließend werden die Antworten der Interviewpartner aggregiert dargestellt, und anschließend ein Fazit aus diesen Interviews gezogen.

Neben den Interviews wurde auch eine quantitative Validierung in Form von A/B-Tests [StWa2018] gewählt, die einen Vergleich zwischen einer bisher existierenden Lösung und der neu entwickelten Lösung ermöglichen. An dieser Stelle bedeutet dies vor allem das bisher existierende SP²IDER-Konzept mit dem durch diese Arbeit neu entwickelten Konzept zu vergleichen (siehe Kapitel 6.3.3).

6.3.1. Aufbau der Validierungsinterviews

Zuerst wird dem Interviewpartner der Hintergrund des Interviews zur Validierung des hier vorgestellten Konzepts erläutert, anschließend in mehreren Fragen konzeptuell nachgefragt, ob dieses Konzept eine sinnvolle Ergänzung ist, um aktuellen Problemen zu begegnen, welche bei Industriepartnern wahrgenommen wurden.

Frage I: Hintergrund der Forschung/Zusammenarbeit mit Industriepartnern

Diese Frage zielt darauf ab, den forschersichen Hintergrund zu verstehen, um die weiteren Antworten in diesem Kontext interpretieren zu können. Insbesondere die Verbindungen zu Industriepartnern und deren Anforderungen an Daten stehen bei dieser Frage im Vordergrund.

Frage II: Was ist über die Daten und die Systemlandschaft bei industriellen Anwendungspartnern bekannt, wo liegen hier aktuelle Problemstellungen?

Als Erweiterung von Frage I soll nochmal gezielt auf die Probleme und Herausforderungen der Industriepartner eingegangen werden. Wie sehen System- und Datenstrukturen aus, wie spielt das in der Forschung eine Rolle?

Frage III: Haben die Nutzer ausreichend Wissen zur Datenverfügbarkeit und zur Bedeutung der existierenden Daten?

Kern des hier vorgestellten Konzepts ist eine methodische Hilfestellung, die es Nutzern ermöglicht die Bedeutung von Daten besser zu verstehen und benötigte Daten schneller aufzufinden. Hier soll nun geklärt werden, ob dieses Problem auch in den Firmen besteht, mit denen die Interviewpartner zusammenarbeiten.

Frage IV: Ist es sinnvoll eine methodische Hilfestellung zur Verfügung zu stellen, um Daten besser zu beschreiben?

Auch unter der Annahme, dass eine zusätzliche Hilfestellung dem Nutzer einen Vorteil bringt, ist mit ihr ebenfalls eine Steigerung der Komplexität in der bestehenden IT-Infrastruktur verbunden,

weswegen diese Frage darauf abzielt, ob ein eventueller Vorteil als ausreichend groß gesehen wird.

Frage V: Wie könnte eine sinnvolle Hilfestellung aussehen?

Sollte eine Hilfestellung als aussichtsreich angesehen werden, so zielt diese Frage darauf ab, in kurzer Zeit evtl. vorhandene Lösungsideen des Interviewpartners abzufragen, bevor eine Vorstellung des Konzepts bereits den möglichen Lösungsraum einschränkt.

Anschließend wird an dieser Stelle im Interview das Konzept im Detail vorgestellt und der Demonstrator anhand eines Beispiels gezeigt.

Frage VI: Welche Nutzer könnten eine solche Hilfestellung nutzen?

Ausgehend von der nun erfolgten Konzeptvorstellung soll reflektiert werden, welche Nutzer im Adressatenkreis dieser Hilfestellung sind und ob in diesem Konzept Nutzer nicht bedacht wurden.

Frage VII: Ist das hier vorgestellte Konzept eine sinnvolle Hilfestellung?

Letztlich ist diese Frage der Kern des Validierungsinterviews, um herauszufinden, ob das hier vorgestellte Konzept einen Beitrag zur Lösung der Probleme beisteuern kann, welche in den Unternehmen erkannt werden.

Gibt es weiteres Feedback?

Auch wenn weiteres Feedback im Rahmen dieser Arbeit nicht mehr umgesetzt werden kann, so sollen trotzdem auch mögliche Verbesserungen für die Zukunft aufgenommen werden, um eine Basis für Weiterentwicklungen zu bilden.

6.3.2. Ergebnisse der geführten Interviews

Die Interviews zeigen die enorme Bandbreite, die durch die verschiedenen Perspektiven der unterschiedlichen Forscher abgedeckt wird. So sind z.B. die Sichtweisen aus der Automobil-, der Halbleiter-, der Haushaltswarenbranche und von mittelständischen Maschinen- und Werkzeugbauern abgedeckt. Die Zusammenarbeit erstreckt sich mit den Forschenden vor allem durch die Zusammenarbeit in Drittmittelprojekten, aber auch durch bilaterale Kooperationen und Normungsgremien, in denen Probleme der Firmen besprochen und gelöst werden sollen. In diesen Projekten und Gremien stehen auch Herausforderungen in Bezug auf Datenstrukturen im Fokus. Eine weitere große Problemstellung, insbesondere hinsichtlich der Daten ist die Kooperation von verschiedenen Unternehmen in klassischen Wertschöpfungsketten oder komplexeren Wertschöpfungsnetzwerken. In einer anderen Perspektive macht sich auch bemerkbar, dass unterschiedliche fachliche Hintergründe zu anderen Perspektiven führen, was durch Forschende mit fachlichem Hintergrund im Maschinenbau, im Wirtschaftsingenieurwesen, sowie in der Informatik deutlich wurde.

Hinsichtlich der Datenstrukturen zeigt sich eine große Heterogenität, welche zu Problemen in der täglichen Arbeit führt. Beispielsweise wird in einem Projekt in einer Wertschöpfungskette der Versuch einer gemeinsamen Produkt-Weiterentwicklung unternommen, was sich stark auf die

internen Daten auswirkt, z.B. in Form von Produktdokumentation oder Anforderungen in Spezialexsystemen, während auf der Kollaborationsebene allerdings nur in Form einer gemeinsamen Datenaustauschplattform und durch die Nutzung klassischer Office-Formate zusammengearbeitet wird. Erschwerend kommt noch hinzu, dass die Datenstrukturen dieser Plattform nicht jedem Nutzer ausreichend bekannt sind und in größeren Unternehmen nicht jede an der Produkt-Weiterentwicklung beteiligte Person auch Zugriff auf die Austauschplattform hat. Ein weiteres Industriebeispiel war die in einem Unternehmen stattfindende parallele Entwicklung der gleichen Systemmodelle, da den Ingenieuren nicht bekannt war, dass in einer anderen Abteilung bereits an solchen Modellen gearbeitet wurde und die entsprechenden Daten auch unbekannt blieben. Auch wurde gezeigt, dass häufig in Firmen ein Vendor-Lock-In Effekt zu beobachten ist. Nach einer Kaufentscheidung für ein spezifisches System ist man stark an diesen Toolanbieter gebunden und nutzt firmenseitig Forschungsprojekte aus Anwendersicht, um neue Konzepte zu erarbeiten und diese dann dem eigenen Tool-Lieferanten zu präsentieren und idealerweise umsetzen zu lassen.

Diese Beispiele zeigen, dass es auf Seiten der (Daten)-Nutzer kein ausreichendes Wissen zur Datenverfügbarkeit gibt, um die eigenen Aufgaben innerhalb der Engineering-Prozesse effizient und effektiv durchzuführen. Nutzern fällt es schwer die Datenbestände im eigenen Unternehmen, aber auch fremde Daten, welche theoretisch verfügbar wären, im Überblick zu halten und zur Erfüllung der eigenen Aufgaben zielgerichtet einzusetzen. Neben dem fehlenden Wissen kann auch mangelnder Tool-Support ein Problem sein, wenn bisherige Partner entsprechende Lösungen zum Beheben der Probleme nicht im Angebot haben. Ein weiteres Problem ist häufig eine zu große oder zu geringe Datenmenge, bei der meistens auch der Kontext, welcher zur Einordnung der Bedeutung der Daten notwendig ist, nicht bekannt ist.

Insofern beantworten alle Interviewten die Frage nach der Notwendigkeit einer Hilfestellung im Bereich der Datenverfügbarkeit im Unternehmen mit einer klaren Feststellung der Notwendigkeit. Diese Notwendigkeit wird insbesondere unterstrichen durch die Menge an zueinander inkompatiblen Insellösungen, welche momentan in den Firmen genutzt werden, um die Komplexität der Daten gerecht zu werden. Ebenfalls zeigt sich die Notwendigkeit auch an der Menge der Standardisierungsaktivitäten, beispielsweise zum Datenaustausch, um Kollaboration zu ermöglichen. Ein weiterer Grund für eine solche Hilfestellung liegt auch in der Beobachtung der Fluktuation von Mitarbeitenden, welche in solchen Einsatzgebieten wie oben beschrieben dazu führt, dass neue Mitarbeitende sehr lange Einarbeitungszeiten benötigen und sich trotzdem nicht aller vorhandenen Daten bewusst sind.

Auf die Frage nach möglichen Lösungsansätzen und Hilfestellungen, bevor im Interview das hier beschriebene Konzept im Detail präsentiert wurde, wurden unter anderem Beschreibungen der Dateisysteme und Austauschstrukturen genannt, sowie die Rolle von Thesauri und Synonymen betont, da häufig auch das Problem besteht, dass Daten in unterschiedlichem Kontext zwar den gleichen Inhalt haben, allerdings unterschiedlich beschrieben wurden und deswegen nicht von allen Mitarbeitenden direkt gefunden werden, bzw. die Bedeutung nicht erkannt wird. Ein

gewünschter Lösungsansatz war eine Visualisierung von Strukturen, bspw. durch einen Baum oder eine Graph-Struktur.

An dieser Stelle im Interview wurde den Interviewten das hier vorgestellte Konzept präsentiert, insbesondere die bereits bestehende Vorarbeit in Form des semantischen Engineering-Repository SP²IDER, die Elemente des Knowledge Graphen, die dahinter stehende Idee und die Methodik, wie der Knowledge Graph aufgebaut und anschließend mit den bestehenden Metadaten des SP²IDER Repository verknüpft wird und wie Nutzer anschließend diese Informationen nutzen können, um ihre tägliche Arbeit in Engineering-Prozessen zu verbessern.

Die Frage nach dem Nutzerkreis, welcher mit einer solchen Lösung arbeiten könnte, wird sehr divers gesehen. Zum einen wird der Vorteil des Systems zwar für jeden Nutzer gesehen, da hier ein bisher nicht vorhandener Überblick über Datenbestände und ihre Speicherorte geschaffen wird, welcher Nutzer inspirieren könnte, dass beispielsweise bisher nicht genutzte Datenbestände auch in der täglichen Arbeit verwendet werden, allerdings wird auch kritisch angemerkt, dass hier wieder ein neues System entsteht, welches auch einen weiteren Nutzungs-Overhead mitbringt, bevor man produktiv an Prozessen arbeiten kann. Insofern wird es sicherlich auch Nutzer geben, für die eine solche Lösung zu viel ist, da nur ein eingeschränktes Subset an Daten benötigt wird, um den täglichen Aufgaben nachzukommen. Allerdings wird auch gesehen, dass z.B. Domäneningenieure oder gerade auch interdisziplinär arbeitende Ingenieure zu den Hauptnutzern des Systems zählen werden. Besonders hervorgehoben wurden hier solche Ingenieure, welche im Engineering bereits heute Schnittstellenpositionen besitzen und z.B. auf Seiten der Produkt-, aber auch der Produktionsentwicklung agieren. Ein Benefit, von dem jeder potenzielle Nutzer in einem Unternehmen profitieren könnte, ist eine Enterprise Search, sprich eine Suche auf dem kompletten verbundenen Datenbestand des Unternehmens über einen Service. Durch dieses Konzept könnten nicht nur (Meta)-Daten, sondern auch der Knowledge Graph Ausgangspunkt einer Suche sein, was zu potenziell noch besseren Ergebnissen führen kann.

Weitere wichtige Nutzer des Systems sind nach Ansicht der Interviewten insbesondere Dateningenieure oder „Datagurus“, welche sich insbesondere mit den Beständen der Daten im Unternehmen, aber auch über Unternehmensgrenzen hinaus befassen. Über die Unternehmensgrenzen hinaus dient diese Aufgabe dazu Kollaboration in konkreten Projekten zu ermöglichen, aber auch andere Datenbestände z.B. in einem öffentlichen Datenraum zu erfassen und sie dem eigenen Unternehmen und dem eigenen Datenbestand hinzuzufügen und über diese Lösung zur Verfügung zu stellen.

