

Otto-von-Guericke-Universität Magdeburg
Fakultät für Elektrotechnik und Informationstechnik

Masterarbeit



Konzepte zur Umsetzung von mehrwertbringenden Use Cases
in Produktionsanlagen mittels I4.0 Verwaltungsschalen

eingereicht: 20. Juni 2022

verteidigt: 11. Juli 2022

von: Melanie Stolze
geboren am 19. Dezember 1996
in Heilbad Heiligenstadt

Kurzreferat

Eines der Ziele der Automatisierung ist es, Produktionsprozesse flexibel, ökonomisch rentabel und ressourcenschonend zu gestalten. Mit der Verknüpfung der Verwaltungsschale und ihrem zugehörigen Asset zu einer Industrie 4.0-Komponente haben Forschende ein Mittel geschaffen, das nicht nur unternehmensintern sondern auch übergreifend die digitale Vernetzung von Betriebsmitteln (Assets) ermöglicht. Der Anwendungsbereich der Verwaltungsschale bei klein- und mittelständischen Unternehmen ist dennoch überschaubar. Gründe dafür sind möglicherweise die Fragen nach dem Mehrwert einer Verwaltungsschale und deren Implementierungskosten.

Um das Potential der Industrie 4.0-Komponente aufzuzeigen, beschäftigt sich diese Masterarbeit mit Use Cases aus der Fertigungsindustrie. Es werden effektivitätssteigernde Use Cases beschrieben und verschiedene praxisnahe Szenarien aus einem ausgewählten Use Case untersucht.

Bei dem ausgewählten Use Case handelt es sich um die Dokumentationspflege bei einem Gerätetausch mit Hilfe von Verwaltungsschalen. Für die Umsetzung des Use Case werden die Beziehungen zwischen den verschiedenen Assets bzw. ihren Verwaltungsschalen betrachtet. Sie bilden das Geflecht, aus dem die gesamte Dokumentation einer Produktionsressource hervorgeht.

Weiterhin spielt die Analyse der erforderlichen Schnittstellenoperationen für den Zugriff auf die Verwaltungsschalen eine wichtige Rolle. Die Schnittstellen ermöglichen den externen Zugriff auf die Daten einer Verwaltungsschale und somit die Aktualisierung der Dokumentation bei einem Gerätetausch.

Die Untersuchungen in dieser Arbeit stützen sich auf die Gegenüberstellung verschiedener Lösungsansätze mit ihren Vor- und Nachteilen und ermöglichen die Auswahl der bestmöglichen Lösung. Die Konzepte geben Aufschluss über die Verwendung der Verwaltungsschalen-Modellelemente und deren Schnittstellen zur Realisierung des Use Case. Sowohl die in den verschiedenen Konzepten erstellten technologieneutralen als auch technologiespezifischen Sequenzdiagramme geben Einsicht über die Verwendung der Verwaltungsschale.

Zur Veranschaulichung der erworbenen Ergebnisse werden die am besten bewerteten Konzepte auf den LIA-I4.0-Demonstrator der Otto-von-Guericke-Universität Magdeburg angewendet. Im Vordergrund steht die prinzipielle Realisierung des Datenaustausches zwischen den am Gerätetausch beteiligten Interaktionsteilnehmern zur Umsetzung der Dokumentationspflege.

Abstract

One of the goals of automation is the development of flexible, economically profitable and resource-saving manufacturing processes. By linking the Asset Administration Shell and its associated asset to form an Industry 4.0 component, researchers have established a tool that enables digital networking of assets not only within the company but also across companies. Nevertheless, the scope of application of the Asset Administration Shell at small and medium-sized companies is manageable. Reasons for this may be the questions regarding the added value of Asset Administration Shells and its implementation costs.

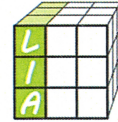
To demonstrate the potential of the Industry 4.0 component, this master thesis deals with use cases from the manufacturing industry. Effectiveness-increasing use cases are described and various practical scenarios from a selected use case are explored.

The selected use case involves the maintenance of documentation for a device replacement with the help of Asset Administration Shells. For the implementation of the use case the relationships between the various assets and their Asset Administration Shells are considered. They form the network from which the entire documentation of a production resource emerges.

Additionally, the analysis of the required interface operations for accessing the Asset Administration Shells plays an important role. The interfaces enable the external access to the data of Asset Administration Shells and thus the updating of the documentation in the event of a device replacement.

The investigations in this thesis are based on the comparison of different solution approaches with their advantages and disadvantages and enable the selection of the best possible solution. The concepts provide information about the use of the Asset Administration model elements and their interfaces for realizing the use case. Both the technology-neutral and technology-specific sequence diagrams created in the different concepts provide insights into the use of the Asset Administration Shell.

To illustrate the acquired results, the best evaluated concepts are applied to the LIA-I4.0 demonstrator of Otto-von-Guericke-University Magdeburg. The focus is on the principal realization of the data exchange between the interaction participants involved in the device exchange for the implementation of the documentation maintenance.



Masterarbeit

für Frau Melanie Stolze

Thema: *Konzepte zur Umsetzung von mehrwertbringenden Use Cases in Produktionsanlagen mittels I4.0 Verwaltungsschalen*

Aufgabenstellung:

Die Betriebsmittel (Assets) einer Maschine oder Anlage werden während des Engineerings zu einem Produktionsprozess verbunden. Diese Prozesse müssen flexibler werden, damit die Auswahl von Assets für eine bestimmte Anwendungsaufgabe im laufenden Betrieb definiert werden kann. Im Zuge der digitalen Transformation industrieller Produktionsprozesse werden physische Assets mit Digitalen Zwillingen zu Softwarekomponenten verbunden. Die Verwaltungsschale (Asset Administration Shell - AAS) ist ein standardisierter digitaler Zwilling, der allgemeine Daten zu den Assets bietet. Diese AAS müssen einzeln oder gemeinschaftlich die Use Cases des Industrie 4.0-Konzeptes erbringen, um dadurch einen Mehrwert im Engineering oder im betriebsnahen Bereich zu erbringen. Das Zusammenwirken der vielfältigen Mittel der AAS bezogen auf konkrete, effektivitätssteigernde Use Cases sollen in dieser Masterarbeit untersucht und detailliert ausgearbeitet werden.

Im Einzelnen sind dafür die folgenden Teilaufgaben zu bearbeiten:

- Literaturrecherche zu Use Cases, die die besonderen charakteristischen Merkmale des I4.0 Konzeptes widerspiegeln und Auswahl und Detaillierung eines oder mehrerer besonders mehrwertversprechender Use Cases
- Konzept für die Umsetzung eines ausgewählten Use Cases und daraus abgeleiteten Szenarien mit Mitteln der I4.0 Verwaltungsschale (AAS) unter besonderer Betrachtung der dafür erforderlichen Schnittstellen und Mittel des Verwaltungsschalen Metamodells
- Herausarbeitung von technologischen Best Practice Lösungsansätzen für die Umsetzung
- Das ausgewählte Beispiel des Use Cases sollte zu einer konkreten Anlage Bezug nehmen, an denen die Szenarien prinzipiell umgesetzt werden können
- Dokumentation der Ergebnisse

Magdeburg, 12.01.2021

Tag der Ausgabe: 19.01.2022


Tag der Abgabe: 22.06.2022

Erstprüfer: Prof. Dr.-Ing. Christian Diedrich

Zweitprüfer: M. Sc. Alexander Belyaev

Betreuer: M. Sc. Chris Urban


Prof. Dr.-Ing. Christian Diedrich
Aufgabensteller


Prof. Dr.-Ing. Roberto Leidhold
Vorsitzender des Prüfungsausschusses

Eidesstattliche Erklärung

Hiermit versichere ich, die vorliegende Arbeit selbstständig und unter ausschließlicher Verwendung der angegebenen Literatur und Hilfsmittel erstellt zu haben.

Die Arbeit wurde bisher in gleicher oder ähnlicher Form keiner anderen Prüfungsbehörde vorgelegt und auch nicht veröffentlicht.

Melanie Stolze

Magdeburg, den 20. Juni 2022

Inhaltsverzeichnis

1	Einleitung	9
1.1	Motivation	9
1.2	Zielstellung	10
1.3	Vorgehen - Forschungsmethodik	11
2	Untersuchung nutzbringender Use Cases für I4.0	14
2.1	Use Case Cluster - Auftragsgesteuerte Produktion	24
2.1.1	Herstellung von individualisierten Produkten	25
2.1.2	Unternehmensinterne Logistik	29
2.2	Use Case Cluster - Wandlungsfähige Fabrik	31
2.2.1	Modularisierung von Produktionssystemen	32
2.2.2	Plug and Produce für Feldgeräte	35
2.3	Verwaltung von Assets	38
2.4	Value-Based Services für Produktionsressourcen	45
2.5	Use Case Cluster - IT-Infrastruktur und Software	49
2.5.1	Dokumentationspflege bei Gerätetausch	50
2.5.2	Human-Machine-Interface	51
2.6	Kreislaufwirtschaft	54
3	konzeptionelle Umsetzung des Use Case „Dokumentationspflege bei Gerätetausch“	58
3.1	Systemarchitektur - Repositories zur Speicherung von reaktiven Verwaltungsschalen	59
3.2	Szenario 1 - Gerätetausch durch Instandhaltungspersonal	66
3.3	Szenario 2 - Gerätetausch durch Hersteller des Produktionsmoduls	70
3.4	Szenario 3 - vollautomatisierter Gerätetausch durch Roboter	75
4	Untersuchung technologischer Best-Practice Lösungen	79
4.1	Konzepte für die Modellierung von Beziehungen zwischen Verwaltungsschalen	80
4.2	Elemente des Verwaltungsschalen-Metamodells zur Modellierung von Beziehungen	84
4.3	Konzepte für die Realisierung der Beziehungsmodellierung in Verwaltungsschalen	88
4.4	Datenaustausch über die Schnittstelle der Verwaltungsschale	98

5	Prinzipielle Umsetzung des Use Case am LIA-I4.0-Demonstrator	118
5.1	Ausgangssituation LIA-I4.0-Demonstrator	118
5.2	Beziehungsmodellierung im AASX-Package-Explorer	120
5.3	Prinzipieller Datenaustausch im LIA-I4.0-Demonstrator	129
6	Zusammenfassung	137
7	Ausblick	139
	Literatur	141
A	Anhang A - VWS-Teilmodell-Templates der IDTA	144

Abkürzungsverzeichnis

AAS	Asset Administration Shell
AG2	Arbeitsgruppe 2
AGP	Auftragsgesteuerte Produktion
API	Application Programming Interface
BMK	Betriebsmittelkennzeichen
BoM	Bill of Material
DIN	Deutsches Institut für Normung
DKE	Deutsche Kommission Elektrotechnik
ERP	Enterprise Resource Planning
HMI	Human Machine Interface
HTTP	Hypertext Transfer Protocol
I4.0	Industrie 4.0
IDTA	Industrial Digital Twin Association e. V.
IEC	International Electrotechnical Commission
IHS	Instandhaltungssystem
IIoT	Industrial Internet of Things
IRI	Internationalized Resource Identifier
KMU	klein- und mittelständische Unternehmen
KPI	Key Performance Indicator
M2M	Machine-to-Machine
MES	Manufacturing Execution System
MQTT	Message Queuing Telemetry Transport

OPC UA Open Platform Communications Unified Architecture

PLM Product Lifecycle Management

PSLM Product System Lifecycle Managment

REST Representational State Transfer

RFID Radio Frequency Identification

SCM Supply Chain Management

SPS speicherprogrammierbare Steuerung

TC Technical Commitee

TM Teilmodell

UML Unified Modelling Language

VBS Value-Based-Service

VDMA Verband Deutscher Maschinen- und Anlagenbau e. V.

VWS Verwaltungsschale

ZVEI Zentralverband für Elektrotechnik- und Elektronikindustrie

Abbildungsverzeichnis

1.1	Forschungsmethodik der Masterarbeit	13
2.1	Wertschöpfungsketten in Industrie 4.0 bezogen auf produzierende Unternehmen	15
2.2	Technische Perspektive des Use Case „Herstellung von individualisierten Produkten“	26
2.3	Use Case Diagramm - Herstellung von individualisierten Produkten	27
2.4	Technische Perspektive des Use Case „Unternehmensinterne Logistik“	30
2.5	Use Case Diagramm - Unternehmensinterne Logistik	31
2.6	Technische Perspektive des Use Case „Modularisierung von Produktionssystemen“	34
2.7	Use Case Diagramm - Modularisierung von Produktionssystemen	35
2.8	Use Case Diagramm - Plug and Produce für Feldgeräte	37
2.9	Technische Perspektive des Use Case „Assetmanagement“	41
2.10	Use Case Diagramm - Assetmanagement	42
2.11	Technische Perspektive des Use Case „Value-Based Services für Produktionsressourcen“	47
2.12	Use Case Diagramm - Value-Based Services für Produktionsressourcen	49
2.13	Aktueller Stand der Dokumentationspflege bei dem Use Case „Dokumentationspflege bei Gerätetausch“	50
2.14	Technische Perspektive des Use Case „Human-Machine-Interface“	53
2.15	Use Case Diagramm - Human Machine Interface	54
2.16	Use Case Diagramm - Kreislaufwirtschaft	57
3.1	Bereitstellung der VWS durch den Komponentenhersteller	61
3.2	Bereitstellung der VWS durch den Maschinenhersteller	62
3.3	Speicherung der VWS durch das produzierende Unternehmen	63
3.4	Repositories zur Speicherung von reaktiven Verwaltungsschalen	65
3.5	Use Case Diagramm Szenario 1 - Gerätetausch durch Instandhaltungspersonal	67
3.6	Sequenzdiagramm Szenario 1 (Teil 1) - Auftrag aufgeben und VWS identifizieren	68
3.7	Sequenzdiagramm Szenario 1 (Teil 2) - Überprüfung der techn. Daten und Ersetzung der VWS	69

3.8	Use Case Diagramm Szenario 2 - Gerätetausch durch den Hersteller des Produktionsmoduls	71
3.9	Sequenzdiagramm Szenario 2 (Teil 1) - Zugriff anfragen und Auftrag aufgeben	72
3.10	Sequenzdiagramm Szenario 2 (Teil 2) - Verwaltungsschalen identifizieren .	73
3.11	Sequenzdiagramm Szenario 2 (Teil 3) - Überprüfung der techn. Daten und Ersetzung der VWS	74
3.12	Sequenzdiagramm Szenario 2 (Teil 4) - Zugriff wieder zurückgeben	75
3.13	Use Case Diagramm Szenario 3 - Gerätetausch durch Roboter	77
3.14	Sequenzdiagramm Szenario 3 - Gerätetausch durch Roboter	78
4.1	Möglichkeiten zur Modellierung von Beziehungen zwischen Verwaltungsschalen	82
4.2	Spezifikation des AAS-Element - Entity	85
4.3	Spezifikation des AAS-Element - Reference-Element	85
4.4	Spezifikation des AAS-Element - Relationship-Element	86
4.5	Übersicht über die Submodels und Submodel Element Collections einer zusammengesetzten Komponente	87
4.6	Ausstattung einer AAS mit Views	87
4.7	Spezifikation des AAS-Element - View	87
4.8	Modellierung von Beziehungen in der AAS mit den gegebenen AAS-Elementen (Teil 1) (in Anlehnung an [27, S. 27–31])	89
4.9	Modellierung von Beziehungen in der AAS mit den gegebenen AAS-Elementen (Teil 2) (in Anlehnung an [27, S. 27–31])	91
4.10	Beispiel für die prinzipielle Umsetzung der Beziehungsmodellierung	97
4.11	Prinzipielles Vorgehen für den Eingriff in den AAS-Aufbau zur Dokumentationspflege bei einem Gerätetausch	99
4.12	HTTP/REST-spezifische Zugriffe auf die AAS-API zur Identifizierung einer AAS	104
4.13	HTTP/REST-spezifisches Sequenzdiagramm zur Umsetzung der Interaktion „GET Beziehung“	107
4.14	HTTP/REST-spezifisches Sequenzdiagramm für die Speicherung einer Entity	109
4.15	HTTP/REST-spezifisches Sequenzdiagramm für die Einbindung von Entity, Relationship- oder Reference-Element in eine AAS	111
4.16	HTTP/REST-spezifisches Sequenzdiagramm für das Aktualisieren eines Reference-Elements	113
4.17	HTTP/REST-spezifisches Sequenzdiagramm für das Aktualisieren einer Entity	114
4.18	HTTP/REST-spezifisches Sequenzdiagramm für das Hinzufügen eines Elements in einen View	115

4.19	HTTP/REST-spezifisches Sequenzdiagramm für das Löschen eines Elements aus der AAS und dem dazugehörigen View	117
5.1	Ausgangsmodell des LIA-I4.0-Demonstrators	119
5.2	Übersicht über die Implementierung des konzeptionellen AAS-Aufbaus im AASX-Package-Explorer	122
5.3	Detaillierte Darstellung der Daten für ein Relationship-Element und der mit diesem in Verbindung stehenden AAS-Elemente	123
5.4	Detaillierte Darstellung der Daten für ein Reference-Element und der mit diesem in Verbindung stehenden AAS-Elemente	125
5.5	Detaillierte Darstellung der Daten für Submodel, Submodel Element Collection und View	126
5.6	Technologiespezifisches Sequenzdiagramm für das Aktualisieren eines vorgegebenen Reference-Elements am Beispiel des LIA-I4.0-Demonstrators . .	130
5.7	JSON-Codeausschnitt der Metainformationen zum Relationship-Element <i>PLC1ControlsDrillingMachine1</i> aus der AAS <i>PLC1</i>	131
5.8	JSON-Codeausschnitt der Entity <i>DrillingMachine1</i> aus der AAS <i>PLC1</i> . .	132
5.9	Technologiespezifisches Sequenzdiagramm zur Identifizierung der AAS <i>DrillingMachine1</i> des LIA-I4.0-Demonstrators	133
5.10	JSON-Codeausschnitt der Metainformationen von der AAS <i>DrillingMachine1</i> und deren AAS Descriptor	133
5.11	JSON-Codeausschnitt des View <i>RelationshipsFunctional</i> der AAS <i>DrillingMachine1</i>	134
5.12	JSON-Codeausschnitt des Reference-Element <i>PLC1ControlsDrillingMachine1</i> der AAS <i>DrillingMachine1</i>	135

Tabellenverzeichnis

2.1	Übersicht der nutzbringenden Use Cases	22
4.1	Vor- und Nachteile der unterschiedlichen Abbildungen einer Hierarchie in Verwaltungsschalen	83
4.2	Vor- und Nachteile der Konzepte zur Umsetzung der Beziehungsmodellierung	92
4.3	Merkmale und Rechenaufwand zur Identifizierung der AAS eines Geräts über HTTP/REST-API	103
5.1	Modellierung von Beziehungen in einer Tabelle	120
A.1	Auflistung der durch die IDTA entwickelten bzw. noch zu entwickelnde Teilmodell-Templates	144

1 Einleitung

1.1 Motivation

Die Verwaltungsschale (VWS) bietet als digitaler Zwilling die Möglichkeit, Daten eines Betriebsmittels (Assets) zentral zu sammeln. Bei den Assets kann es sich um physisch vorhandene Objekte (z.B. Produktionsanlagen) und virtuelle Objekte (z.B. Software) handeln. Zu den Daten, die für z.B. physisch vorhandene Assets gespeichert werden, gehören nicht nur das digitale Typenschild oder Messwerte von Sensoren. Vielmehr erfolgt die Datenerfassung in der VWS über Bereiche wie geschäftliche Rahmenbedingungen, mechanische Aspekte, bereitgestellte Funktionalitäten und Leistungsfähigkeit sowie den Standort des Assets hinaus [1, S. 26 f.].

Die zentral gespeicherten Daten können für Zwecke der Datenanalyse oder zur Kommunikation mit anderen VWS genutzt werden. Die Plattform Industrie 4.0 (I4.0) gibt den Entwicklern dafür Konzepte an die Hand, mit denen sie die VWS in Unternehmen erfolgreich implementieren können. Dabei muss sowohl die semantische als auch die syntaktische Interoperabilität zwischen den VWS als I4.0-Komponenten gewährleistet sein, um spätere übergreifende Vernetzungen von VWS zu ermöglichen.

Trotz der vielen nutzbringenden Möglichkeiten, die die VWS bietet, ist ihr Einsatz in der Praxis noch nicht flächendeckend. Das kann möglicherweise auch an offenen Fragen der Entwickler und Anwender liegen, bezüglich der standardisierten Umsetzung und des Mehrwertes einer solchen I4.0-Komponente.

Fragen wie:

Wozu brauche ich eine VWS und in welchen Bereichen bringt sie mir einen Mehrwert?

Wie gestaltet sich der Zugriff auf die Daten der VWS und welche Daten und Schnittstellen werden wie genutzt?

treiben mitunter die Unternehmen und Entwickler in ihrem Handeln voran.

Im Allgemeinen ist die Motivation hinter dieser Masterarbeit, den Stakeholdern anhand eines ausgewählten Use Cases die Konzepte für teilautomatisierte Interaktionen mit sowohl proaktiven als auch reaktiven VWS zu liefern, die zudem noch einen Mehrwert bieten.

1.2 Zielstellung

Diese Masterarbeit will Antworten auf die oben gestellten Fragen finden. Zum einem warum der Einsatz einer VWS in bestimmten Bereichen der Fertigungstechnik nutzbringend ist und wo dieser stattfindet. Zum anderen, wie die Datenstruktur der Verwaltungsschale und deren Elemente durch ihre Schnittstellen zu beeinflussen sind und wie die prinzipielle Überführung des Datenzugriffs an einem Demonstrator in die Realität stattfindet. Mit dem Zugriff auf eine Schnittstelle wird der Datenaustausch zwischen VWS und anderen Systemen in einem Kommunikationsnetz ermöglicht.

Das Vorgehen zur Erreichung des beschriebenen Gesamtziels gliedert sich in folgende Schwerpunkte:

- Zu Beginn verschafft eine systematische Literaturrecherche nach Anwendungsfällen (engl. Use Cases) im Bereich der Fertigungsindustrie einen Überblick, in denen vor allem der Einsatz von VWS nutzbringend ist. Die Recherche bezieht sich auf zukunftsorientierte Use Cases, da die VWS in der Praxis bisher nicht weit verbreitet ist und die Umsetzung der Use Cases zur Zeit nur an Prototypen stattfindet. Die Entscheidung, welcher Use Case in diese Arbeit aufgenommen wird beruht auf festgelegten Gesichtspunkten. Diese geben Aufschluss über das Aufwand/Nutzen-Verhältnis und den Einsatzzweck der VWS. Die Sinnhaftigkeit des Einsatzes der VWS gegenüber anderen I4.0-Technologien für die Realisierung der Use Cases soll mit einbezogen werden.
- Nach der Ermittlung der Use Cases verschafft eine nähere Beschreibung der Funktion Einblicke in diese. Verweise auf Forschungsgruppen, die die Umsetzung einzelner Use Cases unter Verwendung von VWS an Prototypen testen, unterstreichen die Rolle der I4.0-Komponente in der diskreten Fertigung.
- Die nähere Betrachtung einer der Use Cases soll dessen Einsatzmöglichkeiten aufzeigen. Dafür werden Szenarien von dem Use Case abgeleitet, die wiederkehrend in den Wertschöpfungsketten der Fertigungsindustrie auftreten. Die Struktur der Szenarien in Bezug auf die Interaktionen zwischen denen am Szenario teilnehmenden Rollen und den dafür benötigten Elementen der VWS spiegeln technologieneutrale Sequenzdiagramme wider.
- Mit der Gestaltung verschiedener Konzepte, zur Spezifizierung der entworfenen Sequenzdiagramme, wird die Einbindung der VWS-Schnittstelle und der einzelnen VWS-Elemente gezeigt. Die Konzepte werden auf ihre Vor- und Nachteile analysiert und abschließend die Best-Practice Lösung in technologiespezifischen Sequenzdiagrammen dargestellt.

- Zum Ende wird eines der abgeleiteten Szenarien ausgewählt, das zur Veranschaulichung der Best-Practice Lösung dient. Als Praxisbezug für die Umsetzung dient der LIA-I4.0-Demonstrator der Otto-von-Guericke-Universität Magdeburg (kurz OvGU Magdeburg). Er soll den Aufwand zur Integration des ausgewählten Use Case, mit Bezug auf die in der Praxis vorhandenen Gegebenheiten, verdeutlichen.

Schlussendlich richtet sich diese Masterarbeit an Leser, die nach einer Übersicht über nutzbringende Use Cases zur Umsetzung von Industrie 4.0 mittels VWS suchen. Der Leser bekommt damit eine Vorstellung von den Einsatzmöglichkeiten einer VWS. Mit den erstellten Konzepten wird ihm gezeigt, wie ein Use Case prinzipiell in die Praxis umzusetzen ist. Das heißt zum Beispiel:

- das Suchen, Ändern, Hinzufügen und Löschen von Daten einer bereits existierenden Verwaltungsschale
- das Anstoßen von Operationen in Verwaltungsschalen über die API (kurz für Application Programming Interface) des VWS-Metamodells
- das Modellieren von Beziehungen zwischen Verwaltungsschalen mit Hilfe der Elemente des VWS-Metamodells

1.3 Vorgehen - Forschungsmethodik

Das Vorgehen während der Arbeit gliedert sich in vier Hauptbestandteile. Der Erste Abschnitt (Kapitel 2) beschäftigt sich mit der systematischen Literaturrecherche nach praxisnahen Use Cases, die das I4.0-Konzept widerspiegeln und bei denen eine VWS als Lösungsansatz zweckdienlich ist. Dabei beschränkt sich die Recherche für diese Arbeit vor allem auf den Bereich der Fertigungstechnik. Damit wird jedoch nicht ausgeschlossen, dass ein genannter Use Case nicht auch in der Verfahrenstechnik realisierbar ist.

Die nähere Beschreibung der Use Cases gibt Aufschluss über deren Funktion. Eine Zusammenstellung von Forschungsgruppen, die einzelne Use Cases mit der VWS anhand von Prototypen realisieren, hebt den Nutzen der VWS zusätzlich hervor. Ein Use Case wird nur in diese Arbeit aufgenommen, sobald sich der wirkliche Nutzen der VWS in diesem widerspiegelt.

Aus denen im ersten Abschnitt zusammengetragenen Use Cases erfolgt in Kapitel 3 die Auswahl genau eines Use Cases, aus dem konkrete Szenarien abgeleitet werden. Die Wahl der Szenarien richtet sich dabei nach der Häufigkeit des Vorkommens in der Wertschöpfungskette von Industrieunternehmen. Des Weiteren soll die Durchsetzung des I4.0-Konzepts möglichst kosteneffizient erreichbar und ohne größere Anpassung der Umgebungsbedingungen möglich sein. Die Szenarien beinhalten dabei eine Auswahl an Mitteln, die von der

VWS zur Verfügung gestellt werden und zur Lösung des Problems beitragen.

Die Visualisierung der Interaktionen in den Szenarien erfolgt mittels technologieneutralen UML-Sequenzdiagrammen, wobei die Umsetzung der Schnittstellenaufrufe durch eine konkrete API noch außer Betracht gelassen wird. Durch die Sequenzdiagramme ist kurzerhand ersichtlich, welche Interaktionsteilnehmer zugehörig zum Szenario sind und wie diese die Daten untereinander austauschen.

Aufbauend auf den im zweiten Abschnitt gewonnenen UML-Sequenzdiagrammen, erfolgt im dritten Abschnitt (Kapitel 4) die Abbildung der Szenarien durch technologiespezifische UML-Sequenzdiagramme. Für die Lösung eines Szenarios gibt es keine feste Vorgehensweise, sodass verschiedene Konzepte zur Realisierung des Szenarios mit ihren Vor- und Nachteilen beschrieben und anhand diesen das beste Konzept weitergeführt wird. Das heißt, es wird diskutiert, welche Methoden der VWS-API am besten auf das Szenario anwendbar sind und wie diese dann syntaktisch richtig in Sequenzdiagrammen dargestellt werden. Darunter fällt zudem die Entwicklung von Modellen, die Beziehungen zwischen VWS zweckmäßig abbilden, sowie die genaue Festlegung aller zur Umsetzung notwendigen Modellelemente des VWS-Metamodells. Das Resultat dieses Abschnitts ist eine Best-Practice Lösung, die in Kapitel 5 aufgegriffen wird.

Der letzte Abschnitt fasst die aus Kapitel 3 und 4 gesammelten Erkenntnisse zusammen und veranschaulicht die prinzipielle Realisierung der Best-Practice Lösung am LIA-I4.0-Demonstrator des Lehrstuhls für integrierte Automation (kurz LIA) der OvGU Magdeburg. Hierbei wird in einem ersten Schritt ein Vorgang im Demonstrator ausgewählt, der eines der vorher definierten Szenarien bestmöglich abbildet. Daran angeknüpft erfolgt die Spezifizierung der an der Interaktion teilnehmenden Rollen. Anschließend wird ein Sequenzdiagramm, entsprechend der Best-Practice Lösung aus dem dritten Abschnitt, mit konkreten Parametern des Demonstrators angereichert.

Die beschriebene Forschungsmethodik ist in Abbildung 1.1 übersichtlich dargestellt.

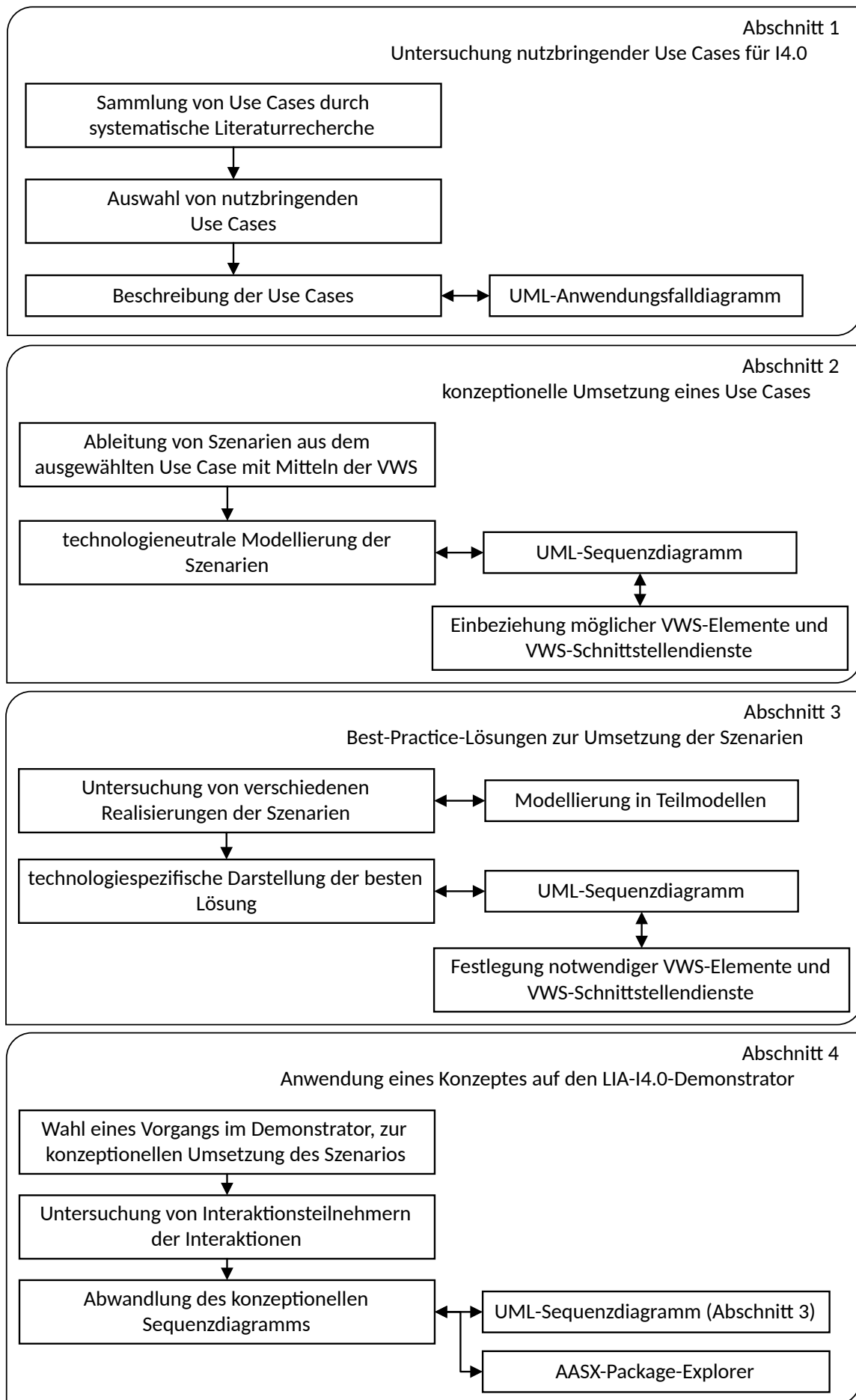


Abbildung 1.1: Forschungsmethodik der Masterarbeit

2 Untersuchung nutzbringender Use Cases für I4.0

Der Nutzen einer Integration des I4.0 Konzeptes geht mitunter aus dem Ergebnisbericht der Plattform Industrie 4.0 [2, S. 9] hervor, in dem folgende Vorteile erwähnt werden:

- Eine anhand der Flexibilisierung der Prozesse wirtschaftlich lohnende Fertigung bis zur Losgröße 1, demzufolge auch die Berücksichtigung individueller Kundenwünsche.
- Die flexible Reaktion auf Störungen.
- Eine durch die digitale Vernetzung standortübergreifende Optimierung der Nutzung von Ressourcen (z.B. Produktionsanlagen, Material, Energie) und der damit einhergehenden Steigerung der Produktionseffizienz.
- Die Entwicklung neuer Potenziale bezüglich nachgelagerter Dienstleistungen, die dem Nutzer bei Verwendung des Produkts zusätzlich angeboten werden können.

Des Weiteren entstehen in Bezug auf die Vernetzung der Wertschöpfungsketten laut dem Ergebnisbericht [2] weitere Vorteile in Bezug auf logistische Aspekte. Es ist denkbar, die Positionsdaten der Produkte in der Fertigung sowie die Materialverbräuche in Echtzeit an den Einkauf zu übertragen. Damit ist der Organisationseinheit Einkauf die Einsicht in die aktuellen Bestände und die Verortung der gefertigten Produkte möglich. Zusätzlich ist es vorstellbar, dem Kunden Informationen über den aktuellen Fertigungsstand seines bestellten Produkts zu liefern [2, S. 46].

Eine Basis für die systematische Literaturrecherche schafft die Forschungsroadmap der Arbeitsgruppe 2 (AG2) der Plattform Industrie 4.0 [3]. Darin beschrieben sind Use Cases, die unter anderem mit vorhandenen I4.0-Technologien ausführbar sind oder noch Zukunftsbilder darstellen.

Eine zweite Basis bildet der Use Case Katalog [4] des technischen Komitees 65 (engl. Technical Committee, abgek. TC 65) der IEC. Dieser Katalog umfasst 45 Use Cases gegliedert in acht Use Case Cluster, die den Begriff des „Smart Manufacturing“ repräsentativ beschreiben und sich auf die Wertschöpfungsketten in produzierenden Unternehmen beziehen [4, S. 8].

Aufbauend auf den umfassend ermittelten Use Cases aus [3] und [4] unterstreicht eine Suche nach Prototypen, die die Umsetzung einiger aufgeführter Use Cases in der Praxis erforscht, den Mehrwert dieser.

Eine Eingliederung der Use Cases erfolgt sowohl in der Forschungsroadmap [3] als auch im Use Case Katalog [4] in den Wertschöpfungsketten eines produzierenden Unternehmens. Die Darstellung der I4.0-Wertschöpfungsketten wurden aus den Literaturen [3] und [5] zu einer Grafik zusammengefasst und in Abbildung 2.1 dargestellt. Für die Illustration der Wertschöpfungsprozesse, die in den Wertschöpfungsketten enthalten sind, wurde der Statusreport für das durchgängige Engineering in I4.0-Wertschöpfungsketten der VDI/VDE Gesellschaft [5] herangezogen.

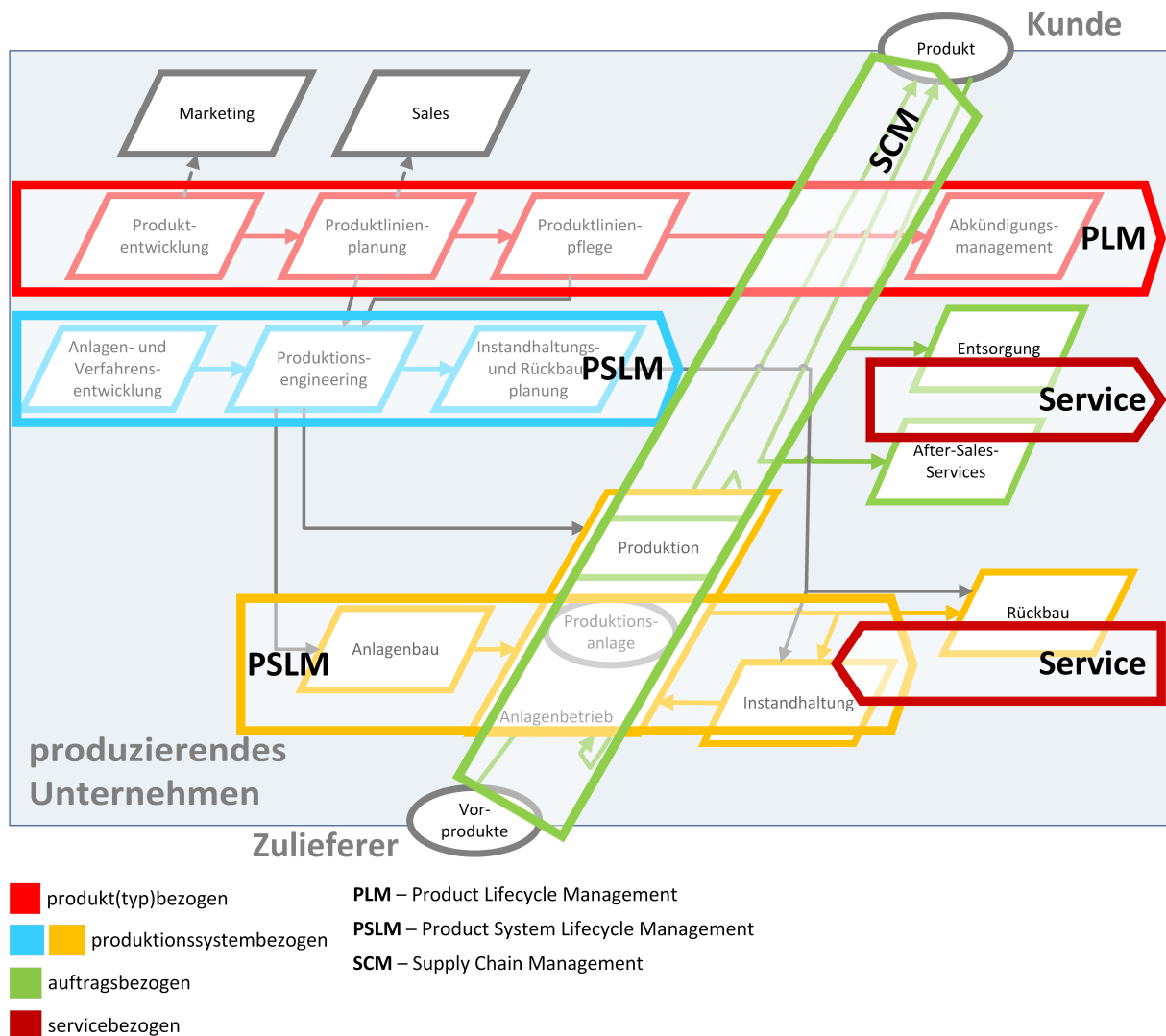


Abbildung 2.1: Wertschöpfungsketten in Industrie 4.0 bezogen auf produzierende Unternehmen (in Anlehnung an [3, S. 5] und [5, S. 3])

Die in Abbildung 2.1 den Wertschöpfungsprozessen überlagerten Wertschöpfungsketten werden in [3, S. 5f.] wie folgt beschrieben:

Product Lifecycle Management (PLM): Darin enthalten sind alle Prozesse, die zur Entwicklung eines Produkttyps erforderlich sind. Angefangen bei der Ideensammlung über die Aufstellung der Anforderungen, der Entwicklung der Produktlinienplanung und -pflege

bis hin zum Recycling finden diese Schritte meist virtuell statt.

Product System Lifecycle Management (PSLM): Das PSLM überspannt die Wertschöpfungsprozesse des Produktionssystems in dem das Produkt hergestellt wird und ist in Abb. 2.1 durch zwei Farben getrennt. Die blaue Wertschöpfungskette fasst die Prozesse der virtuellen Welt zusammen. Alle Prozesse, die in der realen Welt stattfinden, sind durch die orange Wertschöpfungskette markiert.

Die Prozesse der virtuellen Welt umfassen unter anderem das Engineering von Produktionslinien und die Instandhaltungsplanung, während in der realen Welt Prozesse wie die Errichtung einer Anlage, die Durchführung der Instandhaltung und das zum Lebenszyklusende der Anlage notwendige Recycling stattfinden.

Supply Chain Management (SCM): Diese Wertschöpfungskette bezieht sich vor allem auf die notwendigen Schritte zur Produktherstellung. Darunter fallen das Management der aktiven Aufträge und deren Steuerung durch die Produktion, sowie die logistische Planung zum Transport der Ressourcen und das Management der zugekauften Ressourcen.

Service: Die Services beziehen sich zum einen auf das hergestellte Produkt und zum anderen auf die Anlagen, die zur Herstellung des Produkts notwendig sind. Die Produktservices umfassen beispielsweise die Wartung der Produkte oder die Lieferungen von Ersatzteilen an den Kunden. Hingegen sind mit den Anlagenservices die Dienste von z.B. Anlagenherstellern, die Wartungen durchführen, gemeint oder bei Rückbau der Anlage die Inanspruchnahme von in dieser Branche niedergelassenen Dienstleistern.

Die beschriebenen Wertschöpfungsketten sollen vor allem zur Einordnung der individuellen unternehmensinternen Wertschöpfungsprozesse dienen. Zur Verbesserung der Übersichtlichkeit erfolgt am Ende dieses Kapitels eine Eingliederung der beschriebenen Use Cases in die oben erwähnten Wertschöpfungsketten.

Neben der Beschreibung der Wertschöpfungsketten ist auch ein gemeinsames Verständnis für die in dieser Arbeit verwendeten Begriffe „Use Case“ und „Szenario“ sehr wichtig. Das ist daraus ersichtlich, dass z.B. in der Forschungsroadmap der Plattform Industrie 4.0 Begriffe wie „Anwendungsszenarien“ und „Anwendungsbeispiele“ (s. [3, S. 4]) Verwendung finden. Während hingegen im Use Case Katalog der TC 65 nur der Begriff „Use Case“ definiert ist [4, S. 11].

Um eine Verwechslung der Bedeutungen von den verwendeten Begriffen zu vermeiden, hat die Normungsroadmap der DIN und DKE [6] einen Vorschlag unterbreitet, der sowohl national als auch international Akzeptanz findet. Dabei handelt es sich um die Kategorisierung von Use Cases, die wie folgt vorgenommen wird:

„→ **Business Szenarien**, bei denen aus einer geschäftlichen Perspektive Wertschöpfungsbeziehungen zwischen Firmen sowie deren Geschäftsmodelle beschrieben

werden.

→ **Use Cases**, bei denen ein technisches System in seinem Anwendungskontext beschrieben wird, und zwar wie Akteure außerhalb des technischen Systems mit diesem und untereinander interagieren.

→ **Praxisbeispiele**, bei denen ein konkreter Lösungsansatz beschrieben wird.“ [6, S. 24]

Diese Arbeit ordnet den verwendeten Begriff „Use Case“ in die oben genannte gleichnamige Kategorie ein, wie es die Taskforce TC 65 der IEC ebenfalls vornimmt [4, S. 151]. Der Begriff „Szenario“ ist nicht definiert und wird in dieser Arbeit als ein konkreter Problemraum mit Rahmenbedingungen betrachtet, in dem sich der Use Case abspielt. Die Best-Practice Lösungen, die in Kapitel 4 auftreten, sind in die oben genannte Kategorie der Praxisbeispiele einzuordnen und zeigen somit den bestmöglichen Lösungsansatz für ein Szenario.

Mit der Festlegung der Wertschöpfungsketten und der Einordnung der in dieser Arbeit verwendeten allgemeinen Begrifflichkeiten, fehlt noch die Festlegung von grundlegenden Eigenschaften für die objektive Bewertung der Use Cases im Hinblick auf die Einbeziehung des I4.0-Konzeptes. Diese Eigenschaften zur Bewertung der Use Cases werden anhand folgender Fragestellungen definiert:

a) Existieren bereits entwickelte und veröffentlichte Teilmodell-Templates der VWS, die die Umsetzung des Use Case erleichtern?

Ja: Wenn zur Lösung des Problems nur die bereits veröffentlichten Teilmodell-Templates notwendig sind, sinkt der Implementierungsaufwand, da eine konkrete Datenstruktur vorhanden ist. Durch die Verwendung ausschließlich dieser Teilmodell-Templates wird die Interoperabilität sowohl unternehmensintern als auch -übergreifend gewährleistet. Die Akteure sind jederzeit in der Lage, durch öffentlich zugängliche Spezifikationen, die Daten semantisch zu interpretieren und wiederzuverwenden.

Teilweise: Sobald es nicht möglich ist, alle Anforderungen durch die bereits veröffentlichten Teilmodell-Templates abzudecken, sind eigens konzipierte Teilmodelle notwendig. Da die intern erstellten Teilmodelle keiner öffentlichen Spezifikation unterliegen, ist die unternehmensübergreifende Interpretation der Daten nicht immer gewährleistet. Der Aufwand für die Implementierung muss infolgedessen nicht unbedingt höher sein, als wenn nur bereits veröffentlichte Teilmodell-Templates Verwendung finden. Allerdings ist es notwendig eine Spezifikation zu formulieren, wenn es anderen Unternehmen möglich sein soll die Daten der selbst erstellten Teilmodelle zu interpretieren.

Nein: Ist die Lösung mit keinem der veröffentlichten Teilmodell-Templates realisierbar, steigt der Implementierungsaufwand hinsichtlich der Analyse von praktikablen Datenstrukturen. Es müssen Teilmodelle designt werden, die zum einen die Daten aus den

Wertschöpfungsprozessen und zum anderen die Zugriffsberechtigungen durch Nutzer berücksichtigen. Eine Interpretation der Daten außerhalb der Unternehmensgrenze ist nur durch die semantische Bedeutung des einzelnen Datums, jedoch nicht über den Zusammenhang der Datenstruktur möglich (sofern keine Spezifikation des Teilmodells existiert).

b) In welchen Wertschöpfungsketten gliedert sich der Use Case ein und erzielt mit der VWS demzufolge einen Nutzen?

c) Ist der Use Case auch mit anderen technologischen Mitteln bzw. Software-Tools außer der VWS realisierbar?

Enterprise Resource Planning (ERP): Das Unternehmen SAP [7] stellt ein ERP-System als ein Softwaresystem dar, das die Leitung eines Unternehmens unterstützt. Zu den unterstützten Bereichen gehören unter anderem die Produktion, die Beschaffung von Hilfs- und Betriebsstoffen, das Assetmanagement und die Verfolgung wirtschaftlicher Zielsetzungen. Weiterhin ist auch ein Supply Chain Management (SCM) implementierbar, das die Verfolgbarkeit der Produkte ermöglicht und aktuelle Lagerbestände an die Geschäftsebene übermittelt [7].

Die Sammlung der Informationen aus den verschiedenen Geschäftsbereichen im ERP-System erfolgt nicht zuletzt durch die Integration von Internet of Things (IoT)-Technologien. Bringt der Einsatz einer VWS zusätzliche Funktionalitäten in Kombination mit den bereits vorhandenen Technologien, kann die VWS vorrangig eingesetzt und die bestehende IoT-Technologie dadurch abgelöst werden.

Manufacturing Execution System (MES): Die Siemens AG [8] umschreibt ein MES als eine Softwarelösung, die vor allem in der Steuerung von Produktion und Fertigungsprozessen Einsatz findet. Die Software unterstützt die zentrale Datenerfassung von Produktionsanlagen, Steuerungen und Geschäftsanwendungen in Unternehmen sowie von firmenzugehörigen Werken an unterschiedlichen Standorten. Die Erfassung von Echtzeitdaten ermöglicht Optimierungen in der Fertigung und die Steigerung der Produktivität. Das MES bildet eine Schnittstelle zwischen der Feldebene und einem ERP-System. Produkt- und Auftragsdetails werden im MES erfasst, die zum einen an das ERP-System weitergeleitet und zum anderen in Form von Arbeitsanweisungen an Produktionsmitarbeiter elektronisch übermittelt werden [8].

Ein weiteres Ziel eines MES ist die Produktionsplanung papierlos zu gestalten und somit redundante Informationen zu vermeiden. Ursachen von Störungen und Problemlösungen können durch die Echtzeitüberwachung schnell identifiziert werden (ebd.).

Die VWS kann das MES durch die Erfassung und Aufbereitung der Informationen von Anlagenkomponenten bis hin zu Produktionslinien unterstützen. Dafür müssen der VWS Zugänge, in Form von Schnittstellen, zum MES bereitgestellt werden. Wie bei dem ERP-

System kann eine VWS nützlich sein, wenn diese mehr Funktionalitäten als die bisher eingesetzte Applikation bietet.

Machine-to-Machine-Kommunikation (M2M-Kommunikation): Der Autor V. S. Chakravarthi [9] beschreibt diese Technologie als die direkte Kommunikation zwischen Maschinen, ohne den Eingriff von Menschen. Die Kommunikation findet drahtgebunden über Kommunikationskabel oder drahtlos via WLAN, Bluetooth oder Funk statt. Das Ziel der Kommunikation zwischen Maschinen ist die gegenseitige Überwachung, Steuerung und Übermittlung von Informationen [9, S. 151].

Durch die Einführung der M2M-Kommunikation ist es möglich, dass Anlagen in einer Produktionslinie sich eigenständig in ihren Aufgaben beeinflussen. Damit ist z.B. bei Anlagenstörungen eine rechtzeitige Reaktion der vor- und nachgelagerten Anlagen realisierbar. Diese können auf Störungen durch beispielsweise eine Verlangsamung der Produktion reagieren.

Die Ersetzung aller vorhandenen Technologien für die M2M-Kommunikation, durch die VWS, ist nicht möglich. Jedoch kann die VWS bei der Kommunikation, über z.B. WLAN, die Semantik und Syntax für den Datenaustausch vereinheitlichen und in diesem Bereich einen Mehrwert schaffen.

Customer-Relationship-Management (CRM): Das Unternehmen GEDYS IntraWare GmbH informiert auf seiner Website [10] über die Definition, Ziele und Aufgaben eines CRM. Ein CRM rückt den Kunden in den Vordergrund und kann sowohl als Software, Strategie oder Prozess umgesetzt werden, wobei hier durch den technischen Bezug nur die Software betrachtet wird. Die CRM-Software hilft einem Unternehmen durch Datenerfassung, Analysemethoden und Darstellungen der Ergebnisse, sich mehr an den Bedürfnissen seiner Kunden auszurichten. Dazu zählen beispielsweise die individuelle Angebotserstellung für Kunden aber auch das Angebot von Services. Einsatz findet die CRM-Software in den Bereichen Vertrieb, Marketing, Service und IT. Eine Verbindung mit ERP-Systemen ist häufig anzutreffen [10].

Die VWS kann unterstützend wirken, da einzelne Use Cases auf Services eingehen, die Unternehmen ihren Kunden anbieten möchten. Die Services können in der VWS integriert und bei Auslieferung des Produkts vom Kunden entsprechend genutzt werden.

Industrial Internet of Things (IIoT): Das IIoT ermöglicht laut [11] die Integration von physischen Objekten wie Produktionssystemen, Maschinen, Sensoren etc. in digitale Netzwerke. Verbunden mit diesen digitalen Netzwerken sind unter anderem ERP-Systeme oder MES, die durch die Vernetzung Zugriff auf alle Daten der vorhandenen Produktionsressourcen erhalten. Geschäftsmodelle im Zusammenhang mit IIoT sind unter anderem die Bereitstellung von Services, wie z.B. das Condition Monitoring, oder die Verwendung von IIoT-Plattformen für einen unternehmensübergreifenden Datenaustausch. [11]

zusätzliche Softwareapplikationen: Nicht für alle Anforderungen bieten größere Softwaresysteme eine Lösung. Oftmals besetzen die individuellen Softwareapplikationen Nischenbereiche, die zur Lösung eines einmalig vorkommenden Problems entwickelt werden. Das geschieht entweder durch Dienstleister für Softwareentwicklung oder unternehmensintern im IT-Bereich.

Die Entscheidung über die Verwendung einer VWS anstatt einer spezifischen Softwareapplikation ist nicht leicht zu treffen. Prinzipiell ist aber zu sagen, dass bei komplexen Methoden zur Aufbereitung und Auswertung von Daten die individuelle Softwareapplikation mehr Vorteile bietet. Bei dem Transport von Daten zu überlagerten Softwaresystemen mit einfacher Datenaufbereitung kann man allerdings den Einsatz einer VWS erwägen.

d) Welche Rolle nimmt die VWS bei der Umsetzung des Use Case ein?

passiv: Die Daten der passiven VWS werden entsprechend dem Metamodell serialisiert und als Datei zwischen Teilnehmern ausgetauscht. Zu den Dateiformaten zählen beispielsweise XML, JSON, RDF, OPC UA und AutomationML [12, S. 31 f.].

aktiv: Eine aktive VWS gliedert sich laut [12] in reaktive und proaktive VWS auf. Die Daten der reaktiven VWS werden durch eine IP/API-Schnittstelle zugänglich gemacht. Die reaktive VWS agiert ähnlich einem Server und veröffentlicht ihre Daten bei Anfragen durch externe Softwareapplikationen.

Die proaktive VWS besitzt zusätzlich zu einer IP/API-Schnittstelle die Möglichkeit, über die I4.0-Sprache mit anderen VWS zu interagieren. Die proaktive VWS verhält sich im Gegensatz zu der reaktiven VWS autonom [12, S. 31 f.].

Mit den aufgelisteten Fragen ist eine objektive als auch subjektive Einordnung der nutzbringenden Use Cases aus den folgenden Abschnitten möglich. Die Übersicht in Tabelle 2.1 verweist auf die Use Cases, die in dieser Arbeit näher beschrieben werden. Zu den Informationen dieser Übersicht gehören unter anderem eine Kurzbeschreibung der Use Cases und eine Einordnung derer in die genannten Wertschöpfungsketten. Weiterhin befasst sich eine Spalte der Tabelle mit der Beantwortung der Frage a) hinsichtlich bereits veröffentlichter bzw. sich in der Entwicklung befindlicher Teilmodell-Templates. Zusätzlich sind alternative Technologien bzw. Software-Tools aus Frage c), die neben einer VWS auch zur Umsetzung des Use Case genutzt werden können, vermerkt. Die Auflistung der Alternativen sind nur Vorschläge, die keinen Anspruch auf Vollständigkeit erheben.

Wie zukunftsorientiert ein Use Case ist, zeigt eine Fortschritt-Skala unter dem Titel des jeweiligen Use Case. Dabei beschreibt eine voll ausgefüllte Skala einen Use Case mit einem bereits vorhandenen Prototyp, der Verwaltungsschalen zur Umsetzung nutzt und damit eine stärkere Anwendung im industriellen Umfeld in den nächsten Jahren fördert.

Je weiter der Fortschritt in der Skala am Anfang steht, desto zukunftsorientierter ist der Use Case. Für einen Use Case mit einem Fortschritt von über 50% der Skala sind bereits

konzeptionelle Forschungsergebnisse vorhanden. Use Cases, zu denen hingegen noch keine Forschungsarbeiten existieren oder die sich in der Konzeptphase befinden, befinden sich am Anfang der Fortschritt-Skala. Die Zuordnung der Use Cases in die Fortschritt-Skala basiert auf bekannten Forschungen und schließt weitere nicht aus, die zu einer Veränderung der Skala führen können.

Hinweis: Die ersten Teilmodell-Templates der VWS wurden durch die Plattform Industrie 4.0 in Zusammenarbeit mit dem Zentralverband für Elektrotechnik- und Elektronikindustrie (ZVEI) entwickelt. Zu diesen Teilmodell-Templates gehören das „ZVEI Digital Nameplate for Industrial Equipment“ [13] und der „Generic Frame for Technical Data for Industrial Equipment in Manufacturing“ [14], die als einzige bisher veröffentlicht sind. Eine Standardisierung dieser Teilmodelle fand nicht statt, da es sich bei den Organisationen nicht um Standardisierungsorganisationen handelt.

Im Jahr 2020 schloss sich der Verband Deutscher Maschinen- und Anlagenbau e. V. (VDMA) mit dem ZVEI, der Bitkom und weiteren Industrieunternehmen zusammen. Gemeinsam gründeten sie die Industrial Digital Twin Association e. V. (IDTA) [15].

Als Ziel setzt sich die IDTA unter anderem die Entwicklung von Teilmodell-Templates [16]. Darunter fällt auch die Weiterentwicklung der beiden zuvor genannten Teilmodell-Templates des ZVEI. Zusätzlich zu diesen Teilmodell-Templates befinden sich 17 weitere in Entwicklung oder Planung (vgl. [16]).

Aufgrund der unterschiedlichen Entwicklungsstände, werden in der Frage a) nur Teilmodell-Templates einbezogen, die sich bereits durch die IDTA in der Entwicklung befinden. Das heißt, die sich in Planung befindlichen Teilmodell-Templates werden ausgeschlossen. Die gesamte Liste der 21 Teilmodell-Templates mit einer kurzen Beschreibung sind im Anhang A aufgeführt.

Tabelle 2.1: Übersicht der nutzbringenden Use Cases

Use Case Cluster	Use Case	Use Case Beschreibung	Einordnung in Wertschöpfungskette				Möglichkeit zur Nutzung von Teilmodell-Templates	technol. Alternativen zur Umsetzung des Use Case
			PLM	PSLM	SCM	Service		
Auftrags-gesteuerte Produktion	Herstellung von individualisierten Produkten	Durch die Verwendung einer VWS können individuelle Kundenwünsche in einem Produktauftragskonfigurator berücksichtigt werden. Die Bestandteile des Produkts, betriebswirtschaftliche Informationen (z.B. Lieferdatum, Menge) sowie notwendige Fertigungsschritte werden in neu generierten VWS gespeichert. Diese Informationen können in der Produktion Anwendung finden.			●		teilweise	ERP, zusätzliche Software
	Unternehmensinterne Logistik	Produktionsmaschinen, wie auch Transport- und Lagersysteme erhalten proaktive VWS und bilden damit ein interaktionsfähiges System. Das automatische Auffüllen von Materialbeständen zur Fertigung eines Produkts sowie der Transport von Produkten, zwischen Maschinen mit externer Produktaufnahme, werden ermöglicht.			●		nein	M2M-Kommunikation, MES
Wandlungsfähige Fabrik	Modularisierung von Produktionssystemen	Das Ziel ist ein zusammengesetztes Produktionssystem aus Produktionsmodulen, dass unkomplizierte und teilautomatisierte Produktionsumstellungen zulässt. Die VWS dient als Informationsträger für Schnittstellenbeschreibungen und sorgt somit für einen reibungslosen Austausch von Produktionsmodulen unterschiedlicher Hersteller.	●	●			teilweise	-
	Plug and Produce für Feldgeräte	Ähnlich einem PC-Netzwerk wird die automatische Konfiguration von Kommunikationsteilnehmern bei Ab- bzw. Ankopplung von Feldgeräten vorgenommen. Mit der VWS wird die automatisierte Integration von Feldgeräten unterschiedlicher Hersteller in ein echtzeitfähiges Kommunikationsnetzwerk erlaubt.		●	●		nein	-

Use Case Cluster	Use Case	Use Case Beschreibung	Einordnung in Wertschöpfungskette				Möglichkeit zur Nutzung von Teilmodell-Templates	technol. Alternativen zur Umsetzung des Use Case
			PLM	PSLM	SCM	Service		
Asset-management	Verwaltung von Assets [Redacted]	Es werden beschreibende Assetinformationen über mehrere Lebenszyklusphasen hinweg in einem Assetmanagementsystem gespeichert. Diese Informationen finden auch in den Use Case Clustern „Optimierung des Produktionsablaufs“ und „Energieeffizienz“ Anwendung.	●	●	●	●	teilweise	ERP, MES
Produkt- und Produktionsdienstleistungen	Value-Based Services für Produktionsressourcen [Redacted]	Hersteller möchten ihren Kunden wertorientierte Dienstleistungen für die erzeugten Produkte bereitstellen. Dazu gehören z.B. Updates und Diagnosemöglichkeiten für Assets, die durch die VWS zugänglich sind.	●	●	●	●	abhängig vom Service	IIoT
IT-Infrastruktur und Software	Dokumentationspflege bei Gerätetausch [Redacted]	Ziel ist die Aktualisierung von Dokumentationen bei einem herstellerunabhängigen Gerätetausch durch den Einsatz von VWS. Das neue Gerät muss dabei mindestens über den gleichen Funktionalitätsumfang wie das zu ersetzende Gerät verfügen.	●	●	●	●	teilweise	ERP, zusätzliche Software
	Human-Machine-Interface [Redacted]	Bei modulbasierten Produktionssystemen soll nach z.B. einer Rekonfiguration des Systems eine automatische Anpassung der betroffenen Human Machine Interfaces (HMI's) stattfinden. Damit sollen Ausfallzeiten bedingt durch Umrüstungen minimiert und das Personal in seinen Arbeiten unterstützt werden. Die VWS stellt die Navigationsstruktur für die HMI's bereit.	●	●	●	●	teilweise	zusätzliche Software
Sonstige	Kreislaufwirtschaft [Redacted]	Die Berücksichtigung der Wiederverwendbarkeit von Materialien bereits bei der Produktentwicklung sowie die Speicherung dieser produktbezogenen Informationen schaffen einen Mehrwert bei der umweltfreundlichen Produktion und Wiederverwendung von Produkten. Die VWS dient als Speicher für die notwendigen Informationen des Materialkreislaufs.	●		●		teilweise	IIoT

2.1 Use Case Cluster - Auftragsgesteuerte Produktion

Die Auftragsgesteuerte Produktion (AGP) befasst sich laut [3] mit der werks- und unternehmensübergreifenden autonomen und automatisierten Verbindung von Produktionsressourcen. Gemeint ist die flexible Verknüpfung der Fähigkeiten von Produktionsanlagen zur Herstellung kundenindividueller Produkte. Vor allem durch die rasante Verkürzung der Zeitintervalle zwischen neu entwickelten Produkten und der Verfügbarkeit neuer Produktionstechnologien ist eine AGP für Unternehmen entscheidend um sich am Markt zu etablieren [3, S. 8].

Die Basis zur Realisierung der AGP bilden vor allem die Standardisierung von Prozessschritten sowie die standardisierte Beschreibung der Funktionalitäten von Produktionsressourcen. Das Ergebnis der Standardisierung ist ein einheitliches Verständnis der Fähigkeiten für alle Akteure. Damit sind technische Systeme realisierbar, die eine automatisierte Auftragsplanung, -vergabe und -steuerung durch flexible Verknüpfung von eigenen als auch fremden Prozessmodulen zulassen (ebd.).

Laut [3] wandeln sich die Wertschöpfungsketten von Unternehmen durch die AGP von starren und vordefinierten zu zergliederten und flexiblen Wertschöpfungsketten um. Diese stehen je nach Auftrag in unterschiedlichen Beziehungen zueinander. Durch die unkomplizierte Auslagerung von Fertigungskapazitäten, wird den produzierenden Unternehmen die Konzentration auf spezielle Wertschöpfungsprozesse ermöglicht. Damit ist die Hervorhebung der eigenen Kompetenzen, gegenüber Wettbewerbern, am Markt gewährleistet.

Die AGP lässt aber nicht nur Vorteile entstehen. Durch die Konkurrenzsituation am Markt wird der Druck auf produzierende Unternehmen, schneller und günstiger bei gleichbleibend hoher Qualität zu produzieren, immer stärker. Des Weiteren sinkt durch die Auslagerung von Fertigungsaufgaben die Nachfrage nach neuen Maschinen mit neuen Funktionalitäten, was eine Schwächung der Marktposition von Maschinenlieferanten zur Folge hat [3, S. 9].

Zwei Forschungsberichte verdeutlichen die Aspekte der auftragsgesteuerten Produktion. Der erste Bericht [17] betrifft den deutschlandweiten I4.0-Demonstrator, der durch den Zusammenschluss von mehreren deutschen Instituten und Lehrstühlen entstand. Der Demonstrator veranschaulicht praxisnah die Konzepte von Industrie 4.0 für die Wirtschaft. Das technische Konzept und die Implementierung sind im Forschungsbericht [17] des Lehrstuhl für Automatisierung und Informationssysteme an der technischen Universität München nachzulesen.

Der zweite Bericht der Plattform Industrie 4.0 [18] ist theoretischer Art und analysiert die Prozessphasen, die während der AGP am Beispiel eines individuellen Fahrradlenkers durchlaufen werden. Darin sind nicht nur die teilnehmenden Akteure, sondern auch Themen wie Standards und Forschung, Sicherheit, Recht sowie Arbeit und Qualifizierung näher beschrieben [18].

2.1.1 Herstellung von individualisierten Produkten

Die kundenindividuelle Fertigung von Produkten geringer Bestellmengen hat in den letzten Jahren bei den produzierenden Unternehmen immer mehr an Bedeutung gewonnen. Mit den noch teilweise starren Kopplungen von Produktionslinien und den je nach Produktvariante resultierenden Produktionsaufträgen ist die Einführung einer individualisierten Produktion nur wirtschaftlich beschränkt möglich.

Dieser Use Case beschäftigt sich mit den Anforderungen an eine individualisierte Produktion, damit diese nutzbringend in die Realität überführt werden kann. Die TC 65 [4] legt als eine Voraussetzung für den Use Case ein bestehendes anpassungsfähiges Produktionssystem fest. Zusätzlich müssen standardisierte Produktionsaufträge spezifiziert werden, die von Maschinen interpretierbar sind und eine flexible Steuerung des Produkts durch die Fertigung gewährleisten. Wie erwähnt sind jedoch viele der bereits vorhandenen Produktionssysteme in diesem Zusammenhang nur bis zu einem gewissen Grad anpassungsfähig. Des Weiteren ist die Standardisierung eines Produktionsauftrags, der für alle Produktvarianten von der Unternehmensebene über die angeknüpften Leitebenen bis in die Feldebene reicht, eine Herausforderung [4, S. 24].

Der Use Case ist laut [4, S. 23] im Supply Chain Management integriert. Sein Rahmen schließt als Akteure zum einen den Kunden und zum anderen den Produktionskoordinator des produzierenden Unternehmens ein. Die Schnittstelle zwischen dem Kunden und dem Unternehmen bildet ein Software-Tool, in dem der Kunde seine Produktaufträge individuell konfiguriert. Im produzierenden Unternehmen bildet die Anfrage-/Auftragsverwaltung sowie ein Tool zur Unterstützung der Fertigungsplanung die Schnittstelle zwischen dem Produktionskoordinator und den Produktionsressourcen. Ausgeschlossen von der Betrachtung des Use Case sind die ausgelagerten Produktionsschritte und die unternehmensinterne Logistik [4, S. 25].

Die TC 65 [4] beschreibt als ein Ziel des Use Case die Reduzierung von manuellen Tätigkeiten durch leistungsfähige Optimierungsalgorithmen, die eine höhere Auslastung der Fertigung ermöglichen. Dadurch verringern sich die Routineaufgaben des Produktionskoordinators, sodass dieser sich bedeutsameren Aufgaben zuwenden kann. Die Algorithmen gewährleisten bei unerwarteten Ereignissen eine schnellere Reaktion sowie eine schnellere Konzipierung neuer Lösungen für die Produktion [4, S. 27].

Die Interaktionen, die im Use Case stattfinden, sind in Abbildung 2.2 dargestellt und laut [4, S. 25-27] folgendermaßen konkretisiert:

1. Der Kunde generiert über das Produktauftrags-Konfiguratorool eine standardisierte Produktionsanfrage an das produzierende Unternehmen. In der Anfrage stellt der

Kunde mit Hilfe des Produktkatalogs sein Produkt zusammen und gibt weitere Informationen wie die Bestellmenge, das Lieferdatum, besondere Qualitätsanforderungen etc. an. Nach einer anschließenden Validierung der eingegebenen Daten, sendet der Kunde seine Produktionsanfrage an das produzierende Unternehmen.

2. Der Produktionskoordinator erhält über die Anfrage-/Auftragsverwaltung die Produktionsanfrage und prüft diese auf die Umsetzbarkeit bezüglich vorhandener Produktionsressourcen, dem gewünschten Liefertermin und anderen Anforderungen. Unter Zuhilfenahme des Tools zur Fertigungsplanung erstellt der Produktionskoordinator einen Fertigungsauftrag und setzt somit einen geplanten Liefertermin fest. Die gewonnenen Informationen bilden die Basis für das Angebot an den Kunden. Die eingeplanten Produktionsressourcen werden für eine festgelegte Zeitspanne der Angebotsprüfung und -annahme durch den Kunden freigehalten.

3. Nimmt der Kunde das Angebot an, gibt der Produktionskoordinator den Fertigungsauftrag an die eingeplanten Produktionsressourcen frei. Gleichzeitig sorgt er für die Bereitstellung aller zur Fertigung benötigten Materialien und Ressourcen. Der Kunde kann während der Produktion über den Fertigungsstand seines Produkts informiert werden.

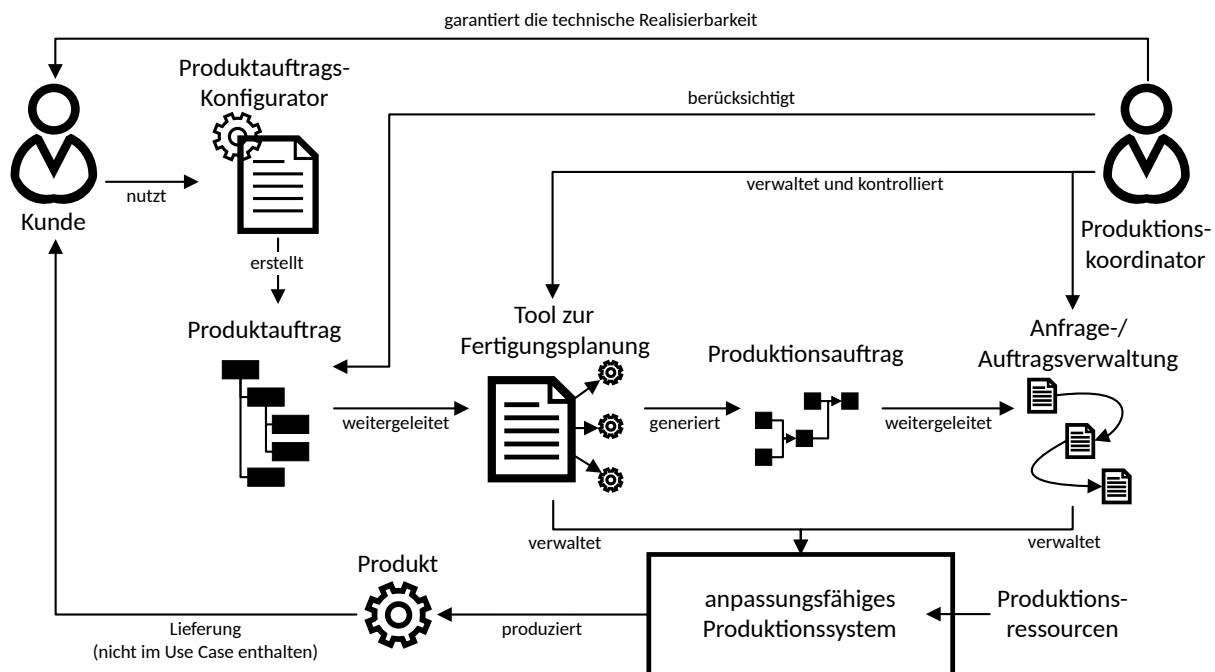


Abbildung 2.2: Technische Perspektive des Use Case „Herstellung von individualisierten Produkten“ (in Anlehnung an [4, S. 25])

Die Idee, eine Verwaltungsschale in diesem Use Case einzusetzen, bezieht sich nur auf den ersten der eben beschriebenen drei Interaktionsschritte. Dazu soll das Produktauftrags-Konfiguratortool eine VWS erzeugen, die alle Bestandteile des Produkts sowie Angaben zu Bestellmenge, Lieferdatum usw. enthält. Damit wird eine erste Stufe zur Herstellung

individueller Produkte bewältigt. Darauf aufbauend ist eine Wiederverwendung der Informationen aus der individuellen Produkt-VWS in der Produktion und dessen Anreicherung mit zusätzlichen Produktionsdaten denkbar. Aufgrund dessen ist im weiteren Vorgehen der Use Case auf das Produktauftrags-Konfigurator tool begrenzt. Alle weiteren Schritte wie die Fertigungsplanung und -ausführung, unter Verwendung einer VWS, werden ausgelassen. Abbildung 2.3 zeigt den Use Case unter Verwendung einer VWS. Die Akteure sind wie bei dem originalen Use Case weiterhin der Kunde und der Produktionskoordinator. Zusätzlich zu den menschlichen Akteuren sind auch die VWS und die Anfrage-/Auftragsverwaltung am Use Case beteiligt. Die Anfrage-/Auftragsverwaltung beschreibt Aufgaben, die einem ERP-System ähneln, weshalb für diesen Use Case im folgenden die Bezeichnung ERP-System genutzt wird. Der gesamte Use Case der Auftragskonfigurierung (Abb. 2.3 orange hinterlegt) ist bildlich durch die Zusammensetzung einzelner Teilschritte beschrieben. Sie unterstützen die detailliertere Beschreibung des Use Case.

Die Konfiguration des Produktauftrags beginnt für den Kunden mit der Wahl der Produktkomponenten im Konfigurator tool. Die Realisierbarkeit der individuellen Produktvariante wird bei jedem Auswahlschritt durch einen Produktkatalog mit technischen Anforderungen validiert. Damit ist abgesichert, dass nur technisch realisierbare Produkte vom Kunden bestellt werden können. Nach der Zusammenstellung gibt der Kunde unter anderem die Bestellmenge, Qualitätsanforderungen und das gewünschte Lieferdatum an. Sind alle Daten vorhanden folgt eine nochmalige Validierung des Produktauftrags durch den Kunden. Mit dem Abschluss der Auftragskonfiguration wird der Auftrag als Anfrage an den Produktionskoordinator, stellvertretend für das produzierende Unternehmen, gesendet. Gleichzeitig wird vom Konfigurator tool eine VWS erstellt, die das Produkt mit seinen Eigenschaften und den Auftragsdaten als digitalen Zwilling repräsentiert.

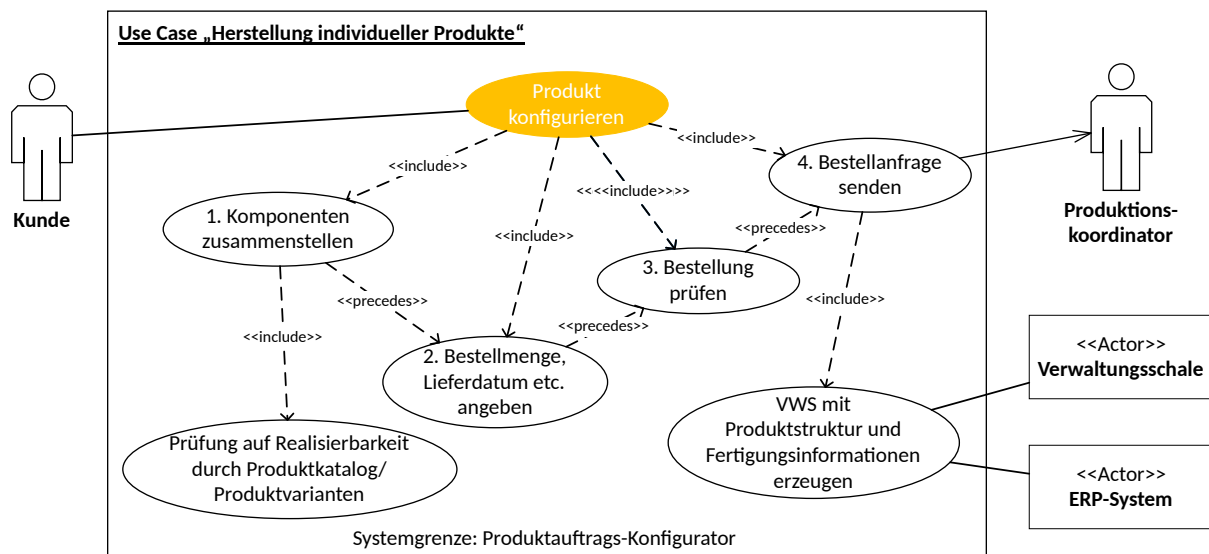


Abbildung 2.3: Use Case Diagramm - Herstellung von individualisierten Produkten

Die technischen Produktdaten können in diesem Fall in einer reaktiven VWS gespeichert werden. Teilmodell-Templates wie die „Bill of Material“ und der „Generic Frame for Technical Data for Industrial Equipment in Manufacturing“ (s. Anhang A) stehen für die Umsetzung zur Verfügung. Für die allgemeinen Auftragsdaten existieren seitens der IDTA jedoch noch keine Teilmodell-Templates. Das ERP-System wird in die Lage versetzt, die Auftragsdaten über die VWS-API anzufragen und für die Produktion weiter zu verarbeiten. Das Splitten des originalen Use Case in kleinere Teilschritte ermöglicht den produzierenden Unternehmen die sukzessive Anpassung ihrer Produktionssysteme. Die Umsetzung des originalen Use Case wäre erreicht, wenn mit der Produktkonfiguration eine VWS erzeugt wird, die von der Auftragsbestätigung bis zur Fertigstellung des Produkts Einsatz findet. Die VWS trägt dabei alle relevanten Fertigungsinformationen und Dokumentationen zusammen.

Die Aufgaben zur Umsetzung des ersten Schritts umfassen neben der Integration der VWS-API in das ERP-System auch die oben genannte Spezifikation eines standardisierten Produktauftrags. Der standardisierte Produktauftrag kann anschließend zur Spezifikation eines VWS-Teilmodells genutzt werden. Die Speicherung allgemeiner Informationen aus dem Produktauftrag ist in bereits veröffentlichten Teilmodell-Templates, wie sie in diesem Kapitel bereits benannt wurden, möglich.

Zu den Anforderungen an die Standardisierung gehören unter anderem die nachfolgenden Aspekte, die in [4, S. 27] näher beschrieben sind:

- Standardisierung der Produktaufträge (Struktur, Randbedingungen, Qualitätskriterien)
- Standardisierung der Produktionsaufträge (Struktur, Randbedingungen, Fertigungsdokumentation)
- Standardisierung der Produktionsfähigkeiten von Produktionsressourcen (Vergleichbarkeit von Fähigkeiten siehe Use Case „Standardisierung von Produktionstechnologien“ [4, S. 50–53])
- Standardisierung der Basisarchitekturen (Schnittstellen und Interaktionsprotokolle)

Für die Umsetzung des Use Case Diagramms aus Abbildung 2.3 ist als alternative Technologie die direkte Verknüpfung des Produktauftrags-Konfiguratortools mit dem ERP-System möglich. Das ERP-System erhält somit alle Konfigurationsinformationen zu den bestellten Produkten von dem Konfiguratortool. Durch dieses Verfahren entsteht jedoch keine I4.0-Komponente, die zukünftig die Informationen bezogen auf eine Produktinstanz individuell über die gesamte Produktion hinweg speichert.

2.1.2 Unternehmensinterne Logistik

Der Transport von Produkten zwischen Produktionssystemen und die interne Beförderung von Betriebsstoffen oder Werkzeugen wird schon heute z.B. in Versandzentren und der Automobilindustrie überwiegend durch autonome Transportsysteme erledigt. Der Use Case gehört in Verbindung mit der Herstellung individueller Produkte zu dem Cluster der auftragsgesteuerten Produktion und gliedert sich ebenfalls in die Wertschöpfungskette des SCM ein [4, S. 23].

Das besondere an diesem Use Case ist der geringe Einfluss des Menschen als Akteur, auch wenn dieser noch die Hoheitsgewalt über alle Vorgänge hat. Die technischen Systeme werden in die Lage versetzt, selbstständig untereinander zu kommunizieren und ihre Abläufe entsprechend zu planen.

Die TC 65 [4] verweist darauf, dass zusätzlich zur individuellen Produktherstellung auch die Use Cases „Modularisierung von Produktionssystemen“ (s. Kapitel 2.2.1) und „Flexible Terminierung und Ressourcenzuweisung“ (s. [4, S. 27–30]) zur konzeptionellen Umsetzung der unternehmensinternen Logistik genutzt werden können. Die Logistik ist durch ihre Rolle als Zulieferer der Produktionssysteme als ein integraler Bestandteil dieser zu betrachten. Das Ziel dieses Use Case ist den Fluss von Materialien sowie deren Lagerung so effizient (optimale Nutzung der vorhandenen Ressourcen) und effektiv (Erfüllung von Zusagen bzgl. Preis, Qualität, Reaktionszeit etc.) zu steuern und zu planen. Die Flexibilität spielt ebenso eine bedeutende Rolle, da sie die Reaktion auf unerwartete Ereignisse durch schnelle Anpassung von Transportrouten gewährleistet [4, S. 37].

Voraussetzungen für die Umsetzung des Use Case werden von der TC 65 nicht genannt. Jedoch sollten neben den autonomen Transportsystemen auch Zugänge zu Produktionssystemen bzw. Lagern für die automatisierte Anlieferung/Abholung von Produkten und Betriebsstoffen vorhanden sein. Des Weiteren müssen Kommunikationsmöglichkeiten zwischen allen am Use Case beteiligten Akteuren zur Verfügung stehen. Die technische Perspektive auf den Use Case ist in Abbildung 2.4 zu sehen.

Die Basisanforderungen an die Standardisierung bilden in der unternehmensinternen Logistik die Verknüpfung der Standardisierungen von den drei genannten Use Cases individuelle Produktherstellung, flexible Terminierung und Modularisierung von Produktionssystemen. Zusätzlich zu diesen Anforderungen nennt die TC 65 [4, S. 39] weitere Bedingungen:

- Standardisierung von Schnittstellen der Transport-, Lager- oder Produktionsressourcen für den informellen Datenaustausch bei Warenübergabe
- Standardisierung der Identifizierung und Lokalisierung von Transportsystemen

- Standardisierung von Karten und Navigationssystemen zur Planung des bestmöglichen Transportwegs

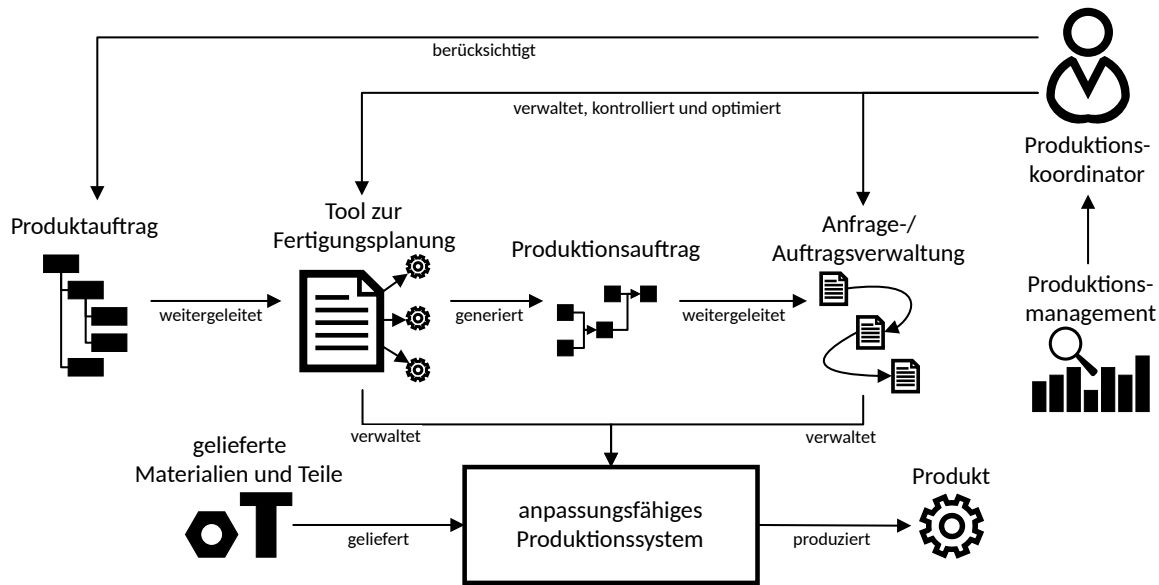


Abbildung 2.4: Technische Perspektive des Use Case „Unternehmensinterne Logistik“ (in Anlehnung an [4, S. 38])

Die konzeptionelle Einbindung einer proaktiven VWS in den Use Case wird im Folgenden für die Initiierung eines Transports betrachtet. Dabei nehmen als Akteure die Transport-, Produktions- und Lagersysteme sowie das herzustellende Produkt teil. Jeder der genannten Akteure bekommt eine VWS zugewiesen, über die die Transporte der Produktionsressourcen mittels dem standardisierten Ausschreibungsverfahren [19] initiiert werden.