Eine weitere relevante Rolle könnte ein Datenhändler sein, welcher angedockt an den Knowledge Graph und SP²IDER als Metadaten-Repository den Zugang zu weiteren Daten gegen eine Monetarisierung ermöglicht. Eine weitere betroffene Rolle sind Prozessmanager, welche den Knowledge Graph und das Prozessmodell synchron halten und bidirektionale Abgleiche vornehmen.

Nicht zuletzt wird als Rolle auch noch ein externer Nutzer gesehen. Ein weiteres Lösungsszenario könnte das unternehmensübergreifende Teilen von Daten sein. Aufgrund der Sichten im

Knowledge Graph ist es möglich eine gezielte Informationsbereitstellung für weitere Firmen in einem Wertschöpfungsnetzwerk durchzuführen und über ein solches Tool ggf. auch den Datenzugriff zu organisieren.

Ansonsten wird das hier vorgestellte Konzept als eine sinnvolle Hilfestellung für die geschilderten Herausforderungen gesehen, vor denen Unternehmen stehen. Um für den Nutzer durchweg eine Verbesserung zu erzielen, sollte laut Ansicht der Interviewten vor allem weiter in die Einfachheit der Bedienung und eine Hilfestellung bei der Einführung investiert werden, um die Einstiegshürde so gering wie möglich zu halten. Ein starker Vorteil, welcher in diesem Konzept gesehen wird, ist die Ganzheitlichkeit des Ansatzes in der Kombination mit SP²IDER als Metadaten-Repository, welches ebenfalls einen ganzheitlichen Anspruch hat. Bisherige Hilfestellungen und Teillösungen waren meistens eher Insellösungen, die keine Optimierung eines Gesamtsystems im Fokus hatten. Ebenfalls soll das Rollenkonzept so gestaltet sein, dass interdisziplinär arbeitende oder in einer Domäne arbeitende Ingenieure nicht belastet werden, indem Rollen wie der Dateningenieur die Hauptlast der Einführung und Datenverknüpfung übernehmen. Die Konzeptlösung ist aus Sicht der Graph-Forschung in der Informatik sicherlich keine vollständige, aber eine pragmatische Einschränkung, um den Kern der Graph-Technologie zu nutzen.

Kritisch angemerkt werden die unterschiedlichen Strategien zum Auffinden ähnlicher Daten. Die Nutzung von Thesauri oder Glossaren ist plausibel, allerdings müssten hier Informationen auch über Unternehmensgrenzen hinweg geteilt werden, um den Mehrwert zu verstärken und von solch aufgebauten Instrumenten zu profitieren. Diese Bereitschaft zum Datenaustausch wird als eher schwierig zu realisieren und dementsprechend kritisch gesehen.

6.3.3. Quantitative Validierung durch Tests

Neben der qualitativen Validierung durch die geführten Interviews mit Forschenden soll auch eine quantitative Validierung stattfinden, welche den Geschwindigkeitsvorteil durch die Nutzung einer Hilfestellung im Bereich des Auffindens ähnlicher Daten aufzeigt.

6.3.4. Ergebnis der Validierung

Mit Blick auf die verschiedenen durchgeführten Aktivitäten im Bereich der Validierung kann diese als erfolgreich abgeschlossen gelten. Die Interviews mit verschiedenen Forschenden haben aufgezeigt, dass hier auf Seiten der bekannten Industriepartner die Notwendigkeit besteht, eine Hilfestellung beim Handling der Daten zu erhalten. Die Nachfragen nach bekannten Problemen und Bedarfen haben ergeben, dass die entsprechenden Bedarfe genau dort gesehen werden, an denen die konzeptuellen Anforderungen ansetzen, z.B. AN-K01 und AN-K02, der Bedarf zur Modellierung von Daten und die Verknüpfung mit der Ist-Sicht der genutzten Datenquellsysteme. Auch wird z.B. die Unterstützung verschiedener Nutzer wie in AN-K05 direkt gefordert, eine universelle Einsetzbarkeit mit einem Fokus auf Engineering-Prozessen wie in AN-K04 und AN-K03 beschrieben ist allerdings auch entsprechend der Ansicht der Interviewten sinnvoll.

Über die bestehenden konzeptuellen Anforderungen hinaus konnte ein Test mit potenziellen Nutzern zeigen, dass im Vergleich zu einem manuellen Suchen von Datentypen eine enorme Zeitersparnis durch eine automatisierte Ermittlung von Ähnlichkeiten und ein Finden der im Knowledge Graph abgebildeten Datentypen in den realen IT-Systemen möglich ist und Nutzer hiervon einen Vorteil durch schnellere und genauere Datenaufbereitung haben.

Insgesamt konnte in diesem Kapitel gezeigt werden, dass die Verifikation und der Validierung des Konzepts eines Knowledge Graph basiertes Assistenzsystem zur Datenanalyse-Unterstützung für das Engineering Smarter Produkte erfolgreich durchgeführt werden konnte. Die Anforderungen an das Konzept wurden erfolgreich mithilfe des Software-Demonstrators umgesetzt, die Experteninterviews zeigen, dass die Anforderungen als sinnvoll gewählt bewertet werden.

7. Zusammenfassung und Ausblick

Durch die zunehmende Interdisziplinarität der Produkte und Trends wie den digitalen Zwilling werden im Engineering-Kontext immer mehr Daten aus immer mehr Datenquellen genutzt. Gleichwohl fehlt den Nutzern häufig das Wissen welche Daten für ihre Aufgaben relevant sein könnten und wo diese abgespeichert sind.

Gerade in größeren Unternehmen und Konzernen ist die Situation dabei besonders komplex, hier werden häufig mehrere ähnliche Systeme konzernweit eingesetzt, um die gleiche Aufgabe zu erfüllen, ohne dass dies in den jeweiligen Abteilungen zwangsläufig bekannt wäre.

Aufbauend auf dieser Feststellung ergibt sich die Frage, wie man Abhilfe schaffen könnte durch die Nutzung von Technologien und einem neuen Konzept, welches die Anforderungen erfüllt, welche sich aus der initialen Recherche, Interviews mit Firmen und dem Austausch mit anderen Forschenden ergeben. Eine der potenziellen Lösungsmöglichkeiten ist das am Lehrstuhl VPE entwickelte Metadaten-Repository SP²IDER (Semantic Product & Process, Digitized Engineering Repository), welches IT-Systeme und die darin enthaltenen Datentypen und Objekte beschreibt. Hier zeigt sich, dass bereits verschiedene Forschungsansätze verfolgt werden und es einige Lösungen gibt, welche von Firmen vermarktet werden, welche allerdings häufig einen anderen Fokus haben und die hier beschriebenen Anforderungen nicht erfüllen (können).

Darauf aufbauend wurde dann ein Konzept vorgeschlagen, welches bestehend aus neuen Prozessen, Methoden und einem IT-Tool einen Beitrag liefern soll zur besseren Beschreibung der Datenbestände in Unternehmen. Methodisch setzt das Konzept auf SP²IDER als Grundlage auf, nimmt die dort gespeicherten Daten und ergänzt diese um einen Knowledge Graph, welcher auf einer abstrakten Ebene die im Unternehmen vorhandenen Prozesse als Ausgangsbasis nimmt und ein Datenmodell aufbaut, welches unabhängig von IT-Systemen beschreibt, welche Daten in den (Engineering)-Prozessen genutzt bzw. erstellt werden. Dieses Datenmodell wird anschließend von einem Dateningenieur bzw. SP²IDER-Experten ausgewertet und die realen IT-Systeme mit den abstrahierten Systemen verbunden.

Auf Basis dieser Verbindungen ist ein Algorithmus in der Lage die vorhandenen Daten aus dem Knowledge-Graph und dem Metadaten-Repository auszuwerten und Ähnlichkeiten zu finden, diese dem Dateningenieur vorzuschlagen und sofern er diese bestätigt auch eine Verbindung zu ziehen. So entsteht ein aufgewertetes Datenmodell, welches sowohl eine abstrahierte Sicht (den Knowledge Graph), als auch die reale IT-Sicht (das Metadaten-Repository) enthält und von weiteren Nutzern auf der Suche nach Daten ausgewertet werden kann. Ein weiteres methodisches Konstrukt erlaubt es, Sichten auf diesen Graph zu definieren, welche dazu dienen weiteren Nutzern die Vorteile der Graph-Darstellung zu verwenden, um Daten zu finden und in ihre täglichen Prozesse einzubinden.

Der Algorithmus zur Ähnlichkeitssuche kombiniert verschiedenen Strategien und auch verschiedene Suchräume, deren Inhalte verglichen werden, um möglichst viele Übereinstimmungen zwischen den abstrahierten Datenobjekten und den vorhandenen

Datentypen aus realen IT-Quellsystemen festzustellen. Auch kann basierend auf den Ergebnissen noch die Sensibilität des Algorithmus durch den Nutzer eingestellt werden, falls zu viele oder zu wenige Übereinstimmungen vorgeschlagen werden.

Zur Realisierung dieses Ansatzes wurde auf verschiedene semantische Technologien zurückgegriffen, welche auch im Bereich des Semantic Web eingesetzt werden. Unter anderem wurde eine server- und servicebasierte Architektur gewählt, eine Kommunikation über REST-Schnittstellen gewährleistet und mit einem IT-Demonstrator auch prototypisch umgesetzt. Die Ergebnisse dieser Arbeit wurden anschließend mithilfe des Demonstrators und den erhobenen Anforderungen erfolgreich verifiziert und das Konzept in verschiedenen Interviews mit Forschenden auch erfolgreich validiert.

Im Rahmen dieser Arbeit sind weiterhin auch einige Anknüpfungspunkte für zukünftige Arbeiten entstanden, da natürlich im Rahmen einer Arbeit nicht alle Aspekte ausreichend tief erforscht werden können. So besteht weiterhin Forschungsbedarf in einem Kern der vorgestellten Methode zum Auffinden von Ähnlichkeiten in (Engineering)-Datenbeständen. Die hier angewendeten Strategien der Ähnlichkeitssuche sind relativ einfach, wenngleich bereits eine massive Hilfestellung zur vorherigen Situation. Gleichwohl können hier noch Verbesserungen erzielt werden, insbesondere auch durch das Einbeziehen weiterer Informationen oder auch der Objektdaten selbst.

Zu den weiteren Informationen, die man einbeziehen kann, zählen Prozessbeschreibungen für einen automatisierten Übersetzungsservice, welcher eine direkte Verbindung zwischen dem Knowledge-Graphen und der Prozessmodellierung schafft. Durch einen solchen Service und die bidirektionale Schnittstelle können auch Fehler in Prozessbeschreibungen erkannt und verbessert werden, wodurch nicht nur ein besserer Überblick über den Ist-Zustand, sondern auch ein besserer Soll-Zustand erreicht werden kann.

Ein weiterer Forschungsbedarf besteht bei der Ermöglichung unternehmensübergreifender Kollaboration. Das hier vorgestellte Konzept bietet zwar einen Ansatz für ein Sichtenkonzept, welches den Datenzugriff beschränkt, ein Szenario über Unternehmensgrenzen hinweg ist allerdings weitaus komplexer und verlangt eine neue Methodik. Bei Daten ergibt sich immer das Problem, dass sie schwer zurückzuziehen sind, sobald sie einmal bekannt sind. Ein möglicher Ansatz wäre hierbei ein unabhängiger Dritter, welcher Daten verarbeitet, ohne sie dem Partnerunternehmen bekannt zu geben. Ein Tool zum Visualisieren der Datenbestände, wie hier entwickelt könnte Unternehmen bei der Freigabe bzw. der Auswahl freigegebener Daten helfen, ohne diese vollständig zu übermitteln.

Ebenfalls im Bereich der Kollaboration mehrerer Unternehmen angesiedelt sind Verbesserungen an den Informationsgrundlagen beim Finden von Ähnlichkeiten. So ist angedacht, dass als ähnlich identifizierte Typen auch im Glossar angelegt werden. Dieses würde so bei einer erneuten Ähnlichkeitssuche direkt ein entsprechendes Paar finden und vorschlagen. Allerdings ist fraglich, inwieweit dieser Prozess mehrfach in einem Unternehmen durchgeführt wird, während andere Unternehmen, die erstmalig Ähnlichkeiten suchen hiervon profitieren könnten. Ein möglicher

Forschungsansatz könnte hier gezielt untersuchen, wie eine Kooperation mehrerer Firmen zum allseitigen Vorteil möglich ist, ohne kritisches Wissen oder kritische Daten freizugeben.