Die Anfrage an einen Transport kann von dem herzustellenden Produkt, dem Lager oder indirekt von einer Produktionsressource (über die Anfrage zur Bereitstellung von Betriebsstoffen an das Lager) ausgehen. Liegt ein Ereignis vor, für das ein Transport benötigt wird, stellt der anfragende Akteur zu Beginn seine Informationen über die Herkunft und das Ziel aber auch z.B. die Dringlichkeit des Transports oder Eigenschaften der zu transportierenden Ware bereit. Auf die Anfrage hin beginnt ein Ausschreibungsverfahren, in dem alle freien Transportsysteme ihr Angebot abgeben. Nach der Auswertung und Auswahl des bestmöglichen Angebots, wird zwischen dem anfragenden Akteur und dem Transportsystem eine Vertragsbindung gemäß der Ausschreibung eingegangen. Abbildung 2.5 zeigt das Use Case Diagramm mit seinen Akteuren und Teilaufgaben.

Von der IDTA veröffentlichte Teilmodell-Templates für diesen speziellen Use Case existieren derzeit nicht [16]. Deshalb müssen Teilmodelle für die Verwaltung von Standortdaten sowie relevanter Informationen für die Warenübergabe wie Gewicht, Temperatur, geometrische Maße etc. spezifiziert werden. Des Weiteren ist die Ausstattung aller Akteure mit proaktiven VWS, die über die I4.0-Sprache autonom kommunizieren, mit einem Mehraufwand verbunden. Hinzukommend sind nicht alle Produktionsressourcen für die Kopplung

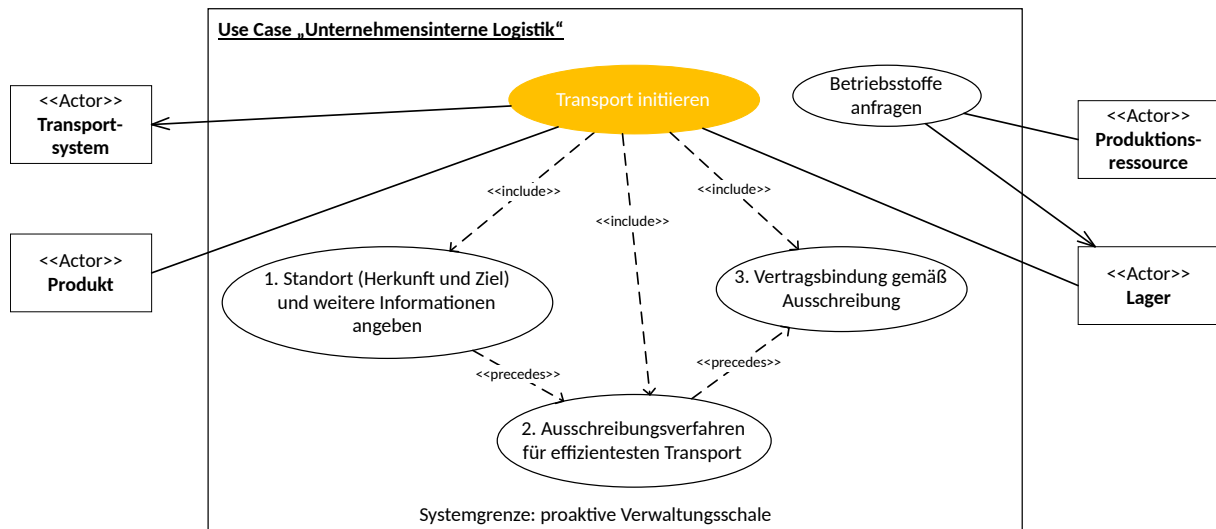


Abbildung 2.5: Use Case Diagramm - Unternehmensinterne Logistik

mit einer VWS auf dem dafür notwendigen technischen Stand, weshalb dieser Use Case noch zukunftsorientiert ist.

Die derzeitige Alternative zur VWS ist die Nutzung eines MES sowie der M2M-Kommunikation. Diese Technologien bieten jedoch keine Möglichkeiten zur autonomen Verhandlung zwischen den Akteuren für die Erlangung des effektivsten und effizientesten Transportwegs.

2.2 Use Case Cluster - Wandlungsfähige Fabrik

Die wandlungsfähige Fabrik wird im Ergebnispapier der Plattform Industrie 4.0 [3] mit dem Begriff „Plug and Produce“ gleichgesetzt, der in dieser Arbeit allerdings als einzelner Use Case Beachtung findet (s. Kapitel 2.2.2). Im Allgemeinen handelt es sich bei diesem Use Case Cluster um den schnellen, unkomplizierten und teilweise automatisierten Austausch von Produktionsmodulen als Reaktion auf regelmäßige Prozessumstellungen. Die Produktionssysteme sollen sich bei An- bzw. Abkopplung von Produktionsmodulen weitestgehend automatisch in ihrem Produktionsablauf anpassen [3, S. 10 f.].

Ein Grund für die Notwendigkeit einer wandlungsfähigen Fabrik liegt laut [3] in den immer kürzer werdenden Produkt- und Innovationszyklen, die statische Anlagen mit einer Lebensdauer von mehr als 25 Jahren nicht praktikabel werden lassen. Weiterhin ist mit der Flexibilität auch eine optimalere Produktionsauslastung bei wachsender Produktvielfalt verbunden [3, S. 10 f.].

Im Hinblick auf die Wertschöpfungsketten des produzierenden Unternehmens verlagern sich [3] zufolge einzelne Wertschöpfungsprozesse in den Bereich der Maschinenhersteller, die dem produzierenden Unternehmen ihre Produktionsmodule anbieten. Der Tätigkeitsbereich des Systemintegrators im produzierenden Unternehmen, der die technische Integration der verschiedenen Produktionsmodule umfasst, kann durch diesen Wandel abkömmlich werden.

Festgelegte Standards ermöglichen es dem Maschinenhersteller, Schnittstellen für die Interoperabilität mit Maschinen anderer Hersteller bei Auslieferung des Produktionsmoduls bereitzustellen. Damit nimmt der Systemintegrator im produzierenden Unternehmen nur noch eine organisatorische Rolle ein. Der Vorteil für das produzierende Unternehmen, das bisher noch auf einzelne Maschinenhersteller für Produktionslinien angewiesen ist, liegt in der Kombination von Produktionsmodulen verschiedener Hersteller. Des Weiteren ist der reibungslose Ersatz von Produktionsmodulen durch andere Maschinenhersteller der gleichen Technologie möglich [3, S. 11].

Die aufgeführten Use Cases dieses Clusters werden durch die Arbeitsgruppe 2 „Digitale Fabrik“ des Vereins Smart Factory^{KL} erforscht und an dem Prototyp „Industrie 4.0-Produktionsanlage“ bereits umgesetzt [20]. Mit Hilfe des Forschungsberichts [21] der Plattform Industrie 4.0 und dem RAMI4.0-Modell veranschaulicht das Whitepaper der Smart Factory^{KL} [20] die Implementierung eines modularen Produktionssystems. Die Rekonfiguration des Produktionssystems, durch den Austausch von Produktionsmodulen, geschieht über die Anwendung des „Plug and Produce“ für Produktionsmodule. Die exemplarische Beschreibung in dem Whitepaper, von notwendigen Teilmodellen und deren Elementen, ermöglicht einen tieferen Einblick in die Funktionsweise eines Modultauses [20].

2.2.1 Modularisierung von Produktionssystemen

Produktionslinien sind bereits modulbasiert konstruiert, mit dem Aspekt, dass ein aufwandsarmer Tausch von Modulen nur durch genau baugleiche Module möglich ist. Dieser Austausch findet statt, sobald ein Modul defekt ist oder es einer Wartung unterzogen werden muss. Die Zuständigkeiten dafür liegen meistens im Bereich der Instandhaltung. Aus der bereits existierenden Modulbauweise, die laut [4] Voraussetzung für diesen Use Case ist, wird die Gestaltung einer wandlungsfähigen Fabrik vorstellbar. Die Motivation hinter der Umgestaltung einer starren Fabrik in eine wandlungsfähige Fabrik lässt sich vor allem auf die steigende Anzahl von Produktvarianten, als Resultat der Beachtung individueller Kundenwünsche, zurückführen. Des Weiteren spielt die schnelle Einführung der Produkte am Markt eine immer größer werdende Rolle, die durch starre Produktionssysteme derzeit eingeschränkt wird [4, S. 41 f.].

Der TC 65 [4] zufolge existieren in einem produzierenden Unternehmen viele Produktionssysteme unterschiedlicher Maschinenhersteller, die wiederum elektronische Komponenten und Steuerungen unterschiedlicher Komponentenhersteller beinhalten. Das hat zur Folge, dass ein Systemintegrator jedes Produktionssystem individuell betrachten und in das übergeordnete Leitsystem integrieren muss [4, S. 41 f.]. Das führt zu erhöhten Stillstandszeiten, sobald der Austausch eines Produktionsmoduls stattfinden soll.

Die beschriebene Situation ist jedoch klar von dem Use Case „Plug and Produce für Feldgeräte“ (s. Kapitel 2.2.2) abzugrenzen. Bei „Plug and Produce für Feldgeräte“ steht die vollautomatisierte An- bzw. Abkopplung von Feldgeräten im Vordergrund [3, S. 10], während es sich in diesem Use Case um die vereinfachte Integration von Produktionsmodulen in ein übergeordnetes Leitsystem handelt [4, S. 41 f.].

Abgegrenzt von der auftragsgesteuerten Produktion, die ihren Mehrwert im SCM des produzierenden Unternehmens hat, verknüpft die Modularisierung von Produktionssystemen die Wertschöpfungsketten unterschiedlicher Stakeholder. Von dem Produktdesign beim Maschinenhersteller (PLM) über die Produktionsplanung und der Produktionstechnik (PSLM) des produzierenden Unternehmens, erzeugt dieser Use Case einen Mehrwert für die beteiligten Akteure [3, S. 6].

Für die Übergabe eines Produktionsmoduls vom Maschinenhersteller an das produzierende Unternehmen und einer vereinfachten Integration ist in diesem Use Case laut [4] eine standardisierte Schnittstellenbeschreibung für das Produktionsmodul notwendig. Die Schnittstelle wird vom Hersteller mit Informationen beschrieben und bei Anlieferung an den Kunden (das produzierende Unternehmen) in einem Produktionssystem-Engineeringtool veröffentlicht. Die Informationen der standardisierten Schnittstelle [4, S. 43] umfassen Eigenschaften wie:

- Informationen bezüglich der Material und Energieversorgung sowie dazugehörige mechanische Anschlüsse
- physische und logische Schnittstellen zu Informationen, wobei logische Schnittstellen die Semantik und den Interaktionsablauf beschreiben
- Parametrierungsmöglichkeiten
- Bedienbilder mit Maschineninformationen für die Integration in ein übergeordnetes Human-Machine-Interface (HMI)
- Anforderungen an die Wartung
- Konfigurationsvorgaben und technische Fähigkeiten
- Modelle wie Prozess- und Betriebsmittelbeschreibungen
- Mehrsprachigkeit der Schnittstellenbeschreibung

Die technische Perspektive wird durch die TC 65 [4] wie folgt beschrieben. Für das Design eines Produktionssystems nutzt der Produktionssystemarchitekt das Produktionssystem-Engineeringtool. Zum Design gehört auf der einen Seite die Erstellung eines Layouts für das Produktionssystem und auf der anderen Seite die Prozessbeschreibung und die damit verbundenen Anforderungen an das anpassungsfähige Produktionssystem. Durch

die Trennung in Layout und Prozess wird das Hinzufügen, Ersetzen oder Entfernen von Produktionsmodulen vereinfacht und die Modularität deutlich. Mit Fertigstellung des Designs ist der Produktionssystemarchitekt in der Lage, die Bestellung der Produktionsmodule zu veranlassen. Dafür zieht er wieder das Produktionssystem-Engineeringtool zur Hilfe, in dem Maschinenhersteller die Schnittstellenbeschreibungen ihrer Produktionsmodule veröffentlichen. Anhand eines Vergleichs der veröffentlichten Schnittstellenbeschreibungen mit seinen eigenen Anforderungen kann der Produktionssystemarchitekt das für ihn zweckmäßige Produktionsmodul ordern. Bei Ersetzen oder Hinzufügen von Produktionsmodulen in ein bestehendes Layout ist das Vorgehen das gleiche [4, S. 43 f.].

Mit der Anlieferung des Produktionsmoduls durch den Maschinenhersteller gibt dieser die auf das Produktionsmodul angepasste Schnittstellenbeschreibung nochmals an den Produktionssystemarchitekten weiter. Die Informationen der Schnittstelle werden im übergeordneten Leitsystem (der Integrationsschicht) der Produktionsressourcen verarbeitet. Die Integrationsschicht wird bei Erstinstallation des Produktionssystems durch den Systemintegrator integriert, da dieser die Produktionsmodule auch hardwareseitig zu einem Produktionssystem verbindet. Bei Anpassung des Produktionssystems hingegen wird die Integrationsschicht automatisiert durch das Produktionssystem-Engineeringtool überarbeitet (ebd.). Die gesamte technische Perspektive für den Use Case ist in Abbildung 2.6 zu sehen.

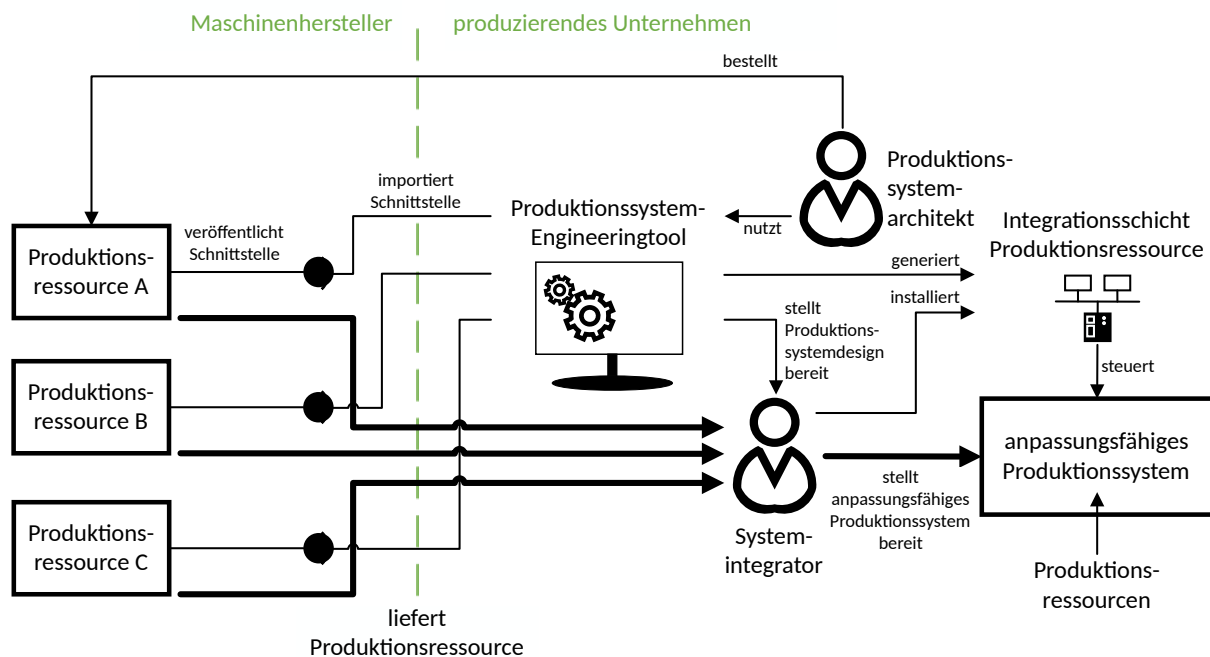


Abbildung 2.6: Technische Perspektive des Use Case „Modularisierung von Produktionssystemen“ (in Anlehnung an [4, S. 42])

Für die Integration der I4.0-Komponente in diesen Use Case ist die Verwendung einer reaktiven VWS von Vorteil, da in diesem Fall die VWS keine Kommunikation zu anderen VWS herstellen muss. Die VWS nimmt die Rolle des Informationsvermittlers ein. Sie

agiert unternehmensübergreifend, da sie alle Informationen der standardisierten Schnittstellenbeschreibung des Maschinenherstellers enthält. Die beteiligten Akteure schließen insbesondere den Maschinenhersteller und das Produktionssystem-Engineeringtool ein. Der in der technischen Perspektive (s. Abbildung 2.6) dargestellte Produktionssystemarchitekt und Systemintegrator spielen bei der Verwendung der VWS eine implizite Rolle. Grund dafür ist der alleinige Zugriff des Produktionssystem-Engineeringtools auf die Informationen der VWS.

Für die Umsetzung enthält die VWS Teilmodell-Templates, die sich noch in der Entwicklung oder Planung durch die IDTA befinden. In der Zukunft kann jedoch ein Großteil der im Anhang A aufgelisteten Teilmodell-Templates zum Einsatz kommen. Dazu zählen z.B. die Teilmodell-Templates „Maintenance“, „Simulation“ und „Functional Safety and Reliability“. Aufgrund der Vielfalt der Teilmodelle (TM) sind nur einige Beispiele im Use Case Diagramm (Abbildung 2.7) zu sehen.

Die Teilmodelle müssen spezifiziert werden und sowohl bei den Herstellern der Produktionsmodule als auch den produzierenden Unternehmen Akzeptanz finden, was den Use Case stark zukunftsorientiert werden lässt.

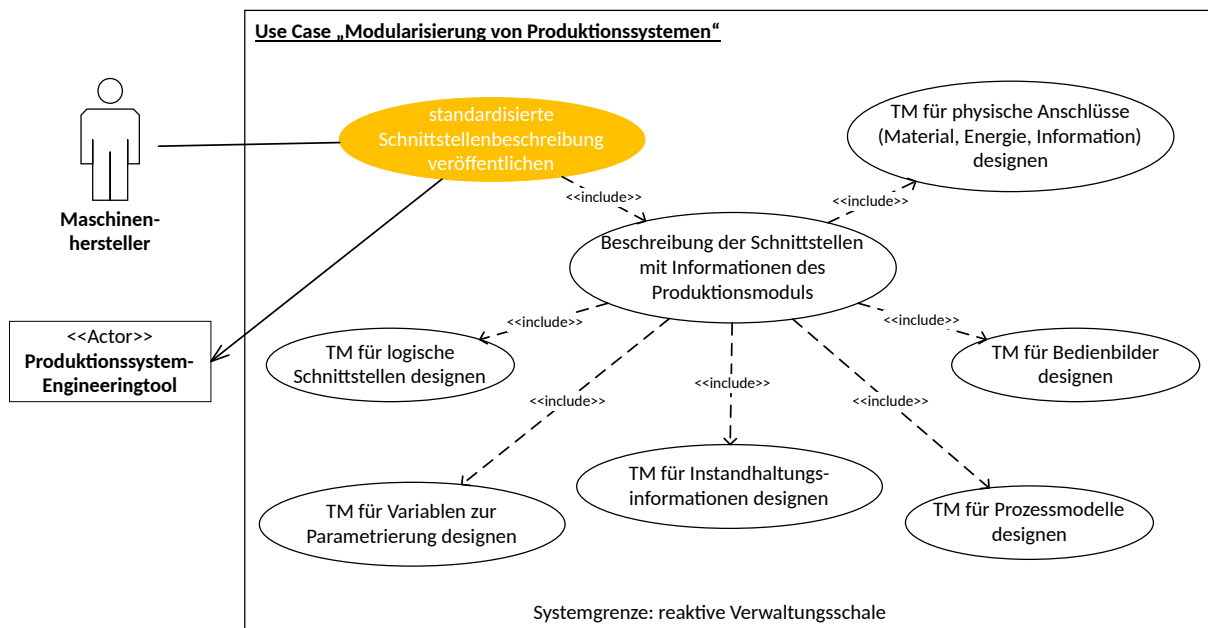


Abbildung 2.7: Use Case Diagramm - Modularisierung von Produktionssystemen

2.2.2 Plug and Produce für Feldgeräte

Der Use Case „Plug and Produce“ beschreibt laut [3] einen Austausch von Produktionsressourcen in der Industrie, ähnlich dem Austausch eines USB-Datenspeichers an einem Computer. Das heißt, die Abkopplung eines Gerätes und die Ankopplung eines neuen Gerätes wird automatisch erkannt, wobei das neue Gerät entsprechend seiner Beschreibung im überlagerten System automatisiert integriert und konfiguriert wird. Der

Austausch kann sich sowohl auf Feldgeräte, steuerungstechnische Funktionen oder ganze Produktionsmodule beziehen. Bei dem Wechsel eines Feldgeräts muss dieses automatisch in das Kommunikationsnetz integriert und parametrierbar werden. Die Weiterleitung steuerungstechnischer Änderungen in überlagerten Leitsystemen muss an alle betreffenden unterlagerten Steuerungssysteme stattfinden. Der Austausch von Produktionsmodulen als Reaktion auf Prozessänderungen ist ohne starke Beeinträchtigungen der gesamten Produktion zu gewährleisten [3, S. 10 f.]. Damit sind nur einige Beispiele des Use Case genannt. Im Zusammenhang mit der VWS als konzeptionelle Lösung wird im Folgenden der Feldgerätetausch näher beschrieben.

Ein manueller Feldgerätetausch nimmt Zeit in Anspruch, in der das Produktionssystem keine Leistung erbringt und somit Verluste für das produzierende Unternehmen entstehen. Mit dem automatisierten Feldgerätetausch sollen diese Stillstandszeiten minimiert werden und zu einem flexibleren Produktionsprozess führen [21, S. 10]. Beispielsweise können intelligente Feldgeräte wie Laser oder Drucker zur Beschriftung von Produkten, je nach Produktvariante, ohne größere Verzögerungen gewechselt werden [3, S. 10 f.]. Hiermit wird die Kopplung zwischen den verschiedenen Use Cases ersichtlich, die letztendlich die Möglichkeit zur Herstellung individueller Produkte bietet.

Hinweis: Die Einordnung des Use Case „Plug and Produce“ kann in unterschiedlichen Use Case Clustern erfolgen. Die AG2 der Plattform Industrie 4.0 ordnet den Use Case dem Cluster der wandlungsfähigen Fabrik zu [3, S. 10 f.].

Die TC 65 der IEC hingegen ordnet den Use Case in den Bereich der IT-Infrastruktur und Software zu und nennt diesen Use Case „Gerätekonfiguration“ [4, S. 121 ff.]. Die TC 65 verbindet damit vor allem die Softwareanwendungen, die Informationen der Feldgeräte verarbeiten und unabhängig vom Feldgerätehersteller funktionieren sollen. Für die konkrete Realisierung hinsichtlich der Gerätekonfiguration verweist die TC 65 auf einen Forschungsbericht der Plattform Industrie 4.0. Dieser Forschungsbericht [21] der in Zusammenarbeit der Plattform Industrie 4.0 mit dem ZVEI zu „Plug and Produce“ entstand, zeigt mit Hilfe von Modellen, technologischen Mappings und Lösungsansätzen detailliert den Vorgang eines solchen Gerätetausches.

Aufgrund der genannten Situation wird der Use Case in dieser Arbeit formell in das Cluster der wandlungsfähigen Fabrik eingeordnet (s. Tabelle 2.1).

Der Ablauf zur automatischen Erkennung, Integration und Konfiguration von Feldgeräten gliedert sich im Forschungsbericht [21] in sechs Schritte. Zu Beginn ist die physische Verbindung zwischen dem Feldgerät und dem Kommunikationsnetz herzustellen. Dafür kann es notwendig sein, laufende Funktionen oder Systeme anzuhalten, um einen sicheren Wechsel oder Neuanschluss zu gewährleisten.

Anschließend muss die Ankopplung des Geräts in das Netz von z.B. einem Gerätemanagementsystem erkannt werden. Das ist unter anderem mittels einer dynamischen Na-

mensauflösung oder einer Abfrage auf freie Adressen möglich. Mit der Abfrage erhält das Gerätemanagementsystem eine Information, sobald ein neues Gerät eine Adresse belegt. Wurde das Gerät im zweiten Schritt mit seiner eindeutigen Adresse im Netz erkannt, sind erste Nachrichtenaustausche möglich. Dabei wird meistens die herstellerunabhängige Gerätebeschreibung des neuen Geräts an das Gerätemanagementsystem übertragen. Mit der Gerätebeschreibung ist das Gerätemanagementsystem im vierten Schritt imstande, Einblicke in Kommunikationsrandbedingungen, Variablen mit deren Beschreibungen, Ein- und Ausgänge sowie bereitgestellte Funktionalitäten des Geräts zu nehmen. Mit den veröffentlichten Funktionalitäten des Geräts kann das Gerätemanagementsystem anschließend einen Abgleich der angebotenen und benötigten Funktionalitäten vornehmen.

Nach der Bewertung der Fähigkeiten kann das neue Gerät, durch die Einbindung der Geräteinformationen in das bestehende Netz, die Echtzeitkommunikation aufnehmen. Für die Informationsverarbeitung in überlagerten Systemen müssen eventuelle Anpassungen an Anlageninformationsmodellen sowie die Parametrierung des Geräts vorgenommen werden. Die Parametrierung kann dabei entweder manuell durch einen Systemintegrator geschehen oder (teil-)automatisiert durch die Ableitung von Parametern aus den übergeordneten Informationsmodellen.

Auf die erfolgreiche Integration des Feldgeräts in das Echtzeitkommunikationsnetz folgt der letzte Schritt, die Rekonfiguration von Systemen der Leitebene (z.B. Prozessleitsysteme) und der Betriebsleitebene (z.B. Ressourcenverwaltungssysteme) [21, S. 11]. Das Diagramm, das die beschriebenen Schritte und beteiligten Akteure für den gesamten Use Case beschreibt, ist in Abbildung 2.8 detailliert zu sehen.

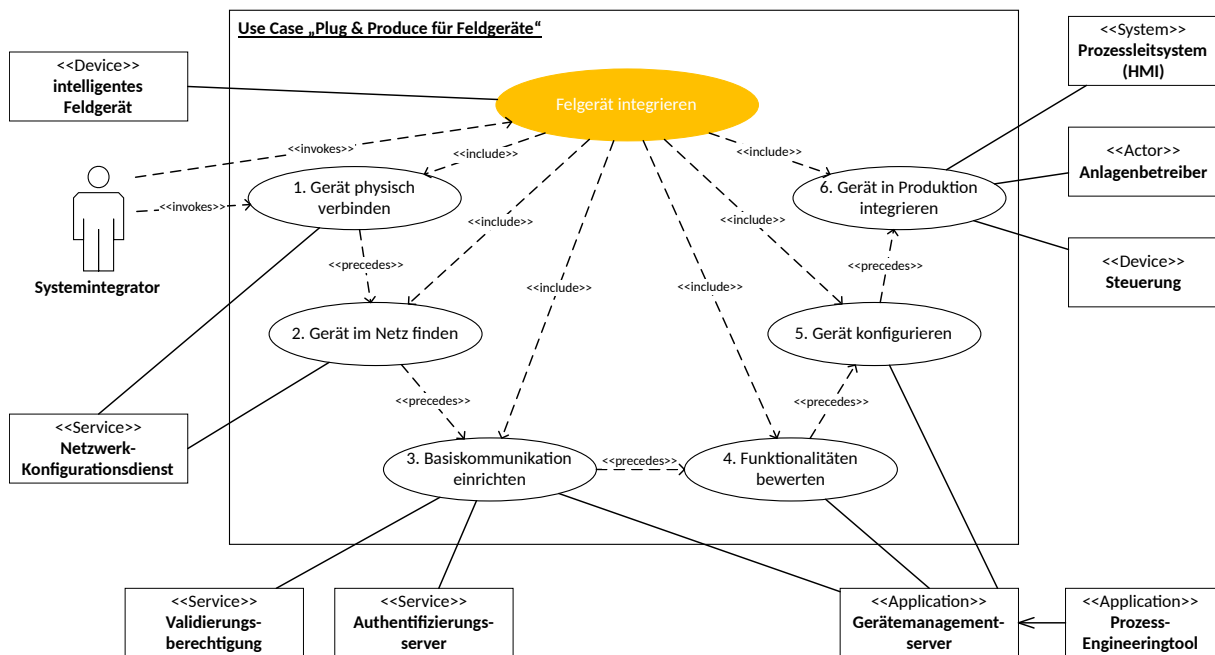


Abbildung 2.8: Use Case Diagramm - Plug and Produce für Feldgeräte (in Anlehnung an [21, S. 12])

In diesem Use Case erhalten das Feldgerät, das Gerätemanagementsystem, die Steuerung, das Prozessleitsystem und das Prozess-Engineeringtool eine VWS. Für die Umsetzung der Interaktionen zwischen den Akteuren müssen proaktive VWS eingesetzt werden. Während das Gerätemanagementsystem und die Steuerung ausschließlich Kommunikationsdienste bereitstellen, enthält das Feldgerät neben Kommunikationsdiensten auch Informationen über seine Eigenschaften sowie Dienste für den Lese- und Schreibzugriff auf die gespeicherten Eigenschaften [21, S. 29 f.]. Eine beispielhafte Interaktion zwischen den genannten Akteuren ist im Forschungsbericht [21] durch ein UML-Sequenzdiagramm (s. [21, S. 31 f.]) sehr gut veranschaulicht.

2.3 Verwaltung von Assets

Die Verwaltung eines Assets ist die Kernaufgabe einer VWS und somit ist dieser Use Case prädestiniert für den Einsatz von VWS. Er zeigt die Leistungsfähigkeit der VWS über den Lebenszyklus eines Assets hinweg.

Die verwalteten Assets können gemäß der TC 65 [4] sowohl physisch als auch virtuell existieren. Jeder Unternehmensbereich stellt Informationsmodelle des Assets mit Hilfe von Engineeringtools bereit. Die Informationsmodelle werden mitunter als Asset-Typen implementiert und erzeugen durch eine nachfolgende Instanziierung konkrete Asset-Instanzen [4, S. 56].

Die individuellen Informationen, die über den gesamten Lebenszyklus eines Assets entstehen, werden in der Asset-Instanz fortlaufend aktualisiert (s. Use Case „virtuelle Repräsentation von physischen Assets“ [4, S. 59–62]). Der Zusammenschluss aller existierenden Assets, mit deren VWS, erfolgt in einem Assetmanagementsystem [4, S. 56–59].

Der Mehrwert des Use Case zeigt sich vor allem in den Wertschöpfungsketten, in denen dieser eingesetzt werden kann. Alle vier Wertschöpfungsketten, von dem PLM und PSLM über das SCM bis hin zum Service, profitieren von einem Assetmanagementsystem und entwickeln sich durch dieses gleichzeitig zu einem zusammenhängenden Wertschöpfungsnetzwerk [4, S. 23]. Die Informationen der Assets werden einheitlich erfasst und ausgewertet. Voraussetzung für die Verwirklichung eines Assetmanagementsystems ist allerdings ein einheitlicher Ansatz zur Speicherung der Daten im produzierenden Unternehmen [4, S. 56–59].

Viele der im Use Case Katalog [4] dokumentierten Use Cases nutzen das Assetmanagementsystem als Datenbasis weiterer Datenanalysen und sind ebenso wie dieser Use Case dem Cluster „Assetmanagement“ zugeordnet. Auf diese Use Cases wird zu Ende des Kapitels kurz eingegangen.

Die Informationsmodelle für das Assetmanagement kommen aus verschiedenen Engineeringtools, die in den verschiedenen Wertschöpfungsprozessen zur Speicherung von Assetinformationen genutzt werden. Beispiele solcher Informationsmodelle werden durch die technische Perspektive der TC 65 [4] genannt und im Folgenden beschrieben.

Beginnend bei der Produktentwicklung legt der Produktentwickler durch die Nutzung des Produktentwicklungstools die Speicherung seiner Ergebnisse als virtuelles Asset fest. Dazu gehören z.B. Konstruktionsmodelle, Simulationsergebnisse und technische Details. Aber auch Einträge bezüglich des Asset-Lebenszyklus wie Gantt-Diagramme, deren aktueller Stand in den Assetinformationen aufgenommen werden, repräsentieren den Asset-Typ in der Produktentwicklung.

Mit der Einführung eines neuen Produkts muss das Design eines Produktionssystems für die Fertigung erfolgen. Die Zuständigkeit hierfür liegt bei einem internen bzw. externen Produktionssystemarchitekten, der mit Hilfe des Produktionssystem-Engineeringtools ein Gesamtlayout designt. Weiterhin wählt er Produktionsmodule gemäß den Fertigungsanforderungen aus und erstellt Pläne für die spätere Instandhaltung. Diese Informationen werden im Assetmanagementsystem abgelegt. Für den Lebenszyklus des Produktionssystems werden Fortschritte bei der Errichtung gespeichert, von der Planung über die Beschaffung und Lieferung bis zur Inbetriebnahme. Aber auch die zeitliche Dokumentation der durchgeführten Wartungen gemäß dem Wartungsplan ist Bestandteil des Asset-Lebenszyklus. Bezüglich der Modularisierung von Produktionssystemen (s. Kapitel 2.2.1) gehört auch die Rekonfiguration von Produktionssystemen dazu.

Der für die Inbetriebnahme verantwortliche Systemintegrator dokumentiert in einem vom Produktionssystemarchitekten konfigurierten Tool alle Vorgänge während der Montage und Inbetriebnahme des Produktionssystems. In dem bereitgestellten Tool wird die vollständige Hinterlegung der Daten im Assetmanagementsystem garantiert.

Nach Errichtung des Produktionssystems und mit dem Beginn der Produktion des neuen Produkts trägt der Produktionskoordinator über das Produktionsmanagement ebenfalls Informationen zum Assetmanagementsystem bei. Zu den Assets zählen Fertigungsaufträge, Prüfprotokolle aus der Qualitätssicherung, aber auch Dokumente von Zulieferteilen und schließlich Informationen und Dokumentationen zum Produkt selbst. Während des Lebenszyklus, z.B. bezogen auf das herzustellende Produkt, wird der Fertigungsstand, der Weg durch das Produktionssystem (welche Produktionsressource war an der Produktion beteiligt) etc. dokumentiert. Aber auch die Erfassung des Lebenszyklus von Betriebsstoffen, wie der Zeitpunkt der Lieferung und die Verfolgung bis zum Produktionssystem, das den Betriebsstoff verwertet, ist über das Assetmanagementsystem möglich. Durch die Kopplung des Assetmanagementsystems mit einem ERP-System und einem MES ist die Aktualisierung der Daten in diesen Systemen denkbar.

Mit Beginn der Produktion sind unter anderem wiederkehrende Anpassungen und Beobachtungen von Anlagenparametern, die Behebung und Dokumentation von kleineren

Störungen, aber auch die Aufnahme der Stückzahlen notwendig. Die genannten Informationen werden durch die Tätigkeiten des Anlagenführers gewonnen. Die Strukturierung solcher Nutzungsinformationen liegt jedoch nicht in der Verantwortlichkeit des Anlagenführers. Vielmehr übernehmen der Produktionssystemarchitekt und der Produktionskoordinator die Erstellung von entsprechenden Informationsmodellen. Für größere Störungen oder zyklisch wiederkehrende Wartungen der Produktionssysteme ist das Service-/Instandhaltungspersonal, sowohl aus dem produzierenden Unternehmen als auch von den Maschinenherstellern, verantwortlich. Eine zweckmäßige Dokumentation der durchgeführten Wartungsarbeiten erfolgt über Systeme zur Instandhaltungsverwaltung. Die darin gespeicherten Informationen werden anschließend im Assetmanagementsystem abgelegt. Alle verwendeten Engineeringtools und auch das Assetmanagementsystem werden durch die IT-Administrationsabteilung bzgl. einer Benutzerverwaltung und Zugriffsberechtigungen verwaltet. Jede Informationsänderung soll durch die Engineeringtools unmittelbar mit dem Assetmanagementsystem synchronisiert werden [4, S. 58 f.]. Die gesamte technische Perspektive der beschriebenen Informationsverwaltung ist Abbildung 2.9 zu sehen.

Für die Vergleichbarkeit der Informationen über die Grenzen der einzelnen Wertschöpfungsprozesse hinweg sind unternehmensinterne Standardisierungen unumgänglich. Laut der TC65 zählen zu den Anforderungen an die Standardisierungen [4, S. 59] folgende Punkte:

- Standardisierung der Lebenszykluseinträge eines Assets
- Standardisierung von Assetinformationen sowie deren Bedeutung durch semantische Beschreibungen, die die korrekte Auswertung der Daten für jeden Akteur ermöglichen
- Standardisierung eines unternehmensweiten Konzepts zur einheitlichen Verwendung des Assetmanagementsystems
- branchenspezifische Standardisierung von konkreten Lebenszykluseinträgen (für nähere Informationen siehe Use Case „Standardisierung von Produktionstechnologien“ [4, S. 50–53])

Es ist erkennbar, dass der Aufwand zur Integration eines solchen Assetmanagementsystems gerade durch die aktuell noch überwiegend papierbasierte Dokumentation der Produktion sehr hoch ist. Deshalb ist dieser Use Case, auch bei existierenden Prototypen, stark zukunftsorientiert.

Die Möglichkeit zur Einbindung von VWS in diesen Use Case ist signifikant. Es existiert ein System, in dem Assets durch ihren digitalen Zwilling in einem Assetmanagementsystem verwaltet werden. Die Repräsentation des digitalen Zwillings ist durch eine reaktive VWS realisierbar.

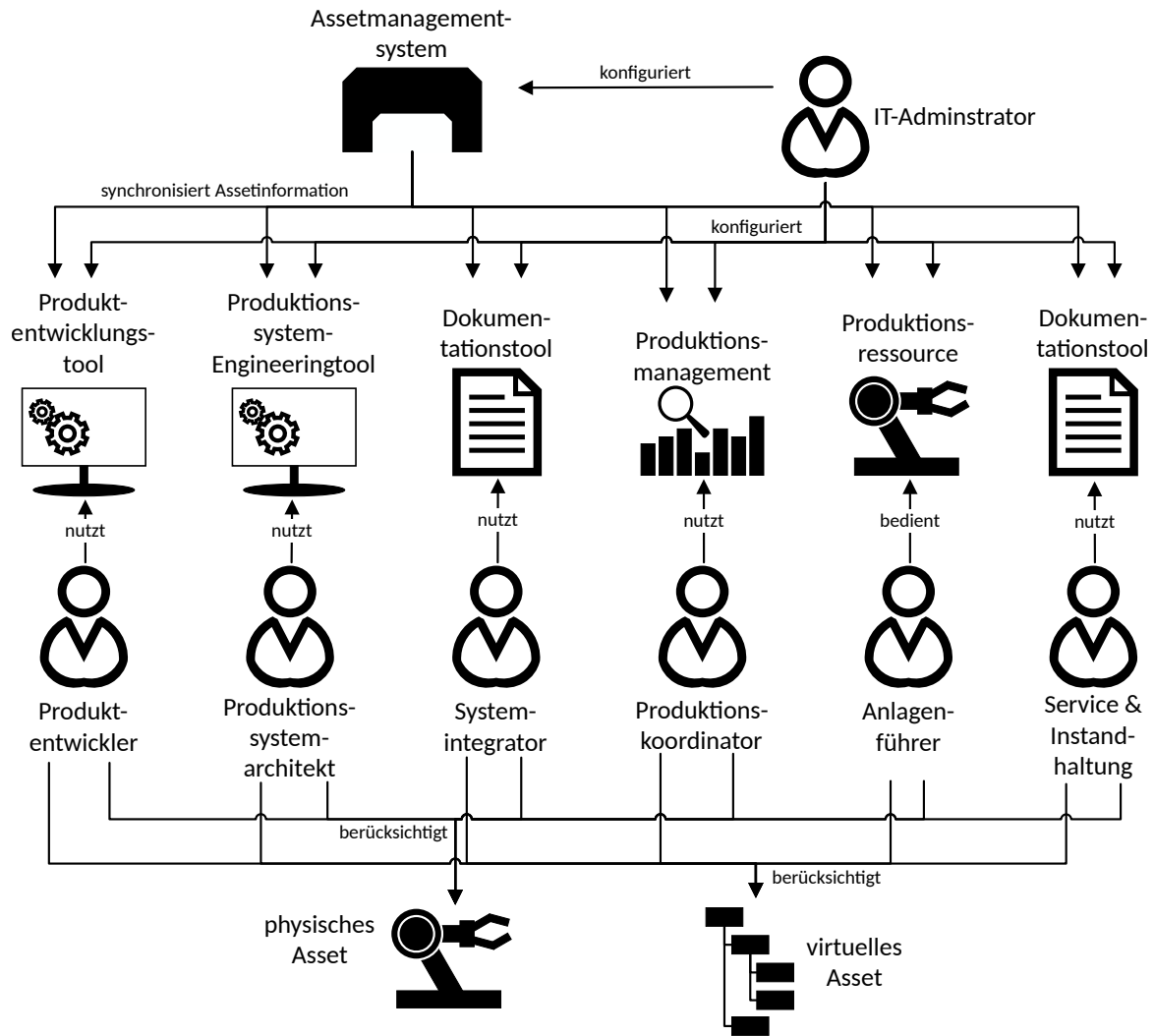


Abbildung 2.9: Technische Perspektive des Use Case „Assetmanagement“ (in Anlehnung an [4, S. 57])

Bereits die veröffentlichten Teilmodell-Templates „ZVEI Digital Nameplate for Industrial Equipment“ und „Generic Frame for Technical Data for Industrial Equipment in Manufacturing“ als auch das sich in der Planung befindliche Teilmodell-Template „Plant Asset Management“ können für diesen Use Case genutzt werden (s. Anhang A). Diese Teilmodell-Templates bilden jedoch nur eine Basis, die durch viele weitere Teilmodelle erweitert werden muss, um die volle Funktion eines Assetmanagementsystems zu gewährleisten.

Im Assetmanagementsystem werden die VWS erzeugt und den Assets zugewiesen. Alle Informationen, die dem Asset neu hinzugefügt, geändert oder entfernt werden, sind durch das Assetmanagementsystem mit der VWS synchron zu halten. Damit ist ein stets aktueller Datenzugriff für alle am Use Case beteiligten Akteure auf die digitale Repräsentation des Assets verfügbar. Die konzeptionelle Einbindung der VWS in den Use Case ist in Abbildung 2.10 dargestellt.

Eine Alternative zur VWS ist ein ERP-System, das zur Verwaltung aller im Unternehmen

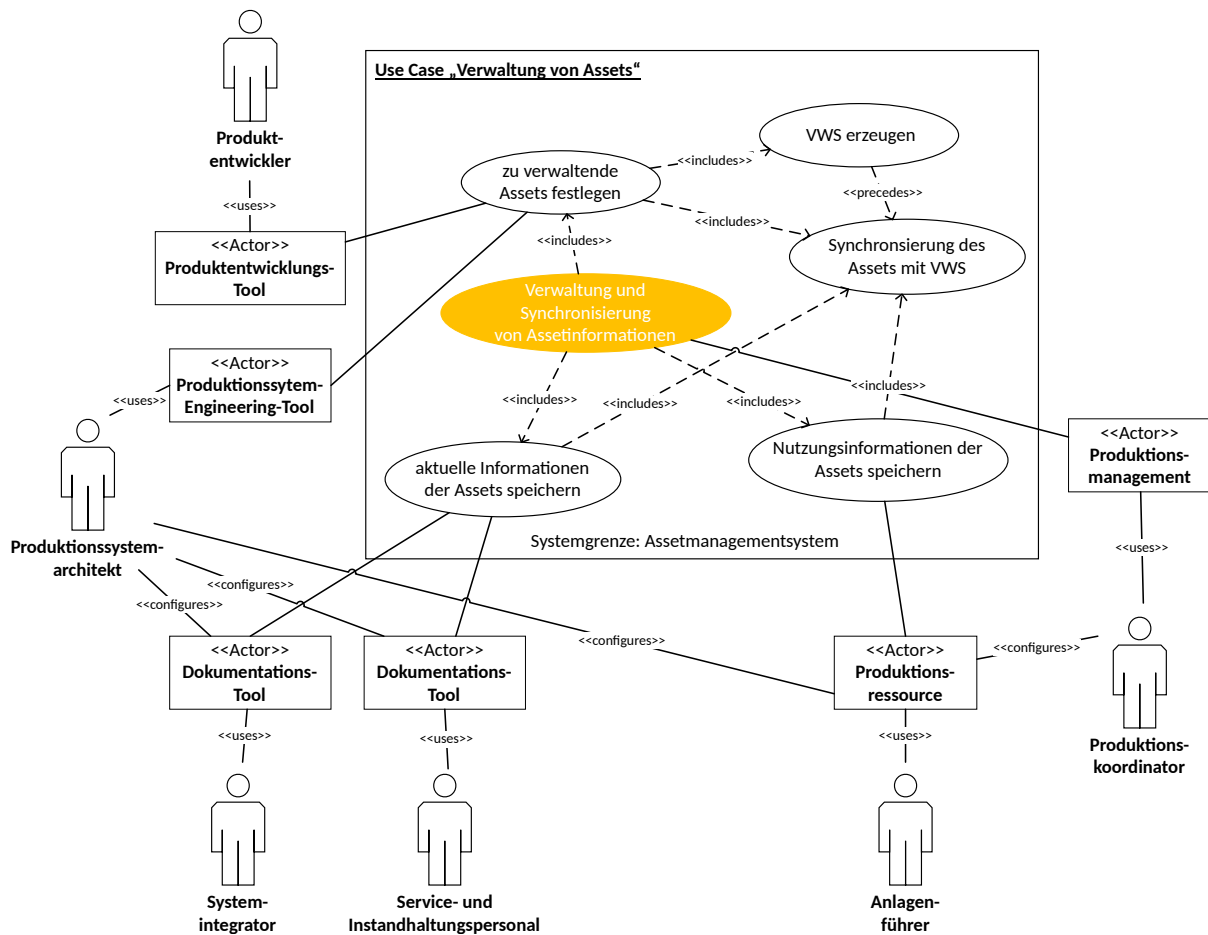


Abbildung 2.10: Use Case Diagramm - Assetmanagement

existierenden Produktionsressourcen eingesetzt wird. Des Weiteren kann ein MES Daten zu den aktuellen Nutzungsinformationen der Produktionsressourcen sowie Ergebnisse von Instandhaltungsdienstleistungen erfassen und an das ERP-System übertragen.

Ist ein solches Assetmanagementsystem im produzierenden Unternehmen vorhanden, kann es als Datenbasis für viele weitere Use Cases, die die TC 65 [4] beschreibt, dienen. In den Use Cases können die Daten entweder aus dem Assetmanagementsystem gelesen oder in dieses geschrieben werden. Zusätzlich zu dem weiter oben erwähnten Use Case „virtuelle Repräsentation von physischen Assets“ sind einige weitere dieser Use Cases im Folgenden genannt. Begonnen wird mit den Use Cases, in denen das Assetmanagementsystem mit weiteren Informationen angereichert wird.

Aktualisierung und funktionale Skalierbarkeit von Produktionsressourcen:

Dieser Use Case beschreibt die Aufrüstung von Produktionsressourcen, sowohl hardware- als auch softwareseitig. Aus der Sicht eines produzierenden Unternehmens soll ein Update einer Produktionsressource möglichst schnell und ohne größere Ausfallzeiten der Ressource ausgeführt werden. Aus der Sicht eines Maschinenherstellers möchte dieser für seine vermarkteten Maschinen diverse Funktionalitäten modulbasiert anbieten und dem Kunden

eine unkomplizierte Implementierung der Funktionen während des Produktionsbetriebs ermöglichen. Jedes dieser vorgenommenen Updates wird für die Rückverfolgbarkeit in dem Assetmanagementsystem zu dem entsprechendem Asset synchronisiert [4, S. 65 ff.].

Condition Monitoring von Produktionsressourcen: Das Konzept des Condition Monitoring wird schon viele Jahre in produzierenden Unternehmen angewandt. Eine kontinuierliche Erfassung von Anlagenparametern und die darauffolgende Analyse der Daten ermöglicht eine frühzeitige Erkennung und somit auch Vermeidung von Störungen, bevor diese tatsächlich eintreten [4, S. 67–70].

Durch die Vielfalt von Produktionsressourcen in einem produzierenden Unternehmen können die Mitarbeiter nicht alle technologischen Abläufe einer Produktionsressource in seinen Details kennen. Deshalb ist es sinnvoll, dass ein Maschinenhersteller grundlegende Analysemethoden für die Vorhersage von Störungen dem produzierenden Unternehmen bereitstellt. Die Ergebnisse der Analysemethoden werden in einem zentralen Zustandsvorhersage-Tool des produzierenden Unternehmens gesammelt und sowohl dem Produktionskoordinator als auch dem Anlagenführer übermittelt. Jede der Vorhersagen wird gleichzeitig mit dem Assetmanagementsystem synchronisiert, um die Informationen auch unternehmensweit nutzen zu können (ebd.).

Selbstoptimierung von Produktionsressourcen: Dieser Use Case wird durch die TC 65 [4] als eine Erweiterung des Condition Monitoring für Produktionsressourcen genannt. Mit Hilfe des maschinellen Lernens werden Daten der Produktionsressourcen genutzt, um diese nutzbringend zu optimieren und Ausfallzeiten zu reduzieren. Die Algorithmen zur Bewertung der Daten werden, wie bei dem Condition Monitoring, vom Maschinenhersteller mitgeliefert. Er verfügt über das Know-How der einzelnen technologischen Vorgänge. Die Ergebnisse der Algorithmen aus dem laufenden Produktionsbetrieb werden vom produzierenden Unternehmen beachtet und etwaige Anpassungen am Produktionssystem vorgenommen. Auch diese Informationen werden zeitgleich im Assetmanagementsystem aktualisiert [4, S. 70 ff.].

Service Workflow Management für Produktionssysteme: Im Produktionsbetrieb fallen wiederkehrende Services an, deren Ausführung und Dokumentation von Bedeutung sind. Dabei geht es nicht nur um einen Instandhaltungsservice oder die Rekonfiguration des Produktionssystems durch den Produktionssystemarchitekten. Tatsächlich geht es um die automatisierte Bearbeitung und Dokumentation von Tätigkeiten, die in verschiedenen Bereichen des Unternehmens, im Zusammenhang mit dem Produktionssystem, auftreten. Zuständig für die zentrale Überwachung ist eine Workflow-Engine. Die beteiligten Akteure können mit ihren spezifischen Engineeringtools standardisierte Serviceanfragen über die Workflow-Engine an andere Akteure senden und deren aktuellen Stand verfolgen. Eine wichtige Datenbasis dafür bildet das Assetmanagementsystem in dem die Ergebnisse der Workflows gespeichert werden [4, S. 80 ff.].

Nachfolgend sind die Use Cases beschrieben, die das Assetmanagementsystem nicht nur mit weiteren Informationen anreichern, sondern auch dessen Informationen nutzbringend verwenden.

Optimierung des Betriebs: Dieser Use Case ordnet sich dem Cluster „Optimierung des Produktionsablaufs“ ein. Er ähnelt dem oben beschriebenen Use Case „Selbstoptimierung von Produktionsressourcen“, wobei es sich hier nicht um die Optimierung einer spezifischen Ressource handelt, sondern um die Optimierung von komplexen Produktionssystemen. Ziel ist die Betriebsoptimierung (auch werksübergreifend) und zusammenhängend damit die Optimierung vorliegender Key Performance Indicators (KPIs). Bezogen auf das Ressourcen-, Instandhaltungs- und Rekonfigurationsmanagement eines Produktionssystems werden Maßnahmen zur Verbesserung der Produktivität getroffen. Eine Datenbasis zur Bildung dieser Maßnahmen ist das Assetmanagementsystem [4, S. 72–75].

Optimierung des Energieverbrauchs: Dieser zum Cluster „Energieeffizienz“ gehörende Use Case ist eine Erweiterung des Use Case „Optimierung des Betriebs“ und hat als Ziel die Senkung des Energieverbrauchs unter Verwendung von KPIs. Die Verantwortlichkeit für die Verbesserung der Energieeffizienz liegt bei dem Produktionskoordinator. Mit Hilfe des Produktionsmanagementsystems und den aktuellen Energiepreisen auf dem Markt, legt der Produktionskoordinator Maßnahmen zur Energieeinsparung fest. Beispiele für solche Maßnahmen sind die Vermeidung von Energiespitzen oder die Aktivierung eines Stand-By-Modus für Produktionsressourcen. Die dafür notwendigen Informationen erhält das Produktionsmanagementsystem unter anderem von der laufenden Produktion, dem Assetmanagementsystem und der Anfrage-/Auftragsverwaltung zur Analyse der zukünftigen Produktionsauslastung [4, S. 88 ff.].

Beteiligung an dezentralen Energienetzen: Zusätzlich zu dem Use Case „Optimierung des Energieverbrauchs“ zieht der Produktionskoordinator auch die Verträge mit den Energielieferanten für seine Entscheidungsfindung zu Hilfe. Dabei betrachtet der Produktionskoordinator nicht wie zuvor den aktuellen Energiepreis sondern den zeitlichen Verlauf dessen. Mit den gewonnenen Erkenntnissen aus den zeitlichen Betrachtungen und den vorliegenden Vertragslaufzeiten kann er Maßnahmen für eine Gesamtoptimierung festlegen [4, S. 93 f.].

Unterstützung bei der taktischen und strategischen Entscheidungsfindung: Das Assetmanagementsystem kann in diesem Use Case unterstützend bei der Entscheidungsfindung bzgl. der Konfiguration einer Betriebsinfrastruktur dienen. Die Daten aus dem Assetmanagementsystem werden von einer Software verarbeitet, die die Entscheidungsfindung für verschiedene Gruppen von Akteuren erleichtert. Die Konfigurationsplanung kann sich in diesem Fall auf die aktuelle Situation als auch auf eine zukünftige Situation

beziehen. Eine Relation zu diesem Use Case bildet im operativen Bereich die „Optimierung des Betriebs“ [4, S. 143–146].

Ein Prototyp des Assetmanagementsystems wurde durch die Zusammenarbeit der Unternehmen Lenze SE und HEGLA GmbH & Co. KG realisiert. Die IDTA stellt auf ihrer Website [22] einen Überblick über das entwickelte Assetmanagementsystem bereit. In dem Assetmanagement enthalten sind unter anderem Informationen zu den verbauten Produktionsressourcen sowie deren Zustandsgrößen und Prozessdaten. Alle Produktionsressourcen werden automatisch und herstellerunabhängig in das Assetmanagementsystem abgebildet. Weiterhin kann für ein Assetmanagementsystem, mit Blick auf eine umweltschonende Produktion, der CO₂-Fußabdruck eines Produkts dokumentiert werden. Die IDTA beschreibt auf ihrer Website den Use Case „Product Carbon Footprint“ [23], der sich bisher noch in der Entwicklungsphase befindet. Die Funktionen des Use Cases umfassen beispielsweise eine Berechnung ausgestoßener Treibhausgase, zur Erleichterung der Einhaltung von EU-Vorlagen.

2.4 Value-Based Services für Produktionsressourcen

Hersteller von Produktionsressourcen (z.B. Aktoren, Sensoren, Maschinen) möchten ihren Kunden gerne Dienstleistungen anbieten, die unter anderem die Produktivität der eingesetzten Produktionsressourcen erhöhen [4]. Die Idee dieses Use Case basiert auf dem Angebot von bedarfsgerechten Services über eine Softwareapplikation. Die Applikationen können von den Kunden selbst durchgeführt oder alternativ von Servicemitarbeitern des Herstellers aus der Ferne gesteuert werden [4, S. 113]. Ein solcher Service würde bei einer Produktionsmenge von beispielsweise einer Million Produktionsressourcen pro Jahr, eine Betreuung jedes einzelnen Kunden ermöglichen. Die Serviceabteilung des Herstellers wäre aufgrund der begrenzten Personalkapazitäten dazu nicht in der Lage.

Zu den Services zählen laut der TC 65 [4] z.B. automatisierte Diagnosemöglichkeiten zur Feststellung von verschlissenen Komponenten und einer damit verbundenen Ersatzteillieferung. Weiterhin ist ein Fernzugriff auf die Nutzungsdaten der verkauften Produktionsressourcen durch den Hersteller vorstellbar, um die Produktqualität zu optimieren. Besonders der Fernzugriff durch den Hersteller auf die im Produktionssystem integrierte Ressource benötigt eine besondere Aufmerksamkeit im Bereich der IT-Security. Vertragliche Regelungen zwischen dem Hersteller und dem Kunden bilden die Basis, in der Regulierungen von Zugriffsrechten auf definierte Datenbereiche und deren Nutzung vereinbart sind [4, S. 112 f.].

Der genaue Inhalt der Services ist in diesem Use Case nicht beschrieben, aber Beispiele sind in den Use Cases „Condition Monitoring von Produktionsressourcen“ (s. [4, S. 67–70]) oder „Selbstoptimierung von Produktionsressourcen“ (s. [4, S. 70 ff.]) zu finden.

Den Mittelpunkt der technischen Perspektive bildet laut [4] eine IIoT-Plattform, auf der durch den Kunden erzeugte Nutzungsdaten der Produktionsressourcen gespeichert und an den Value-Based Service (VBS) weitergegeben werden. Die Verbindung zur IIoT-Plattform wird durch den Systemintegrator eingerichtet.

Der vom Hersteller bereitgestellte VBS wird von einem Softwareentwickler designt und durch Verwendung eines Softwareentwicklungstools erzeugt. Hilfe bei der Realisierung des VBS bekommt der Softwareentwickler durch Softwareapplikationen zur Mustererkennung. Diese Softwareapplikationen werden durch einen Datenwissenschaftler entworfen und deren Realisierung vom selben Softwareentwicklungstool übernommen. Die Entwickler des VBS können sowohl bei dem Hersteller der Produktionsressourcen angestellt sein als auch ihre Tätigkeiten als externe Dienstleister bereitstellen.

Sobald die Produktionsressource im Produktionssystem des produzierenden Unternehmens integriert ist, ist die Nutzung der VBS für Produktionssystemarchitekten, Produktionskoordinatoren und Anlagenführer freigegeben. Die Häufigkeit der Verwendung eines Services kann automatisiert in der IIoT-Plattform dokumentiert und über die vertraglichen Regelungen automatisiert in Rechnung gestellt werden [4, S. 114 f.].

In Abbildung 2.11 ist die beschriebene technische Perspektive zu sehen. Rückmeldungen durch den Kunden, die in der Abbildung nicht zu sehen sind, erlauben die stetige Optimierung des VBS und der unterstützenden Softwareapplikationen zur Mustererkennung (ebd.).

Die Einführung von Value-Based-Services vereint drei Akteure miteinander, die jeder für sich ihren Nutzen aus der Kooperation ziehen. Auf der einen Seite steht der Anbieter der IIoT-Plattform, der durch die enge Bindung mit dem Hersteller der Produktionsressource seine Dienste ausbauen kann. Das hilft dem Anbieter der IIoT-Plattform, mehr Kunden für sich zu gewinnen [3, S. 15], [4, S. 114 f.].

Auf der anderen Seite stehen zum einen der Hersteller, der durch die Nutzungsinformationen seiner Kunden die Möglichkeit erhält die eigene Produktpalette zu optimieren und auszuweiten und folglich seinen Nutzen in den Wertschöpfungsketten PLM und Service zu erhöhen. Zum anderen hat der Kunde durch Inanspruchnahme der Services die Gelegenheit seine Produktion effizienter zu gestalten und somit einen Mehrwert in seinen Wertschöpfungsketten, dem PSLM und SCM, zu schaffen [3, S. 15].

Eine Voraussetzung für die Umsetzung dieses Use Case ist laut der TC 65 [4] die Anbindung der Produktionsressourcen an eine IIoT-Plattform. Solche Plattformen, die ihren Kunden verschiedene Dienste und Apps anbieten, existieren bereits auf dem Markt. Durch

die zunehmende Implementierung der Dienste können diese zu immer günstigeren Preisen am Markt angeboten werden [4, S. 115]. Anbieter für IoT-Plattformen sind unter anderem die Konzerne Microsoft mit Azure, Amazon mit AWS und viele weitere wie Schneider Electric mit EcoStruxure, Bosch und Siemens für den industriellen Bereich (IIoT-Plattformen) [24].

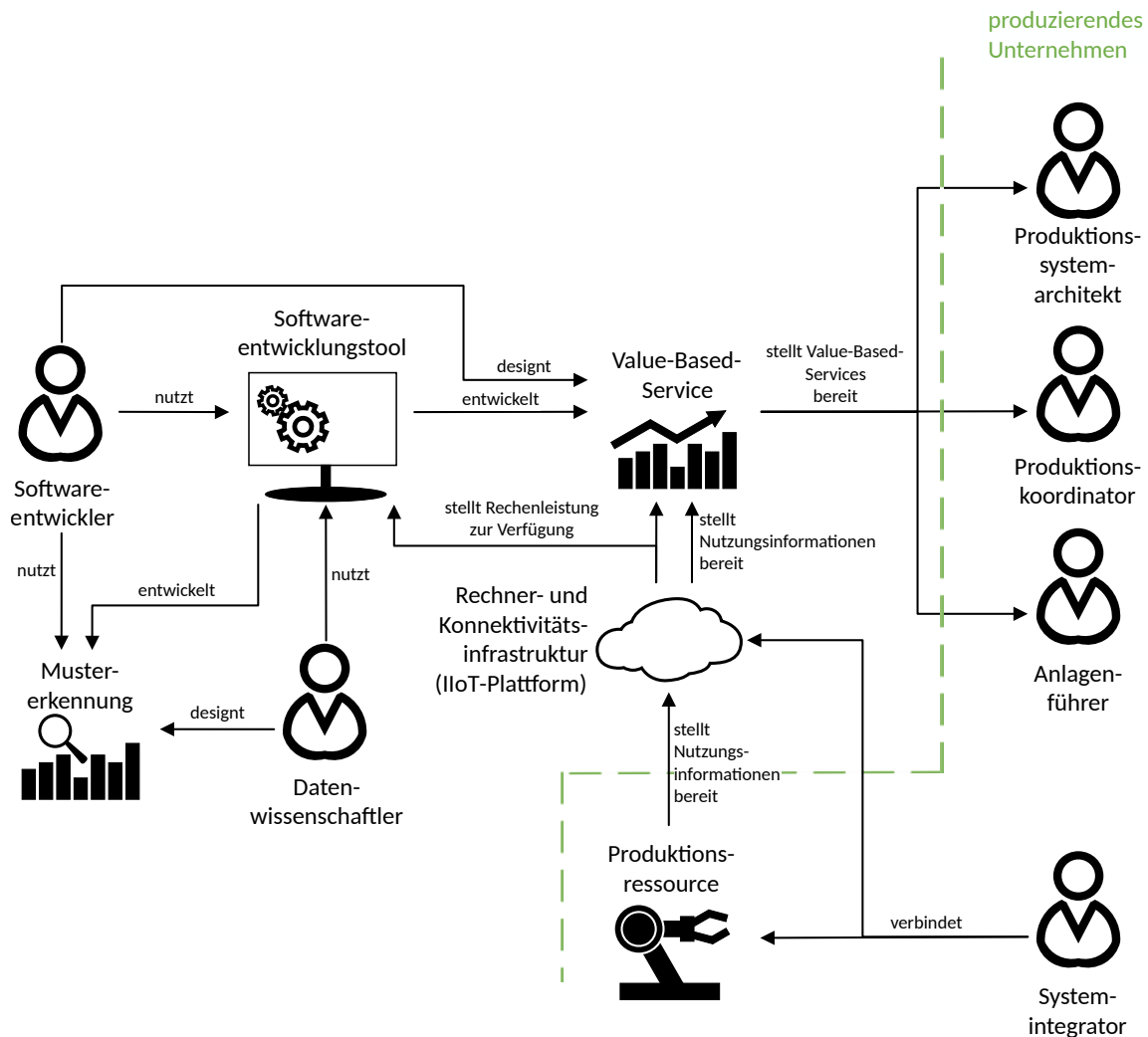


Abbildung 2.11: Technische Perspektive des Use Case „Value-Based Services für Produktionsressourcen“ (in Anlehnung an [4, S. 114])

Als Anforderungen an die Standardisierung stellt die TC65 [4, S. 116] unter anderem folgende Punkte auf:

- Standardisierung der Anbindung von Produktionsressourcen an die IIoT-Plattformen
- Standardisierung der Mechanismen für die Bereitstellung von Informationen und der Zugriffsrechte durch Dritte (z.B. Hersteller der Produktionsressource)
- Standardisierung notwendiger Geschäftsprozesse für z.B. die Abrechnung in Anspruch genommener Dienste

Die Implementierung des Use Case, durch die Anwendung von VWS, kann folgendermaßen geschehen. Die IIoT-Plattform aus der technischen Perspektive in Abbildung 2.11 wird durch eine proaktive VWS ersetzt. Jede Produktionsressource ist bei Auslieferung durch den Hersteller mit einer solchen VWS verbunden. Der Zugriff auf die Daten in der VWS erfolgt z.B. über den Scan eines auf dem Typenschild der Produktionsressource vorhandenen QR-Codes. Über eine Web-Oberfläche steht den Akteuren des produzierenden Unternehmens ein Zugriff auf die VBS, sowie eine Übersicht über die Resultate der durchgeführten VBS zur Verfügung. Die Web-Oberfläche zur Interaktion mit der VWS und den darin enthaltenen VBS wird durch den Hersteller der Produktionsressource bereitgestellt.

Über den Inhalt des VBS in der VWS müssen neben der Verarbeitung der Nutzungsdaten auch die vertraglichen Vereinbarungen verhandelt werden. Dazu gehören unter anderem die Zugriffsrechte des Herstellers auf Nutzungsdaten des Kunden und die Festlegung welche Daten grundsätzlich weitergegeben werden dürfen. Zusätzlich müssen Konditionen vereinbart werden, zu denen der Kunde den VBS in Anspruch nimmt. Die Häufigkeit der Verwendung des VBS kann in der VWS festgehalten werden, um eine automatisierte Abrechnung für den Hersteller zu ermöglichen.

Durch die Ersetzung der IIoT-Plattform durch eine VWS würde die Arbeit des Systemintegrators, der in diesem Use Case die Verbindung der Produktionsressource mit der IIoT-Plattform herstellt, entfallen. Jedoch ist eine Abwandlung seiner Tätigkeit in diesem Fall möglich. Anstatt die Verbindung zur IIoT-Plattform zu schaffen, kann der Systemintegrator für die interne Integration der VWS verantwortlich sein. Abbildung 2.12 zeigt das Use Case Diagramm, in dem die Akteure des Herstellers grün und die des produzierenden Unternehmens blau umrandet sind.

Aus diesem Use Case leitet die TC 65 [4] Spezialfälle ab, die im Folgenden kurz erläutert sind und bei denen die Implementierung einer VWS, wie oben beschrieben, erfolgen kann.

Benchmarking für Produktionsressourcen: Dieser Use Case beschreibt die Bereitstellung von Dienstleistungen, die Datenanalysen anhand eines Vergleichsmaßstabs durchführen. Der Hersteller einer Produktionsressource sammelt von seinen Kunden anonymisiert Nutzungsdaten, die als Vergleichsmaßstab dienen. Ein Kunde, der seine Prozesse effizienter und effektiver gestalten möchte, kann das Benchmarking nutzen und seine Nutzungsdaten beim Hersteller mit den anonymisierten Daten in Beziehung stellen lassen. Damit können Unternehmen, die die gleiche Produktionsressource einsetzen, gegenseitig voneinander lernen [4, S. 116 ff.].

Produktionsressource als ein Service: Ziel eines jeden produzierenden Unternehmens ist die Auslastung seiner Produktion, die abhängig von der Auftragslage ist. Mit diesem Use Case wird unter anderem eine flexible Vergabe von Produktionskapazitäten an externe Unternehmen unterstützt. Über die VWS können produzierende Unternehmen (genannt Eigentümer der Produktionsressource) ihre freien Produktionskapazitäten als

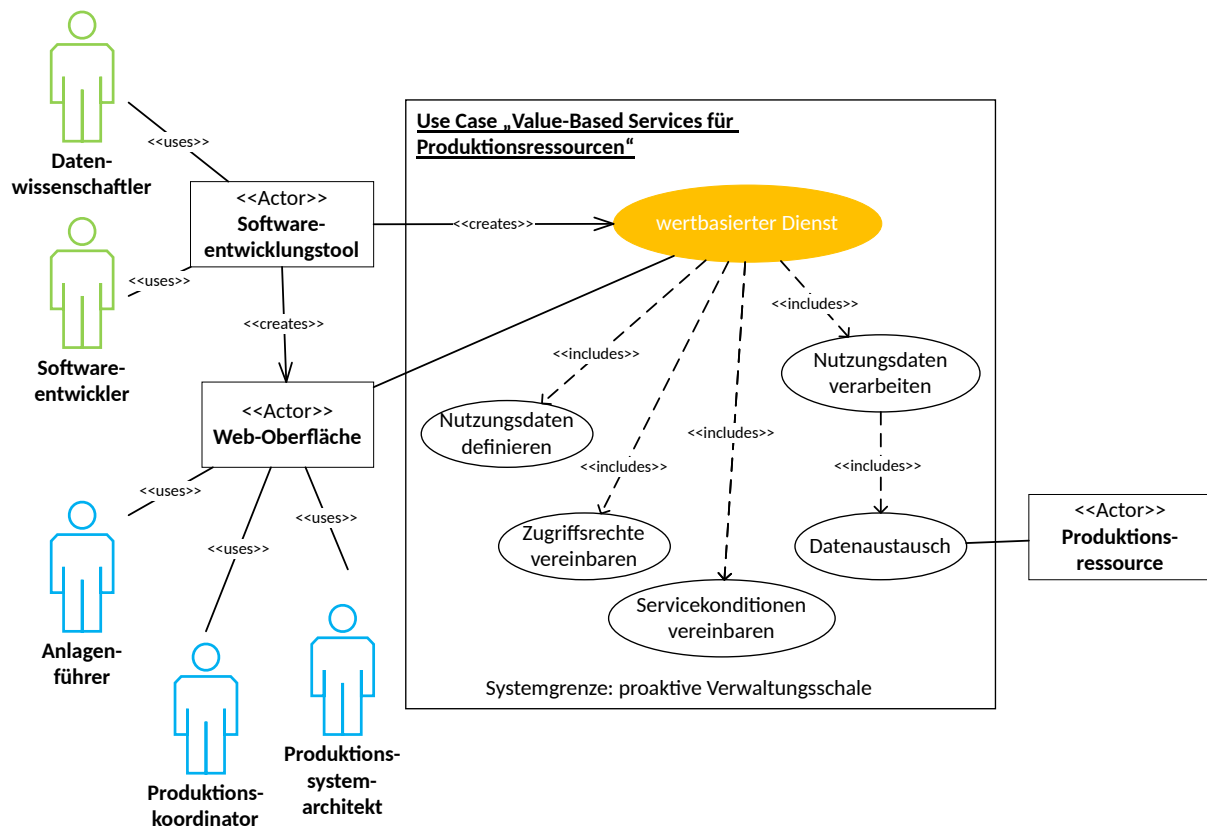


Abbildung 2.12: Use Case Diagramm - Value-Based Services für Produktionsressourcen

einen Service bereitstellen. Kunden (genannt Nutzer der Produktionsressource), die diese Kapazitäten benötigen, gehen mit den produzierenden Unternehmen eine vertragliche Verbindung ein. Mit dem Vertrag erhält der Nutzer der Produktionsressource über die VWS die Möglichkeit, erbrachte Leistungen des Eigentümers zu verfolgen [4, S. 118 ff.].

Ein Unternehmen, das produktbezogene Services anbietet, ist WITTENSTEIN SE mit einem Service-Portal. Das Portal enthält neben den Services auch aktuelle Daten zu erworbenen Produkten, wie Dokumentationen, Zertifikate, technische Daten etc. Mit den Daten ist es nicht nur WITTENSTEIN möglich die Nutzung der Produkte bei dem Kunden zu optimieren sondern auch die eigenen Services effizienter zu gestalten. Die IDTA stellt auf ihrer Website eine kurze Zusammenfassung über die Umsetzung bereit [25].

2.5 Use Case Cluster - IT-Infrastruktur und Software

Industrie 4.0 lebt von der digitalen Vernetzung sowohl physisch existierender als auch virtueller Assets. Aufgrund dessen ist ein Hauptpunkt der I4.0-Technologie die IT-Infrastruktur, die die Interaktionsfähigkeit aller Assets gewährleistet. Für die Verarbeitung der Daten und den benutzergerechten Zugriff stehen Softwareapplikationen zur Verfügung.

Sowohl die IT-Infrastruktur als auch die Softwareapplikationen sorgen für einen fortwährenden Anstieg des Automatisierungsgrads im industriellen und geschäftlichen Umfeld. Sie bringen jedoch auch Gefahren hinsichtlich der Daten- und Angriffssicherheit mit sich, die immer mehr Aufmerksamkeit benötigen. Einige der Use Cases, die in diesem Cluster durch eine VWS realisiert werden können, sind im Folgenden beschrieben.

2.5.1 Dokumentationspflege bei Gerätetausch

Ein Produktionssystem, das in einem produzierenden Unternehmen betrieben wird, besteht meistens aus Produktionsmodulen verschiedener Maschinenhersteller. Die Maschinenhersteller wiederum verbauen in ihren Produktionsmodulen Komponenten verschiedener Komponentenhersteller.

Für ein Produktionssystem sammeln sich in allen Lebensphasen wichtige Dokumente an, die verwaltet werden müssen. Die Verwaltung und Pflege der Dokumente fängt bei den Komponentenherstellern an und endet bei dem produzierenden Unternehmen. Ein Beispiel ist in Abbildung 2.13 dargestellt, in dem ein Produktionssystem aus Produktionsmodulen mehrerer Maschinenhersteller besteht.

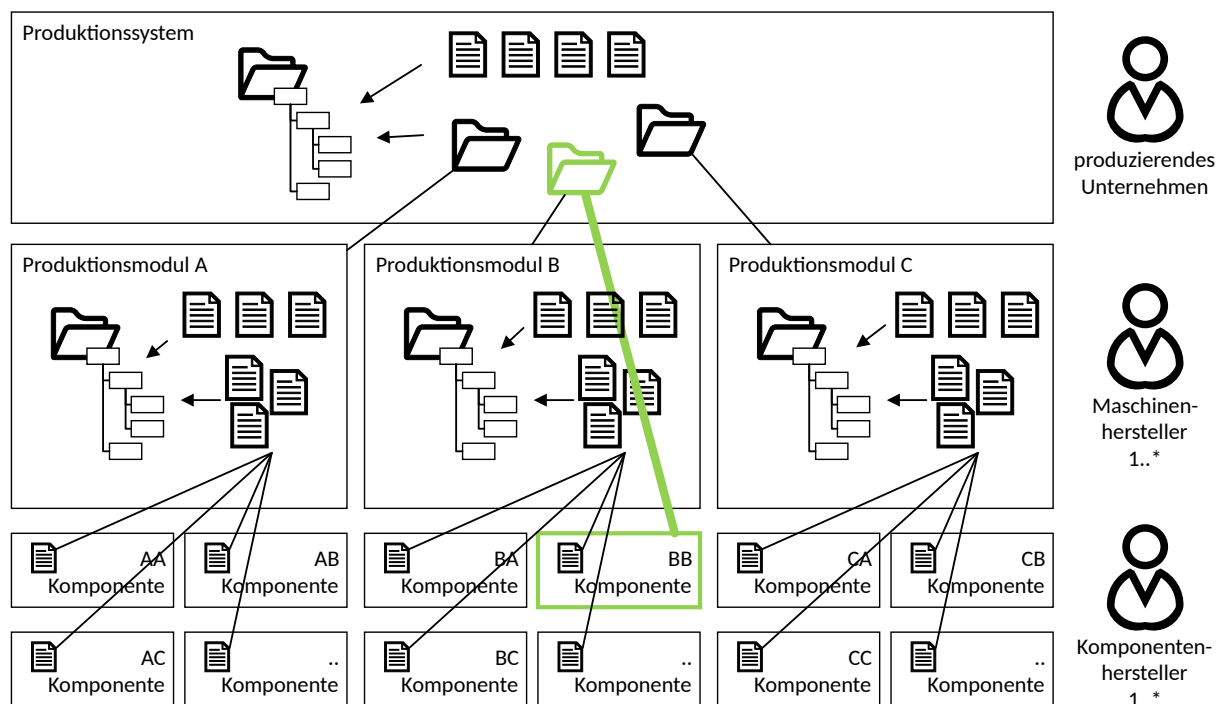


Abbildung 2.13: Aktueller Stand der Dokumentationspflege bei dem Use Case „Dokumentationspflege bei Gerätetausch“

Die Grafik (Abbildung 2.13) zeigt, wie Dokumentationen, Zertifikate, Bescheinigungen usw. bei jeder im Produktionsmodul verbauten Komponente anfallen. Diese Dokumente müssen von allen Akteuren strukturiert abgelegt werden. Nach Fertigstellung des Produktionsmoduls und anschließender Auslieferung dessen an den Kunden, hier das produzierende

Unternehmen, werden die Dokumente in den unternehmensinternen Verwaltungssystemen abgelegt. Da mitunter neben zentralen Verwaltungssystemen auch Dateisysteme in verschiedenen Abteilungen existieren, sind redundante Daten und eine folgende Inkonsistenz der Daten nicht vermeidbar.

Eine Lösung des Problems ist die Ausstattung jedes Assets mit einer reaktiven VWS. Das heißt jedes Feldgerät wird vom Komponentenhersteller bereits mit einer reaktiven VWS ausgestattet und bei Auslieferung an den Maschinenhersteller übergeben. Bei der Errichtung des Produktionsmoduls werden alle vorhandenen VWS der verbauten Komponenten in der VWS des Produktionsmoduls durch Referenzen vereinigt.

Nach der Inbetriebnahme des Produktionsmoduls übergibt der Maschinenhersteller anschließend eine Kopie der Produktionsmodul-VWS mit allen seinen untergeordneten VWS an das produzierende Unternehmen. Das produzierende Unternehmen verbindet wiederum die reaktive VWS mit der übergeordneten Produktionssystem-VWS und falls notwendig mit anderen VWS.

Durch die Verknüpfungen stehen die Verwaltungsschalen, sowohl bei dem Maschinenhersteller als auch bei dem produzierenden Unternehmen in unterschiedlichen Beziehungen zueinander. Tritt der Fall ein, dass eine Komponente ausgetauscht werden muss, so sind die entsprechenden VWS der Geräte auszutauschen. Dabei müssen die Beziehungen von der VWS des alten Geräts auf die VWS des neuen Geräts fehlerfrei übertragen werden, damit die Funktionalität des Gesamtsystems weiterhin gewährleistet ist.

Der Gerätetausch kann unter anderem durch das Instandhaltungspersonal des produzierenden Unternehmens oder bei Optimierung der Maschinenserie durch den Maschinenhersteller stattfinden. Sobald wie im obigen Beispiel ein Asset durch das produzierende Unternehmen ausgetauscht wird, wird mit der automatisierten Aktualisierung der Verwaltungsschalen des Produktionsmoduls indirekt die VWS des betroffenen Produktionssystems angepasst. Damit kann zu jedem Zeitpunkt ein Zugriff auf die Verwaltungsschalen stattfinden und redundante Daten vermieden werden.

Diese Arbeit befasst sich überwiegend mit der Entwicklung von Konzepten zur Umsetzung dieses Use Case. Da ab Kapitel 3 umfassende Betrachtungen durchgeführt werden, wird in diesem Kapitel nicht weiter auf den Use Case eingegangen.

2.5.2 Human-Machine-Interface

Ein Ziel dieses Use Case ist laut [4] die Bereitstellung standardisierter Visualisierungs- und Bedienelemente, die eine automatisierte Anpassung von Benutzeroberflächen (HMIs) entsprechend der aktuellen Situation ermöglichen. Die HMIs dienen sowohl dem Produktionskoordinator als auch dem Anlagenführer und Service-/Instandhaltungspersonal als Unterstützung. Durch das HMI wird für die Akteure die Struktur des Produktionssystems

sichtbar und Eingriffe in den Produktionsprozess möglich. Der Zugriff jedes einzelnen Akteurs unterliegt einer Benutzerrolle, die Einschränkungen der Zugriffsberechtigungen auf das Produktionssystem definiert und in diesem Use Case dementsprechend beachtet werden muss [4, S. 132 f.].

Grundsätzlich werden zwei Konzepte miteinander verknüpft. Es handelt sich dabei zum einen um die Bereitstellung von standardisierten Visualisierungs- und Bedienelementen und zum anderen um die Aktualisierung der Elemente durch neue verfügbare Funktionalitäten des Produktionssystems und dessen Komponenten (ebd.).

Für die Standardisierung der Visualisierungs- und Bedienelemente kann der TC 65 [4] zufolge der Use Case „Modularisierung von Produktionssystemen“ (s. Kapitel 2.2.1) herangezogen werden. Die Anordnung der Elemente jedes einzelnen Produktionsmoduls in einem HMI erfolgt durch die Kenntnis von der baulichen Struktur des Produktionssystems. Diese Struktur gleicht in den meisten Fällen auch der Navigationsstruktur im HMI. Sobald also z.B. eine Rekonfiguration des Produktionssystems stattfindet, kann auf Grund der gegebenen baulichen Struktur die automatisierte Anpassung der Navigationsstruktur mit seinen Visualisierungs- und Bedienelementen stattfinden. Das zieht auch eine Verringerung von Stillstandszeiten mit sich.

Als eine Voraussetzung für die Realisierung des Use Case sieht die TC 65 [4] die Implementierung einer Präsentationslogik. Sie übernimmt die Entkopplung der logischen Informationen der Visualisierungs- und Bedienelemente von den technologiebedingten Informationen der eingesetzten Feldgeräte. Beispielsweise besitzt jeder Motor die Funktionen „Ein“ und „Aus“, wodurch die logische Information für jeden Motor gleich bleibt. Die Parametrierung kann sich aber je nach Art des Motors ändern und somit technologiebedingte Änderungen herbeiführen (z.B. unterschiedliche Startrampen) [4, S. 132 f.].

Eine weitere Voraussetzung laut [4] ist das Vorhandensein eines Assetmanagementsystems, das als Informationsquelle für die Visualisierungs- und Bedienelemente dient. Einen Mehrwert bringt der Use Case vor allem in den Wertschöpfungsketten der Produktionsplanung (PSLM), des SCM und des Service [4, S. 23], [4, S. 134].