Im Rahmen dieser Arbeit wurde das Konzept prototypisch umgesetzt und der Demonstrator so weit erstellt, wie es insbesondere für die Absicherung der Anforderungen notwendig war. Sollte das Konzept industriell getestet und eingesetzt werden, muss vorher nochmal die Programmierung überarbeitet werden, um evtl. noch verbleibende Fehler zu finden und zu beheben, insbesondere aber um alle Themen rund um den Bereich Security zu überarbeiten, da diese noch nicht ausreichend abgesichert sind, um industriell eingesetzt zu werden. Aufgrund des Charakters als Prototyp sind auch viele Details des User-Interface oder auch Komfortfunktionen außerhalb des Fokus. Beispiele hierfür sind unter anderem die gesamte Funktionalität rund um das User-Management, sei es das Anlegen oder Ändern von Usern, die Verwaltung von Passwörtern oder auch die Verbindung mit bereits bestehenden Login-Providern.

Weiterer Forschungsbedarf besteht ebenfalls im Bereich Wissensmanagement. Der Knowledge Graph ist ein Mechanismus, welcher Wissen rund um die Datenzugänglichkeit bereitstellt, allerdings nicht in die Tiefe geht. Insbesondere in Bezug auf die Erarbeitung neuer smarter Produkte ist auch weiteres Wissen von größter Relevanz. Auf welcher Wissensbasis sind Entscheidungen in Vorgängerprodukten getroffen worden, was war die Grundlage für Strategien usw. Dieses Wissen ist im hier präsentierten Konzept nicht enthalten, gleichwohl könnte eine weitere Wissensmanagement-Lösung, welche sich mit Wissen im Bereich der eigentlichen Engineering-Daten befasst eine gute Ergänzung sein.

In der gleichen Forschungsrichtung ließen sich aufbauend auf dem hier vorgestellten Konzept auch weitere Management-Services erstellen, wie z.B. ein verbessertes Änderungs- oder Freigabemanagement. Generell ist mit dem hier geschaffenen Wissen eine Überarbeitung von Engineering-Prozessen denkbar und interessanter Forschungsgegenstand.

Zuletzt ist es von enormer Wichtigkeit, dass nicht nur weiter an den Methoden geforscht wird, die dem neuen Konzept zugrunde liegen, sondern auch der Einführungsprozess untersucht und unterstützt wird, damit die Realisierungschance des Konzepts steigt und typische Hürden einer IT-Einführung oder eines IT-Change Projekts direkt adressiert werden.

8. Literaturverzeichnis

- [AbGD2016] ABRAMOVICI, M., J.C. GÖBEL, and H.B. DANG. Semantic data management for the development and continuous reconfiguration of smart products and systems. *CIRP Annals*, 2016 65(1), 185-188.
- [AbGG2016] ABRAMOVICI, M., P. GEBUS, J.C. GÖBEL, and H.B. DANG. A Semantic Information Retrieval Framework within the Scope of IPS2-PLM. *Procedia CIRP*, 2016 47, 294-299.
- [AbHe2016] ABRAMOVICI, M., and O. HERZOG, eds. *Engineering im Umfeld von Industrie 4.0. Einschätzungen und Handlungsbedarf*. München: Herbert Utz Verlag GmbH; acatech, 2016. Acatech Studie. 3831645019.
- [AbMN2008] ABRAMOVICI, M., J. MICHELE, and M. NEUBACH. Erweiterung des PLM-Ansatzes für hybride Leistungsbündel. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 2008 103(9), 619-622.
- [Abra2014] ABRAMOVICI, M. Smart Products. In: T.I.A.f. PRODU, L. LAPERRIÈRE, and G. REINHART, eds. *CIRP Encyclopedia of Production Engineering*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 1-5. 978-3-642-35950-7.
- [AdAB2016] ADOLPHS, P., S. AUER, H. BEDENBENDER, M. BILLMANN, M. HANKEL, R. HEIDEL, M. HOFFMEISTER, H. HUHLE, M. JOCHEM, M. KIELE-DUSCHE, G. KOSCHNICK, H. KOZIOLEK, L. LINKE, R. PICHLER, F. SCHEWE, K. SCHNEIDER, and B. WASER. *Struktur der Verwaltungsschale. Fortentwicklung des Referenzmodells für die Industrie 4.0 Komponente*, 2016.
- [AiDA2017] AIDI, Y., H.B. DANG, M. ABRAMOVICI, P. GEBUS, and J.C. GÖBEL. Lifecycle Management industrieller Produkt-Service-Systeme. In: H. MEIER AND E. UHLMANN, eds. *Industrielle Produkt-Service Systeme*: Springer Berlin Heidelberg, 2017, pp. 245-272. 9783662480175.
- [Alas2009] ALASEM, A. An Overview of e-Government Metadata Standards and Initiatives based on Dublin Core. *Electronic Journal of e-Government*, 2009 7(1), 1-10.
- [AlDE2019] ALT, R., H. DEMIRKAN, J.F. EHMKE, A. MOEN, and A. WINTER. Smart services: The move to customer orientation. *Electronic Markets*, 2019 29(1), 1-6.
- [Alt2012] ALT, O. *Modellbasierte Systementwicklung mit SysML*, München: Carl Hanser Fachbuchverlag, 2012. 9783446431270.

- [AnLM2018] ANGÉE, S., S.I. LOZANO-ARGEL, E.N. MONTOYA-MUNERA, J.-D. OSPINA-ARANGO, and M.S. TABARES-BETANCUR. Towards an Improved ASUM-DM Process Methodology for Cross-Disciplinary Multi-organization Big Data & Analytics Projects. In: L. UDEN, B. HADZIMA, and I.-H. TING, eds. *Knowledge Management in Organizations. 13th International Conference, KMO 2018, Žilina, Slovakia, August 6-10, 2018, Proceedings*. Cham: Springer International Publishing, 2018, pp. 613-624. 978-3-319-95203-1.
- [apig2017] APIGEE. *The State of APIS. 2017 Report: How APIs Power Digital Ecosystems*, 2017.
- [ArGM2018] ARNARSSON, Í. Ö., E. GUSTAVSSON, J. MALMQVIST, and M. JIRSTRAND. Analysis of Engineering Change Requests using Markov Chains. In: D. MARJANOVIĆ, M. ŠTORGA, S. ŠKEC, N. BOIČETIĆ, and N. PAVKOVIĆ, eds. *Design 2018. Proceedings of the 15th International Design Conference, May 2018, Dubrovnik, Croatia*. Zagreb: Fac. of Mechanical Engineering and Naval Architecture Univ, 2018a, pp. 523-532. 978-953-7738-59-4.
- [ArGM2018] ARNARSSON, Í. Ö., E. GUSTAVSSON, J. MALMQVIST, and M. JIRSTRAND. Design analytics is the answer, but what questions would product developers like to have answered? In: A. MAIER, H. KIM, J. OEHMEN, F. SALUSTRI, S. ŠKEC, and M. KOKKOLARAS, eds. *Design theory and research methodology*. Red Hook, NY: Curran Associates Inc, 2018b, pp. 71-80. 978-1-904670-95-7.
- [ArMG2016] ARNARSSON, Í. Ö., J. MALMQVIST, E. GUSTAVSSON, and M. JIRSTRAND. Towards big-data analysis of deviation and error reports in product development projects. In: C. BOKS, ed. *Proceedings of NordDesign 2016. August 10-12, 2016, Trondheim, Norway*. Bristol, United Kingdom: The Design Society, 2016, pp. 83-92. 978-1-904670-80-3.
- [AuZa2015] AUGUSTIN, M., and H. ZADEK. Kooperatives Änderungsmanagement. *Deutscher Logistik-Kongress: Eine Welt in Bewegung*, 2015, 260-291.
- [AzSa2008] AZEVEDO, A., and M.F. SANTOS. KDD, SEMMA AND CRISP-DM: A PARALLEL OVERVIEW. In: H. WEGHORN AND ABRAHAM, AJITH, P., eds. *Proceedings of INFORMATICS 2008 and DATA MINING 2008*, 2008, pp. 182-185. 978-972-8924-63-8.
- [BaAG2011] BAUER, F., H. ANACKER, T. GAUKSTERN, J. GAUSEMEIER, and V. JUST. ANALYZING THE DYNAMIC BEHAVIOR OF MECHATRONIC SYSTEMS WITHIN THE CONCEPTUAL DESIGN. In: S.J. CULLEY, B. HICKS, T.C. MCALOONE, T.J. HOWARD, and U. LINDEMANN, eds. *Product and systems design*. Glasgow: Design Society, 2011, pp. 329-336. 978-1-904670-24-7.

- [BaEF2020] BARTH, L., M. EHRAT, R. FUCHS, and J. HAARMANN. Systematization of Digital Twins. *Proceedings of the 2020 The 3rd International Conference on Information Science and System*. New York, NY, USA: ACM, 2020, pp. 13-23. 9781450377256.
- [BaTy2012] BATRA, S., and C. TYAGI. Comparative analysis of relational and graph databases. *International Journal of Soft Computing and Engineering (IJSC)*, 2012 2(2).
- [BeGK2015] BENTLAGE, A., F. GEHLER, W. KASTENHUBER, C. KIND, L. MENG, and M. ZENKER. *Akquisition und Nutzung von Lebenszyklus-Wissen für Produktinnovation - LeWiPro : gemeinsamer Abschlussbericht : Berichtszeitraum: 01. Januar 2013 bis 30. Juni 2015*, 2015.
- [BeLD2011] BERGER, O., S. LABBENE, M. DHAR, and C. BAC. Introducing OSLC, an open standard for interoperability of open source development tools. *ICSSSEA*, 2011 (ISSN-0295-6322).
- [BePV2012] BECKER, J., W. PROBANDT, and O. VERING. *Grundsätze ordnungsmäßiger Modellierung. Konzeption und Praxisbeispiel für ein effizientes Prozessmanagement*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2012. BPM kompetent. 9783642304125.
- [Berl2007] BERLINER KREIS. Berliner Kreis-Mitteilungen. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 2007 102(5), 262.
- [BeZe2014] BENTLAGE, A., and M. ZENKER. Data Mining of lifecycle information. In: L. OVERMEYER AND V.P. SHKODYREV, eds. *AST - Symposium on Automated Systems and Technologies. 15 - 16 October 2014*. Garbsen: PZH-Verl. TEWISS - Technik und Wissen, 2014, pp. 15-20. 978-3-944586-84-7.
- [BeZK2014] BENTLAGE, A., M. ZENKER, and C. KIND. Analyse von Lebenszyklusdaten. *ERP Management*, 2014 (3), 50-53.
- [Bied2015] BIEDERMANN, W. *A minimal set of network metrics for analysing mechatronic product concepts*. Dissertation, München, 2015.
- [BiEF2018] BITZER, M., M. EIGNER, K.-G. FAIßT, C. MUGGEO, and T. EICKHOFF. Framework of the evolution in virtual product modelling and model management towards digitized engineering. In: A. MAIER, H. KIM, J. OEHMEN, F. SALUSTRI, S. ŠKEC, and M. KOKKOLARAS, eds. *Design theory and research methodology*. Red Hook, NY: Curran Associates Inc, 2018, pp. 345-354. 978-1-904670-95-7.
- [BiMM2012] BIFFL, S., R. MORDINYI, and T. MOSER. Anforderungsanalyse für das integrierte Engineering. Mechanismen und Bedarfe aus der Praxis. *atp edition*, 2012 (5), 28-35.