Die technische Perspektive wird durch die TC 65 [4] wie folgt beschrieben und ist in Abbildung 2.14 zu sehen. Die genannte Präsentationslogik wird während der Planung des Produktionssystems von einem Softwareentwickler des produzierenden Unternehmens konzipiert. Die Entwicklung der Präsentationslogik umfasst die oben genannte Entkopplung der HMI-Elemente von den Informationsquellen. Die Informationsquellen bilden zum einen das Assetmanagementsystem und zum anderen die Integrationsschicht der Produktionsressourcen, die das Produktionssystem steuern. Weiterhin muss der Softwareentwickler Dokumentations- und Explorationstools bereitstellen, deren Navigationsstruktur durch die Präsentationslogik automatisiert erzeugt wird.

Das Dokumentations- und Explorationstool sind für die Nutzung durch den Produktionskoordinator, den Anlagenführer und das Service-/Instandhaltungspersonal in der Produktion

vorgesehen. Das Dokumentationstool erlaubt den schreibenden Zugriff und das Explorationsstool den lesenden Zugriff auf das Assetmanagementsystem und die Integrationsschicht. Die Zugriffsrechte der verschiedenen Akteure auf die Informationen werden durch den IT-Administrator geregelt [4, S. 134].

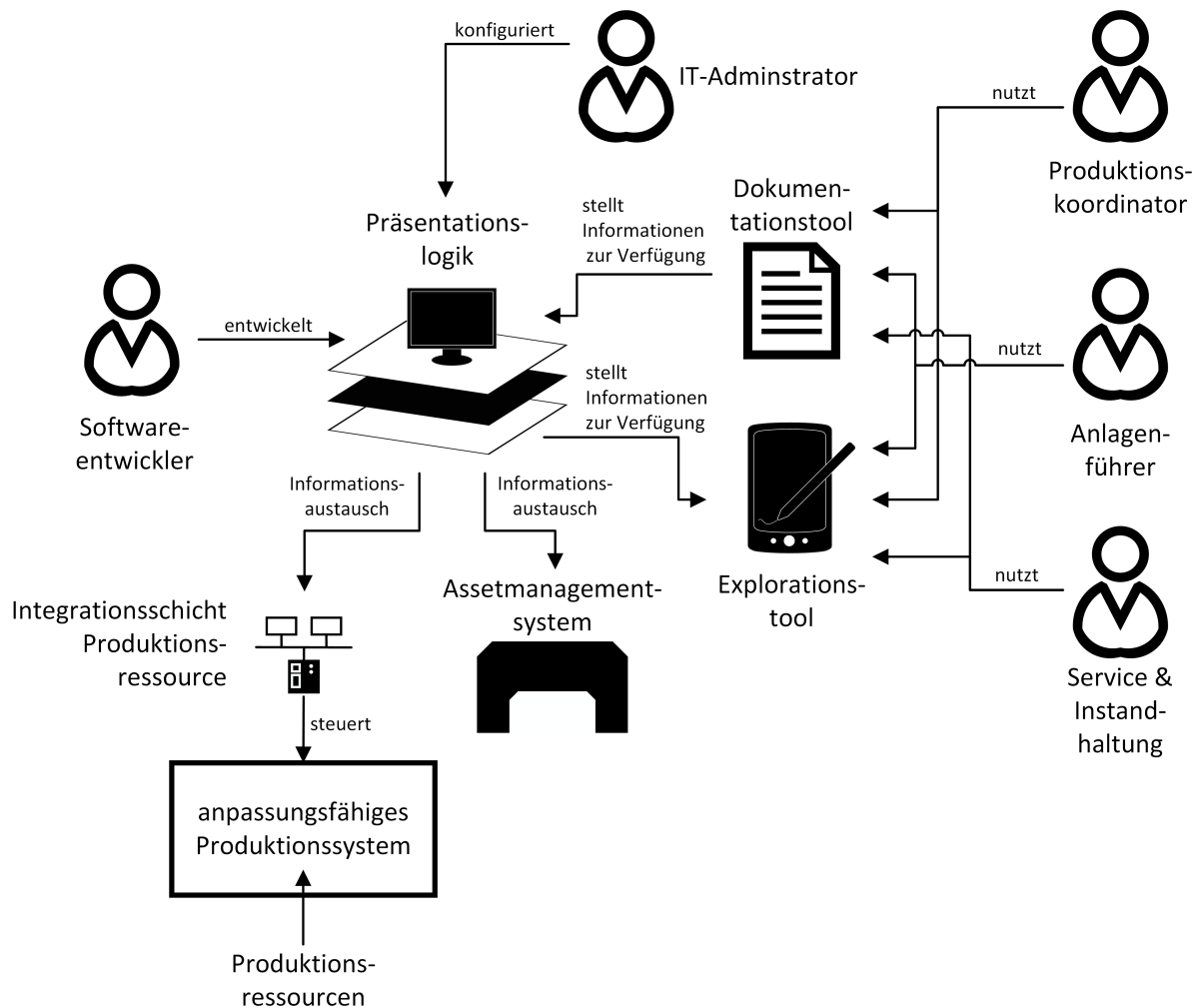


Abbildung 2.14: Technische Perspektive des Use Case „Human-Machine-Interface“ (in Anlehnung an [4, S. 134])

Der zusätzliche Aufwand, zur Entkopplung der Visualisierungs- und Bedienelemente von den Informationsquellen, wirkt sich jedoch nach Einführung senkend auf den Engineering-Aufwand aus. Grund dafür ist, dass bei Rekonfiguration des Produktionssystems eine automatisierte Aktualisierung des HMI stattfindet. Zur Umsetzung des Use Case sind von der TC 65 Standardisierungen gefordert [4, S. 135], die im Nachfolgenden aufgelistet sind:

- Standardisierung von Visualisierungs- und Bedienelementen des HMI in Bezug auf den Zugriff und die Darstellung für den Nutzer
- Standardisierung von Schnittstellen zum Datenaustausch zwischen den Visualisierungs-/Bedienelementen und den Feldgeräten

- Standardisierung von Stilrichtlinien für die Gestaltung von HMIs

Da das Assetmanagementsystem in diesem Use Case eine Rolle spielt, werden VWS bereits indirekt verwendet. Die Idee ist nun, z.B. die VWS des Produktionssystems um ein Teilmodell zu erweitern, das die Navigationsstruktur für das HMI enthält. Die Navigationsstruktur bildet sich automatisiert durch Informationen aus der baulichen Struktur des Produktionssystems. Voraussetzung ist, dass die VWS des Produktionssystems die bauliche Struktur enthält. Die Präsentationslogik ist damit in der Lage, auf die Navigationsstruktur zurückzugreifen und basierend darauf, die Visualisierungs- und Bedienelemente anzuordnen.

Die Wahl der Rolle der VWS fällt auf eine reaktive VWS, da die Präsentationslogik eine Softwareapplikation darstellt, die nicht zwingend über eine I4.0-Sprache mit der VWS kommunizieren muss. Die Alternative zur VWS wäre der Entwurf der Navigationsstruktur intern in der Präsentationslogik. Das entsprechende Use Case Diagramm ist in Abbildung 2.15 zu sehen.

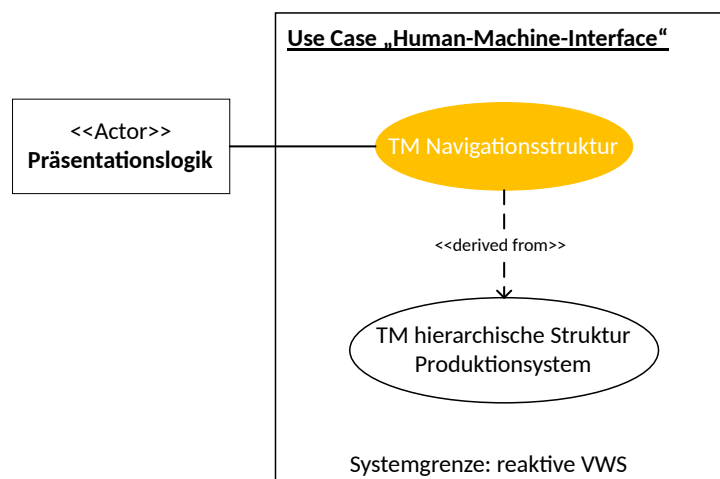


Abbildung 2.15: Use Case Diagramm - Human Machine Interface

2.6 Kreislaufwirtschaft

Durch die Ausbreitung der Automatisierungs- und Informationstechnik in den letzten Jahrzehnten und der damit einhergehenden Möglichkeit der Massenproduktion sanken die Preise der Produkte stetig. Als Folge steigerte sich der Absatz und die Spirale der Produktionssteigerung, zum Nachteil unserer Ressourcen, dreht sich bis heute weiter.

Für viele der hergestellten Produkte sind Rohstoffe notwendig, die endlich sind. Die Knappheit der Rohstoffe wird immer bedeutender für die zukünftige Produktion. Aufgrund dessen besteht ein großes Interesse an der Wiederverwertung und -verwendung der Rohstoffe und Materialien aus den bereits existierenden Produkten und Produktionsprozessen.

Weiterhin möchten immer mehr Unternehmen mit ihrer Produktion auch auf den daraus hervorgehenden CO₂-Fußabdruck achten und diesen so klein wie möglich halten. So ist es das Ziel dieses Use Case, Rohstoffe und Materialien fortwährend in einem geschlossenen Materialkreislauf zirkulieren zu lassen [3, S. 24f.].

Als ein wichtiger Punkt geht aus dem Ergebnispapier [3] die Wiederverwendbarkeit von Materialien hervor, die schon bei der Produktentwicklung zu beachten ist. Dazu gehört auch die Speicherung der Informationen zu den verwendeten Materialien und Rohstoffen und auf welche Art diese wiederverwendet/-verwertet werden können. Eine Dokumentation der Informationen zugeschnitten auf jedes hergestellte Produkt ermöglicht die konsequente Verfolgung in der Kreislaufwirtschaft [3, S. 24 f.].

Zu den Arten der Wiederverwendung/-verwertung in der Kreislaufwirtschaft gehören laut [3, S. 24] unter anderem:

Remanufacturing: Das Remanufacturing beschreibt die Zerlegung von gebrauchten Geräten und den anschließenden Austausch von verschlissenen Einzelteilen. Das Verfahren versucht ein gebrauchtes Gerät wieder in seinen Neuzustand zu versetzen.

Reuse: Reuse beschreibt die Wiederverwendung von Produktionsressourcen an einer anderen Stelle. Die Produktionsressourcen sind in diesem Fall noch funktionstüchtig, werden aber z.B. aufgrund der Umrüstung eines Produktionssystems an dieser Stelle nicht mehr benötigt.

Repair: Defekte Produktionsressourcen werden repariert und in einen funktionstüchtigen Zustand zurückversetzt.

Upcycling: Das Upcycling umfasst die Aufbereitung von Rohstoffen und Materialien, sodass aus diesen neue, höherwertige Produkte entstehen.

Für die Dokumentation der verwendeten Materialien und Ressourcen ist der Einsatz einer VWS denkbar. Die VWS nutzt dafür alle bisher veröffentlichten Teilmodell-Templates. Zu Beginn der Produktentwicklung werden in der VWS in einem neu designten Teilmodell Informationen über die Komponenten, aus denen das Produkt besteht, gespeichert. Dazu gehören unter anderem Informationen zu den Materialien bzw. Rohstoffen sowie deren Lebensdauer und Art der Wiederverwendbarkeit.

Am Beispiel eines Produktionsmoduls wird versucht die oben beschriebenen Arten der Wiederverwendung/-verwertung durch den Einsatz einer VWS näher zu beschreiben. Ausgangspunkt für das Beispiel ist die VWS eines Produktionsmoduls, die Verweise auf die VWS der verbauten Komponenten enthält und die wiederum Informationen über ihre Einzelteile abbilden. Sowohl das Produktionsmodul als auch die Komponenten verfügen in ihren VWS über Teilmodell-Templates wie das digitale Typenschild, die technischen

Daten, die Dokumentation und das genannte Teilmodell mit Informationen für die Kreislaufwirtschaft. Mit diesen Informationen ist beispielsweise eine Analyse des Ursprungsorts, der Fähigkeiten, der Lebensdauer und der stofflichen Zusammensetzung einer Komponente möglich. Im Folgenden sind die oben genannten Arten der Wiederverwendbarkeit am Beispiel des Produktionsmoduls beschrieben.

Remanufacturing: Während das Produktionsmodul bei einem Einzelteil, das das Ende seiner Lebensdauer erreicht, automatisiert eine Bestellung aufgibt, kann es gleichzeitig eine Rücksendung dessen beantragen. Durch das digitale Typenschild ist der Hersteller des Einzelteils bekannt.

Ist das Einzelteil bei dem Hersteller angekommen ist dieser in der Lage, durch die Daten der VWS die Komponenten herauszufiltern, die die Lebensdauer bestimmen. Diese Komponenten können, falls möglich, durch den Hersteller ersetzt und das Feldgerät damit in einen Neuzustand versetzt werden.

Reuse: Baugruppen des Produktionsmoduls werden durch eine Umrüstung dessen nicht mehr benötigt. Mit den Informationen über die Art der Wiederverwendung, ist ein Abgleich mit z.B. einem ERP-System möglich, das den gesamten Bestand an Produktionsressourcen dokumentiert. In diesem Abgleich werden Funktionalitäten der Baugruppe und deren Einzelteile mit bereits existierenden Komponenten in den Produktionssystemen abgeglichen. Wird eine Funktionalität gefunden, für die eine Wiederverwendbarkeit der Baugruppe oder deren Einzelteile besteht, können diese für die spätere Verwendung eingelagert oder auch sofort an einer anderen Stelle genutzt werden.

Repair: Ein Produktionsmodul enthält in seiner VWS, auch durch Verweise auf andere VWS, alle technischen Informationen zu seinen verbauten Komponenten (z.B. Lebensdauer). Das System kann die Lebensdauer der Einzelteile überwachen und vor Ablauf dieser eine automatisierte Bestellung bei einem überlagerten Managementsystem auslösen. Die Bestelldetails können aus den Teilmodellen der technischen Daten sowie dem digitalen Typenschild zusammengetragen werden. Sobald das Ersatzteil zur Verfügung steht, kann die Dokumentation zu dem Gerät den Systemintegrator in der Reparatur anleiten und unterstützen.

Upcycling: Die Materialien und Rohstoffe des im Remanufacturing ausgetauschten Einzelteils können, da es nicht mehr wiederverwendet werden kann, aufbereitet werden. Die Daten für die Art der Aufbereitung stellt dessen VWS bereit. Damit ist es dem Hersteller möglich aus dem defekten Produkt ein höherwertiges herzustellen.

Das Use Case Diagramm, das die beschriebene Situation zu Teilen widerspiegelt, ist in Abbildung 2.16 dargestellt. Dabei ist der Produktentwickler des Herstellers, der die VWS des Produkttyps erstellt, grün umrandet. Alle Akteure die mit dem produzierenden Unternehmen in Zusammenhang stehen sind durch eine blaue Umrandung gekennzeichnet.

Ohne Umrandung bleibt das Service- und Instandhaltungspersonal, da dieses sowohl auf der Seite des Herstellers als auch auf der Seite des produzierenden Unternehmens vorhanden ist.

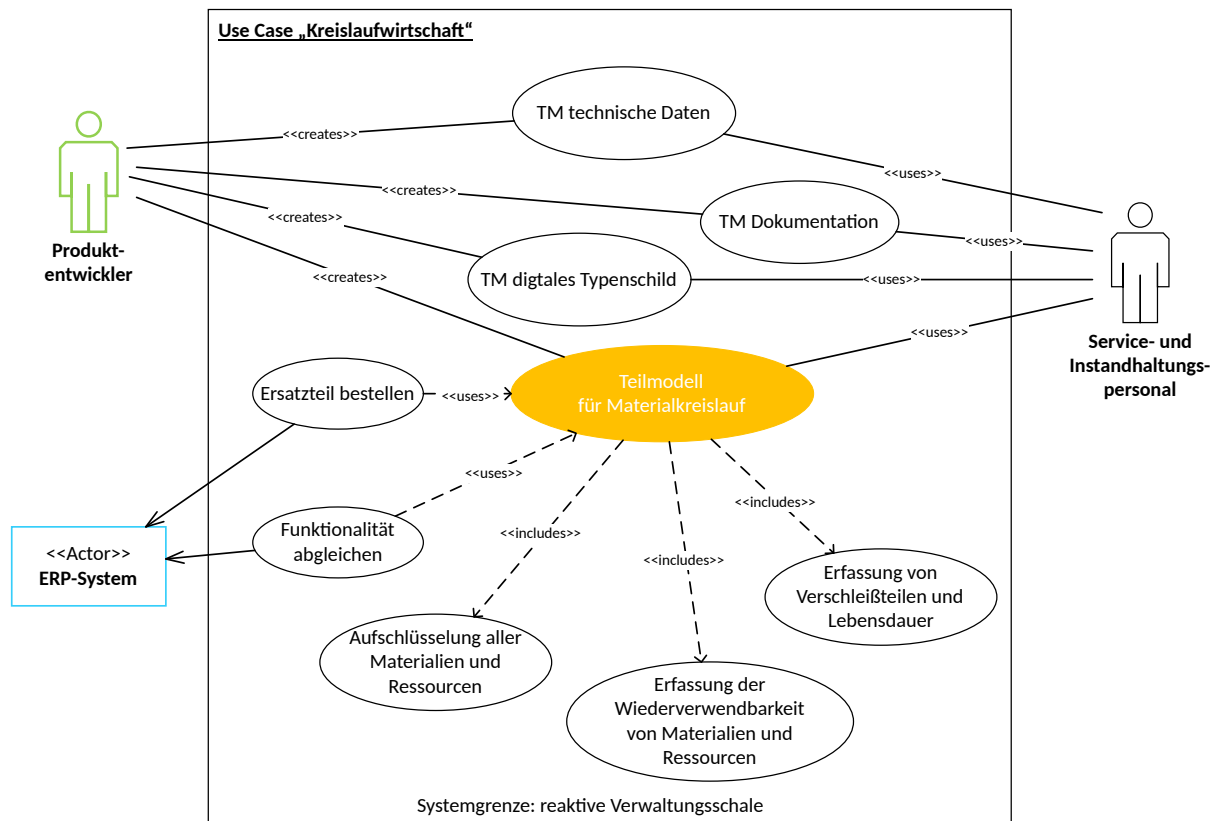


Abbildung 2.16: Use Case Diagramm - Kreislaufwirtschaft

Es können bereits veröffentlichte Teilmodell-Templates wie das „ZVEI Digital Nameplate for Industrial Equipment“ und der „Generic Frame for Technical Data for Industrial Equipment in Manufacturing“ sowie die Teilmodell-Templates „Handover Documentation“ und „Contact Information“, die sich in Prüfung befinden, genutzt werden. Aber auch das sich in Entwicklung befindliche Teilmodell-Template „Service Order Creation“ kann Verwendung finden, um beispielsweise die automatisierte Bestellung eines Ersatzteils beim ERP-System einzuleiten (s. Anhang A).

Das genannte Beispiel begrenzt sich auf das produzierende Unternehmen und die Hersteller der Produktionsressourcen. Weiterhin kann aber auch der private Kunde von so einem Materialkreislauf profitieren. Sollte das gekaufte Produkt nicht mehr gebraucht, defekt oder aber nicht mehr verwendbar sein, kann der Kunde durch die VWS herausfinden, an wen er das Produkt zur Weiterverwendung, Reparatur, oder zum Upcycling senden kann. Durch die Aufbringung eines dauerhaften QR-Codes an dem Produkt, seitens des Herstellers, ist es dem Kunden über eine Web-Oberfläche möglich, entsprechende Informationen zu seinem Anliegen zu erhalten. Als Alternative zu der VWS ist eine IIoT-Plattform denkbar, in der die Speicherung aller Informationen bezogen auf jedes individuelle Produkt stattfindet.

3 konzeptionelle Umsetzung des Use Case „Dokumentationspflege bei Gerätetausch“

Das Ziel des Use Case „Dokumentationspflege bei Gerätetausch“ liegt in der Aktualisierung von Verwaltungsschalen und deren Beziehungen zu anderen VWS. Nach einem erfolgten Gerätetausch muss die gesamte Dokumentation zu dem Produktionssystem auf dem neuesten Stand sein.

In vielen industriellen Unternehmen befinden sich die Dokumente gegenwärtig sowohl in physischen als auch virtuellen Dateordnern, von denen Kopien in verschiedenen Abteilungen existieren. Das führt zu unterschiedlichen Versionsständen und birgt Gefahren, die das Arbeiten mit veralteten Kenntnisständen nach sich ziehen kann. Aber auch unternehmensübergreifend können unterschiedliche Versionsstände zu Schwachstellen führen. Wenn etwa das Instandhaltungspersonal des produzierenden Unternehmens eine Komponente eines Produktionsmoduls tauscht, ohne den Maschinenhersteller einzubeziehen, führt das zu inkonsistenten Daten zwischen beiden Akteuren. Muss eine aufwändigere Wartung durch den Maschinenhersteller erfolgen, ist dieser mit seinen Informationen zu den verbauten Komponenten nicht mehr auf dem aktuellen Stand. Erst ein Abgleich mit den Informationen aus der Instandhaltung schafft bei dem Maschinenhersteller konsistente Daten. Das kostet jedoch Zeit, die eingespart werden kann. Eine bildliche Beschreibung des Problems erfolgte bereits in Kapitel 2.5.1.

Ein Lösungsansatz bietet die Ausstattung eines jeden Assets mit einer reaktiven VWS. Diese Verwaltungsschalen stehen in unterschiedlichen Beziehungen zueinander und bilden dadurch VWS-Strukturen. Ist der Tausch eines Geräts notwendig, dann muss die dazugehörige VWS durch die VWS des neuen Geräts ersetzt werden. Die Speicherung der reaktiven VWS kann in Repositories erfolgen und ist in Kapitel 3.1 veranschaulicht. Im Zusammenhang damit stehen auch die am Use Case beteiligten Akteure, die die Informationen der Verwaltungsschalen beeinflussen (s. Abbildung 2.13 aus Kapitel 2.5.1).

Die Aufgabenstellung des Use Case wird durch ein Beispiel dargestellt, das sich durch die Kapitel 3.2 bis Kapitel 3.4 zieht. Eine Beschreibung des Beispiels ist nachfolgend festgehalten:

Ein Produktionssystem besitzt ein Produktionsmodul, das sich mit der Qualitätssicherung des darin hergestellten Produkts beschäftigt. Dieses Produktionsmodul besteht aus einer Roboterzelle mit einem Roboter, dessen Greifer eine hochauflösende Kamera hält.

Damit werden Bilder von dem Produkt erstellt, die in einem Bildverarbeitungssystem ausgewertet werden.

In rauen Produktionsumgebungen besteht die Wahrscheinlichkeit, dass die Linse der Kamera nach einem gewissen Zeitraum verunreinigt und somit keine sichere Qualitätsprüfung mehr möglich ist. Die benutzte Kamera (im weiteren „alte Kamera“ genannt) muss durch eine Kamera aus dem Bestand ersetzt werden. Dabei kann es sich um eine neue Kamera oder um eine Kamera handeln, die durch beispielsweise eine Reinigung der Linse wieder instand gesetzt wurde. (In den weiteren Kapiteln wird dafür der Begriff „neue Kamera“ verwendet.)

Jedes Asset, von dem Produktionssystem über das Produktionsmodul bis zum Feldgerät, besitzt eine VWS mit allen wichtigen Assetinformationen. Mit dem Tausch der Kamera müssen auch die dazugehörigen VWS ausgetauscht werden, sodass die Dokumentation des Produktionssystems auf dem aktuellen Stand ist. Voraussetzung in der Aufgabenstellung ist, dass die VWS beider Kameras bereits in Repositories des produzierenden Unternehmens abgelegt und in einer Registry eingetragen sind.

Aus dem beschriebenen Beispiel lassen sich verschiedene Szenarien ableiten. Dabei besitzt jedes Szenario unterschiedliche Akteure, die den Tausch der Kamera durchführen und folglich die Aktualisierung der Verwaltungsschalen initiieren.

Die Szenarien umfassen unter anderem den Tausch der Kamera durch das Instandhaltungspersonal des produzierenden Unternehmens bzw. durch das Servicepersonal des Herstellers des Produktionsmoduls. Als drittes Szenario wird ein vollautomatisierter Kameratausch betrachtet, der durch den Roboter erfolgt, dessen Greifer die Kamera hält.

Der Fokus in den Szenarien liegt auf dem Tausch der Verwaltungsschalen und nicht auf dem physischen Tausch und der damit verbundenen Konfiguration der Kamera. Das heißt, wenn im Folgenden der Begriff Gerätetausch verwendet wird, ist damit vordergründig die Aktualisierung der VWS-Struktur gemeint. Zur Beschreibung der Szenarien werden Use Case Diagramme und Sequenzdiagramme herangezogen.

3.1 Systemarchitektur - Repositories zur Speicherung von reaktiven Verwaltungsschalen

Im Lebenszyklus einer Komponente entstehen Informationen, die nicht immer in ein und derselben VWS gespeichert werden sollen. Ein Grund dafür ist unter anderem, dass in den Lebensphasen eines Assets, von der Herstellung einer Komponente über den Verkauf und den Einbau der Komponente bis hin zu dessen Nutzung, verschiedene Akteure beteiligt sind. Alle Akteure steuern dabei individuelle Informationen zu den VWS bei, die nur teilweise für andere Akteure zugänglich sein sollen. Als Lösung erlaubt die DIN SPEC 91345 [1]

(Referenzarchitekturmodell Industrie 4.0) die Verwaltung eines Assets durch mehrere VWS. Die Verwaltungsschalen des Assets müssen sich jedoch gegenseitig aufeinander beziehen, um die Verfolgung der zu einem Asset dazugehörigen Verwaltungsschalen zu garantieren [1, S. 39].

Die Verwaltungsschalen des Assets können, wenn sie passiver Art sind, als Dateien in Repositories abgelegt und deren Inhalte durch die VWS-API zugänglich gemacht und somit zu reaktiven VWS umgewandelt werden. Sowohl der Komponentenhersteller, der Maschinenhersteller als auch das produzierende Unternehmen verfügen über solche Repositories. Jeder Akteur kann für sich entscheiden, ob er den Zugriff auf eines seiner Repositories für seine Kunden bereitstellt oder nicht.

Die folgende Möglichkeit zur Speicherung von VWS stellt nur eine Option von vielen dar und erhebt keinen Anspruch auf Vollständigkeit. Sie dient lediglich zur Veranschaulichung einer potentiellen Systemarchitektur, die die Basis für die Ergebnisse der folgenden Abschnitte und Kapitel 4 bildet.

Hinweis: In den folgenden grafischen Beschreibungen werden die VWS-IDs in Form von Ziffernfolgen abgebildet. Es dient der Übersichtlichkeit und der einfachen Verfolgbarkeit einer speziellen VWS über die verschiedenen Schaubilder hinweg.

Begonnen wird mit dem Aufgabenbereich des Komponentenherstellers, der Feldgeräte für die Errichtung des Produktionsmoduls an den Maschinenhersteller liefert. Er steht am Anfang der Lieferkette und entwickelt die VWS zu jedem seiner Produkte. Jede produzierte Komponente stammt dabei von einem Produkttyp ab und dementsprechend stammt auch die VWS einer Komponente von einer Typ-VWS ab. Die Speicherung der Typ-VWS und der daraus entstehenden Instanz-VWS kann auf verschiedene Arten geschehen. Eine Möglichkeit ist in Abbildung 3.1 dargestellt.

In der Grafik ist ein Repository sowohl zur Verwaltung der Typ-VWS als auch der Instanz-VWS zu sehen. Dabei gehört jede Instanz-VWS zu einem Asset und erhält eine global eindeutige VWS-ID. Bei Verkauf der Komponente an den Maschinenhersteller wird die entsprechende Instanz-VWS aus dem Repository des Komponentenherstellers serialisiert und als Datei (passive VWS) an den Maschinenhersteller übergeben. Dieser wiederum pflegt die VWS-Dateien in sein Repository ein und wandelt sie durch Hinzufügung der VWS-API in reaktive VWS um. Die VWS der gekauften Komponenten sind in der Grafik grau hinterlegt, da diese nach Auslieferung der Komponenten nicht mehr im Repository des Komponentenherstellers existieren.

Die Teilmodelle und Views¹ mit den VWS-Elementen (s. Kapitel 4.2), die für den Maschinenhersteller bestimmt sind, werden an diesen bei Kauf der Komponente weitergegeben. Grafisch sind die Teilmodelle und Views, die der Komponentenhersteller erstellt hat, als

¹Views ermöglichen die Filterung von referenzierbaren VWS-Elementen, um diese bestimmten Benutzergruppen bereitzustellen [26, S. 86]

rechteckige Kästen in der VWS gekennzeichnet. Die Beziehung „derived from“, die in der Instanz-VWS gespeichert wird, lässt immer den Rückschluss auf die Typ-VWS zu.

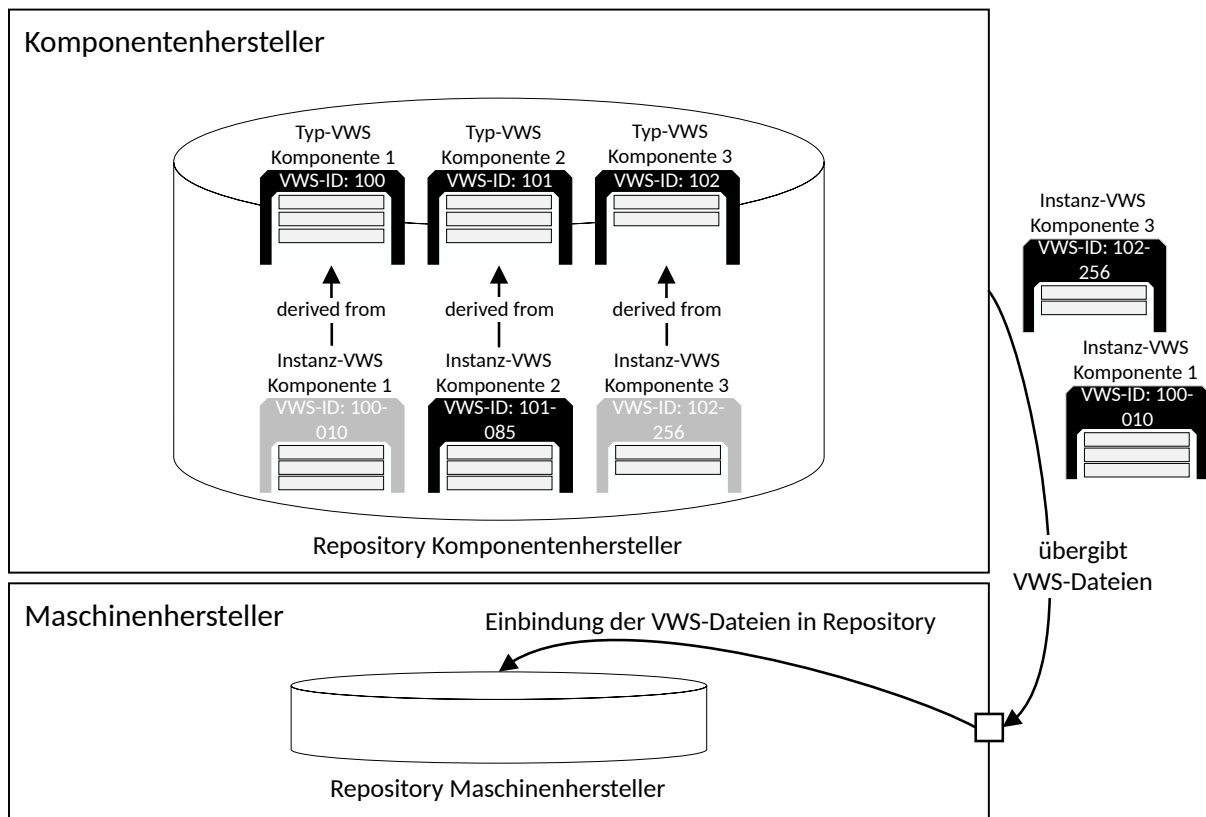


Abbildung 3.1: Bereitstellung der VWS durch den Komponentenhersteller

Mit dem Kauf der Komponenten ist der Maschinenhersteller in der Lage, sein Produktionsmodul zusammen zu setzen. Während der Errichtung des Produktionsmoduls entstehen weitere Informationen seitens des Maschinenherstellers, mit denen die VWS der gekauften Komponenten sowie seine selbst designten VWS angereichert werden. Zu diesen Informationen zählen auch die erzeugten Beziehungen zwischen den einzelnen Komponenten. Diese Beziehungen betreffen beispielhaft die bauliche Zusammensetzung, durch die am Ende eine Art Hierarchie in Form einer Strukturliste entsteht. Weiterhin können auch funktionale Beziehungen vorhanden sein, die z.B. eine speicherprogrammierbare Steuerung (SPS) mit den Parametern von einzelnen Feldgeräten zur Erfüllung einer definierten Funktion verbindet. Alle diese Beziehungen können in den VWS modelliert werden (s. Kapitel 4). Dargestellt ist die beschriebene Situation in Abbildung 3.2. Das Bild zeigt das Repository des Maschinenherstellers. Darin gespeichert sind seine hergestellten Produktionsmodule, wobei eines dieser Produktionsmodule abgebildet ist. Zu sehen sind die vom Maschinenhersteller entwickelten VWS in blau und die VWS der zugekauften Komponenten in schwarz. Durch blaue rechteckige Kästen sind die Teilmodelle und Views hervorgehoben, die durch den Maschinenhersteller hinzugefügt wurden. Des Weiteren sind in diesem Bild erstmals die oben genannten Beziehungen zwischen VWS grafisch beschrieben (blau umrandete

Raute mit der Beschriftung „Verweis-VWS“). Es ist denkbar, diese Beziehungen zwischen einer VWS und der zu ihr in Verbindung stehenden VWS in dafür vorgesehene Teilmodelle zu integrieren.

Nach der Fertigstellung des Produktionsmoduls, wird dieses an das produzierende Unternehmen ausgeliefert. Mit der Übergabe des Produktionsmoduls werden auch die VWS an das produzierende Unternehmen übergeben. Dabei erhält der Kunde aber nur die Informationen, die vertraglich vereinbart wurden. Urheberrechtlich geschützte Teilmodelle, die das Know-How des Maschinenherstellers beinhalten, werden aus den VWS für den Kunden entfernt. Der Maschinenhersteller behält folglich die originalen VWS bei sich und übergibt dem Kunden nach Auslieferung eine Art Duplikat der Produktionsmodul-VWS mit seinen untergeordneten VWS als serialisierte Dateien. Die duplizierten VWS müssen dabei eine Referenz auf die originalen VWS zur Rückverfolgbarkeit besitzen, wie es in der DIN SPEC 91345 [1] definiert ist. Dieser Verweis ist in Abbildung 3.2 beispielhaft an der VWS des Produktionsmoduls durch einen blauen gestrichelten Pfeil verdeutlicht.

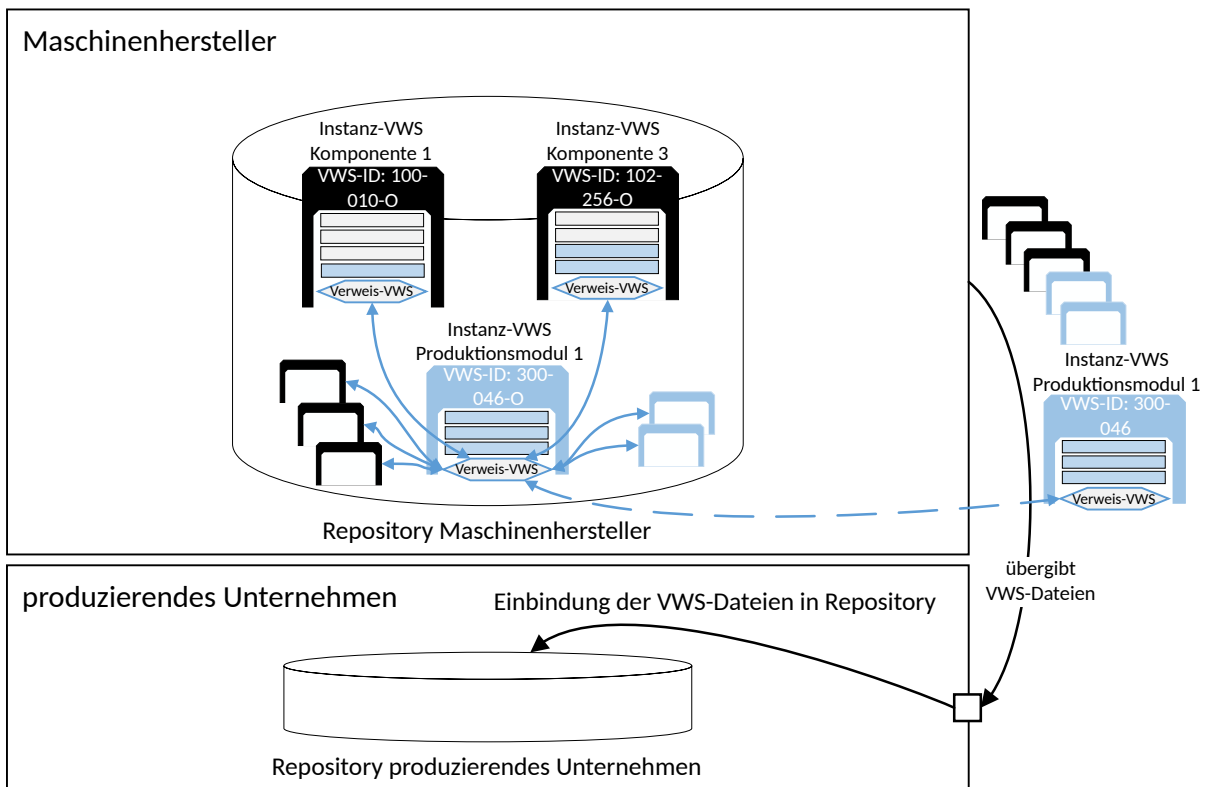


Abbildung 3.2: Bereitstellung der VWS durch den Maschinenhersteller

Am Ende der Lieferkette befindet sich das produzierende Unternehmen, das den Gerätetausch während der Nutzungsphase des Produktionsmoduls veranlasst. Wie beschrieben, wird dem produzierenden Unternehmen die VWS des Produktionsmoduls mit seinen dazugehörigen untergeordneten VWS von dem Maschinenhersteller als Dateien (passive VWS) übergeben. Dadurch ist es dem produzierenden Unternehmen möglich, nach Einbindung der Dateien in sein Repository, diese in reaktive VWS umzuwandeln und eigene

Informationen in den VWS zu ergänzen. Dazu zählen beispielsweise Beziehungen zu der VWS des übergeordneten Produktionssystems oder die Einfügung neuer Teilmodelle in die Verwaltungsschalen des Produktionsmoduls.

Abbildung 3.3 zeigt das Repository des produzierenden Unternehmens und stellt die Informationen in grün dar, die durch dieses hinzugefügt wurden. Zur Erhaltung der Übersichtlichkeit, sind die durch den Maschinenhersteller eingefügten Beziehungen nicht explizit durch Pfeile dargestellt. Sie sind durch die Abbildung der blau umrandeten Rauten mit der Beschriftung „Verweis-VWS“ implizit gekennzeichnet.

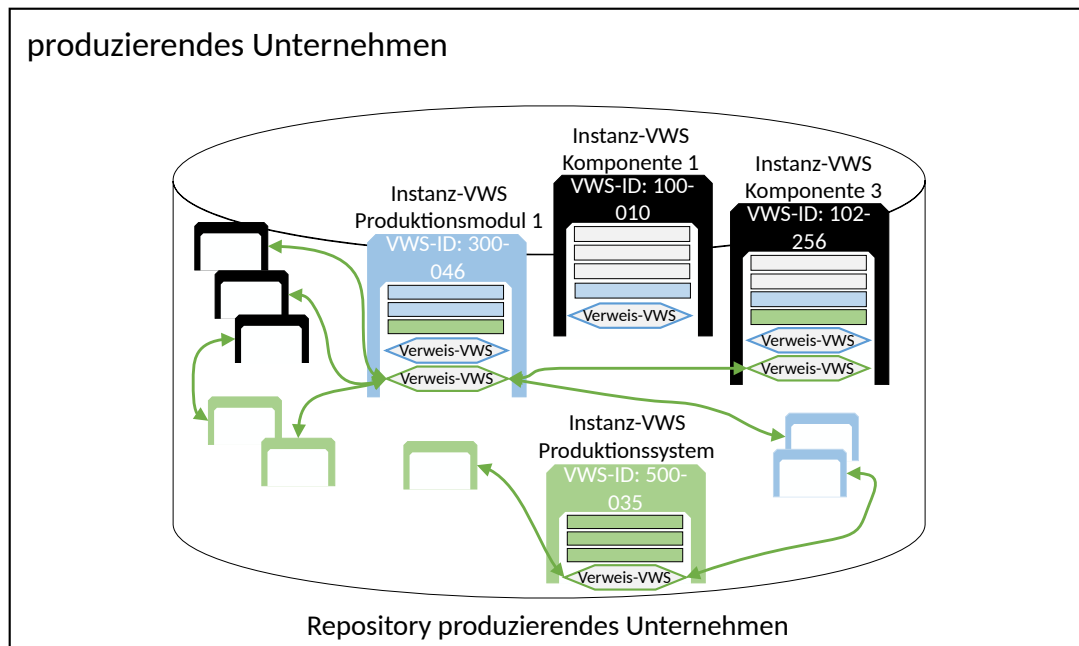


Abbildung 3.3: Speicherung der VWS durch das produzierende Unternehmen

Mit den Repositories der drei genannten Akteure ist es diesen möglich, den digitalen Zwilling des Assets mit individuellen Informationen zu erweitern. Die Informationen können dann je nach Situation für den jeweiligen Kunden zur Verfügung gestellt werden. Dadurch wird eine klare Abgrenzung der Aufgabenbereiche aller drei Akteure vorgenommen, die unabhängig voneinander zu handhaben sind.

Diese Architektur bietet sich durch zusätzliche Erweiterungen auch für den Use Case „Value-Based Services für Produktionsressourcen“ (s. Kapitel 2.4) an. Dafür werden seitens des Komponenten- und Maschinenherstellers Zugänge zu separaten Repositories gelegt, die eine Interaktion zwischen den Akteuren ermöglichen.