- [BiMS2017] BIFFL, S., R. MORDINYI, H. STEININGER, and D. WINKLER. Integrationsplattform für anlagenmodellorientiertes Engineering. In: B. VOGEL-HEUSER, T. BAUERNHANSL, and M. ten HOMPEL, eds. *Handbuch Industrie 4.0 Bd.2. Automatisierung*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2017, pp. 189-212. 978-3-662-53247-8.
- [Bitz2008] BITZER, M. A. *Entwicklung einer Methode zur prozessorientierten Planung und Optimierung von Product Lifecycle Management Lösungen. Am Beispiel der Automobilindustrie*. Als Ms. gedr, Kaiserslautern: Techn. Univ, 2008. Schriftenreihe VPE. 6. 978-3-941438-00-2.
- [BlCh2009] BLESSING, L. T., and A. CHAKRABARTI. *DRM, a Design Research Methodology*, London: Springer London, 2009. 978-1-84882-587-1.
- [BlPe2006] BLUMAUER, A., and T. PELLEGRINI. Semantic Web und semantische Technologien: Zentrale Begriffe und Unterscheidungen. In: T. PELLEGRINI AND A. BLUMAUER, eds. *Semantic Web. Wege zur vernetzten Wissensgesellschaft ; mit 4 Tabellen*. Berlin: Springer, 2006, pp. 9-25. 978-3-540-29324-8.
- [BlPM2022] BLEISINGER, O., T. PSOTA, J. MASIOR, M. PFENNING, A. ROTH, A. REICHWEIN, Y. HOOSHMAND, C. MUGGEO, and M. HUTSCH. *Killing the PLM Monolith - the Emergence of cloud-native System Lifecycle Management (SysLM)*, 2022.
- [BoHH2019] BOUZIDI, A., N. HADDAR, and K. HADDAR. Traceability and Synchronization Between BPMN and UML Use Case Models. *Ingénierie des systèmes d information*, 2019 24(2), 215-228.
- [BoRJ1999] BOOCH, G., J. RUMBAUGH, and I. JACOBSON. *Das UML-Benutzerhandbuch. Aktuell zur Version 2.0*, 1999.
- [BrFR2021] BRAUD, A., G. FROMENTOUX, B. RADIER, and O. LE GRAND. The Road to European Digital Sovereignty with Gaia-X and IDSA. *IEEE Network*, 2021 35(2), 4-5.
- [BrLL2021] BRÜNNHÄUßER, J., K. LINDOW, P. LÜNNEMANN, L. KIRSCH, and K. WRASSE. Automatisches Vergleichen von Anforderungen in natürlicher Sprache. *ProduktDatenJournal*, 2021 (1), 58-63.
- [BrTD2016] BRETZ LUKAS, TSCHIRNER CHRISTIAN, and DUMITRESCU ROMAN. A concept for managing information in early stages of product engineering by integrating MBSE and workflow management systems. *2016 IEEE International Symposium on Systems Engineering (ISSE)*, 2016, pp. 1-8.

- [ChDG2021] CHEN, W., M. DAS, V.J.L. GAN, and J.C.P. CHENG. Integrated Data Model and Mapping for Interoperable Information Exchange Between BIM and Energy Simulation Tools. In: E. TOLEDO SANTOS AND S. SCHEER, eds. *Proceedings of the 18th International Conference on Computing in Civil and Building Engineering*. Cham: Springer International Publishing, 2021, pp. 496-506. 978-3-030-51294-1.
- [ChFo2013] CHANDA, S., and D. FOGGON. Application Lifecycle Management. In: S. CHANDA AND D. FOGGON, eds. *Beginning ASP.NET 4.5 databases. All you need to know to build data-driven web sites using ASP.NET 4.5, ADO.NET Entity Framework 5, and C*. New York, NY.: Apress / Springer, 2013, pp. 235-249. 978-1-4302-4380-9.
- [ChJX2020] CHEN, X., S. JIA, and Y. XIANG. A review: Knowledge reasoning over knowledge graph. *Expert Systems with Applications*, 2020 141, 112948.
- [ChTr2012] CHINOSI, M., and A. TROMBETTA. BPMN: An introduction to the standard. *Computer Standards & Interfaces*, 2012 34(1), 124-134.
- [Conr1997] CONRAT NIEMERG, J.-I. *Änderungskosten in der Produktentwicklung*, 1997.
- [Cors2004] CORSTEN, H. *Produktionswirtschaft. Einführung in das industrielle Produktionsmanagement*. 10., vollst. überarb. Aufl., München: Oldenbourg, 2004. Lehr- und Handbücher der Betriebswirtschaftslehre. 3486274740.
- [CrAD2003] CRNKOVIC, I., U. ASKLUND, and A. P. DAHLQVIST. *Implementing and integrating product data management and software configuration management*, Boston: Artech House, 2003. Artech House computing library. 9781580534987.
- [Dame1964] DAMERAU, F. J. A technique for computer detection and correction of spelling errors. *Communications of the ACM*, 1964 7(3), 171-176.
- [Dang2017] DANG, H. B. *Erweiterung des Product Lifecycle Managements für industrielle Produkt-Service Systeme*. Dissertation, 2017.
- [DeER2016] DEMARY, V., B. ENGELS, K.-H. RÖHL, and C. RUSCHE. *Digitalisierung und Mittelstand. Eine Metastudie*, 2016. IW-Analysen.
- [Deng2012] DENGEL, A., ed. *Semantische Technologien. Grundlagen - Konzepte - Anwendungen*, Heidelberg: Spektrum Akademischer Verlag, 2012. 978-3-8274-2663-5.
- [Deo1974] DEO, N. *Graph theory with applications to engineering and computer science*, Englewood Cliffs, N. J.: Prentice-Hall, 1974. Prentice-Hall series in automatic computation. 0-13-363473-6.
- [DeRE2020] DETZNER, A., R. RUCKSCHLOS, and M. EIGNER. Root-Cause Analysis with Interactive Decision Trees. *2020 24th International Conference Information Visualisation (IV)*: IEEE, 2020, pp. 322-327. 978-1-7281-9134-8.
- [Detz2022] DETZNER, A. *Digital Twin and Data Analytics for Product Quality Monitoring and Root-Cause Analysis*. 1. Auflage, Düren: Shaker, 2022. 9783844087192.

- [DIN 2014] 60050 - 351, *Internationales Elektrotechnisches Wörterbuch - Teil 351: Leittechnik*, Berlin: Beuth, 2014.
- [DIN 2015] 9000, *Qualitätsmanagementsysteme - Grundlagen und Begriffe*, Berlin: Beuth, 2015a.
- [DIN 2015] 9001, *Qualitätsmanagementsysteme: Anforderungen*, 2015b.
- [DIN2009] DIN 69901-2, *Projektmanagement - Projektmanagementsysteme - Teil 2: Prozesse, Prozessmodell*, Berlin: Beuth, 2009.
- [DIN2016] 91345, *Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)*, Berlin: Beuth, 2016.
- [EiAA2019] EICKHOFF, T., H. APOSTOLOV, J.C. AURICH, C. DONGES, M. FISCHER, J.C. GÖBEL, C. HERDER, P. KÖLSCH, C. SCHRANK, S. TRENDLER, and V. ZIMMERMANN. Anwendungsfall John Deere. In: J.C. AURICH, W. KOCH, P. KÖLSCH, and C.F. HERDER, eds. *Entwicklung datenbasierter Produkt-Service Systeme. Ein Ansatz zur Realisierung verfügbarkeitsorientierter Geschäftsmodelle*. Berlin: Springer Vieweg, 2019. 978-3-662-59642-5.
- [EiAD2019] EICKHOFF, T., H. APOSTOLOV, DONGES, CHRISTIAN, M. FISCHER, J.C. GÖBEL, M. GREHLER, J. IMWALLE, R. MATTUKAT, L. LIU, K. PANKRATH, F. ZEIHSEL, and J. RICHTER. Intelligentes Informationsmanagement für verfügbarkeitsorientierte Geschäftsmodelle. In: J.C. AURICH, W. KOCH, P. KÖLSCH, and C.F. HERDER, eds. *Entwicklung datenbasierter Produkt-Service Systeme. Ein Ansatz zur Realisierung verfügbarkeitsorientierter Geschäftsmodelle*. Berlin: Springer Vieweg, 2019, pp. 45-108. 978-3-662-59642-5.
- [EiAE2017] EIDEN, A., H. APOSTOLOV, and M. EIGNER. Crowd Engineering - Bringing full cloud CAD into the lab. In: M. VAN DEN BERG AND M. VAN STEENBERGEN, eds. *Building an enterprise architecture practice. Tools, tips, best practices, ready-to-use insights*. Glasgow, Scotland: The Design Society, Institution of Engineering Designers, 2017, pp. 170-175. 1904670849.
- [EiAS2016] EIGNER, M., U. AUGUST, and M. SCHMICH. *Smarte Produkte erfordern ein Umdenken bei Produktstrukturen und Prozessen. Digitalisierung, Integration, Interdisziplinarität und Föderation*, 2016.

- [EiDA2014] EIGNER, M., T. DICKOPF, H. APOSTOLOV, P. SCHAEFER, K.-G. FAIßT, and A. KEßLER. System Lifecycle Management: Initial Approach for a Sustainable Product Development Process Based on Methods of Model Based Systems Engineering. In: S. FUKUDA, A. BERNARD, B. GURUMOORTHY, and A. BOURAS, eds. *Product Lifecycle Management for a Global Market*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2014, pp. 287-300. 978-3-662-45936-2.
- [EiDS2019] EIGNER, M., A. DETZNER, P.H. SCHMIDT, and R. THARMA. Definition des Digital Twin im Produktlebenszyklus. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 2019 114(6), 345-350.
- [EiEG2020] EICKHOFF, T., A. EIDEN, J.C. GÖBEL, and M. EIGNER. A Metadata Repository for Semantic Product Lifecycle Management. *Procedia CIRP*, 2020 91, 249-254.
- [EiEG2021] EICKHOFF, T., A. EIDEN, J. GRIES, and J.C. GÖBEL. Data Model Canvas für die IT-Systemübergreifende Integration von Datenmodellen zur Unterstützung von Datenanalyse-Anwendungen im Produktlebenszyklus. In: R.H. STELZER AND J. KRZYWINSKI, eds. *Entwerfen Entwickeln Erleben in Produktentwicklung und Design 2021*. Dresden: TUDpress, 2021a, pp. 99-109. 9783959084505.
- [EiEG2021] EIDEN, A., T. EICKHOFF, J. GRIES, J.C. GÖBEL, and T. PSOTA. Supporting semantic PLM by using a lightweight engineering metadata mapping engine. *Procedia CIRP*, 2021b 100, 690-695.
- [EiEG2022] EIDEN, A., T. EICKHOFF, J.C. GÖBEL, C. APOSTOLOV, P. SAVARINO, and T. DICKOPF. Data Networking for Industrial Data Analysis Based on a Data Backbone System. *Proceedings of the Design Society*, 2022 2, 693-702.
- [EiGE2020] EIDEN, A., J. GRIES, T. EICKHOFF, and J.C. GÖBEL. Anforderungen an ein Daten-Backend-System zur Unterstützung industrieller Datenanalyse-Anwendungen in digitalen Engineering-Prozessen dynamischer Wertschöpfungsnetzwerke. In: D. KRAUSE, K. PAETZOLD, and S. WARTZACK, eds. *Proceedings of the 31st Symposium Design for X (DFX2020)*: The Design Society, 16 Dec. 2020, pp. 81-90.
- [Eign2013] EIGNER, M. Modellbasierte Virtuelle Produktentwicklung auf einer Plattform für System Lifecycle Management. In: U. SENDLER, ed. *Industrie 4.0. Beherrschung der industriellen Komplexität mit SysLM*. Berlin: Springer Vieweg, 2013, pp. 91-110. 978-3-642-36916-2.
- [Eign2020] EIGNER, M. Digitaler Zwilling – Stand der Technik. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 2020 115(special), 3-6.
- [Eign2021] EIGNER, M. *System Lifecycle Management*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2021. 978-3-662-62182-0.

- [EiGZ2012] EIGNER, M., T. GILZ, and R. ZAFIROV. Systems Engineering VPE Data Schema: PDM Integration of a functional product description model in SysML. *The 13th Mechatronics Forum International Conference. Proceedings; September 17 - 19, 2012, Johannes Kepler University Linz, Austria*. Linz: Trauner, 2012, pp. 651-657. 978-3990330456.
- [EiHS1991] EIGNER, M., C. HILLER, S. SCHINDEWOLF, and M. SCHMICH. *Engineering database. Strategische Komponente in CIM-Konzepten*, München: Hanser, 1991. Praxiswissen CA...-Techniken. 3446164383.
- [EiRZ2014] EIGNER, M., D. ROUBANOV, and R. ZAFIROV. *Modellbasierte virtuelle Produktentwicklung*, Berlin: Springer Vieweg, 2014. 978-3-662-43815-2.
- [EiSA2014] EIGNER, M., P. SCHAEFER, and H. APOSTOLOV. Improving the Life Cycle Eco Performance of a Wheeled Excavator. in the Context of "ERMA - Energy and Resource Efficient Mobile Working Machines". In: C. SCHINDLER, K. DREBLER, B. JÖRG, R. KALMAR, and G. ZOLYNSKI, eds. *Commercial vehicle technology 2014. Proceedings of the 3rd Commercial Vehicle Technology Symposium (CVT 2014), March 11-13, 2014, University of Kaiserslautern, Kaiserslautern, Germany*. Aachen: Shaker, 2014, pp. 244-253. 978-3-8440-2573-6.
- [EiSt2009] EIGNER, M., and R. STELZER. *Product Lifecycle Management. Ein Leitfaden für Product Development und Life Cycle Management*, 2009. 978-3-540-68401-5.
- [EiER2020] ELSALEH, T., S. ENSHAEIFAR, R. REZVANI, S.T. ACTON, V. JANEIKO, and M. BERMUDEZ-EDO. IoT-Stream: A Lightweight Ontology for Internet of Things Data Streams and Its Use with Data Analytics and Event Detection Services. *Sensors (Basel, Switzerland)*, 2020 20(4).
- [EiNe2013] ELAASAR, M., and A. NEAL. Integrating Modeling Tools in the Development Lifecycle with OSLC: A Case Study. In: D. HUTCHISON, T. KANADE, J. KITTLER, J.M. KLEINBERG, F. MATTERN, J.C. MITCHELL, M. NAOR, O. NIERSTRASZ, C. PANDU RANGAN, B. STEFFEN, M. SUDAN, D. TERZOPOULOS, D. TYGAR, M.Y. VARDI, G. WEIKUM, A. MOREIRA, B. SCHÄTZ, J. GRAY, A. VALLECILLO, and P. CLARKE, eds. *Model-Driven Engineering Languages and Systems*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2013, pp. 154-169. 978-3-642-41532-6.
- [Enge2017] ENGELS, B. Bedeutung von Standards für die digitale Transformation. Befunde auf Basis des IW-Zukunftspanels. *IW-Trends*, 2017 2(44), 21-40.
- [Erns2016] ERNST, J. *Systemübergreifendes Änderungsmanagement*, 2016. Schriftenreihe VPE. Band 17. 9783959740364.

- [FaDa2007] FAIßT, K.-G., and C.W. DANKWORT. New Extended Concept for the Usage of Engineering Objects and Product Properties in the Virtual Product Generation Process. In: J.-C. BOCQUET, ed. *DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07.2007*. Glasgow?: The Design Society, 2007, pp. 689-690.
- [Fahr1995] FAHRWINKEL, U. *Methode zur Modellierung und Analyse von Geschäftsprozessen zur Unterstützung des business process reengineering*. 1. Aufl., Paderborn: Heinz Nixdorf Inst, 1995. HNI-Verlagsschriftenreihe, 1. 3931466000.
- [FeGr2013] FELDHUSEN, J., and K.-H. GROTE, eds. *Pahl/Beitz Konstruktionslehre. Methoden und Anwendung erfolgreicher Produktentwicklung*. 8. Aufl. 2013, Berlin, Heidelberg: Springer Berlin Heidelberg, 2013. 9783642295690.
- [FeŞA2020] FENSEL, D., U. ŞİMŞEK, K. ANGELE, E. HUAMAN, E. KÄRLE, O. PANASIUK, I. TOMA, J. UMBRICH, and A. WAHLER. *Knowledge Graphs. Methodology, Tools and Selected Use Cases*. 1st ed. 2020, Cham: Springer International Publishing; Imprint Springer, 2020. Springer eBook Collection. 978-3-030-37438-9.
- [FiSS2012] FISCHER, L., R. SHAPIRO, and B. SILVER, eds. *BPMN 2.0 handbook second edition. Methods, concepts, case studies and standards in business process management notation*. 2nd ed., Lighthouse Point, FL.: Future Strategies, 2012. 9780984976409.
- [FoEW2022] FORTE, S., T. EHEMANN, K. WIEGAND, and J.C. GÖBEL. Enabling Collaborative Lifecycle Engineering of Smart Products and Services by an Adaptive Innovation Infrastructure. In: O. CANCELIERI JUNIOR, F. NOËL, L. RIVEST, and A. BOURAS, eds. *Product Lifecycle Management. Green and Blue Technologies to Support Smart and Sustainable Organizations. 18th IFIP WG 5.1 International Conference, PLM 2021, Curitiba, Brazil, July 11–14, 2021, Revised Selected Papers, Part I*. Cham: Springer International Publishing; Imprint Springer, 2022, pp. 347-357. 978-3-030-94334-9.
- [FoGD2021] FORTE, S., J.C. GÖBEL, and T. DICKOPF. SYSTEM OF SYSTEMS LIFECYCLE ENGINEERING APPROACH INTEGRATING SMART PRODUCT AND SERVICE ECOSYSTEMS. *Proceedings of the Design Society*, 2021 1, 2911-2920.
- [Gada2008] GADATSCH, A. *Grundkurs Geschäftsprozess-Management. Methoden und Werkzeuge für die IT-Praxis ; eine Einführung für Studenten und Praktiker ; [mit Online-Service zum Buch*. 5., erw. und überarb. Auflage, Wiesbaden: Vieweg, 2008. OnlinePlus. 978-3-8348-0363-4.
- [Gaia2021] GAIA-X EUROPEAN ASSOCIATION FOR DATA AND CLOUD AISBL. *Gaia-X Architecture Document. 22.04 Release*, 2021.

- [Gali2006] GALINSKI, C. Wozu Normen? Wozu semantische Interoperabilität? In: T. PELLEGRINI AND A. BLUMAUER, eds. *Semantic Web. Wege zur vernetzten Wissensgesellschaft ; mit 4 Tabellen*. Berlin: Springer, 2006, pp. 47-72. 978-3-540-29324-8.
- [GaSM2019] GARTZEN, T., S. SCHLOESSER, and M. MENDL-HEINISCH. *Engineering 4.0. Eine Studie über die Effizienzpotenziale und den Einsatz von CAE-Software in der Elektrokonstruktion für das Engineering von Maschinen und Anlagen*, 2019.
- [GaTD2013] GAUSEMEIER, J., C. TSCHIRNER, and R. DUMITRESCU. Der Weg zu Intelligenten Technischen Systemen. *Industrie Management*, 2013 29(1).
- [GaTS2014] GAUSEMEIER, J., A. TRÄCHTLER, and W. SCHÄFER. *Semantische Technologien im Entwurf mechatronischer Systeme. Effektiver Austausch von Lösungswissen in Branchenwertschöpfungskette*. [Elektronische Ressource], München: Hanser, 2014. Hanser eLibrary. 9783446438453.
- [Gaus2006] GAUSEMEIER, J., ed. *Vernetzte Produktentwicklung. Der erfolgreiche Weg zum Global Engineering Networking*, München: Hanser, 2006. 3-446-22725-3.
- [GeGa2022] GEHRING, H., and R. GABRIEL. *Wirtschaftsinformatik*, Wiesbaden: Springer Gabler, 2022. Lehrbuch. 978-3-658-37701-4.
- [GiEi2013] GILZ, T., and M. EIGNER. Ansatz zur integrierten Verwendung von SysML Modellen in PLM zur Beschreibung der funktionalen Produktarchitektur. In: M. MAURER AND S.-O. SCHULZE, eds. *Tag des Systems Engineering. Stuttgart 6. - 8. November 2013 ; Der Weg zu den technischen Systemen von morgen*. s.l.: Carl Hanser Fachbuchverlag, 2013, pp. 293-302. 978-3-446-43915-3.
- [Gier2000] GIERHAKE, O. *Integriertes Geschäftsprozeßmanagement. Effektive Organisationsgestaltung mit Workflow-, Workgroup- und Dokumentenmanagement-Systemen*. 3., verb. und erw. Aufl., Braunschweig: Vieweg, 2000. Edition Wirtschaftsinformatik. 3528256656.
- [Gilz2014] GILZ, T. *PLM-Integrated Interdisciplinary System Models in the Conceptual Design Phase Based on Model-Based Systems Engineering*, Kaiserslautern: Kaiserslautern Techn. Univ, 2014. 9783941438732.
- [GLSt2013] GLAESSGEN, E., and D.S. STARGEL. The Digital Twin Paradigm for Future NASA and U.S. Air Force Vehicles. *53rd AIAA/ASME/ASCE/AHS/ASC Structures, Structural Dynamics and Materials Conference - Special Session on the Digital Twin*, 2013.
- [GöEi2020] GÖBEL, J. C., and T. EICKHOFF. Konzeption von Digitalen Zwillingen smarter Produkte. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 2020 115(special), 74-77.

- [GóFC2004] GÓMÉZ-PÉREZ, A., M. FERNÁNDEZ-LÓPEZ, and O. CORCHO. *Ontological Engineering*, London: Springer-Verlag, 2004. 1-85233-551-3.
- [GoHT2020] GOPSILL, J., M. HUMPHREY, D. THOMPSON, and E. GARCIA. CO-WORD GRAPHS FOR DESIGN AND MANUFACTURE KNOWLEDGE MAPPING. *Proceedings of the Design Society: DESIGN Conference*, 2020 1, 1275-1284.
- [GöMü2020] GÖCKEL, N., and P. MÜLLER. Entwicklung und Betrieb Digitaler Zwillinge. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 2020 115(special), 7-10.
- [GrBa2020] GROSSER, T., and C. BANGE. *Linked Data: A Promising Approach for Data Governance and Advanced Analytics*, 2020.
- [GrBU2015] GROSS, D., B. BREDEHORST, and D. ULRICH. *Wissen der Organisation. DIN EN ISO 9001:2015 interpretiert und für die Praxis erläutert*, 2015.
- [GrEE2021] GRIES, J., T. EICKHOFF, A. EIDEN, and J.C. GÖBEL. Supporting Linked Engineering Data Management of Smart Product Systems Through Semantic Platform Services. In: F. NYFFENEGGER, J. RÍOS, L. RIVEST, and A. BOURAS, eds. *PRODUCT LIFECYCLE MANAGEMENT ENABLING SMART. 17th ifip wg 5.1*. [S.l.]: Springer, 2021, pp. 241-255. 978-3-030-62806-2.
- [Grie2021] GRIEVES, M. *Intelligent Digital Twins: The Role of AI and ML in the Future of Digital Twins Chief Scientist of Advanced Manufacturing Florida Institute of Technology*, 2021.
- [GrKL2014] GROSS, D., C. KIND, and D. LANGENBERG. *Software für Wissensmanagement. Vergleich von Wikis, Wissensdatenbanken, Groupware und Suchmaschinen*, 2014.
- [GrSt2020] GREFEN, K., and W. STROMMER. Die Auflösung der Automatisierungspyramide. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 2020 115(special), 15-18.
- [Grub1995] GRUBER, T. R. Toward principles for the design of ontologies used for knowledge sharing? *International Journal of Human-Computer Studies*, 1995 43(5-6), 907-928.
- [GrWE2018] GROTHOFF, J. A., C. A. WAGNER, and U. EPPLÉ. *BaSys 4.0: Metamodell der Komponenten und Ihres Aufbaus; 1st ed*, 2018.
- [GuBM2016] GUHA, R. V., D. BRICKLEY, and S. MACBETH. Schema.org: Evolution of Structured Data on the Web. *Communications of the ACM*, 2016 59(2), 44-51.
- [Hamm1950] HAMMING, R. W. Error Detecting and Error Correcting Codes. *Bell System Technical Journal*, 1950 29(2), 147-160.
- [Harr2016] HARRINGTON, J. L. *Relational Database Design and Implementation*. 4. Aufl., s.l.: Elsevier Reference Monographs, 2016. 978-0-12-804399-8.