Durch die freigegebenen Zugänge und die Referenzen zwischen den VWS eines Assets kann beispielsweise der Komponentenhersteller eine Benachrichtigung an den Maschinenhersteller senden, sobald neue Dokumente eingefügt oder anderweitige Änderungen in den Typ-VWS der verkauften Komponente vorgenommen wurden. Der Maschinenhersteller ist daraufhin in der Lage, die Daten bei dem Komponentenhersteller zu lesen und in seine VWS zu übertragen.

Eine weitere Option ist, dass der Maschinenhersteller regelmäßig bei dem Komponentenhersteller nach neuen Dokumenten oder Änderungen fragt. Als Antwort erhält er dementsprechend die Informationen. Beide genannten Interaktionen können genauso zwischen dem Maschinenhersteller und dem produzierenden Unternehmen stattfinden. Der Nachrichtenaustausch für beide Optionen ist in Abbildung 3.4 nachgebildet. Dabei bildet der gesendete Trigger die Option, Benachrichtigungen über Änderungen an das System des Kunden zu senden.

Es ist zu erwähnen, dass die Verwaltungsschalen nicht in einem gemeinsamen Repository liegen müssen, wie das in den Beispielen veranschaulicht ist. Vielmehr ist es wichtig, dass eine Registry die Endpunkte der VWS kennt, um einer Applikation den Zugriff auf die VWS-API zu ermöglichen.

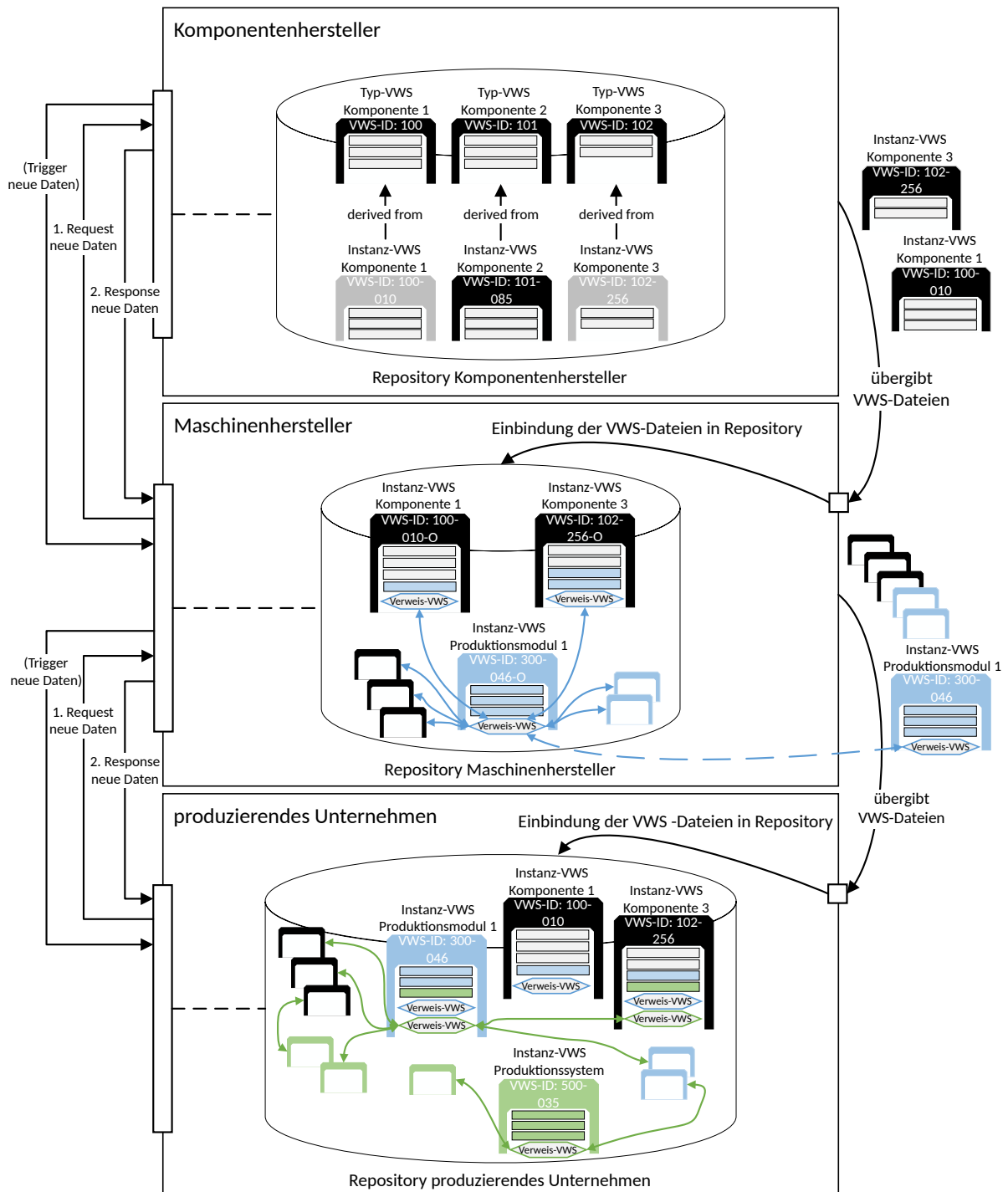


Abbildung 3.4: Repositories zur Speicherung von reaktiven Verwaltungsschalen

3.2 Szenario 1 - Gerätetausch durch Instandhaltungspersonal

Es kommt häufig vor, dass Wartungsarbeiten von dem Instandhaltungspersonal des produzierenden Unternehmens durchgeführt werden. Für solche Wartungsarbeiten wird in diesem Szenario ein Instandhaltungssystem genutzt, das Zugriff auf die Verwaltungsschalen aller Assets im Unternehmen hat. Es ermöglicht einen teilautomatisierten Gerätetausch und setzt als Entscheidungsträger das Instandhaltungspersonal ein, das den korrekten Ablauf des Gerätetausches überwacht.

Der Beginn des Austausches der Kamera wird durch die Erstellung eines Auftrags durch den Instandhalter an das Instandhaltungssystem initiiert. Voraussetzung für den Auftrag ist, dass sich die Roboterzelle in einem sicheren Zustand befindet, damit grundsätzlich ein sicherer Gerätetausch stattfinden kann. Ist die Voraussetzung erfüllt, werden durch den Scan der QR-Codes auf dem Typenschild der alten und neuen Kamera dem Instandhaltungssystem die entsprechenden Asset-IDs übermittelt. Die Suche und Anzeige beider zu den Assets zugehörigen VWS in einer Bedienoberfläche ermöglicht dem Instandhalter die Identifizierung der Assets auf Richtigkeit zu kontrollieren. Eine anschließende Überprüfung der technischen Daten beider Kameras durch das Produktionsmodul², soll den Einsatz eines nicht kompatiblen Geräts verhindern. Die Prüfung garantiert, dass die funktionalen Anforderungen des alten Geräts mit denen des neuen Geräts übereinstimmen.

Mit der Identifizierung der alten Kamera ist das Instandhaltungssystem in der Lage, alle Verweise auf diese zu finden. Wurden alle Verweise ermittelt, werden diese dem Instandhalter grafisch angezeigt, sodass dieser die Vollständigkeit und Richtigkeit prüfen kann. Gibt der Instandhalter dem Instandhaltungssystem seine Zustimmung zu den geplanten Änderungen, werden diese von dem System im letzten Schritt durchgeführt. Damit ist die Dokumentation der Roboterzelle und aller überlagerten Systeme wieder auf dem aktuellen Stand.

Das Use Case Diagramm zu der Beschreibung ist in Abbildung 3.5 zu sehen. Auf Grund der Übersichtlichkeit, ist die Abbildung aller Beziehungen zwischen den Akteuren und dem Instandhaltungssystem (orange hervorgehobene Verbindungen) nicht möglich. Diese werden erst mit den Sequenzdiagrammen ersichtlich.

Das Sequenzdiagramm des Szenarios ist aufgrund des Umfangs in mehrere kleinere Sequenzdiagramme gegliedert, die die einzelnen Teilaufgaben aus dem Use Case Diagramm (s. Abbildung 3.5) widerspiegeln. Jede der Nachrichten während der Interaktion ist mit einer Nummer ausgestattet. Zur Nachvollziehbarkeit, erscheinen einzelne Nachrichtennummern an der entsprechenden Stelle im Text und sind kursiv gekennzeichnet.

Das erste Sequenzdiagramm befasst sich mit der Auftragsvergabe und Identifizierung der

²Die Identifizierung der Parameter ist mit Hilfe der semanticId möglich, sodass die Parameterwerte anschließend auf Gleichheit geprüft werden können (vgl. [26]).

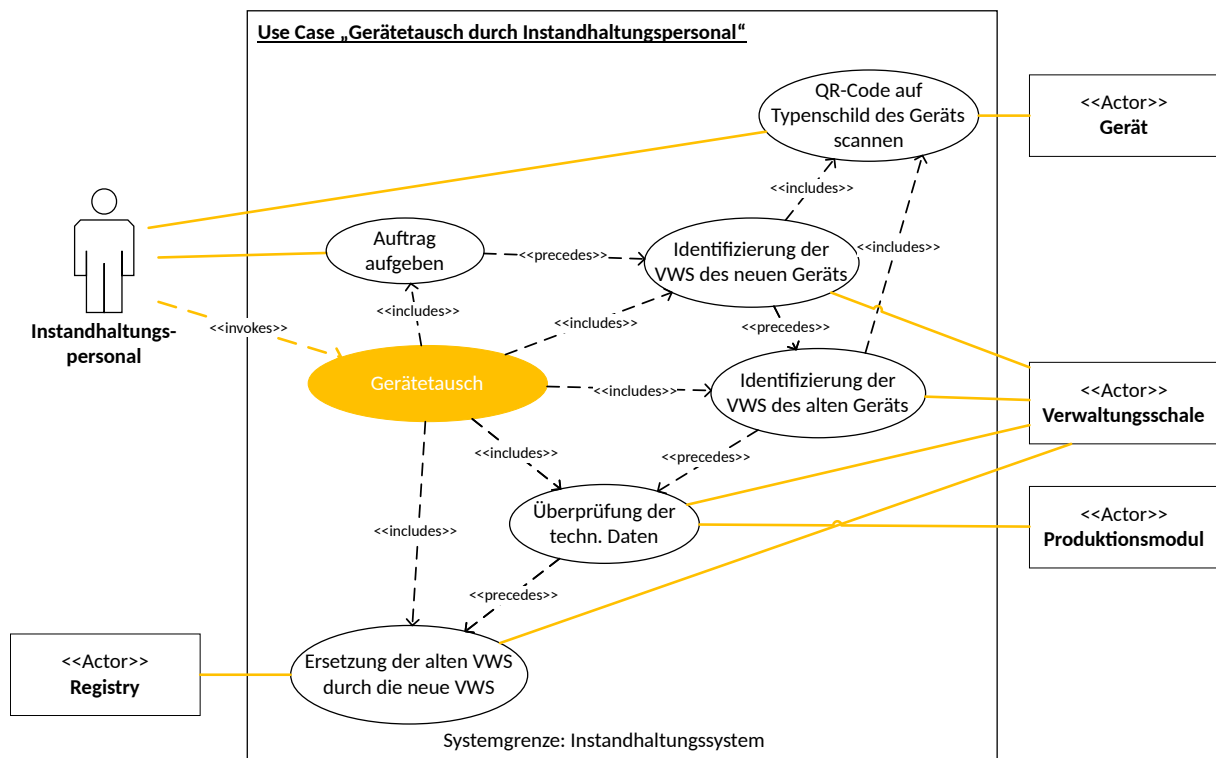


Abbildung 3.5: Use Case Diagramm Szenario 1 - Gerätetausch durch Instandhaltungspersonal

VWS der zu tauschenden Geräte und ist in Abbildung 3.6 zu sehen. Die Interaktion wird seitens des Instandhalters mit einer Auftragsvergabe begonnen (1.0). Danach fordert das Instandhaltungssystem (im folgenden IHS genannt) den Instandhalter auf, die QR-Codes der Typenschilder nacheinander zu scannen (1.1), (1.9). Mit dem Scan erhält das IHS die Asset-IDs der Kameras (1.4), (1.12) und schickt diese weiter an die Registry, um den Endpunkt der zu dem Gerät gehörenden VWS zu erhalten (1.5), (1.13). Sind die Endpunkte bekannt, kann das IHS die VWS über deren API erreichen und sich die notwendigen Informationen beschaffen (1.7), (1.15). Die Informationen dienen anschließend dem Instandhalter, damit er die Richtigkeit der gefundenen VWS bestätigen kann (1.17). Sollte eine der gefundenen VWS die Prüfung durch den Instandhalter nicht bestehen, so muss die beschriebene Interaktion nochmal ausgeführt oder nach möglichen Fehlern gesucht werden. Dieser Fall ist im Sequenzdiagramm aus Abbildung 3.6 nicht explizit dargestellt.

Mit der Bestätigung der gefundenen VWS durch den Instandhalter wird die Überprüfung der technischen Daten und die Ersetzung der VWS des alten Geräts durch die des neuen Geräts vorgenommen. Das dazugehörige Sequenzdiagramm ist in Abbildung 3.7 dargestellt. Für die Überprüfung der technischen Daten, sendet das IHS die Daten beider VWS an das Produktionsmodul (1.0). Die Steuerung bzw. eine Softwareapplikation des Produktionsmoduls übernimmt den anschließenden Datenabgleich (1.1). Ist die Kompatibilität zwischen dem alten und neuen Gerät nicht gegeben, endet das Sequenzdiagramm mit

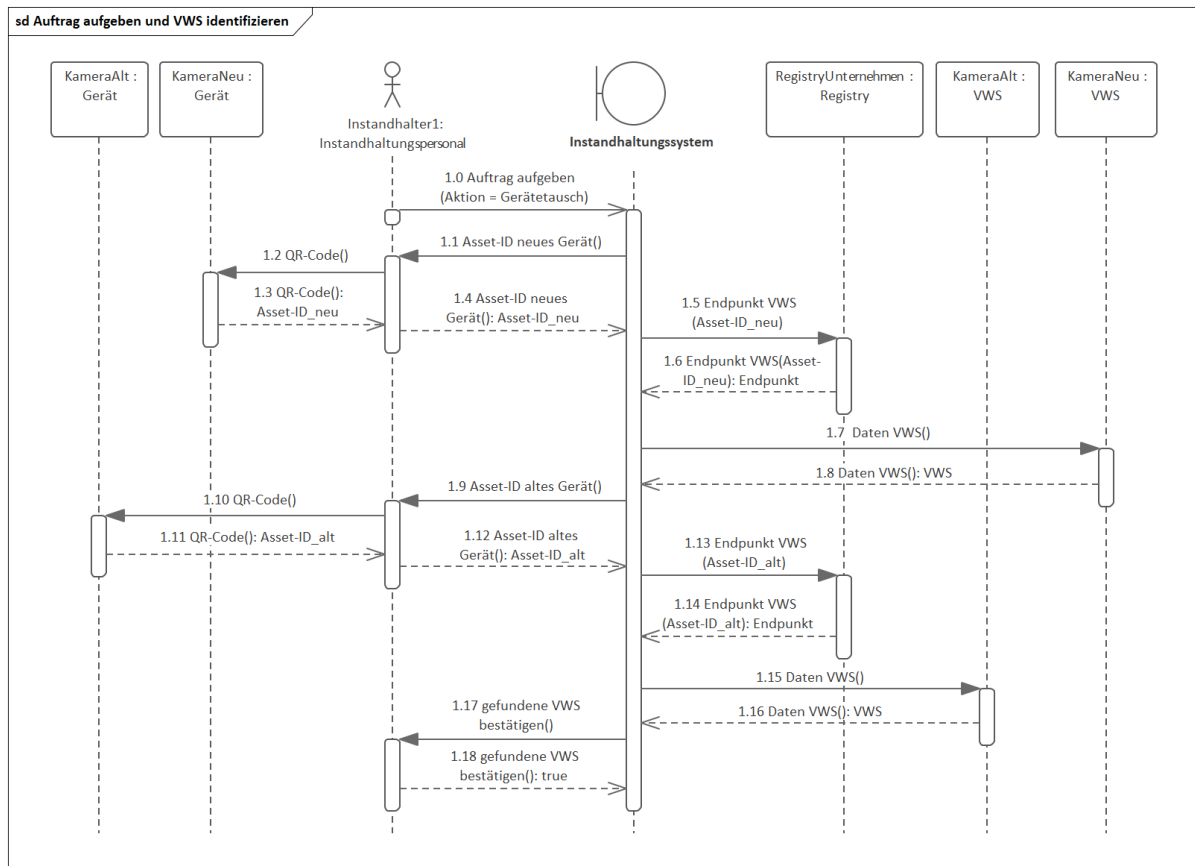


Abbildung 3.6: Sequenzdiagramm Szenario 1 (Teil 1) - Auftrag aufgeben und VWS identifizieren

einem Abbruch des Gerätetausches (nicht in der Abbildung dargestellt). Sind die beiden Geräte wiederum kompatibel, findet daraufhin die Suche nach VWS, die in Beziehung mit der VWS der alten Kamera stehen, statt (1.3). Der Interaktionsrahmen, der die Suche umschließt ist mit „GET Beziehung“ bezeichnet.

Die Suche in dem Interaktionsrahmen läuft in einer Schleife ab und endet, sobald keine Beziehung mehr in der VWS der alten Kamera gefunden wurde. Der Grund, weshalb in diesem Beispiel nur in der VWS des alten Geräts nach Beziehungen gesucht wird, erschließt sich in Kapitel 4.1.

Ist die Suche nach Beziehungen erfolgreich beendet, werden die Ergebnisse dem Instandhalter angezeigt (1.5). Der Instandhalter weiß wiederum, in welchen Zusammenhängen die alte Kamera mit dem Produktionsmodul und überlagerten Systemen steht und kann somit eine Verifizierung der Ergebnisse vornehmen. Bestätigt der Instandhalter dem IHS die Richtigkeit (1.6), beginnt die Aktualisierung der betroffenen Verwaltungsschalen, die in dem Interaktionsrahmen „UPDATE Beziehung“ dargestellt ist. Darin wird iterativ eine der zuvor gefundenen Beziehungen genommen und von der VWS „KameraAlt“ in die VWS „KameraNeu“ übertragen (1.7). Daraufhin werden die anderen VWS nach der selben Beziehung zur VWS „KameraAlt“ durchsucht (1.9). Die Anzahl der VWS, die eine

Beziehung zu der VWS der alten Kamera unterhalten sind durch die Multiplizität in der rechten oberen Ecke des Objekts „VWSMitVerweis“ gekennzeichnet. Ist die Beziehung in der „VWSMitVerweis“ gefunden wird diese dahingehend aktualisiert, sodass sie auf die VWS „KameraNeu“ zeigt (1.11). Anschließend wird die Löschung der ursprünglichen Beziehung in der VWS „KameraAlt“ veranlasst (1.13), da die Kamera nach dem physischen Gerätetausch auch nicht mehr in der Roboterzelle verbaut ist. Damit wird eine Möglichkeit geschaffen, die alte Kamera nach der Instandsetzung an einem anderen Ort in der Produktion weiter zu verwenden. Daraufhin benachrichtigt das IHS den Instandhalter über die erfolgreiche Ersetzung der VWS des alten Geräts mit der VWS des neuen Geräts (1.15). Die Aufgabenstellung des Use Case wurde damit konzeptionell erfüllt.

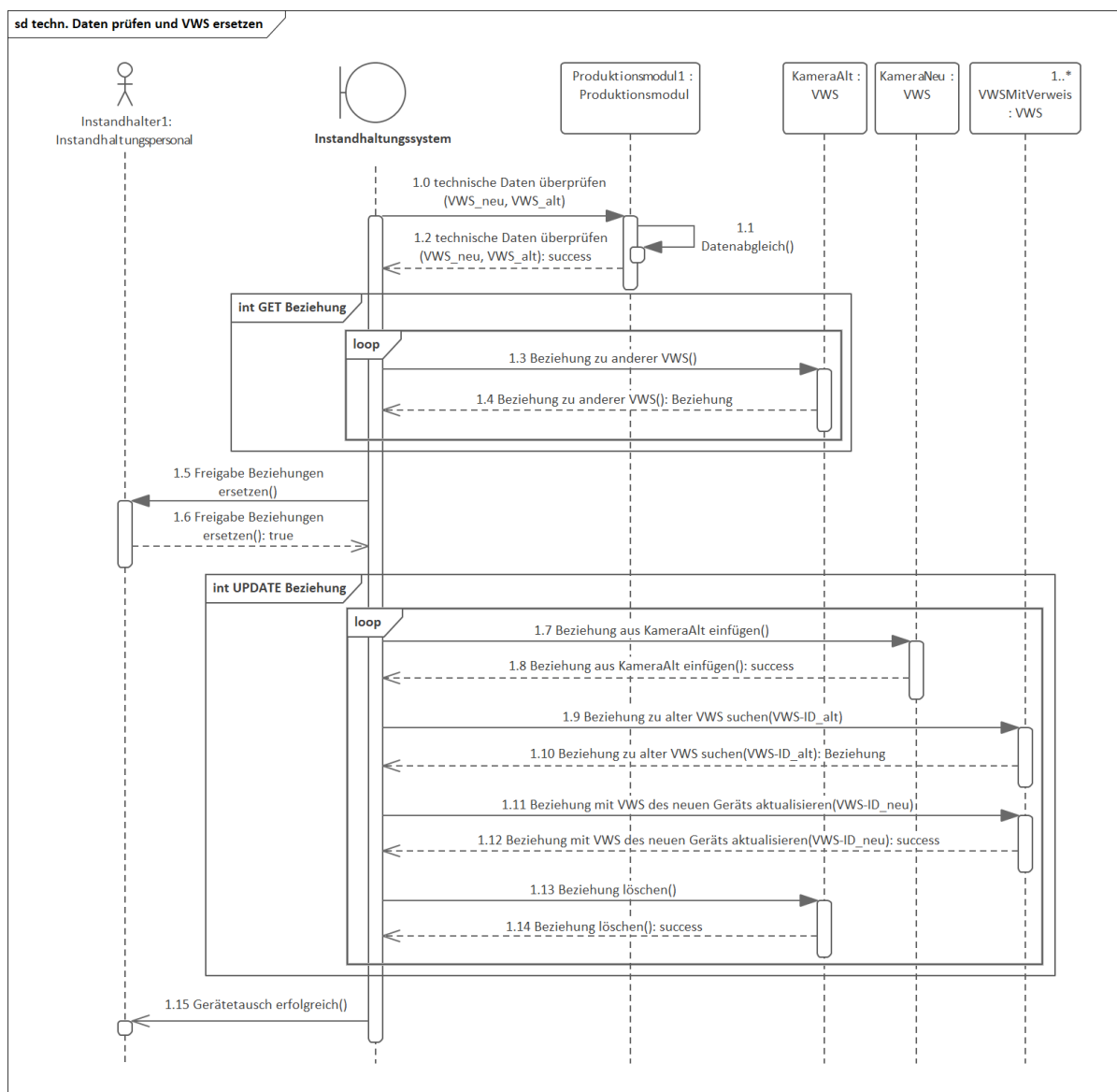


Abbildung 3.7: Sequenzdiagramm Szenario 1 (Teil 2) - Überprüfung der techn. Daten und Ersetzung der VWS

Die Umsetzung der Suchverfahren in den Interaktionsrahmen „GET Beziehung“ und „UPDATE Beziehung“ gestalten sich unterschiedlich, je nach Modellierung der Beziehungen zwischen Verwaltungsschalen. Auf die verschiedenen Beziehungsmodellierungen mit ihren Vor- und Nachteilen wird in Kapitel 4.1 näher eingegangen.

Wichtig für den gesamten Vorgang ist die Verwendung von VWS-Elementen, die die Darstellung von Beziehungen zwischen den VWS ermöglichen. Darunter fallen z.B. das Relationship-Element, das Reference-Element, die Entity aber auch andere VWS-Elemente, deren Eigenschaften in Kapitel 4.2 analysiert werden.

3.3 Szenario 2 - Gerätetausch durch Hersteller des Produktionsmoduls

Besteht ein Wartungsvertrag zwischen dem produzierenden Unternehmen und dem Hersteller des Produktionsmoduls, ist das Servicepersonal des Herstellers für künftige Gerätetausche verantwortlich. Dabei ist es aus Sicht der IT-Security des produzierenden Unternehmens wichtig, Benutzerrollen für das Servicepersonal festzulegen. Die Benutzerrollen legen die Zugriffe auf die Verwaltungsschalen des produzierenden Unternehmens fest.

Es ist vorstellbar, dass ähnlich dem Instandhaltungssystem aus Szenario 1 in diesem Fall ein sogenanntes Servicetool den Gerätetausch durchführt. Da das Servicetool im Besitz des Herstellers des Produktionsmoduls ist, wird die Aktualisierung der VWS nicht nur bei dem produzierenden Unternehmen durchgeführt. Das Servicetool ist auch in der Lage die VWS-Struktur in den Repositories des Herstellers zu aktualisieren. Dadurch sind die Systeme beider Akteure immer auf dem aktuellen Stand. Die Ausgestaltung der Systemarchitekturen mit Repositories zur Speicherung von VWS ist in Kapitel 3.1 nachzulesen.

Bevor der Austausch der Kamera durch das Servicepersonal des Herstellers möglich ist, muss dieses erst den Zugriff bei dem produzierenden Unternehmen anfragen. Das geschieht über das Servicetool des Herstellers. Wird der Zugriff durch das System des produzierenden Unternehmens erteilt kann die Durchführung des Gerätetausches beginnen. Das Servicepersonal muss dazu die Roboterzelle in einen sicheren Zustand bringen, um einen reibungslosen Gerätetausch zu ermöglichen.

Nach der Zugriffsanfrage, gibt das Servicepersonal über das Servicetool einen Auftrag bei dem System des produzierenden Unternehmens auf. Mit der Freigabe durch das System, kann das Servicepersonal die QR-Codes auf den Typenschildern der alten und neuen Kamera scannen. Die Informationen werden an das Servicetool übergeben und die Suche nach den zugehörigen VWS und den Beziehungen zur VWS der alten Kamera angestoßen.

Die Suche wird dabei nicht nur bei der Registry des produzierenden Unternehmens, sondern auch bei der Registry des Herstellers ausgeführt. Alle Ergebnisse des Servicetools werden dem Servicepersonal ähnlich dem Szenario 1 angezeigt.

Nach der Identifizierung der VWS beider Geräte, findet auch in diesem Szenario die Prüfung auf Kompatibilität der Geräte statt. Nach Bestätigung der Richtigkeit, werden dem Servicepersonal die gefundenen Beziehungen zur VWS der alten Kamera angezeigt. Mit einer weiteren Bestätigung durch das Servicepersonal stimmt dieses dem Tausch der alten und neuen VWS und den damit einhergehenden Änderungen zu. Ist der Gerätetausch erfolgreich beendet gibt das Servicetool den Zugriff auf das System des produzierenden Unternehmens wieder ab.

Das komplette Use Case Diagramm mit seinen Teilschritten ist in Abbildung 3.8 zu sehen. Zu Zwecken der Übersichtlichkeit wurden nicht alle Beziehungen der Akteure zu dem Servicetool eingezeichnet (orange hervorgehobene Verbindungen).

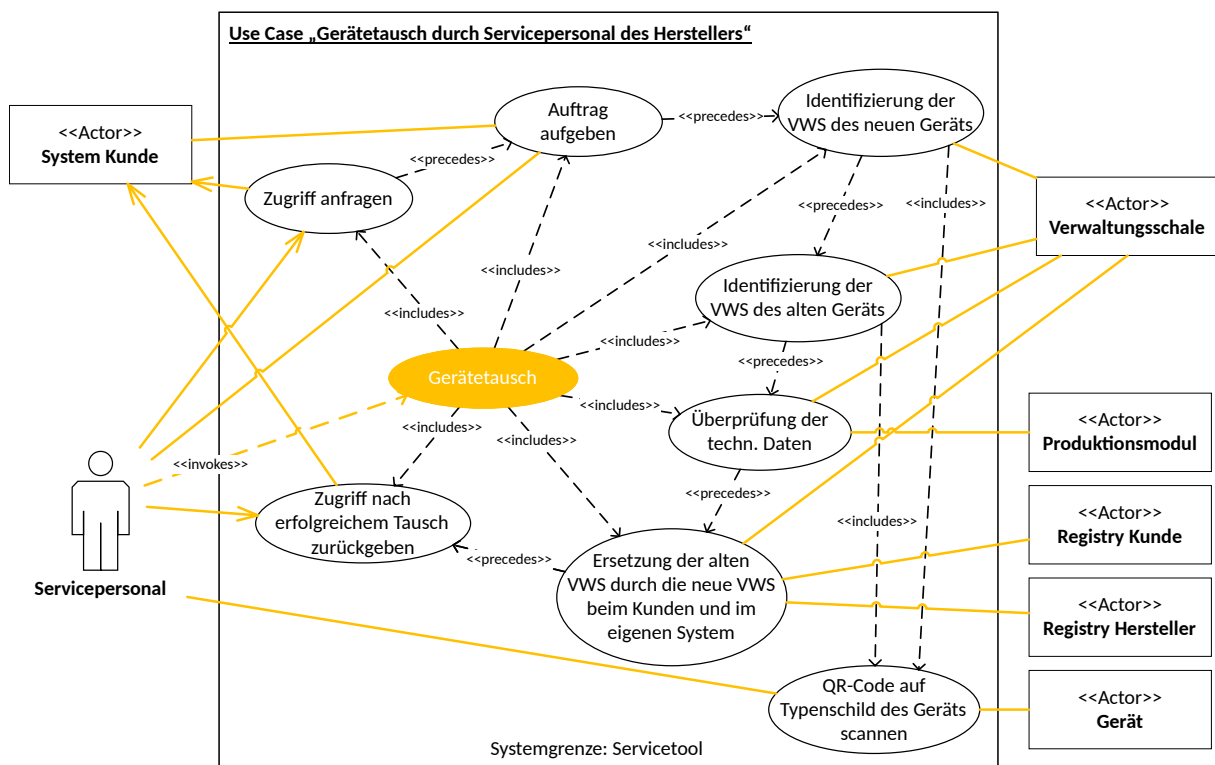


Abbildung 3.8: Use Case Diagramm Szenario 2 - Gerätetausch durch den Hersteller des Produktionsmoduls

Das Sequenzdiagramm des Szenarios ist aufgrund des Umfangs in drei kleinere Sequenzdiagramme gegliedert, die die einzelnen Teilaufgaben aus dem Use Case Diagramm (s. Abbildung 3.8) widerspiegeln. Zur Erfüllung der Aufgabenstellung, können sich die aufgegliederten Sequenzdiagramme, im Gegensatz zu Szenario 1, wiederholen. Sobald das notwendig ist, wird es im folgenden Text beschrieben. Jede der Nachrichten während der Interaktion ist nummeriert und wird für die Rückverfolgung an der entsprechenden Stelle im Text kursiv gekennzeichnet.

Der erste Abschnitt der Interaktion wird in dem Sequenzdiagramm aus Abbildung 3.9 realisiert. Darin geht es um die Zugriffsanfrage und Auftragsvergabe durch das Servicepersonal. Dazu meldet sich das Servicepersonal vor dem Gerätetausch über das Servicetool bei dem System des produzierenden Unternehmens an (1.0). Das System verifiziert den Zugriff (1.2) und gibt ihn bei erfolgreicher Prüfung frei (1.3). Danach ist das Servicepersonal in der Lage einen Auftrag über das Servicetool (1.5) an das System zu senden (1.6). Dieses gleicht anschließend den Auftrag z.B. mit den Daten aus einem ERP-System ab und bestätigt den Auftrag dem Servicetool. Damit ist der erste Abschnitt der Interaktion beendet. Sollten Fehler bei der Überprüfung des Zugriffs oder der Verifizierung des Auftrags auftreten, wird der Gerätetausch abgebrochen und muss von vorne begonnen werden. Diese Möglichkeiten sind in dem Sequenzdiagramm aus Abbildung 3.9 nicht eingezeichnet.

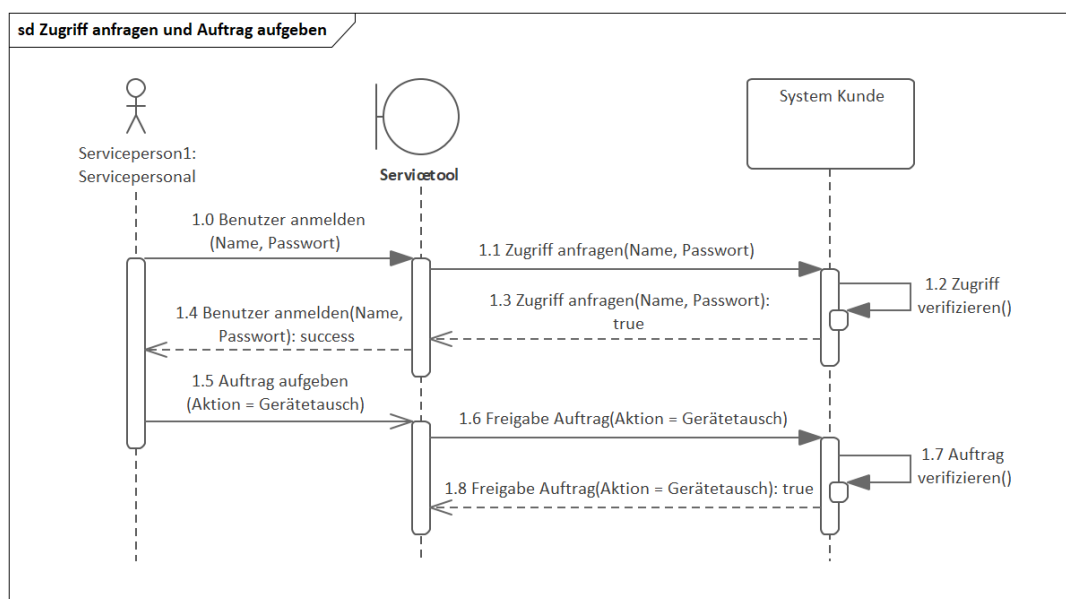


Abbildung 3.9: Sequenzdiagramm Szenario 2 (Teil 1) - Zugriff anfragen und Auftrag aufgeben

Nach der Auftragsvergabe erfolgt im zweiten Abschnitt die Identifizierung der VWS beider Geräte. Der Ablauf ist, bis auf die erste Nachricht zur Auftragsvergabe, derselbe wie in Szenario 1 (s. Kapitel 3.2 Abb. 3.6) und kann dort nachgelesen werden. Der einzige Unterschied zwischen beiden Sequenzdiagrammen besteht in der Bezeichnung der Registry. In diesem Szenario erhält die Registry keinen eindeutigen Namen, da das Sequenzdiagramm sowohl zur Suche der VWS im System des produzierenden Unternehmens als auch im System des Herstellers erfolgt. Damit muss das Sequenzdiagramm zweimal durchlaufen werden. Einmal um durch die Registry die Endpunkte der VWS in den Repositories des produzierenden Unternehmens zu erhalten. Ein zweites Mal um die Endpunkte der VWS in den Repositories des Herstellers festzustellen. Eine Speicherung der Ergebnisse in dem Servicetool ist für die nachfolgenden Interaktionen notwendig. Die Sequenz ist zur Vollständigkeit in Abbildung 3.10 wiedergegeben.

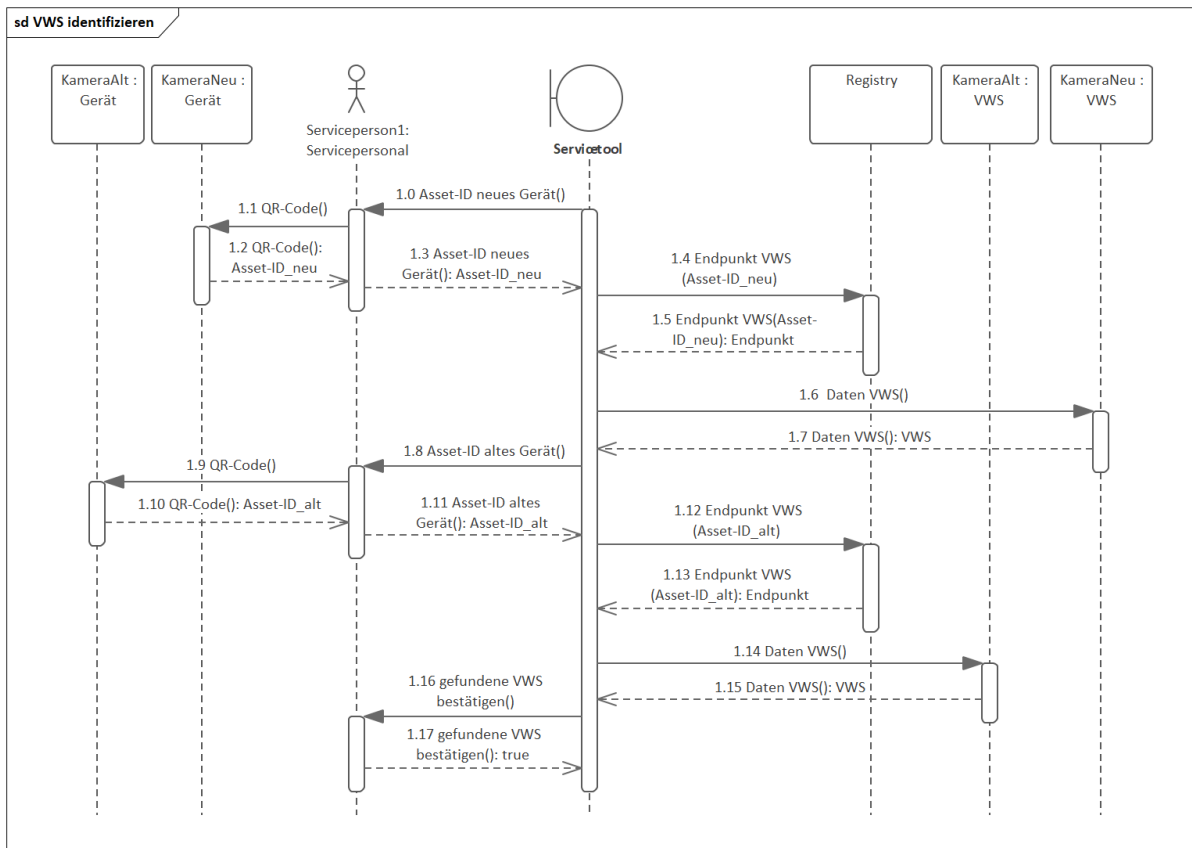


Abbildung 3.10: Sequenzdiagramm Szenario 2 (Teil 2) - Verwaltungsschalen identifizieren

Mit der Bestätigung der korrekten Identifizierung der VWS beider Geräte schließt sich die Aktualisierung der Beziehungen zwischen den VWS an. Auch hier ist der Ablauf derselbe wie in Szenario 1, dessen ausführliche Beschreibung in Kapitel 3.2 nachgelesen werden kann. Abbildung 3.11 zeigt wiederholt das Sequenzdiagramm für die Aktualisierung der VWS-Struktur. Zur Erfüllung der Aufgabenstellung ist eine Wiederholung des Sequenzdiagramms notwendig, da dieses nur die Aktualisierung in einem System vornimmt.

Es wird ersichtlich, dass durch die Anwendung einer Modulbauweise eine gewisse Flexibilität zwischen der Abfolge der Sequenzdiagramme entsteht. Man erhält die Möglichkeit, sich die Sequenzdiagramme aus den Szenarien individuell zusammenzustellen. Beispielsweise ist eine weitere Option für den Ablauf der Interaktion, zuerst die VWS in dem System des produzierenden Unternehmens zu identifizieren und sie anschließend auszutauschen und danach die gleiche Prozedur für das System des Herstellers durchzuführen.

Nach der erfolgreichen Aktualisierung der Verwaltungsschalen in beiden Systemen kann sich das Servicepersonal von dem Servicetool abmelden und den Zugriff auf das System des produzierenden Unternehmens wieder freigeben. Das Sequenzdiagramm zur Erfüllung dieser Aufgabe ist in Abbildung 3.12 zu sehen.