- [HeKi2016] HELMS, B., and M. KISSEL. Engineering Intelligence – Von der graphenbasierten Modellierung zur wissensbasierten Datenanalyse. In: U. LINDEMANN, ed. *Handbuch Produktentwicklung*. München: Hanser, 2016, pp. 979-1012. 978-3-446-44518-5.
- [HeTE2019] HELLGE, V., T. THIELEN, A. EIDEN, and N. OBRESCHKOVA. Die Einführung eines ERP-/PLM-Systems in den frühen Phasen der digitalen Transformation erfolgreich vorbereiten. In: C.K. BOSSE AND K.J. ZINK, eds. *Arbeit 4.0 im Mittelstand*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2019, pp. 219-237. 978-3-662-59473-5.
- [HiKK2020] HILDEBRANDT, C., A. KOCHER, C. KUSTNER, C.-M. LOPEZ-ENRIQUEZ, A.W. MULLER, B. CAESAR, C.S. GUNDLACH, and A. FAY. Ontology Building for Cyber-Physical Systems: Application in the Manufacturing Domain. *IEEE Transactions on Automation Science and Engineering*, 2020, 1-17.
- [HiKR2008] HITZLER, P., M. KRÖTZSCH, S. RUDOLPH, and Y. SURE. *Semantic Web. Grundlagen*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. eXamen.press. 978-3-540-33994-6.
- [HiTB2018] HILDEBRANDT, C., S. TÖRSLEFF, T. BANDYSZAK, B. CAESAR, A. LUDEWIG, and A. FAY. Ontology Engineering for Collaborative Embedded Systems. Requirements and Initial Approach. In: I. SCHAEFER, L. CLEOPHAS, and M. FELDERER, eds. *Workshops at Modellierung 2018. Modellierung in der Entwicklung von kollaborativen eingebetteten Systemen (MEKES)*: CEUR-WS, 2018, pp. 57-66.
- [HiTC2018] HILDEBRANDT, C., S. TORSLEFF, B. CAESAR, and A. FAY. Ontology Building for Cyber-Physical Systems: A domain expert-centric approach. In: B. VOGEL-HEUSER, ed. *2018 IEEE 14th International Conference on Automation Science and Engineering (CASE). 20-24 Aug. 2018*. Piscataway, NJ: IEEE, 2018, pp. 1079-1086. 978-1-5386-3593-3.
- [Horr2011] HORRIDGE, M. *A Practical Guide To Building OWL Ontologies Using Protégé 4 and Co-ODE Tools*, 2011.
- [IEEE1990], *IEEE Standard Glossary of Software Engineering Terminology*, Piscataway, NJ, USA: IEEE, 1990.
- [Iser2008] ISERMANN, R. *Mechatronische Systeme*, Berlin, Heidelberg: Springer Berlin Heidelberg, 2008. 978-3-540-32336-5.
- [ISO2017] ISO 15836-1:2017, *Information and documentation – The Dublin Core metadata element set – Part 1: Core elements*, 2017.
- [ISO2019] ISO 15836-2:2019, *Information and documentation – The Dublin Core metadata element set – Part 2: DCMI Properties and classes*, 2019.

- [IWCO2019] IWCONSULT. *eCl@ss - Nutzen, Ziele und Verwendung. In welchen Bereichen Unternehmen eCl@ss nutzen*, 2019.
- [J.PE2001] J. MADHAVAN, P. BERNSTEIN, and E. RAHM. Generic Schema Matching with Cupid. *VLDB 11*, 2001.
- [JaAB2012] JAFARI, M., P. AKHAVAN, A. BOUROUNI, and H. ROOZBEH. A Framework for the Selection of Knowledge Mapping Techniques. *Journal of Knowledge Management*, 2012 10(1).
- [JaCE2005] JARRATT, T., J. CLARKSON, and C. ECKERT. Engineering change. In: J. CLARKSON AND C. ECKERT, eds. *Design process improvement. A review of current practice*. London: Springer, 2005, pp. 262-285. 978-1-85233-701-8.
- [JaKR2017] JANNIDIS, F., H. KOHLE, and M. REHBEIN, eds. *Digital Humanities. Eine Einführung*, Stuttgart: J.B. Metzler Verlag, 2017. 978-3-476-05446-3.
- [JoAH2017] JORDAN, A., K. ANZENGRUBER, and P. HEHENBERGER. Distributed data management in product development using Git. *International Journal of Product Lifecycle Management*, 2017 10(1), 69-85.
- [Jura1993] JURAN, J. M. *Der neue Juran. Qualität von Anfang an*, Landsberg/Lech: Verl. Moderne Industrie, 1993. 347841010X.
- [KaHD2021] KANNOTH, S., J. HERMANN, M. DAMM, P. RÜBEL, D. RUSIN, M. JACOBI, B. MITTELSDORF, T. KUHN, and P.O. ANTONINO. Enabling SMEs to Industry 4.0 Using the BaSyx Middleware: A Case Study. In: S. BIFFL, E. NAVARRO, W. LÖWE, M. SIRJANI, R. MIRANDOLA, and D. WEYNS, eds. *Software Architecture. 15th European Conference, ECSA 2021, Virtual Event, Sweden, September 13-17, 2021, Proceedings*. Cham: Springer International Publishing; Imprint Springer, 2021, pp. 277-294. 978-3-030-86043-1.
- [KaMK2013] KADIRI, S., A. MILICIC, and D. KIRITSIS. Linked Data exploration in PLM. In: V. PRABHU, M. TAISCH, and D. KIRITSIS, eds. *Advances in Production Management Systems. Sustainable Production and Service Supply Chains. IFIP WG 5. 7 International Conference, APMS 2013, State College, PA, USA, September 9-12, 2013, Proceedings, Part II*. Berlin/Heidelberg: Springer Berlin Heidelberg, 2013, pp. 460-467. 978-3-642-412639.
- [KaRH2014] KAGERMANN, H., F. RIEMENSPERGER, D. HOKE, J. HELBIG, D. STOCKSMEIER, W. WAHLSTER, A.-W. SCHEER, and D. SCHWEER. *Smarte Service Welt. Umsetzungsempfehlungen für das Zukunftsprojekt Internetbasierte Dienste für die Wirtschaft*, 2014.
- [Keen2020] KEEN, A. *Taming your Semantic Dragon. Operationalizing Knowledge Graphs with Multi-Model*, 2020.

- [Kehl2019] KEHL, S. *Marken- und Domänenübergreifendes Management Industrieller Produktdaten*, Wiesbaden: Gabler, 2019. 978-3-658-24448-4.
- [KeNS1992] KELLER, G., M. NÜTTGENS, and A.-W. SCHEER. Semantische Prozeßmodellierung auf der Grundlage "Ereignisgesteuerter Prozeßketten (EPK)". *Veröffentlichungen des Instituts für Wirtschaftsinformatik*, 1992 (89).
- [Kirc2020] KIRCHNER, E. *Werkzeuge und Methoden der Produktentwicklung. Von der Idee zum erfolgreichen Produkt*, Berlin: Springer Vieweg, 2020. 978-3-662-61762-5.
- [KoAk2010] KOCAR, V., and A. AKGUNDUZ. ADVICE: A virtual environment for Engineering Change Management. *Computers in Industry*, 2010 61(1), 15-28.
- [KoER2018] KOLDEWEY, C., J. ECHTERFELD, M. REILENDER, and J. GAUSEMEIER. Business Model Portfolio Planning for Smart Services. In: INTERNATIONAL SOCIETY FOR PROFESSIONAL INNOVATION MANAGEMENT (ISPIM)EDITOR, ed. *Proceedings of the ISPIM connects*, 2018.
- [KüKS2018] KÜGLER, P., P. KESTEL, C. SCHON, M. MARIAN, B. SCHLEICH, S. STAAB, and S. WARTZACK. ONTOLOGY-BASED APPROACH FOR THE USE OF INTENTIONAL FORGETTING IN PRODUCT DEVELOPMENT. *Proceedings of the DESIGN 2018 15th International Design Conference: Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Croatia; The Design Society, Glasgow, UK*, 2018, pp. 1595-1606.
- [Kurb2013] KURBEL, K. *Enterprise resource planning and supply chain management. Functions, business processes and software for manufacturing companies*, Heidelberg: Springer, 2013. Progress in IS. 3642315739.
- [Lee2002] LEE, E. A. Embedded Software. In: M. ZELKOWITZ, ed. *Advances in Computers: Elsevier textbooks*, 2002, pp. 55-95. 9780120121564.
- [LeEZ2012] LENTES, J., H. ECKSTEIN, and N. ZIMMERMANN. A Platform to Integrate Manufacturing Engineering and Product Lifecycle Management. *IFAC Proceedings Volumes*, 2012 45(6), 1071-1076.
- [LeIJ2015] LEHMANN, J., R. ISELE, M. JAKOB, A. JENTZSCH, D. KONTOKOSTAS, P.N. MENDES, S. HELLMANN, M. MORSEY, P. VAN KLEEF, S. AUER, and C. BIZER. DBpedia – A large-scale, multilingual knowledge base extracted from Wikipedia. *Semantic Web*, 2015 6(2), 167-195.
- [LeLH2011] LEE, C., C.S. LEEM, and I. HWANG. PDM and ERP integration methodology using digital manufacturing to support global manufacturing. *The International Journal of Advanced Manufacturing Technology*, 2011 53(1-4), 399-409.

- [LeNa2007] LESER, U., and F. NAUMANN. *Informationsintegration. Architekturen und Methoden zur Integration verteilter und heterogener Datenquellen*. 1. Aufl., Heidelberg: dpunkt-Verl., 2007. 978-3-89864-400-6.
- [LeRM2021] LEIDICH, J., P. ROBL, and J. MROWKA. Effektives Variantenmanagement auf Basis funktional substituierbarer Bauteilmerkmale. *Konstruktion*, 2021 73(6), 48-51.
- [Leve1966] LEVENSHTAIN, V. I. Binary codes capable of correcting deletions, insertions and reversals. *Soviet Physics Doklady*, 1966 10(8), 707-710.
- [LiRR2007] LI, Z., V. RASKIN, and K. RAMANI. A Methodology of Engineering Ontology Development for Information Retrieval. In: J.-C. BOCQUET, ed. *DS 42: Proceedings of ICED 2007, the 16th International Conference on Engineering Design, Paris, France, 28.-31.07.2007*, 2007.
- [Luck2014] LUCKE, D. M. *Ad hoc Informationsbeschaffung unter Einsatz kontextbezogener Systeme in der variantenreichen Serienfertigung*, 2014.
- [LuDa2019] LUTTERS, E., and R. DAMGRAVE. The development of Pilot Production Environments based on Digital Twins and Virtual Dashboards. *Procedia CIRP*, 2019 84, 94-99.
- [MaBS2015] MAHDISOLTANI, F., J. BIEGA, and F.M. SUCHANEK. YAGO3: A Knowledge Base from Multilingual Wikipedias. *CIDR*, 2015.
- [MaRe2009] MARQUES, P. A., and J.G. REQUEIJO. SIPOC: A Six Sigma Tool Helping on ISO 9000 Quality Management Systems. *3rd International Conference on Industrial Engineering and Industrial Management. XII Congreso de Ingeniería de Organización Barcelona-Terrassa, September 2nd-4th 2009*, 2009, pp. 1229-1238.
- [MaWS2019] MAZAROV, J., P. WOLF, J. SCHALLOW, F. NÖHRING, J. DEUSE, and R. RICHTER. Industrial Data Science in Wertschöpfungsnetzwerken. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 2019 114(12), 874-877.
- [MeGA2022] MEHLSTÄUBL, J., E. GADZO, A. ATZBERGER, and K. PÄTZOLD. Herausforderungen datengetriebener Methoden in der Produktentwicklung. *Konstruktion*, 2022 74(6), 60-66.
- [MeUh2012] MEIER, H., and E. UHLMANN. Hybride Leistungsbündel – ein neues Produktverständnis. In: H. MEIER AND E. UHLMANN, eds. *Integrierte Industrielle Sach- und Dienstleistungen*: Springer, Berlin, Heidelberg, 2012, pp. 1-21.
- [MeUK2005] MEIER, H., E. UHLMANN, and D. KORTMANN. Hybride Leistungsbündel: Nutzenorientiertes Produktverständnis durch interferierende Sach- und Dienstleistungen. *wt Werkstatttechnik online*, 2005 95(7/8), 528-532.
- [Milk1998] MIZOGUCHI, R., and M. IKEDA. Towards ontology engineering. *Journal-Japanese Society for Artificial Intelligence*, 1998 (13).

- [MoEE2022] MOLLAHASSANI, D., T. EICKHOFF, A. EIDEN, and J.C. GÖBEL. Wissensbasis zur Förderung von Innovationen Smarter PSS innerhalb eines Wertschöpfungsnetzwerkes. In: D. WILKE, W. KOCH, R. KAFFENBERGER, and S. DREISEITEL, eds. *Tag des Systems Engineering 2022. Tagungsband Paderborn, 16.-18. November 2022: BoD – Books on Demand, 2022*, pp. 162-166. 9783981880588.
- [Mogo2011] MOGO NEM, F. *Engineering networks. Holistic approach for multidisciplinary product lifecycle management*, Kaiserslautern: Technische Universität Kaiserslautern, 2011. Schriftenreihe VPE. Band 9. 978-3-941438-66-8.
- [MrLR2022] MROWKA, J., J. LEIDICH, and P. ROBL. Von der zeichnungsbasierten zur agilen artikelbasierten Stückliste. *Konstruktion*, 2022 74(3), 30-34.
- [MüHH2006] MÜLLER, D., J. HERBST, M. HAMMORI, and M. REICHERT. IT Support for Release Management Processes in the Automotive Industry. In: S. DUSTDAR, J.L. FIADAIRO, and A.P. SHETH, eds. *Business Process Management. BPM. Lecture Notes in Computer Science, vol. 4102*: Springer, Berlin, Heidelberg, 2006, pp. 368-377.
- [Mühl2008] MÜHLHÄUSER, M. Smart Products: An Introduction. In: M. MÜHLHÄUSER, A. FERSCHA, and E. AITENBICHLER, eds. *Constructing Ambient Intelligence*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2008, pp. 158-164. 978-3-540-85378-7.
- [MüLG2019] MÜLLER, P., K. LINDOW, S. GREGORZIK, and R. STARK. *Smart Industrial Products. Smarte Produkte und ihr Einfluss auf Geschäftsmodelle, Zusammenarbeit, Portfolios und Infrastrukturen*, 2019. 978-3-945406-23-6.
- [Müll2005] MÜLLER, A. *Wirtschaftlichkeit der Integration. Eine ökonomische Analyse der Standardisierung betrieblicher Anwendungssysteme*. 1. Aufl., Wiesbaden: Dt. Univ.-Verl., 2005. Gabler Edition Wissenschaft Markt- und Unternehmensentwicklung. 978-3-8350-0118-3.
- [Nort2011] NORTH, K. *Wissensorientierte Unternehmensführung. Wertschöpfung durch Wissen*. 5., aktualisierte und erweiterte Auflage, Wiesbaden: Gabler, 2011. SpringerLink Bücher. 978-3-8349-2538-1.
- [NoSM2020] NOLTE, V., T. SINDRAM, J. MAZAROV, and J. DEUSE. Industrial Data Science erfolgreich implementieren. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 2020 115(10), 734-737.
- [NoTa1997] NONAKA, I., and H. TAKEUCHI. *Die Organisation des Wissens. Wie japanische Unternehmen eine brachliegende Ressource nutzbar machen*, s.l.: Campus Frankfurt / New York, 1997.

- [O'MB2022] O'BRIEN, M. A., D. MOHALLY, G.P. BRASCHE, and A.G. SANFILIPPO. Huawei and International Data Spaces. In: B. OTTO, M. ten HOMPEL, and S. WROBEL, eds. *Designing Data Spaces. The Ecosystem Approach to Competitive Advantage*. Cham: Springer International Publishing; Imprint Springer, 2022, pp. 451-469. 978-3-030-93974-8.
- [OeJT2012] OESTERSÖTEBIER, F., V. JUST, A. TRÄCHTLER, F. BAUER, and S. DZIWOK. Model-Based Design of Mechatronic Systems by Means of Semantic Web Ontologies and Reusable Solution Elements. *Proceedings of the ASME 11th Biennial Conference on Engineering Systems Design and Analysis - 2012. Presented at the ASME 2012 11th Biennial Conference on Engineering Systems Design and Analysis, July 2 - 4, 2012, Nantes, France*. New York, NY: ASME, 2012, pp. 647-656. 978-0-7918-4486-1.
- [OtHW2019] OTTO, B., M. ten HOMPEL, and S. WROBEL. International Data Spaces. In: R. NEUGEBAUER, ed. *Digital Transformation*. Berlin, Heidelberg: Springer Vieweg, 2019, pp. 109-128. 978-3-662-58133-9.
- [OtJa2019] OTTO, B., and M. JARKE. Designing a multi-sided data platform: findings from the International Data Spaces case. *Electronic Markets*, 2019 29(4), 561-580.
- [OtST2019] OTTO, B., S. STEINBUSS, A. TEUSCHER, S. LOHMANN, S. BADER, P. BIRNSTIL, M. BÖHMER, G. BROST, J. CIRULLIES, A. EITEL, T. ERNST, S. GEISLER, J. GELHAAR, R. GUDE, C. HAAS, M. HUBER, C. JUNG, J. JÜRJENS, C. LANGE, D. LIS, C. MADER, N. MENZ, R. NAGEL, F. PATZER, H. PETTENPOHL, J. PULLMANN, C. QUIX, D. SCHULZ, J. SCHÜTTE, and ET AL. *Reference Architecture Model. Version 3.0*, 2019.
- [Pate2014] PATEL-SCHNEIDER, P. F. Analyzing Schema.org. In: T. TUDORACHE, C. WELTY, C. KNOBLOCK, D. VRANDEČIĆ, N. NOY, P. MIKA, A. BERNSTEIN, P. GROTH, K. JANOWICZ, and C. GOBLE, eds. *The semantic web - ISWC 2014. 13th International Semantic Web Conference, Riva del Garda, Italy, October 19-23, 2014; proceedings*. Cham: Springer, 2014, pp. 261-276. 978-3-319-11963-2.
- [Pätz2022] PÄTZOLD, K. Von einer Wissens- zu einer Kompetenzgesellschaft. *Konstruktion*, 2022 (4), 3.
- [Paul2016] PAULHEIM, H. Knowledge graph refinement: A survey of approaches and evaluation methods. *Semantic Web*, 2016 8(3), 489-508.
- [Pico2001] PICOT, A. *Die Grenzenlose Unternehmung. Information, Organisation und Management. Lehrbuch Zur Unternehmensführung Im Informationszeitalter*. 4th ed., Wiesbaden: Springer Gabler. in Springer Fachmedien Wiesbaden GmbH, 2001. 9783322921697.
- [PoHe2014] PORTER, M. E., and J.E. HEPPELMANN. How smart, connected products are transforming competition. *Harvard Business Review*, 2014 92, 64-88.

- [PoHe2015] PORTER, M. E., and J.E. HEPPELMANN. How smart, connected products are transforming companies. *Harvard Business Review*, 2015 114, 96-112.
- [Prob2012] PROBST, G. J. B. *Wissen managen. Wie Unternehmen ihre wertvollste Ressource optimal nutzen*. 7. Aufl., Wiesbaden: Springer Gabler, 2012. 978-3-8349-4562-4.
- [Pros2022] PROSTEP AG. *Digitale Durchgängigkeit zwischen Produktentwicklung und Produktion. Whitepaper*, 2022.
- [PrRU2022] PRINZ, W., T. ROSE, and N. URBACH. Blockchain Technology and International Data Spaces. In: B. OTTO, M. ten HOMPEL, and S. WROBEL, eds. *Designing Data Spaces. The Ecosystem Approach to Competitive Advantage*. Cham: Springer International Publishing; Imprint Springer, 2022, pp. 165-180. 978-3-030-93974-8.
- [Quir2015] QUIRMBACH, O. *Umfrage Änderungsmanagement 2015*, 2015.
- [Reic2022] REICHWEIN, A. Distributed Cross-Domain Link Creation for Flexible Data Integration and Manageable Data Interoperability Standards. *INSIGHT*, 2022 25(1), 38-41.
- [RiDB2019] RIESENER, M., C. DÖLLE, BECKER, and G. A. SCHUH. Framework for the Continuous Increase of Product Performance by Analyzing Product Usage Data. *2019 IEEE International Conference on Industrial Engineering and Engineering Management (IEEM)*, 2019, pp. 1084-1088. 2157-362X.
- [Rodd2019] RODDECK, W. *Einführung in die Mechatronik*. 6., überarbeitete und erweiterte Auflage, Wiesbaden: Springer Vieweg, 2019. Lehrbuch. 978-3-658-27775-8.
- [Rose2006] ROSENKRANZ, F. *Geschäftsprozesse. Modell- und computergestützte planung*, Berlin: Springer, 2006. 3540283439.
- [Ross2014] ROSSBERG, J. *Beginning application lifecycle management*, New York, NY: Apress Springer Science+Business Media, 2014. Books for professionals by professionals. 978-1-4302-5812-4.
- [RoWE2015] ROBINSON, I., J. WEBBER, and E. EIFREM. *Graph databases. [new opportunities for connected data]*. 2nd ed., Sebastopol, CA: O'Reilly, 2015. 978-1-491-93200-1.
- [SaBu2014] SAADATMAND, M., and A. BUCAIONI. OSLC Tool Integration and Systems Engineering -- The Relationship between the Two Worlds. In: R. RABISER, ed. *2014 40th EUROMICRO Conference on Software Engineering and Advanced Applications (SEAA 2014). Verona, Italy, 27 - 29 August 2014*. Piscataway, NJ: IEEE, 2014, pp. 93-101. 978-1-4799-5795-8.

- [Sche1992] SCHEER, A.-W. *Architektur integrierter Informationssysteme. Grundlagen der Unternehmensmodellierung*. Zweite, verbesserte Auflage, Berlin, Heidelberg: Springer Berlin Heidelberg, 1992. Springer eBook Collection Business and Economics. 9783642974038.
- [Schi2002] SCHICHTEL, M. *Produktdatenmodellierung in der Praxis*. [Elektronische Ressource], München: Hanser, 2002. 9783446218574.
- [ScMB2022] SCHWEITZER, G. M., S. MÖRSDORF, M. BITZER, and M. VIELHABER. Detection of Cause-Effect Relationships in Life Cycle Sustainability Assessment Based on an Engineering Graph. *Proceedings of the Design Society, 2022 2*, 1129-1138.
- [ScSe2020] SCHMELZER, H. J., and W. SESSELMANN. *Geschäftsprozessmanagement in der Praxis. Kunden zufrieden stellen, Produktivität steigern, Wert erhöhen*. 9., vollständig überarbeitete Auflage, München: Hanser, 2020. Hanser eLibrary. 9783446467095.
- [ScZe2013] SCHIRRMACHER, S., and M. ZENKER. Wissen, was gefragt ist: Wissensmanagement im Produktlebenszyklus. *phi - Produktionstechnik Hannover informiert*, 2013 14(2), 8-9.
- [Send2009] SENDLER, U. *Das PLM-Kompendium. Referenzbuch des Produkt-Lebenszyklus-Managements*, Berlin: Springer, 2009. Xpert.press. 978-3-540-87898-8.
- [Send2013] SENDLER, U. Industrie 4.0 - Beherrschung der industriellen Komplexität mit SysLM/Systems Lifecycle Management. In: U. SENDLER, ed. *Industrie 4.0. Beherrschung der industriellen Komplexität mit SysLM*. Berlin: Springer Vieweg, 2013, pp. 1-20. 978-3-642-36916-2.
- [ShCD2010] SHAFTO, M., M. CONROY, R. DOYLE, E. GLAESSGEN, C. KEMP, J. LEMOIGNE, and L. WANG. *DRAFT Modeling, Simulation, Information, Technology & Process Roadmap. Technology Area 11*, 2010.