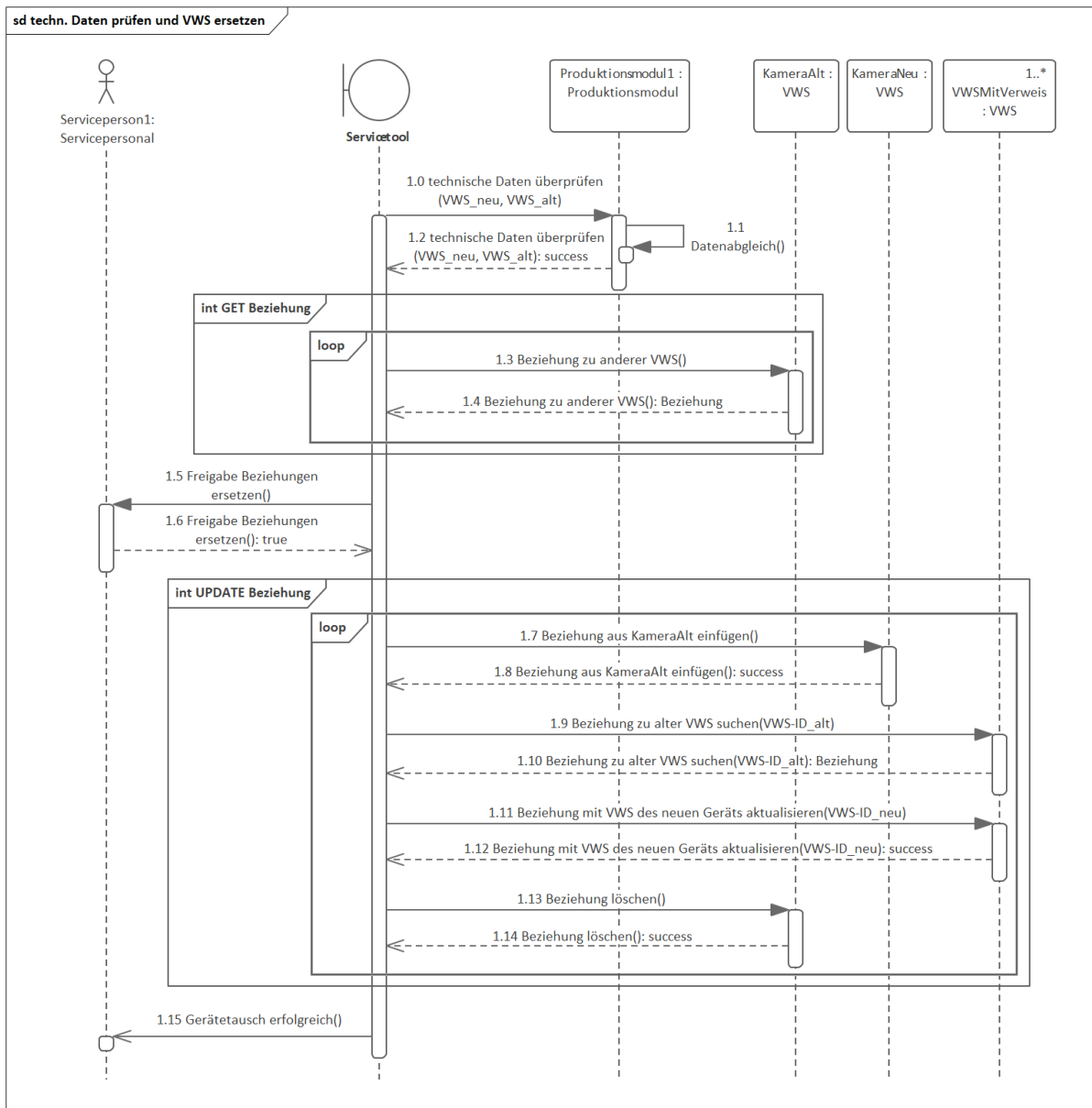


Abbildung 3.11: Sequenzdiagramm Szenario 2 (Teil 3) - Überprüfung der techn. Daten und Ersetzung der VWS

Ausgehend von der Use Case Beschreibung in der Einleitung, zur konzeptionellen Umsetzung der Dokumentationspflege bei einem Gerätetausch, war die Rede von inkonsistenten Daten zwischen dem Maschinenhersteller und dem produzierenden Unternehmen. Der Grund dafür liegt darin, dass in der herkömmlichen Dokumentationspflege keine direkte Verknüpfung der Dokumente mit den Assets besteht. Die Dokumente liegen, abgekoppelt von den in den Produktionssystemen verbauten Assets, in Dateisystemen der Verwaltungsebene. Änderungen am Produktionssystem werden in die Dokumente eingepflegt und sind damit immer nur dem betroffenen Unternehmen, in diesem Fall dem produzierenden Unternehmen, zugänglich. Dies kann beispielsweise die Änderung von Parametern eines Feldgeräts sein, die unter anderem eine qualitative Produktionsverbesserung gewährleisten.

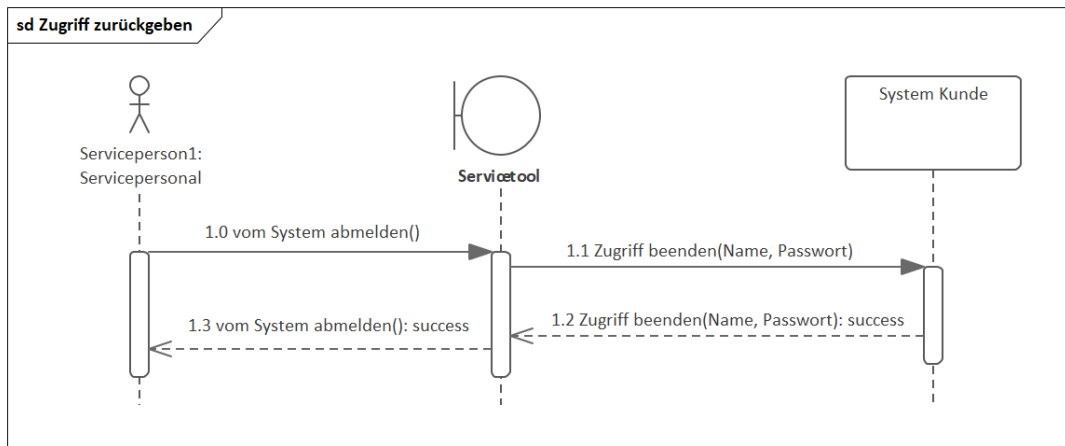


Abbildung 3.12: Sequenzdiagramm Szenario 2 (Teil 4) - Zugriff wieder zurückgeben

Tauscht der Maschinenhersteller im Zuge einer Wartung dieses Feldgerät, hat er nur die Parametrierung, die bei der Inbetriebnahme oder einer zurückliegenden Wartung an das produzierende Unternehmen übergeben wurde. Bei Einsatz eines neuen Feldgeräts wird auf ihm eine veraltete Parametrierung abgespeichert.

Durch die VWS ist es jetzt aber möglich, diese fehlende Kopplung herzustellen. Mit der zusätzlichen Modellierung von Beziehungen zwischen verschiedenen VWS gelingt eine weitere Verknüpfung, die vorher in unterschiedlichen Unterlagen dokumentiert waren. Damit bildet die VWS den Mittelpunkt für den Maschinenhersteller, von dem aus alle für ihn relevanten VWS erreichbar und zusätzlich immer die aktuellsten Informationen zu dem Asset verfügbar sind. Zusätzlich bietet die Verknüpfung der VWS mit dem Asset einen Vorteil für das produzierende Unternehmen. Es muss bei einer Wartung durch den Maschinenhersteller die Daten nicht mehr manuell bereitstellen und senkt damit den Arbeitsaufwand und die Fehlerwahrscheinlichkeit für das Personal. Weiterhin sind damit redundante bzw. inkonsistente Informationen ausgeschlossen, wie z.B. die Speicherung eines Parameters an unterschiedlichen Orten mit verschiedenen Werten.

3.4 Szenario 3 - vollautomatisierter Gerätetausch durch Roboter

In Hinblick auf die vollkommene Vernetzung von Assets sowohl in vertikaler als auch horizontaler Richtung sind eigenständige Wartungsarbeiten durch die Produktionsressourcen selbst denkbar. In diesem Beispiel ist der Roboter in der Lage, die Kamera autonom zu wechseln. Dazu müssen dem Roboter Randbedingungen mitgegeben werden, durch die er erkennt, ab wann die Kamera nicht mehr gebrauchsfähig ist. Für den Austausch der Kamera muss an einer Kamerawechselstation eine zweite Kamera bereitstehen. Die Bereitstellung kann durch das Instandhaltungspersonal oder durch ein autonomes Transportsystem geschehen. Für den Scan in diesem Szenario ist ein RFID-Chip auf dem Gerät

geeigneter als ein QR-Code auf dem Typenschild, da dieser nicht immer an der gleichen Stelle auf den Kameras platziert sein muss.

Vorteil dieses Szenarios ist die Vernetzung unter den Produktionsmodulen. Sobald der Roboter feststellt, dass die Kamera nicht mehr einsatzfähig ist, z.B. aufgrund von äußeren Verschmutzungen, trifft der Roboter Absprachen mit den Produktionsmodulen des Produktionssystems. Dadurch können vor- und nachgelagerte Produktionsmodule besser reagieren und, z.B. durch die Verlangsamung der Produktion, dem Roboter ein Zeitfenster für den sicheren Austausch der Kamera verschaffen.

Der Ablauf für den Austausch der Kamera gestaltet sich ähnlich den ersten beiden Szenarios. Registriert der Roboter eine Beeinträchtigung der Bildaufnahmen, stellt dieser eine Anfrage am Instandhaltungssystem (IHS) zum Wechsel der Kamera. Voraussetzung für den Wechsel ist eine an der Kamerawechselstation vorhandene zweite einsatzfähige Kamera und ein genügend großes Zeitfenster für den physischen Kamerawechsel.

Bekommt der Roboter die Freigabe von dem IHS legt er die alte Kamera in der Kamerawechselstation ab und scannt gleichzeitig den RFID-Chip. Das gleiche geschieht bei Aufnahme der neuen Kamera. Die Daten beider Kamera-VWS werden an das IHS weitergeleitet und die Kameras auf Kompatibilität geprüft. Das IHS nimmt, wie bei den vorherigen Szenarios, nach der bestätigten Kompatibilität den Austausch der VWS vor und übermittelt mit erfolgreicher Beendigung des Vorgangs dem Roboter ein Signal. Dieser kann danach seine Arbeit wieder aufnehmen und die Produktion durch Kommunikation mit den vor- und nachgelagerten Produktionsmodulen wieder hochfahren. Die alte Kamera kann daraufhin, falls möglich, wieder instand gesetzt und für den späteren Einsatz in der Kamerawechselstation platziert werden. Das dazugehörige Use Case Diagramm ist in Abbildung 3.13 dargestellt und ähnelt sehr stark dem Use Case Diagramm aus Abschnitt 3.2 (s. Abbildung 3.5).

Die Verantwortungsbereiche des Roboters fallen in diesem Szenario geringer aus. Die Prüfung der Richtigkeit der gefundenen VWS übernimmt in diesem Fall das IHS, da diesem die Assetinformationen durch den RFID-Chip bereits vorliegen.

Der Roboter könnte auch in die Lage versetzt werden, diese Prüfung durchzuführen. Die Assetinformationen, die das IHS von den VWS erhalten hat, müssen dafür jedoch zusätzlich an den Roboter übermittelt werden. Das führt zu einem höheren Kommunikationsbedarf, der vermeidbar ist. Der größte Teil der Aktualisierung der Verwaltungsschalen wird von dem IHS eigenständig erledigt. Des Weiteren werden die Interaktionen des IHS mit der Registry und den VWS in Kurzform dargestellt, da die ausführlichen Sequenzdiagramme bereits in Szenario 1 und 2 behandelt wurden. Das Sequenzdiagramm für das dritte Szenario ist in Abbildung 3.14 zu sehen. Die Nachrichten des Sequenzdiagramms werden wieder durch ihre Nummerierung in kursiver Schrift referenziert.

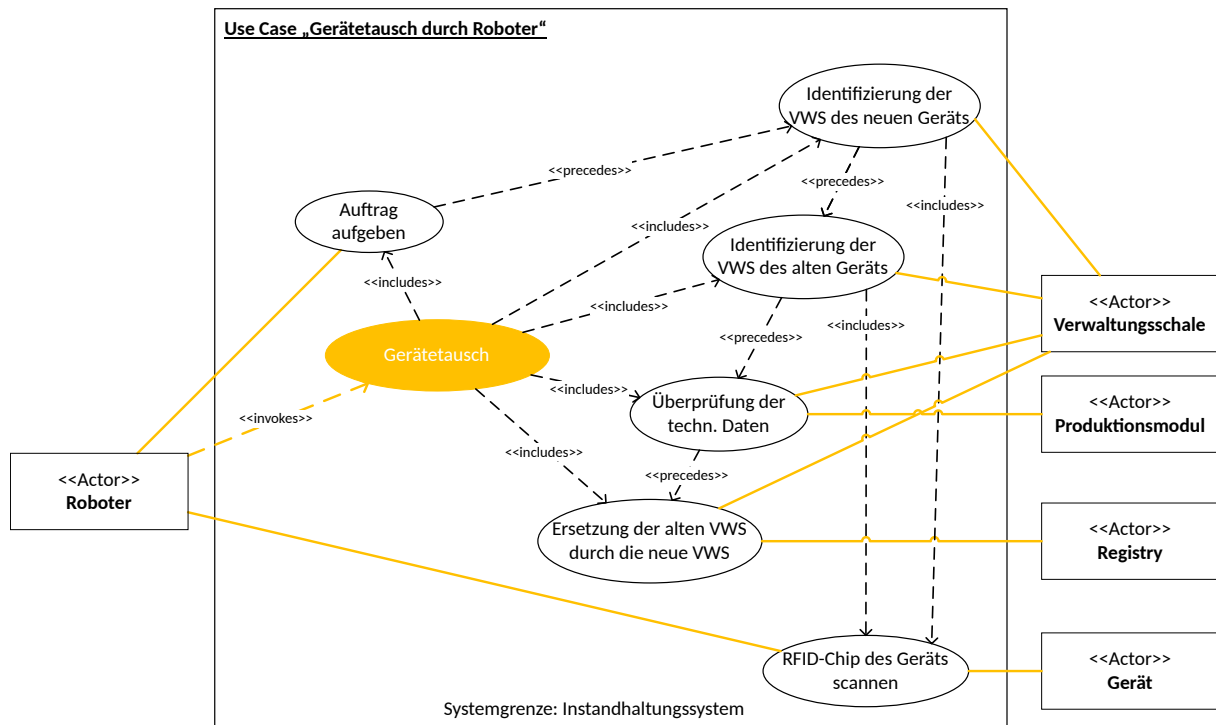


Abbildung 3.13: Use Case Diagramm Szenario 3 - Gerätetausch durch Roboter

Die Interaktion wird von dem Roboter durch eine Auftragsvergabe an das IHS initiiert (1.0). Als Antwort fordert das IHS den Roboter dazu auf, den RFID-Chip der alten Kamera zu scannen (1.1). Der Scan ereignet sich, sobald der Roboter die Kamera in der Kamerawechselstation ablegt. Die aus dem Scan erhaltenen Daten des Assets inklusive der Asset-ID werden an das IHS übermittelt (1.4). Mit der Asset-ID ist das IHS in der Lage, den Endpunkt der entsprechenden VWS bei der Registry zu ermitteln (1.5). Anschließend wiederholt sich das Verfahren bei der Aufnahme der neuen Kamera ((1.6) bis (1.10)).

Liegen dem IHS die Endpunkte beider VWS vor, kann es durch die VWS-API auf die Daten der VWS zugreifen und diese mit den Asset-Daten (z.B. Identifikationsdaten) aus dem RFID-Chip verifizieren (1.11). Diese Aufgabe wurde in den vorherigen beiden Szenarien durch den Instandhalter bzw. das Servicepersonal übernommen. Mit der Verifizierung muss auch die Kompatibilität der beiden Geräte geprüft werden (1.12). Dabei wird das Produktionsmodul von dem IHS kontaktiert und genau wie im Sequenzdiagramm aus Kapitel 3.2 in Abbildung 3.7 die Prüfung durchgeführt. Zu sehen ist in dem Sequenzdiagramm des dritten Szenarios nur die gekürzte Version.

Nach erfolgreicher Verifizierung und Kompatibilitätsprüfung ist die Aktualisierung der Beziehungen durch das IHS realisierbar. Die Interaktion ist im Interaktionsrahmen „UPDATE Beziehung“ zusammengefasst. Anders als in den vorherigen Szenarien, werden die Beziehungen zu anderen VWS erst in dem Interaktionsrahmen „UPDATE Beziehung“ durchgeführt. Dadurch wird eine Zwischenspeicherung der gefundenen Beziehungen umgangen. Der Rest des Interaktionsrahmens fasst die Vorgänge zur Aktualisierung der

Beziehungen übersichtlich zusammen. Die ausführliche Darstellung der Nachrichten (1.14) bis (1.16) ist in den Sequenzdiagrammen aus Abbildung 3.7 und 3.11 enthalten.

Den Abschluss der Aufgabenstellung bildet die Nachricht des IHS an den Roboter, über die erfolgreiche Aktualisierung der VWS-Struktur (1.17). Durch dieses Signal ist es dem Roboter möglich, seine normalen Tätigkeiten wieder aufzunehmen.

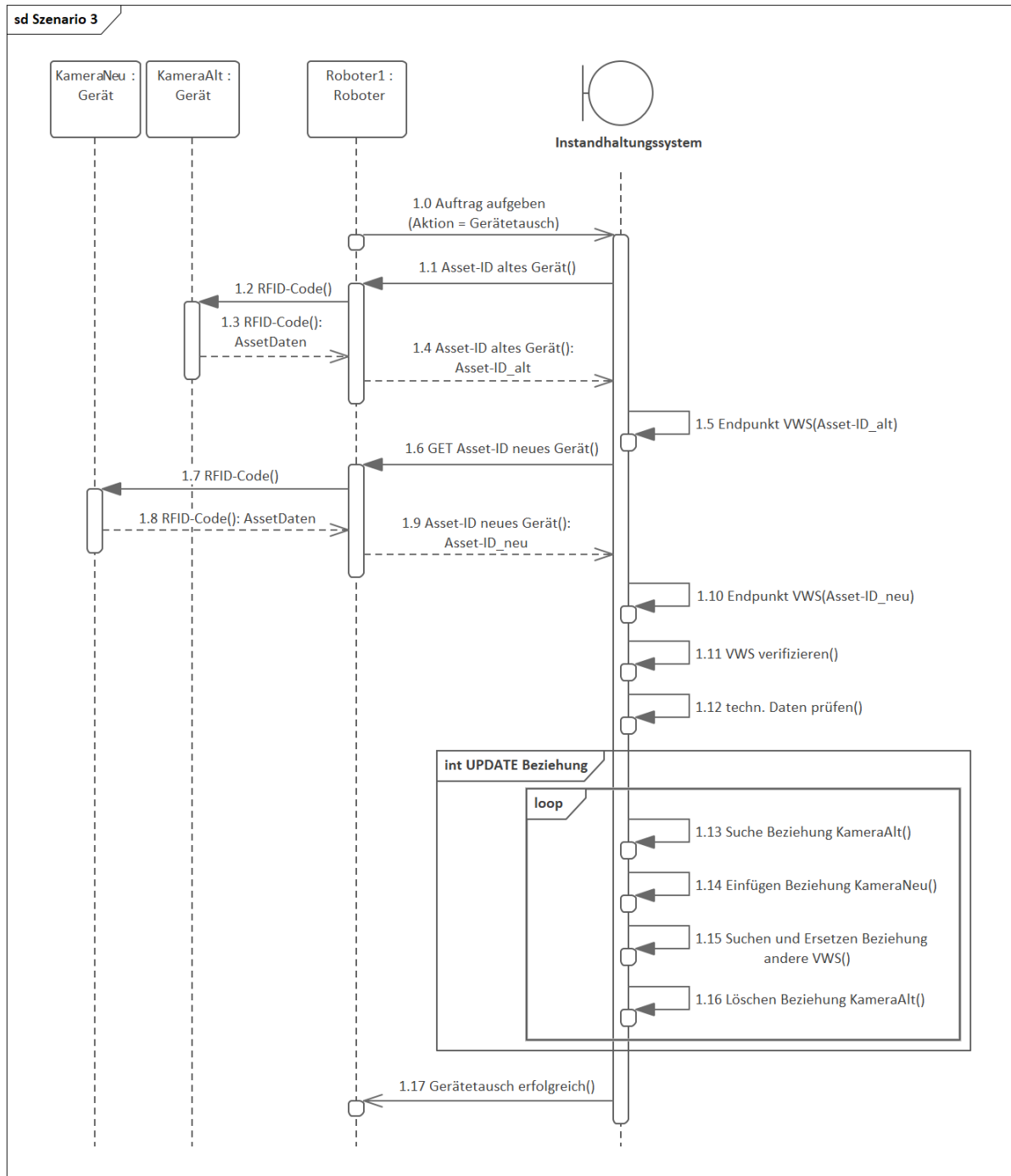


Abbildung 3.14: Sequenzdiagramm Szenario 3 - Gerätetausch durch Roboter

4 Untersuchung technologischer Best-Practice Lösungen

Alle drei Szenarien aus Kapitel 3 haben gezeigt, dass für den Austausch von VWS die Aufgaben des Instandhaltungssystems bzw. des Servicetools im Mittelpunkt stehen. Der Zugriff auf die Softwareapplikationen erfolgt in allen Szenarien nach dem gleichen Schema. Betrachtet werden in diesem Kapitel vor allem die erwähnten Vorgänge zur Identifizierung der VWS und zur Aktualisierung der Beziehungen zwischen den VWS (s. Kapitel 3.2 bis 3.4). Die Umsetzung des Scans der QR-Codes bzw. der RFID-Chips sowie die Zugriffsanfrage am System (s. Kapitel 3.3) und die Prüfung der Kompatibilität zwischen den zu tauschenden Geräten sind nicht Teil dieser Arbeit.

Das Instandhaltungssystem bzw. das Servicetool steht in den Szenarien stellvertretend für eine Softwareapplikation, die die Kommunikation mit der VWS-API übernimmt. Die Interaktionen mit den VWS bilden den Ausgangspunkt für die Untersuchung von geeigneten VWS-Elementen zur Umsetzung des Use Case. Zusammenfassend wird die Ausgangssituation für dieses Kapitel erneut mit den wichtigsten Gegebenheiten beschrieben:

Voraussetzung für die Dokumentationspflege bei einem Gerätetausch ist, dass die VWS des neuen Geräts bereits in einer Registry registriert wurde und in einem Repository gemäß Kapitel 3.1 gespeichert ist.

Zu Beginn des Gerätetausches, werden die QR-Codes bzw. RFID-Chips auf den Typenschildern der auszutauschenden Geräte gescannt. Damit werden einer Softwareapplikation die Asset-IDs der beiden Geräte zur Verfügung gestellt. Sie bilden die Basis, um die zugehörigen VWS der Geräte zu erhalten. Sind der Softwareapplikation nach der Interaktion mit der Registry die entsprechenden VWS bekannt, müssen für die Aktualisierung der gesamten Dokumentation die Beziehungen zu anderen VWS ermittelt und in den betroffenen VWS geändert werden (s. „GET Beziehung“ und „UPDATE Beziehung“ Abbildung 3.7). Mit der erfolgreichen Aktualisierung der VWS-Struktur ist der Gerätetausch abgeschlossen.

Für das bessere Verständnis, wird eine Untergliederung des Kapitels in kleinere Themenkomplexe vorgenommen. Die Gliederung soll strukturiert durch die notwendigen Schritte, zur Erlangung einer Best-Practice Lösung, führen.

Begonnen wird mit der Untersuchung von möglichen Modellierungen der Beziehungen zwischen den VWS. Die am besten geeignete Modellierungsart wird in den darauffolgenden Abschnitten weiterverwendet.

Zur Umsetzung der Beziehungsmodellierung folgt eine Analyse dafür geeigneter VWS-Elemente. Der Abschnitt beschreibt zum einen die vorhandenen Elemente des VWS-Metamodells zur Darstellung einer Beziehung und zum anderen stellt er die Elemente dar, die eine zweckmäßige Strukturierung der Beziehungen ermöglichen.

Darauf folgend werden die Ergebnisse aus den ersten beiden Abschnitten zusammengeführt, um verschiedene Konzepte zur Realisierung der Beziehungsmodellierung zu entwerfen. Ein Beispiel dient zur Veranschaulichung der prinzipiellen Umsetzung.

Der letzte Abschnitt führt die notwendigen Methoden der VWS-API auf, mit der die Softwareapplikation interagieren muss, um die notwendigen Änderungen vorzunehmen. Das Resultat dieses Kapitels bilden technologiespezifische Sequenzdiagramme zur grafischen Beschreibung der Best-Practice Lösung.

Unterstützend bei den Entscheidungsfindungen von mehr oder weniger geeigneten Lösungen, stand die AG1 „Joined Task Force Interaktionen“ der Plattform Industrie 4.0 zur Seite. Sie haben durch die persönlichen Erfahrungen, auch im Zusammenhang mit der Implementierung von VWS, Anregungen zur technischen Umsetzung des Use Case gegeben.

Mit den in diesem Kapitel erstellten Konzepten soll es möglich sein, die Dokumentationspflege bei einem Gerätetausch nicht nur auf produktionsnahe, sondern auch auf andere Bereiche auszuweiten. Das betrifft beispielsweise einen Austausch von Assets und deren VWS in der Betriebs- bzw. Unternehmensleitebene.

4.1 Konzepte für die Modellierung von Beziehungen zwischen Verwaltungsschalen

Die Verwaltungsschalen eines Unternehmens stehen in unterschiedlichen Beziehungen zueinander. Diese Beziehungen können unterschiedlichster Art sein. Einige von der Plattform Industrie 4.0 beschriebenen Beziehungstypen sind im Folgenden aufgeführt:

- „Business relationships
- Constructive relationships
- Functional relationships
- Data relationships
- Communication relationships
- Integration relationships with subrelationships
 - Energy

- Mechanics
- Material flows
- Location relationships
- Temporal relationships
- State relationships
- Asset relationships
- Human-asset relationships“ [27, S. 17]

Nähere Informationen zu den Beziehungstypen sind in dem Bericht [27, S. 17-20] festgehalten. Die Darstellung von konstruktiven Beziehungen ergibt sich beispielsweise durch die Beschreibung der Dekomposition eines Systems. Dadurch ist z.B. die Modellierung eines Systems mit seinen Komponenten in Form einer strukturierten Stückliste denkbar. Anhand dieser modellierten Dekomposition können Rückschlüsse über die bauliche Zusammengehörigkeit und Anordnung von Komponenten gezogen werden. Diese Beziehungen werden in der UML als Aggregation und Komposition bezeichnet. Ein Teilmodell-Template, das die Dekomposition eines Systems erlaubt, ist das sogenannte „Bill of Material“ (BoM)-Teilmodell.

Funktionale Beziehungen, dessen Modellierung in UML durch Assoziationen erfolgt, stellen individuelle Verbindungen zwischen zwei Komponenten her. Ein Beispiel für eine Assoziation liefert die Kamera aus Kapitel 3, die ihre Bilder zur Auswertung an ein Steuerungssystem bzw. eine Softwareapplikation sendet.

Zu Beginn wird ein Blick auf die Modellierung der baulichen Zusammengehörigkeit, d.h. die Dekomposition eines Systems geworfen. Für die Modellierung der Aggregationen und Kompositionen in den VWS existieren verschiedene Möglichkeiten, die in Abbildung 4.1 gegenübergestellt sind.

Den Ausgangspunkt bildet ein einfaches Modell von vier VWS, die durch Aggregationen zu einer Art baulichen Struktur zusammengesetzt sind. Die Aggregation beschreibt eine Ganzes-Teile-Beziehung, wobei die Raute auf das „Ganze“ zeigt. Neben dem Modellbeispiel sind drei Varianten zur Modellierung der Beziehungen in der Abbildung zu sehen. Dabei wird jede VWS mit Elementen des VWS-Metamodells ausgestattet, die die Abbildung einer Beziehung gewährleisten und dessen Beschreibung in Kapitel 4.2 erfolgt.

In der ersten Variante (oben rechts in Abbildung 4.1) enthält eine VWS Beziehungen zu allen ihr untergeordneten VWS. Das heißt, sowohl *VWS 1* als auch *VWS 3* bilden die Beziehung zu *VWS 4* ab, wodurch redundante Informationen entstehen. Hinzukommend wird bei der Betrachtung der Ganzes-Teile-Beziehung (Aggregation) deutlich, dass in dieser

Variante nur das „Ganze“ Kenntnis über seine Teile besitzt. Das einzelne „Teil“ weiß jedoch nicht, zu welchem „Ganzen“ es gehört.

Ein Lösungsansatz zur Vermeidung der redundanten und inkonsistenten Daten ist, in jeder VWS nur die ihm direkt zugeordneten VWS darzustellen. Diese Option ist in Variante 2 zu sehen und wird unter anderem auch in [27, S. 25 f.] erwähnt. Damit wird *VWS 4* nur noch von *VWS 3* referenziert. Die oben erwähnte Schwierigkeit der Ganzes-Teile-Beziehung bleibt jedoch weiterhin bestehen.

Die dritte Variante versucht die Herausforderungen aus Variante eins und zwei komplett zu lösen. Dies gelingt dadurch, dass in der Aggregation auch das „Teil“ Kenntnis über das „Ganze“ hat, zu dem es gehört. Mit dieser Variante wird nicht nur eine Dekomposition optimal modelliert, sondern auch die Darstellung von anderen Beziehungstypen gewährleistet. Das hat bei dem Gerätetausch zur Folge, dass mit einem Blick in eine bestimmte VWS alle mit ihm in Beziehung stehenden VWS bekannt sind.

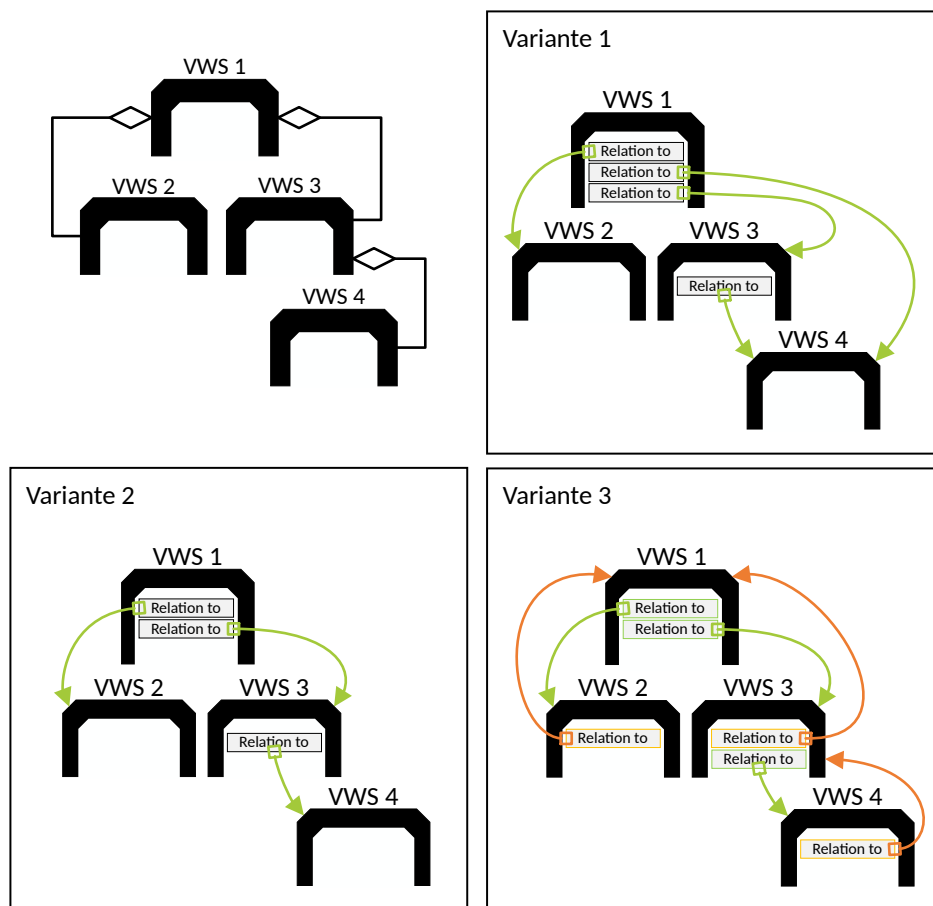


Abbildung 4.1: Möglichkeiten zur Modellierung von Beziehungen zwischen Verwaltungsschalen

Jede der drei genannten und grafisch geschilderten Modellierungen hat ihre Vor- und Nachteile, die bereits kurz beschrieben wurden. Diese sind in Tabelle 4.1 nochmals gegenübergestellt. Es wird ersichtlich, dass die Vorteile der dritten Variante, im Vergleich mit den beiden anderen Varianten, überwiegen. Sie ist sogar vorteilhafter, wenn es um

die Modellierung weiterer Beziehungstypen geht. Aufgrund dessen wird für alle folgenden Abschnitte die Modellierung der Beziehungen entsprechend der dritten Variante aus Abbildung 4.1 als Basis genutzt.

Tabelle 4.1: Vor- und Nachteile der unterschiedlichen Abbildungen einer Hierarchie in Verwaltungsschalen

Art der Beziehungsmodellierung	Vorteile	Nachteile
<p><u>Variante 1</u> Modellierung von Beziehungen zu untergeordneten VWS und mit redundanten Informationen</p>	<ul style="list-style-type: none"> • Die Beziehung zu einer untergeordneten VWS kann sofort über die VWS der obersten Ebene gefunden werden. 	<ul style="list-style-type: none"> • Redundante Informationen, d.h. die Änderung einer Beziehung muss an mehr als zwei Stellen vorgenommen werden. • Erhöhter Aufwand zur Implementierung, durch die Redundanz der Informationen. • Die Kenntnis über die Beziehungen existiert nur von übergeordneten zu untergeordneten VWS.
<p><u>Variante 2</u> Modellierung von Beziehungen zu untergeordneten VWS und ohne redundante Informationen</p>	<ul style="list-style-type: none"> • Keine redundanten Informationen, d.h. die Änderung einer Beziehung ist nur an einer Stelle notwendig. • Geringerer Aufwand zur Implementierung als in Variante 1. 	<ul style="list-style-type: none"> • Als Worst-Case-Szenario muss jede VWS nach der gesuchten VWS bis in die unterste Ebene durchlaufen werden. • Die Kenntnis über die Beziehungen existiert nur von übergeordneten zu untergeordneten VWS.

Art der Beziehungsmodellierung	Vorteile	Nachteile
<u>Variante 3</u> Modellierung von Beziehungen zu über- und untergeordneten VWS und ohne redundante Informationen	<ul style="list-style-type: none"> • Keine redundanten Informationen, d.h. die Änderung einer Beziehung ist nur in den beiden betroffenen VWS notwendig. • Da alle Beziehungen einer VWS zentral in dieser zusammengefasst sind, ist eine Suche nach zusätzlichen Beziehungen nicht notwendig. • Die Kenntnis über die Beziehungen existiert sowohl zu übergeordneten als auch zu untergeordneten VWS. 	<ul style="list-style-type: none"> • Erhöhter Aufwand, da Beziehungen zwischen zwei VWS immer in beiden VWS implementiert werden müssen.

4.2 Elemente des Verwaltungsschalen-Metamodells zur Modellierung von Beziehungen

Für die Abbildung von Beziehungen zwischen VWS stellt das VWS-Metamodell verschiedene Elemente zur Verfügung, dessen nähere Beschreibung in diesem Abschnitt erfolgt. Dazu werden die UML-Klassen aus der Spezifikation [26], die die VWS-Elemente beschreiben, herangezogen. Ausgangspunkt für alle folgenden Ergebnisse ist die Verwendung des Modellierungsprinzips nach Variante drei aus Kapitel 4.1, Abbildung 4.1.

Des Weiteren werden die VWS-Elemente untersucht, die eine Strukturierung der Beziehungen ermöglichen. Dies erlaubt die sinnvolle Trennung von Beziehungen, sowie einen definierten Zugang zu Daten für verschiedene Benutzerrollen.

Hinweis: Für den restlichen Teil dieser Arbeit werden, aufgrund der direkten Verwendung des VWS-Metamodells, alle Begriffe wie sie in der Spezifikation [26] und [28] definiert sind verwendet. Das heißt Teilmodelle werden zu Submodels, Verwaltungsschalen (VWS) zu Asset Administration Shells (AAS) usw.

Entity

Das Submodel-Element *Entity* (s. Abbildung 4.2) dient zur Modellierung einer Entität (Assets). Dabei gibt es zwei Typen von Entities. Zum einen eine *co-managed Entity*, die ein Asset ohne eigene AAS modelliert. Zum anderen eine *self-managed Entity*, die ein Asset mit einer eigenen AAS modelliert. Die Referenzierung einer *self-managed Entity* kann entweder über eine globale oder eine externe Asset-ID geschehen. Eine Anreicherung der Entity mit weiteren Informationen ist durch zusätzliche Submodel-Elements erlaubt [26, S. 75 f.].

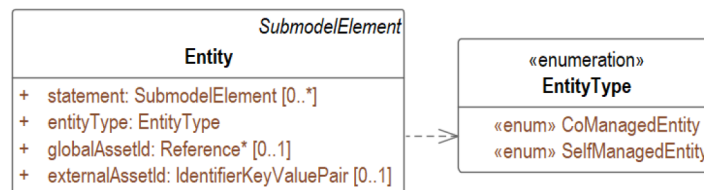


Abbildung 4.2: Spezifikation des AAS-Element - Entity [26, S. 75]

Enthält ein Submodel eine *Entity*, so wird eine Beziehung zu dem dazugehörigen Asset durch Angabe der Asset-ID modelliert (ebd.). Ein Beispiel dafür ist in der Publikation zur AAS-Referenzmodellierung (s. [29, S. 30–34]) aufgeführt. Das Beispiel zeigt ein Entity-Relationship-Modell, das in einem BoM-Submodel gespeichert wird.

Reference-Element

Ein *Reference-Element* (s. Abbildung 4.3) ist ein Datenelement und beinhaltet als Wert, wie der Name es vermuten lässt, eine logische Referenz auf ein referenzierbares Element. Zu den referenzierbaren Elementen gehören unter anderem Propertyts, Files, Entities, Assets, AAS und sogar Reference-Elements selbst. Die Referenz kann dabei auf genau ein Element derselben AAS oder auf das einer externen AAS zeigen. Die Referenz auf einen Endpunkt wird in dem *value* des *Reference-Elements* gespeichert [26, S. 81 f.].

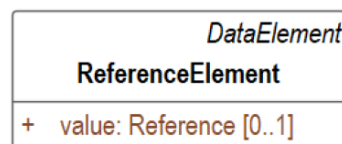


Abbildung 4.3: Spezifikation des AAS-Element - Reference Element [26, S. 81]

Relationship-Element

Das *Relationship-Element* (s. Abbildung 4.4) dient der Modellierung einer Beziehung zwischen zwei referenzierbaren Elementen. Dazu enthält es zwei Attribute, die die Endpunkte der Beziehung beschreiben. Das erste Attribut nimmt die Rolle des Subjekts ein, während das zweite Attribut die Rolle des Objekts einnimmt. Die Endpunkte sind mit *first* für das Subjekt und mit *second* für das Objekt gekennzeichnet. Ein Endpunkt des *Relationship-Elements* kann auf ein referenzierbares Element des AAS-Metamodells zeigen [26, S. 82].

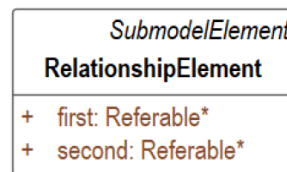


Abbildung 4.4: Spezifikation des AAS-Element - Relationship Element [26, S. 82]

Submodel und Submodel Element Collections

Ein *Submodel* dient als Container für AAS-Elemente und repräsentiert einen besonderen Aspekt der AAS. Die Elemente in einem Submodel können wiederum in Containern, sogenannten *Submodel Element Collections*, eingeordnet werden [26].

Es ergeben sich verschiedene Varianten bezüglich der Modellierung von Beziehungen durch Submodels und Submodel Element Collections. Ein Beispiel ist in Abbildung 4.5 zu sehen, das die AAS des Assets „Asset123“ und die darin enthaltenen Submodels darstellt. Jedes dieser *Submodels* kann *Submodel Element Collections* zur Strukturierung der unterschiedlichen Informationen enthalten. Durch die zu den *Submodels* und *Submodel Element Collections* dazugehörigen semanticIds ist die semantische Interpretation gewährleistet.

View

Views ermöglichen die Filterung der in einer AAS enthaltenen Elemente. Voraussetzung für die Aufnahme eines Elements in einen *View* ist, dass dieses referenzierbar ist. Die Elemente im *View* werden im weiteren als *Referable* bezeichnet. Die semantische Beschreibung der Inhalte erfolgt über die *semanticId* [26, S. 86]. Die Spezifikation des *View* ist in Abbildung 4.7 zu sehen.

Durch die Filterung können nicht nur Informationen schnell gefunden, sondern auch unterschiedlichen Benutzerrollen zugänglich gemacht werden. Beispielsweise ist es möglich, das Modell aus dem Abschnitt der *Submodels* und *Submodel Element Collections* durch

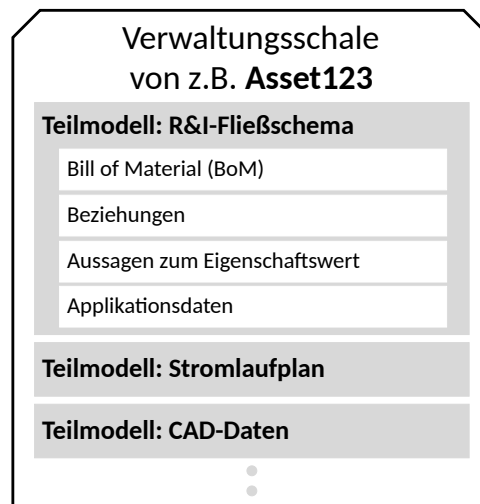


Abbildung 4.5: Übersicht über die Submodels und Submodel Element Collections einer zusammengesetzten Komponente (in Anlehnung an [27, S. 27])

Views zu erweitern. Dabei enthält z.B. ein *View* mehrere Referables, die die einzelnen Submodel Element Collections referenzieren. Für das konkrete Beispiel aus Abbildung 4.6 bedeutet das, dass der View „Bill of Material“ alle Submodel Element Collections mit dem Namen „Bill of Material“ aus den unterschiedlichen Teilmodellen der verschiedenen Gewerke herausfiltert.

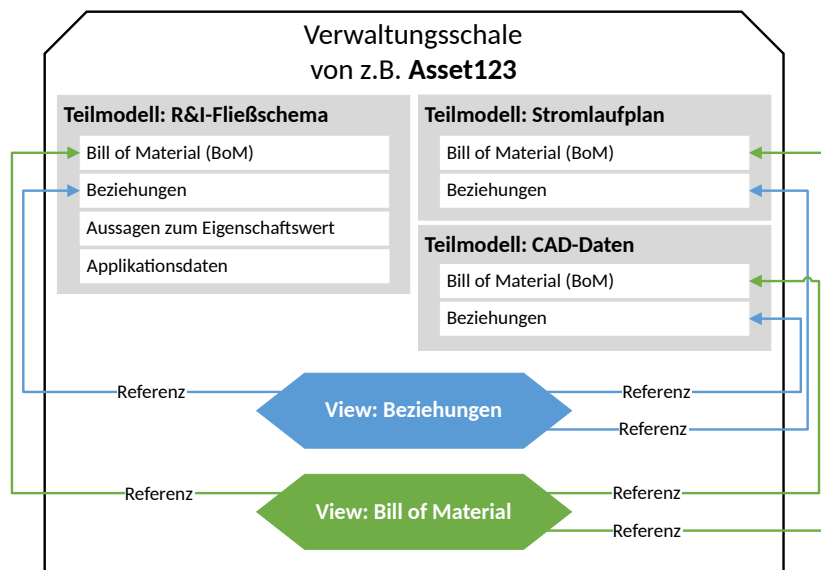


Abbildung 4.6: Ausstattung einer AAS mit Views

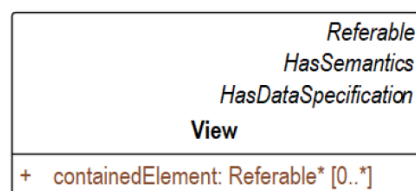


Abbildung 4.7: Spezifikation des AAS-Element - View [26, S. 86]

4.3 Konzepte für die Realisierung der Beziehungsmodellierung in Verwaltungsschalen

Die Elemente des AAS-Metamodells aus dem vorausgegangenen Kapitel werden im Folgenden für die Entwicklung der verschiedenen Konzepte, zur Realisierung der Beziehungsmodellierung, genutzt. Eine anschließende Analyse der entwickelten Konzepte auf ihre Vor- und Nachteile gibt Aufschluss über den besten Lösungsansatz. Teile der in diesem Kapitel entwickelten Konzepte sind von [27] abgeleitet, da darin anschaulich die Beziehungen zwischen Komponenten beschrieben sind.