- [SiAB2019] SIVASOTHY, P., J.C. AURICH, D. BECHEV, H. BREHM, G. BULUN, C. GLENSKE, M. GRETHLER, C. HERDER, J. IMWALLE, A. KEKSEL, P. KÖLSCH, R. MATTUKAT, G. WESSEL, B. SAUER, J. SEEWIG, F. ZEIHSEL, and V. ZIMMERMANN. Anwendungsfall GRIMME. In: J.C. AURICH, W. KOCH, P. KÖLSCH, and C.F. HERDER, eds. *Entwicklung datenbasierter Produkt-Service Systeme. Ein Ansatz zur Realisierung verfügbarkeitsorientierter Geschäftsmodelle*. Berlin: Springer Vieweg, 2019, pp. 109-168. 978-3-662-59642-5.
- [SiBW2022] SIDDHARTH, L., L.T.M. BLESSING, K.L. WOOD, and J. LUO. Engineering Knowledge Graph From Patent Database. *Journal of Computing and Information Science in Engineering*, 2022 22(2).
- [Sind2022] SINDERMANN, S. *SysLM Backbone. Ein Konzept zur durchgängigen Unterstützung des Model-Based Systems Engineering im System Lifecycle*. 1. Auflage, Düren: Shaker, 2022. 9783844084979.
- [Slau1993] SLAUGHTER, R. A. Looking for the real 'megatrends'. *Futures*, 1993 25(8), 827-849.
- [SpCH2017] SPEER, R., J. CHIN, and C. HAVASI. ConceptNet 5.5: An Open Multilingual Graph of General Knowledge. *AAAI 31 (2017)*.
- [Stac1973] STACHOWIAK, H. *Allgemeine Modelltheorie*, Wien: Springer, 1973. 3211811060.
- [StAT2020] STARK, R., R. ANDERL, K.-D. THOBEN, and S. WARTZACK. WiGeP-Positionspapier: „Digitaler Zwilling“. *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb*, 2020 115(special), 47-50.
- [Stek2016] STEKOLSCHIK, A. Engineering Change Management Method Framework in Mechanical Engineering. *IOP Conference Series: Materials Science and Engineering*, 2016 157, 12008.
- [StRR2018] STEINMETZ, C., A. RETTBERG, F.G.C. RIBEIRO, G. SCHROEDER, and C.E. PEREIRA. Internet of Things Ontology for Digital Twin in Cyber Physical Systems. *2018 VIII Brazilian Symposium on Computing Systems Engineering (SBESC)*: IEEE, 5 Nov. 2018 - 8 Nov. 2018, pp. 154-159. 978-1-7281-0240-5.
- [StWa2018] STEIMLE, T., and D. WALLACH. *Collaborative UX Design. Lean UX und Design Thinking: teambasierte Entwicklung menschzentrierter Produkte*. 1. Auflage, Heidelberg: dpunkt.verlag, 2018. 9783864905322.

- [StWH2013] STARK, R., R. WOLL, and H. HAYKA. Ontologiegestützte Informationsbereitstellung für Änderungsentscheidungen im Systems Engineering. In: M. MAURER AND S.-O. SCHULZE, eds. *Tag des Systems Engineering. Paderborn 7. – 9. November 2012: Zusammenhänge erkennen und gestalten*. München: Hanser Verlag, 2013, pp. 159-164. 9783446436039.
- [TeJD2013] TEICHRIB, H., V. JUST, S. DZIWOK, U. POHLMANN, T. SCHIERBAUM, and A. TRÄCHTLER. Modellbasierter Entwurf zweier kooperierender Delta-Roboter. In: M. MAURER AND S.-O. SCHULZE, eds. *Tag des Systems Engineering*. München: Carl Hanser Verlag GmbH & Co. KG, 2013, pp. 217-228. 978-3-446-43435-6.
- [ThWE2018] THARMA, R., R. WINTER, and M. EIGNER. An approach for the implementation of the digital twin in the automotive wiring harness field. *Proceedings of the DESIGN 2018 15th International Design Conference: Faculty of Mechanical Engineering and Naval Architecture, University of Zagreb, Croatia; The Design Society, Glasgow, UK, 2018*, pp. 3023-3032.
- [Tied2006] TIEDEMANN, M. *Konzepte und Technologien zur Anwendungs-Integration*. Dissertation, Lübeck, 2006.
- [Turk2014] TURKI, T. *Bedeutung von Erfahrungswissen in der Produktentwicklung und Ansätze zu dessen Evaluierung und Transfer am Beispiel studentischer Gruppen*, 2014.
- [UhMe2017] UHLMANN, E., and H. MEIER. Produktverständnis im Wandel. In: H. MEIER AND E. UHLMANN, eds. *Industrielle Produkt-Service Systeme*: Springer Berlin Heidelberg, 2017, pp. 1-16. 9783662480175.
- [VDA2010] VDA 4965, *ECM Recommendation Part 0 (ECM)*: Sasig Publications, 2010a.
- [VDA2010] VDA. *Engineering Change Management Reference Process. covering VDA 4965 V3.0, 2010b*.
- [VDI/2020] VDI/VDE 2206, *Entwicklung cyber-physischer mechatronischer Systeme (CPMS)*, Berlin: Beuth, September 2020.
- [VDI2004] 2206, *Entwicklungsmethodik für mechatronische Systeme*, Berlin: Beuth, 2004.
- [VDI2019] VDI 2221 Blatt 1, *Entwicklung technischer Produkte und Systeme - Modell der Produktentwicklung*, Berlin: Beuth, 2019.
- [Völk2006] VÖLKER, R. *Wissensmanagement im Innovationsprozess*, Dordrecht: Springer, 2006. 978-3-7908-1691-4.
- [WaXu2008] WANG, C., and L. XU. Parameter mapping and data transformation for engineering application integration. *Information Systems Frontiers*, 2008 10(5), 589-600.

- [WeWG2015] WEILKIENS, T., C. WEISS, A. GRASS, and K. N. DUGGEN. *Basiswissen Geschäftsprozessmanagement. Aus- und Weiterbildung zum OMG-Certified Expert in Business Process Management 2 (OCEB2) - Fundamental Level. 2.*, aktualisierte Aufl., Heidelberg: dpunkt, 2015. 9783864901935.
- [WhMi2008] WHITE, S. A., and D. MIERS. *BPMN modeling and reference guide. Understanding and using BPMN ; develop rigorous yet understandable graphical representations of business processes*, Lighthouse Point, Fla.: Future Strategies Inc, 2008. 9780977752720.
- [Wies1990] WIESE, H. *Netzeffekte und Kompatibilität. Ein theoretischer und simulationsgeleiteter Beitrag zur Absatzpolitik für Netzeffekt-Güter*, Stuttgart: Poeschel, 1990. Schriftenreihe der Wissenschaftlichen Hochschule für Unternehmensführung Koblenz Forschung. 2. 3791005340.
- [WoHS2015] WOLL, R., H. HAYKA, and R. STARK. Ontologiebasierte Datenintegration für das Modellbasierte Systems Engineering. In: M. MAURER, S.-O. SCHULZE, and J. ABULAWI, eds. *Tag des Systems Engineering. Bremen, 12. - 14. November 2014*. München: Hanser, 2015, pp. 33-42. 9783446443761.
- [YuBo2007] YUJIAN, L., and L. BO. A normalized Levenshtein distance metric. *IEEE transactions on pattern analysis and machine intelligence*, 2007 29(6), 1091-1095.
- [Zage2006] ZAGEL, M. *Übergreifendes Konzept zur Strukturierung variantenreicher Produkte und Vorgehensweise zur iterativen Produktstruktur-Optimierung*. Als Manuskript gedruckt, Kaiserslautern: Verlag Technische Universität Kaiserslautern, 2006. Schriftenreihe VPE / Lehrstuhl für Virtuelle Produktentwicklung, Technische Universität Kaiserslautern. Band 1. 978-3-939432-26-5.
- [Zarn2007] ZARNEKOW, R. *Produktionsmanagement von IT-Dienstleistungen. Grundlagen, Aufgaben und Prozesse*, Berlin, Heidelberg: Springer-Verlag Berlin Heidelberg, 2007. Business Engineering. 978-3-540-47458-6.
- [ZeSc2020] ZEIMETZ, T., and R. SCHENKEL. Sample Driven Data Mapping for Linked Data and Web APIs. In: M. D'AQUIN, S. DIETZE, C. HAUFF, E. CURRY, P. CUDRE MAUROUX, and M. d' AQUIN, eds. *CIKM '20. Proceedings of the 29th ACM International Conference on Information & Knowledge Management : October 19–23, 2020, Virtual Event, Ireland*. New York, NY: The Association for Computing Machinery, 2020, pp. 3481-3484. 9781450368599.
- [ZhMø2014] ZHANG, W., and B. MØLLER-PEDERSEN. Modeling of tool integration resources with OSLC support. In: L. FERREIRA PIRES, ed. *2014 2nd International Conference on Model-Driven Engineering and Software Development (MODELSWARD). 7 - 9 Jan. 2014, Lisbon, Portugal*. Piscataway, NJ: IEEE, 2014, pp. 99-110. 9789897580079.

[ZhTL2018] ZHAN, Y., K.H. TAN, Y. LI, and Y.K. TSE. Unlocking the power of big data in new product development. *Annals of Operations Research*, 2018 270(1-2), 577-595.

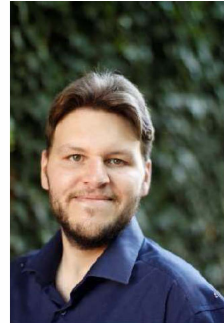
Betreute studentische Arbeiten

| Titel | Name | Jahr |
|---|--------------------|-------------|
| Der Digitale Zwilling – Eine Fallstudie zu den Anforderungen an die Erhebung und Verwertung von Daten | Jens Schlehahn | 2017 |
| Der Digitale Zwilling – Literaturanalyse und Anwendungsbeispiel | Hefron Hesso | 2019 |
| Übersicht über Prozesse, Methoden und Tools des Änderungsmanagements | Eda Gülay | 2019 |
| Funktionsanalyse des PDM-System CIMDatabase und Aufbau einer Wissensdatenbank zur Einführung und Nutzung | Hoshang Vekariya | 2019 |
| Vergleichende Analyse der PLM-Systeme Siemens Teamcenter, PTC Windchill und Aras Innovator | Erkan Cekic | 2019 |
| Erarbeitung eines Prozesses und Integration eines Tools zur Verfolgung von Entwicklungsfortschritten und fachbereichsübergreifender Kollaboration bei der geometrischen Absicherung in der Automobilentwicklung | Dominique Wick | 2019 |
| Entwicklung eines graphbasierten Mechanismus für das Mapping von PLM-Datenmodellen | Marius Holdstein | 2020 |
| Studie zur Informationsaggregation und Datenanalyse in der Produktentwicklung und im Engineering | Christian Walawski | 2021 |
| Aufbau eines Digitalen Zwillings zur Toleranzsicherung und Steigerung der Produktqualität | Mehmet Onur Can | 2021 |
| Aufbau eines Service zur Anzeige und zum Mapping von Metadaten, sowie zum Auswählen von Analysedaten in einem semantischen Engineering-Repository | Kevin Sellers | 2022 |

Lebenslauf

Persönliche Daten

Vor- und Nachname Andreas Eiden
Geburtsort Hermeskeil
Staatsangehörigkeit: Deutsch



Beruflicher Werdegang

Seit 08/2023 Co-Gründer der Inline Process Solutions GmbH
10/2022 – 11/2024 TU Kaiserslautern/RPTU Kaiserslautern-Landau
Lehrstuhl für Fluidverfahrenstechnik
Wissenschaftlicher Mitarbeiter EXIST Forschungstransfer
04/2016 – 09/2022 TU Kaiserslautern
Lehrstuhl für Virtuelle Produktentwicklung
Wissenschaftlicher Mitarbeiter
09/2015 – 03/2016 TU Kaiserslautern
Dekanat Maschinenbau und Verfahrenstechnik
Studiengangmanager Maschinenbau und Verfahrenstechnik
07/2013 – 05/2014 KSB AG, Frankenthal
Arbeitsvorbereitung ETA
Industriepraktikum

Schulische und universitäre Ausbildung

08/2008 – 06/2015 TU Kaiserslautern
Studium Diplom Maschinenbau und Verfahrenstechnik
Vertiefung in Produktionstechnik/Automatisierungstechnik
01/2013 – 06/2013 NTNU Technisch-Naturwissenschaftliche Universität Norwegens,
Trondheim, Norwegen
Auslandstudium im Rahmen des Erasmus-Austauschs
08/1999 – 03/2008 Gymnasium Hermeskeil
Abitur, Leistungskurse Mathematik, Physik und Sozialkunde

Berichte aus der Virtuellen Produktentwicklung (VPE)

bereits veröffentlicht wurden

Band 1 **Damun Mollahassani Madjdabadi**
Thema Methodik zur wissensbasierten Unterstützung von Innovationsprozessen
in kollaborativen Wertschöpfungsnetzwerken
Jahr 2024
ISBN 978-3-95974-236-8
ISSN 3051 - 9284

Band 2 **Andreas Eiden**
Thema Knowledge Graph basiertes Assistenzsystem zur Unterstützung der
Datenanalyse für das Engineering Smarter Produkte
Jahr 2025
ISSN 3051 - 9284

VPE

ISSN 3051 - 9284