Die Idee für das erste Konzept stammt aus [27], woraufhin die BoM und die Beziehungen als Submodel Element Collections in einem Submodel gespeichert werden. Die BoM enthält dabei alle Entities, die in direkter Verbindung mit dem Asset der AAS stehen (s. Kapitel 4.1). Diese können sowohl *co-managed* als auch *self-managed* sein (s. Kapitel 4.2). Ausgeschlossen von der BoM ist die Abbildung von Beziehungen durch zusätzliche AAS-Elemente, da hierfür die Submodel Element Collection „Beziehungen“ zur Verfügung steht [27, S. 27–31].

Die Beziehungen in den Submodel Element Collections werden durch Relationship-Elements und Reference-Elements dargestellt und mit wichtigen Informationen zu dem Beziehungstyp und -kontext angereichert. Einige der Beziehungstypen sind aus Kapitel 4.1 bereits bekannt. Eine Auflistung von allen existierenden Beziehungskontexten ist nicht möglich, da diese individuell im Industriesektor entstehen. Zu diesen Kontexten zählen beispielsweise die Anlagensicherheit, Anlagenstruktur, der Transport, Diagnoseaufgaben und viele mehr. Da die zusätzlichen Informationen zu dem Beziehungstyp und -kontext für jedes folgende vorgestellte Konzept von Relevanz sind, wird auf deren Einbindung zum Ende dieses Abschnitts näher eingegangen.

Veranschaulicht wird das erste Konzept anhand eines Beispiels in Abbildung 4.8. Rot eingerahmt und mit „Variante A“ bezeichnet, ist die Umsetzung der Submodel Element Collection für die Beziehungen zu sehen. Rechts daneben in derselben Abbildung sind zwei weitere Konzepte (Variante B und C) rot eingerahmt hervorgehoben.

Die Konzepte aus Variante B und C strukturieren die Beziehungen anhand von Gruppierungen (zusätzliche Submodel Element Collections). Die Gruppierungen der Beziehungen, entweder nach Beziehungstyp (Variante B) oder Beziehungskontext (Variante C), ermöglichen vor allem dem Anwender die bessere Auffindbarkeit von Informationen. Bei der Suche nach einer Beziehung eines bestimmten Beziehungstyps ist es jedoch notwendig, jedes Submodel zu öffnen, um in der jeweiligen Submodel Element Collection die gesuchte Beziehung zu finden.

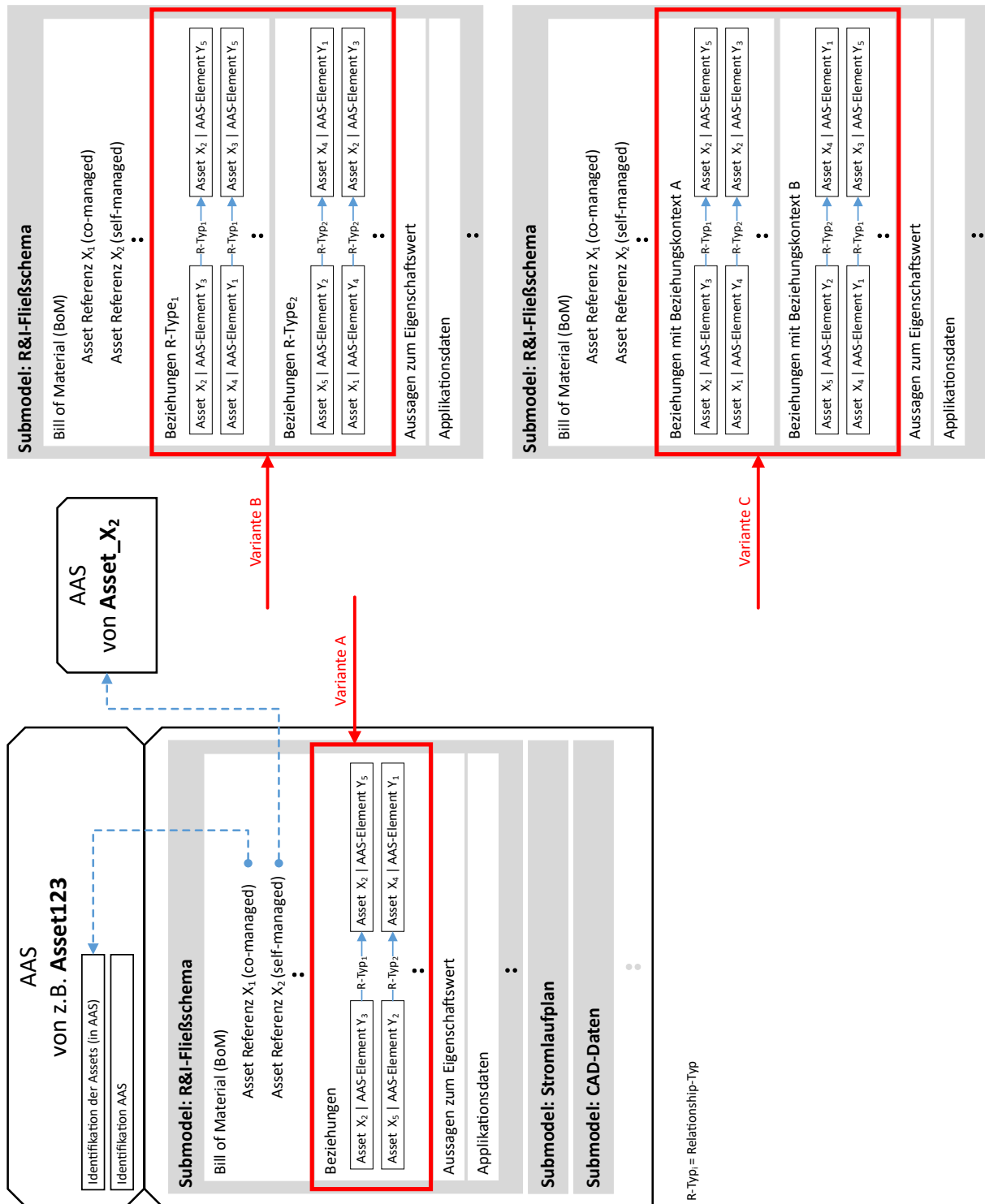


Abbildung 4.8: Modellierung von Beziehungen in der AAS mit den gegebenen AAS-Elementen (Teil 1) (in Anlehnung an [27, S. 27–31])

Zur Verringerung dieses Mehraufwands, dienen die drei folgenden Konzepte aus Abbildung 4.9. Sie beziehen sich auf die zusätzliche Verwendung von Views. Die Implementierung der Beziehungen in den Submodels erfolgt in diesem Beispiel nach dem Schema aus Variante A (Abbildung 4.8). Genauso kann die Umsetzung der Beziehungen aber auch nach Variante B oder C stattfinden.

Das vierte Konzept aus Abbildung 4.9 (Variante D) enthält einen View, der alle Beziehungen aus den unterschiedlichen Submodels referenziert. Das hat sowohl für den Anwender als auch für eine Softwareapplikation, die beide Zugriff auf die AAS haben, einen Vorteil. Durch den einzelnen View werden alle Beziehungen an einem zentralen Punkt referenziert, ohne die Kenntnis über jedes einzelne Submodel haben zu müssen.

Rechts in der Abbildung, neben Variante D, sind zwei weitere Varianten (Variante E und F) verbildlicht, die ähnlich den Varianten B und C eine Gruppierung der Beziehungen vornehmen. Durch Variante E beispielsweise werden Beziehungen eines Beziehungstyps durch einen View zusammengefasst, während in Variante F die Zuordnung der Beziehungen zu den Beziehungskontexten stattfindet.

Jedes der vorgestellten Konzepte, Variante A bis C, bildet für sich einen realisierbaren Lösungsansatz ab, der durch Kombination mit den Konzepten, Variante D bis F, zusätzliche Lösungsansätze ermöglicht. Weiterhin können zu den dargestellten Varianten aus Abbildung 4.8 und 4.9 durchaus weitere Lösungsansätze existieren, die in dieser Arbeit nicht aufgeführt sind.

Die sechs vorgestellten Konzepte (Variante A bis F) weisen alle ihre Vor- und Nachteile in Bezug auf die Implementierung und den Aufwand bei der Suche nach einem bestimmten Element auf. Eine Gegenüberstellung der einzelnen Konzepte soll die Herausarbeitung des besten Lösungsansatzes vereinfachen. Dafür wird eine Tabelle (s. Tabelle 4.2) entwickelt, die sowohl die Merkmale zur Umsetzung des jeweiligen Konzepts als auch dessen Vor- und Nachteile auflistet.

Eine Gemeinsamkeit aller sechs Konzepte ist die Voraussetzung von spezifizierten Beziehungstypen und -kontexten. Außenstehende Softwareapplikationen müssen dazu in der Lage sein, diese beiden Eigenschaften zu interpretieren. Im Fall des Beziehungstyps ist die Definition eines vereinheitlichten Vokabulars denkbar. Die Vielfalt an Beziehungskontexten kann jedoch nicht ohne größeren Aufwand in einzelne Begriffe gefasst werden, weshalb auch die Varianten C und F als Lösungsansatz weniger interessant sind.

Aus der Gegenüberstellung aller sechs Varianten lässt sich zusammenfassend sagen, dass für diesen Use Case eine Kombination aus Variante B und E die bestmögliche Lösung darstellt. Durch die Gruppierung der Beziehungen in Submodel Element Collections (Variante B) wird die Speicherbelegung reduziert. In Kombination mit der Verwendung der Views aus Variante E, erfolgt gleichzeitig die Senkung des Rechenaufwands bei der Suche nach einer vorgegebenen Beziehung. Ein Nachteil des Views ist, dass für diesen kein Interface in der AAS-API für den Zugriff auf dessen Elemente existiert.

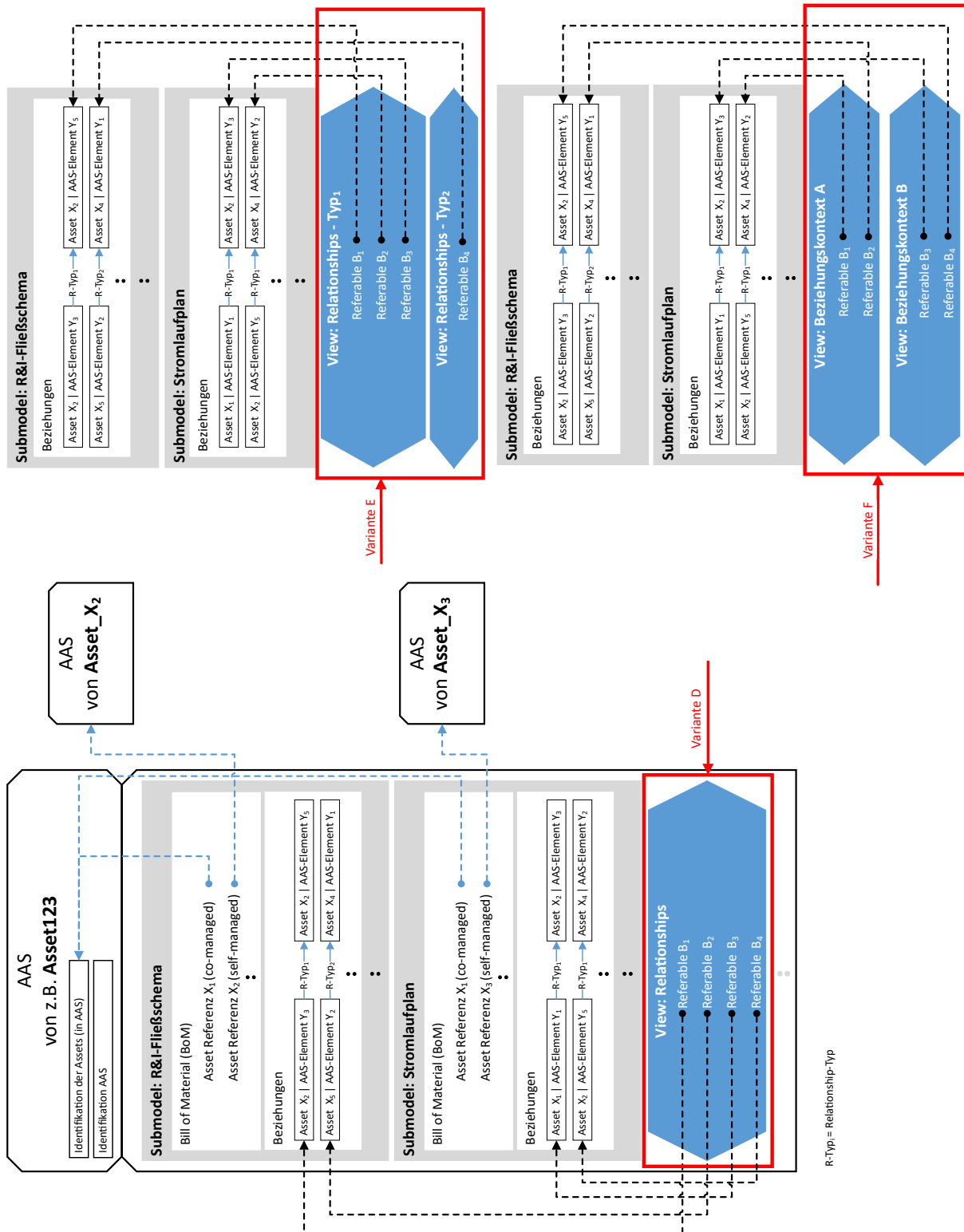


Abbildung 4.9: Modellierung von Beziehungen in der AAS mit den gegebenen AAS-Elementen (Teil 2) (in Anlehnung an [27, S. 27–31])

Tabelle 4.2: Vor- und Nachteile der Konzepte zur Umsetzung der Beziehungsmodellierung

Konzept (Variante)	Merkmale	Vorteile	Nachteile
Variante A	<ol style="list-style-type: none"> 1. Die BoM besteht nur aus Entities zur Darstellung von Assets. Self-managed Entities verwalten zusätzlich die Asset-ID. 2. Alle Beziehungen eines Submodels sind in einer einzigen Submodel Element Collection zusammengefasst und werden durch Relationship- bzw. Reference-Elements modelliert. 3. Die Submodel Element Collection erhält eine semanticId die aussagt, dass darin Beziehungen verwaltet werden. 4. Reference- und Relationship-Elements erhalten Qualifier zur Spezifizierung des Beziehungstyps und -kontextes. 	<ul style="list-style-type: none"> • Es entsteht eine Grundstruktur, in der jedem Submodel seine Beziehungen zugeordnet werden. • Die Zuständigkeiten, bei der Entwicklung der Submodels, überschneiden sich nicht. 	<ul style="list-style-type: none"> • Durch die fehlende Gruppierung nach Beziehungstyp bzw. -kontext muss jedes Reference- bzw. Relationship-Element zwei Qualifier für die Beschreibung beider Eigenschaften erhalten. Das führt zu einer erhöhten Speicherbelegung. • Es existiert kein Überblick über alle in der AAS vorhandenen Beziehungen. Jedes Submodel muss bei der Suche nach einer spezifischen Beziehung durchlaufen werden.
Variante B	<ol style="list-style-type: none"> 1. siehe Punkt 1 aus Variante A 2. Die Beziehungen eines Submodels sind in mehreren Submodel Element Collections nach Beziehungstyp gruppiert und werden durch Relationship- bzw. Reference-Elements modelliert. 3. Die Submodel Element Collections erhalten eine semanticId die aussagt, dass darin Beziehungen verwaltet werden. Zusätzlich wird jeder Submodel Element Collection ein Qualifier zur Definition des Beziehungstyps angefügt. 4. Reference- und Relationship-Elements erhalten einen Qualifier zur Spezifizierung des Beziehungskontextes. 	<ul style="list-style-type: none"> • siehe Vorteile aus Variante A • Senkung der Speicherbelegung, da der Beziehungstyp nur einmalig im Qualifier der Submodel Element Collections für alle darin gespeicherten Beziehungen angegeben werden muss. • Der Beziehungstyp ist einfacher zu standardisieren als der Beziehungskontext. 	<ul style="list-style-type: none"> • Kein Überblick über alle in der AAS existierenden Beziehungen. Jedes Submodel muss bei der Suche nach einer spezifischen Beziehung durchlaufen werden.

Konzept (Variante)	Merkmale	Vorteile	Nachteile
Variante C	<ol style="list-style-type: none"> 1. siehe Punkt 1 aus Variante A 2. Die Beziehungen eines Submodels sind in mehreren Submodel Element Collections nach Beziehungskontext gruppiert und werden durch Relationship- bzw. Reference-Elements modelliert. 3. Die Submodel Element Collections erhalten eine semanticId die aussagt, dass darin Beziehungen verwaltet werden. Zusätzlich wird jeder Submodel Element Collection ein Qualifier zur Definition des Beziehungskontextes angefügt. 4. Reference- und Relationship-Elements erhalten einen Qualifier zur Spezifizierung des Beziehungstyps. 	<ul style="list-style-type: none"> • siehe Vorteile aus Variante A • Senkung der Speicherbelegung, da der Beziehungskontext nur einmalig im Qualifier der Submodel Element Collections für alle darin gespeicherten Beziehungen angegeben werden muss. 	<ul style="list-style-type: none"> • Durch die Vielfalt an Beziehungskontexten werden dementsprechend viele Submodel Element Collections benötigt. • Kein Überblick über alle in der AAS existierenden Beziehungen. Jedes Submodel muss bei der Suche nach einer spezifischen Beziehung durchlaufen werden.
Variante D	<ol style="list-style-type: none"> 1. Alle Beziehungen der Submodels werden durch einen einzigen View referenziert. 2. Die semanticId des View beschreibt, dass sich in dem View nur Beziehungen befinden. Das heißt Referenzen auf Relationship- und Reference-Elements. 	<ul style="list-style-type: none"> • Der View bietet einen Überblick über alle in einer AAS vorkommenden Beziehungen. Damit sinkt der Rechenaufwand bei der Suche nach einer Beziehung, da nicht mehr jedes Submodel explizit durchlaufen werden muss. 	<ul style="list-style-type: none"> • Die Referenz im View lässt keine Rückschlüsse auf den Inhalt des Referables zu. • Während der Suche nach Beziehungen eines vorgegebenen Beziehungstyps muss der Inhalt des Elements, auf den das Referable verweist, näher betrachtet werden. Ein Vergleich zwischen dem Beziehungstyp des Elements und dem vorgegebenen Beziehungstyp ist unumgänglich. Dafür muss der Beziehungstyp bzw. -kontext klar definiert sein

Konzept (Variante)	Merkmale	Vorteile	Nachteile
Variante E	<ol style="list-style-type: none"> 1. Die Beziehungen der Submodels werden durch mehrere Views nach Beziehungstypen gruppiert. 2. Die semanticId des View beschreibt, dass sich in dem View nur Referables eines bestimmten Beziehungstyps befinden. 	<ul style="list-style-type: none"> • Die Views bieten einen gruppierten Überblick über alle in einer AAS vorkommenden Beziehungen. • Durch die Gruppierung nach dem Beziehungstyp sinkt der Rechenaufwand im Vergleich zu Variante D. 	<ul style="list-style-type: none"> • Die Referenz im View lässt keine Rückschlüsse auf den Inhalt des Referables zu. • Für die Suche nach einer Beziehung, müssen für die externe Softwareapplikation die Beziehungstypen jeweils über eine eigene semanticId verfügen und klar definiert sein.
Variante F	<ol style="list-style-type: none"> 1. Die Beziehungen der Submodels werden durch mehrere Views nach Beziehungskontexten gruppiert. 2. Die semanticId des View beschreibt, dass sich in dem View nur Referables mit einem bestimmten Beziehungskontext befinden. 	<ul style="list-style-type: none"> • Die Views bieten einen gruppierten Überblick über alle in einer AAS vorkommenden Beziehungen. • Durch die Gruppierung nach dem Beziehungskontext sinkt der Rechenaufwand im Vergleich zu Variante D. 	<ul style="list-style-type: none"> • Die Referenz im View lässt keine Rückschlüsse auf den Inhalt des Referables zu. • Für die Suche nach einer Beziehung, müssen für die externe Softwareapplikation die Beziehungskontexte jeweils über eine eigene semanticId verfügen und klar definiert sein. • Durch die Vielfalt an Beziehungskontexten, werden dementsprechend viele Views und semanticIds benötigt.

Mit den gewonnenen Erkenntnissen aus Tabelle 4.2 wird die prinzipielle Realisierung der Variante B in Kombination mit Variante E anhand eines kleinen Beispiels betrachtet. Es zeigt eine Möglichkeit für die Realisierung der BoM und der Beziehungen, sowie den Views mit den aus Kapitel 4.2 ermittelten AAS-Elementen.

Das Beispiel enthält drei fiktive AAS (*Asset_X*, *Asset_Y*, *Asset_Z*), die über Beziehungen miteinander verbunden und in Abbildung 4.10 veranschaulicht sind. Erkennbar ist im linken Bereich der Abbildung die Beziehungsmodellierung nach Kapitel 4.1. Im rechten Bereich der Abbildung ist eine Möglichkeit zur prinzipiellen Umsetzung der Beziehungsmodellierung in einem Teilmodell, anhand der in diesem Kapitel vorgestellten Konzepte, zu sehen.

Zu Anfang erfolgt die Erstellung der BoM, deren Inhalt durch co-managed bzw. self-managed Entities zur Referenzierung von Assets realisiert wird. In diesem Beispiel enthält die AAS des *Asset_Y* die self-managed Entity *Asset_X*, die die Asset-ID des *Asset_X* verwaltet.

Nach der Implementierung der BoM folgt die Umsetzung der Beziehungen zwischen den Elementen in derselben AAS sowie zu Elementen externer AAS. Hierfür werden Relationship-Elements und Reference-Elements genutzt. Veranschaulicht wird dies in dem Beispiel aus Abbildung 4.10 anhand der Beziehungen *Asset_Y_Fixiert_Asset_X* und *Asset_Z_Enthält_Asset_Y*, die beide vom Beziehungstyp *konstruktiv* sind. Beide Elemente werden in einer Submodel Element Collection vereint, die durch ihren Qualifier den Beziehungstyp der verwalteten Elemente vorgibt. Die Beziehungskontexte, die wiederum durch einen Qualifier spezifiziert werden, können sich jedoch in den Relationship- und Reference-Elements unterscheiden. In dem angegebenen Beispiel beschreiben beide Beziehungen die Anlagenstruktur.

Das Relationship-Element wird immer dann in eine AAS eingebunden, wenn die aktuelle AAS die Rolle des Subjekts in der Beziehung übernimmt (in Abb. 4.10 mit *first* bezeichnet). Das ist der Fall, wenn von der AAS bzw. dem Asset aktiv Aktionen ausgelöst werden. Die AAS bzw. das Asset, das auf die Aktion reagiert, wird als Objekt bezeichnet (in Abb. 4.10 mit *second* bezeichnet). Damit das Objekt von der Beziehung zwischen ihm und dem Subjekt Kenntnis hat (s. Kap. 4.1), muss die AAS des Objekts eine Referenz auf diese Beziehung erhalten. Das wird durch ein Reference-Element realisiert und ist in Abbildung 4.10 für die Beziehung *Asset_Z_Enthält_Asset_Y*, mit *Asset_Y* als Objekt und *Asset_Z* als Subjekt, zu sehen.

Die *semanticId* des Reference-Elements unterscheidet sich von der des Relationship-Elements. Grund dafür ist, dass durch die *semanticId* der Submodel Element Collection bereits festgelegt ist, dass sich darin nur Elemente zur Modellierung von Beziehungen befinden. Diese Beziehungen können sich jedoch nach Beziehungen (Relationship-Elements) und referenzierten Beziehungen (Reference-Elements) unterscheiden.

Zuletzt werden in der AAS die Views realisiert, die Referables zur Referenzierung aller Beziehungselemente besitzen. Da der View eines der Elemente im AAS-Metamodell ist, dem keine Qualifier hinzugefügt werden können, müssen andere Möglichkeiten zur Spezifizierung des Beziehungstyps betrachtet werden. In dem Beispiel bedeutet dies, dass der View eine spezielle semanticId zur Spezifizierung des Beziehungstyps erhält, in diesem Fall *BeziehungsrubrikKonstruktiv*. Die darin enthaltenen Referables zeigen auf die zwei AAS-Elemente im Submodel *Anlagenstruktur* unter der Submodel Element Collection *Beziehungen vom Beziehungstyp: konstruktiv*.

Hinweis: Qualifier werden zur genaueren Spezifikation eines AAS-Elements genutzt und gehen über die semanticId, zur semantischen Beschreibung des AAS-Elements, hinaus. Jeder Qualifier verfügt über eine eigene semanticId, einen Wertetyp sowie dessen zugehörigen Wert und einer valueId. Vielen AAS-Elementen können Qualifier hinzugefügt werden, dessen Menge durch den verfügbaren Speicherplatz einer VWS begrenzt ist. Ausgenommen davon ist der View, der nur über eine semanticId oder eine Datenspezifikation (HasDataSpecification) verfügt [26, S. 62 f.].

In diesem Use Case dient der Qualifier zur genaueren Spezifikation des Beziehungstyps und -kontextes für die einzelnen AAS-Elemente (Relationship, Reference, Submodel Element Collection etc.). Die semantische Beschreibung der in dieser Arbeit vorkommenden Spezifizierungen muss jedoch durch ein standardisiertes Vokabular vorgegeben werden. Erst damit ist die interoperable Nutzung der Konzepte möglich. Die Entwicklung dieses Vokabulars ist jedoch nicht Teil dieser Arbeit, weshalb im Folgenden nur abstrakte semanticIds und abstrakte Qualifier verwendet werden.

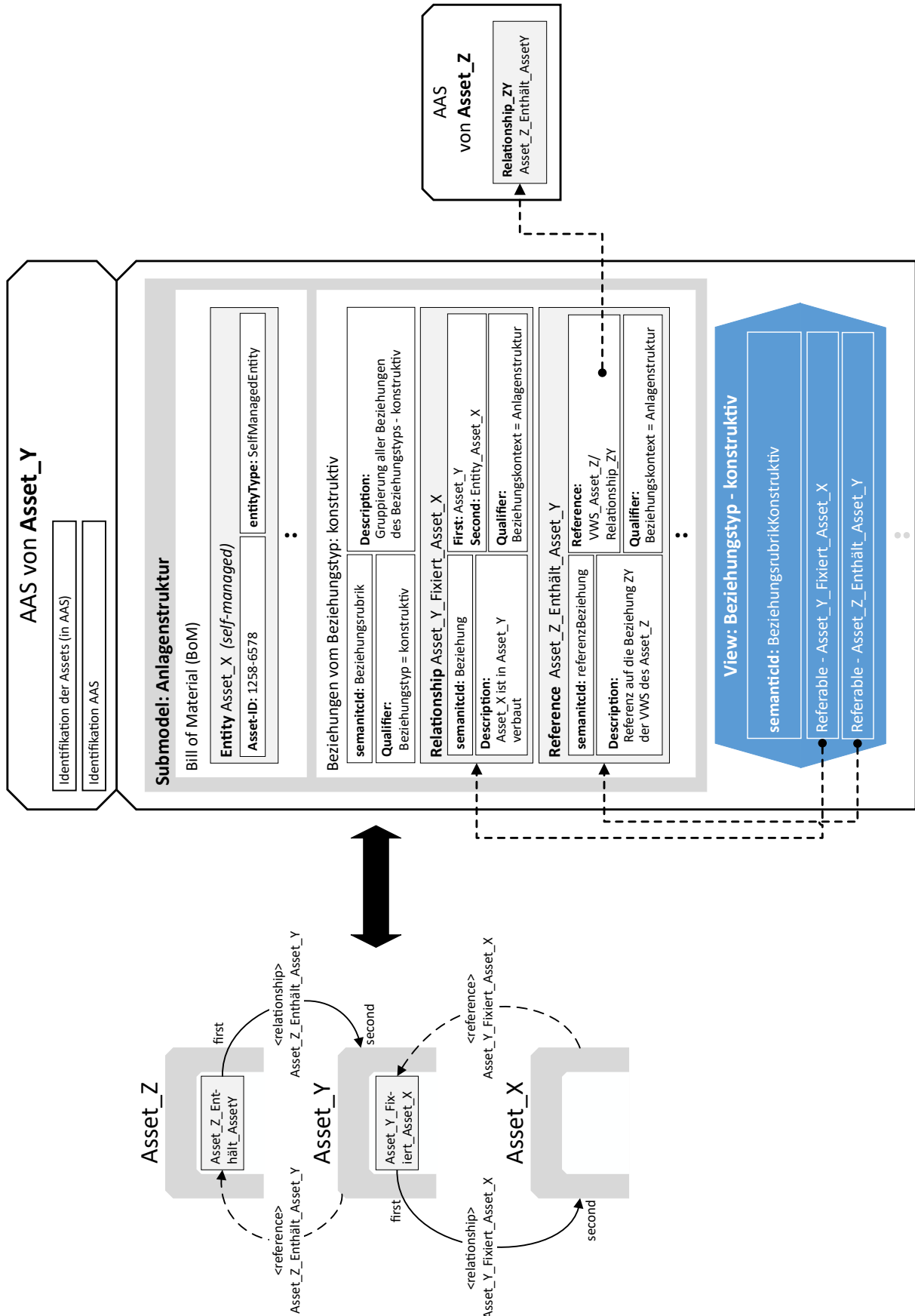


Abbildung 4.10: Beispiel für die prinzipielle Umsetzung der Beziehungsmodellierung

4.4 Datenaustausch über die Schnittstelle der Verwaltungsschale

Die softwaretechnische Implementierung, um eine erfolgreiche Dokumentationspflege bei einem Gerätetausch durchzuführen, folgt nicht einem einzigen vorgeschriebenen Weg. Verschiedene Systemarchitekturen bieten in Kombination mit der AAS-API eine Bandbreite an Möglichkeiten für die Umsetzung des Use Case an. Die in diesem Kapitel entwickelte Lösung basiert auf folgendem Ausgangspunkt:

- Die Verwendung der beschriebenen Systemarchitektur aus Kapitel 3.1.
- Alle AAS, die in dem Use Case eine Rolle spielen, sind vor Beginn des Gerätetausches bereits an einer Registry angemeldet.
- Die Asset-ID des alten und neuen Geräts stehen durch den Scan der Typenschilder bereits zur Verfügung.
- Es gibt kein zusätzliches Repository zur Speicherung einzelner Submodels.
- Die AAS sind entsprechend der aus Kapitel 4.3 entwickelten Best-Practice Lösung designt.
- Übergreifende Planungsunterlagen, wie z.B strukturierte Stücklisten, Stromlaufpläne oder R&I-Fließschemata, die Beziehungen zwischen Assets und damit ihren AAS abbilden, müssen mit den vorgegebenen Datenmodellen aus Kapitel 4.3 in Submodels bereits realisiert sein. Eine automatisierte Umwandlung der meist im pdf-Format vorliegenden Dokumente in Submodels ist nicht Teil dieser Arbeit.

Von diesem Ausgangspunkt aus, werden alle Möglichkeiten beleuchtet, um die Interaktionen aus Kapitel 3 unter Verwendung der AAS-API auf einem technologiespezifischen Level prinzipiell zu verwirklichen. Die Wahl der Technologie, zur Implementierung der AAS-API, fällt in dieser Arbeit auf HTTP/REST. Alle Operationen der AAS-API werden in den Sequenzdiagrammen dieses Kapitels dementsprechend syntaktisch an den HTTP/REST-Konventionen ausgerichtet.

Für die Analyse der notwendigen Operationen, im Zusammenhang mit dem aus Kapitel 4.3 für diesen Use Case erarbeiteten AAS-Aufbau, wird das Beispiel aus Kapitel 4.3 fortgesetzt. Das Beispiel zeigt zu Beginn leicht verständlich die Zusammenhänge zwischen den einzelnen AAS-Elementen, um eine Aktualisierung der Dokumentation erfolgreich durchzuführen. Der Austausch der AAS erfolgt im Folgenden für das Asset *Asset_Y*. Zu sehen ist das Schaubild in Abbildung 4.11. Jeder Schritt auf dem Weg zur vollständigen Aktualisierung der Dokumentation ist durch eine Nummer gekennzeichnet und im Folgenden erläutert. Unberücksichtigt bleibt in dem Ablauf, bestehend aus insgesamt 13 Schritten, die Anzahl

der Wiederholungen der zusammengehörigen Schritte. Erkenntnis darüber bringen die dazugehörigen Sequenzdiagramme.

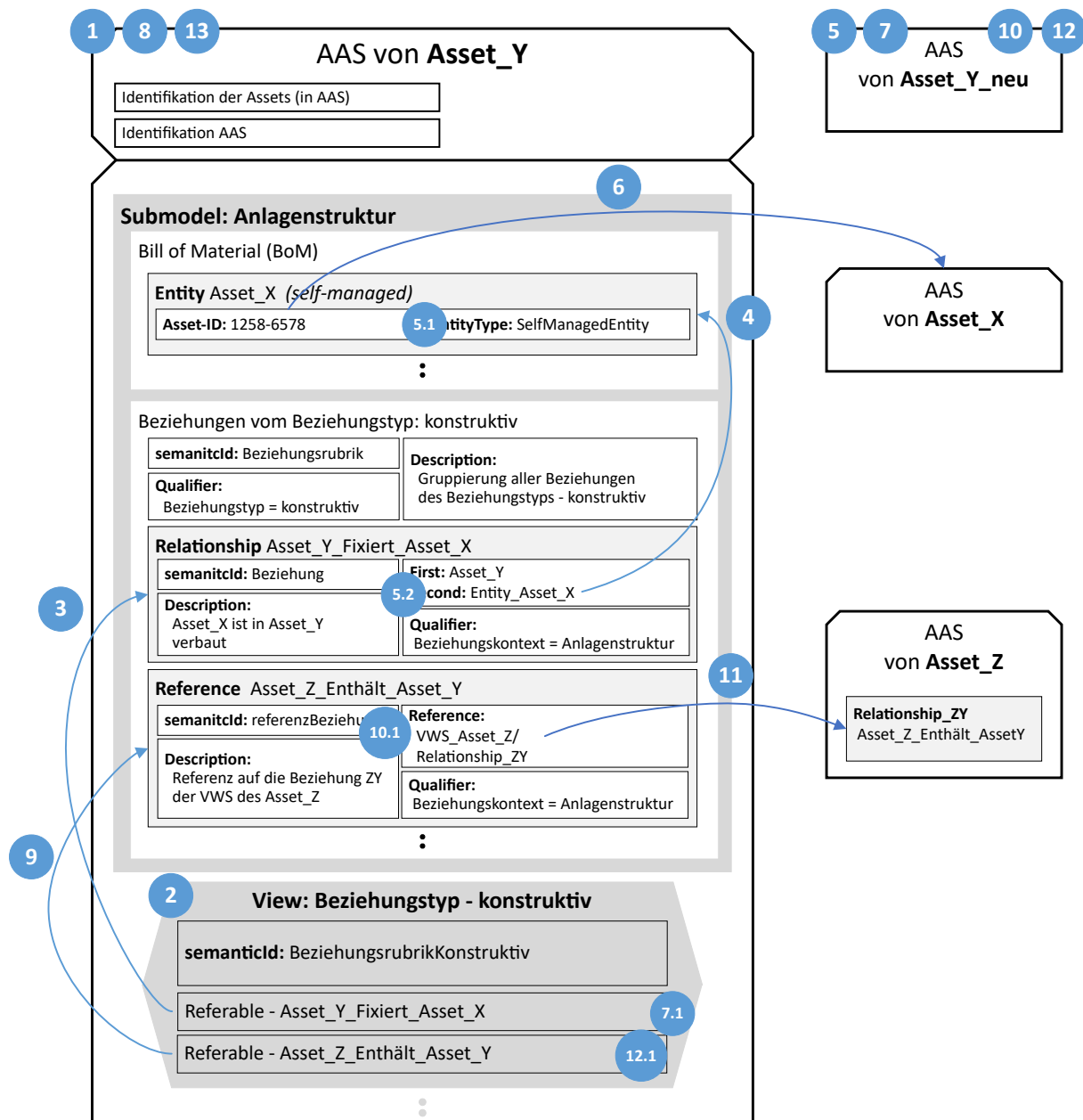


Abbildung 4.11: Prinzipielles Vorgehen für den Eingriff in den AAS-Aufbau zur Dokumentationspflege bei einem Gerätetausch

1. Die AAS von *Asset_Y* identifizieren, d.h. zu der gegebenen Asset-ID einen Endpunkt finden, über den die Daten der AAS zugänglich sind.
2. Die entsprechenden Views in der AAS aus Schritt 1 durchlaufen und alle darin enthaltenen Referables zwischenspeichern.

Schritte zur Aktualisierung des Relationship-Elements

3. Das Relationship-Element über das Referable im View aufrufen und zwischenspeichern. Die beschreibenden Informationen des Submodels und der Submodel Element Collection, in der sich das Element befindet, müssen ebenfalls gespeichert werden.

4. Mit den Informationen aus dem Relationship-Element kann der Referenz zur Entity *Asset_X* gefolgt und diese zwischengespeichert werden. Auch hier ist die Speicherung der beschreibenden Informationen zu dem Submodel und der Submodel Element Collection wie in Schritt drei notwendig.

5. Die beiden gespeicherten Elemente, Entity (**5.1**) und Relationship-Element (**5.2**), in die neue AAS *Asset_Y_neu* einfügen. Dazu muss überprüft werden, ob bereits Submodels und Submodel Element Collections der AAS *Asset_Y* auch in der AAS *Asset_Y_neu* existieren.

In diesem Beispiel muss in der AAS *Asset_Y_neu* ein Submodel *Anlagenstruktur* mit den Submodel Element Collections *Bill of Material* und *Beziehungen vom Beziehungstyp: konstruktiv* vorhanden sein. Die Überprüfung geschieht mit Hilfe der gespeicherten Informationen wie *semanticId* und *Qualifiern* (s. Schritt drei und vier). Kommt das Submodel mit der Submodel Element Collection in der AAS *Asset_Y_neu* nicht vor, müssen diese neu angelegt werden. Andernfalls wird das Element an der entsprechenden Stelle eingefügt.

6. Dieser Schritt wird nur für self-managed Entities ausgeführt. Mit der Asset-ID der Entity *Asset_X*, den Endpunkt der dazugehörigen AAS ermitteln. Daraufhin nach dem Reference-Element in der AAS von *Asset_X* suchen. Den Wert des Reference-Elements mit der Referenz auf das Relationship-Element von der AAS *Asset_Y_neu* ersetzen.

7. Nachdem die AAS *Asset_Y_neu* sowohl das Relationship-Element als auch die Entity enthält, muss in einem entsprechenden View das Referable (**7.1**) für das Relationship-Element wie in der AAS *Asset_Y* hinzugefügt werden. Ist der View mit den vorgegebenen semantischen Bedeutung nicht in der neuen AAS vorhanden, muss dieser neu angelegt werden.

8. Mit der erfolgreichen Aktualisierung des Relationship-Elements und der Entity, werden das Element an sich sowie die Entity aus der AAS *Asset_Y* gelöscht.

Schritte zur Aktualisierung des Reference-Elements

9. Das Reference-Element über das Referable im View aufrufen und zwischenspeichern. Ebenfalls die beschreibenden Informationen des Submodels und der Submodel Element Collection, in der sich das Element befindet, speichern.

10. Das Reference-Element (**10.1**) in das entsprechende Submodel mit der Submodel Element Collection der AAS *Asset_Y_neu* einfügen. Der Vorgang ähnelt dem aus Schritt fünf.

11. Der Referenz im Reference-Element folgen, sodass in der AAS *Asset_Z*, das dazugehörige Relationship-Element gefunden wird. In dem Relationship-Element der AAS *Asset_Z* muss dem Verweis zur Entity, ähnlich dem Schritt vier, gefolgt werden. Danach die Entity *Asset_Y* durch die Entity *Asset_Y_neu* in der entsprechenden BoM der AAS *Asset_Z* ersetzen. Das Relationship-Element bleibt dabei unberührt, da sich bei der Dokumentationspflege nur die Entity und nicht das Beziehungsgerüst ändert.

12. Nach erfolgreicher Einfügung des Reference-Elements in die AAS *Asset_Y_neu*, muss das dazugehörige Referable (**12.1**) durch den entsprechenden View in der AAS *Asset_Y_neu* implementiert werden.

13. Mit der erfolgreichen Aktualisierung des Reference-Elements, ist dieses aus der AAS *Asset_Y* zu löschen.

Die Zusammenführung der Schritte eins bis dreizehn mit den Sequenzdiagrammen aus Kapitel 3.2 vereinfacht die Entwicklung der technologiespezifischen Interaktionen. Für eine leichtere Nachvollziehbarkeit der Erläuterungen zu den Sequenzdiagrammen, werden die Nachrichtennummern aus den Sequenzdiagrammen kursiv in Klammern hervorgehoben. Das für die Entwicklung notwendige Wissen über die AAS-API wird hauptsächlich aus der Spezifikation [28] gewonnen, in der alle bisher entwickelten Operationen der AAS-API definiert sind. Als zusätzliche Quelle dient die Internetplattform „Swagger“, auf der die Entwickler die in [28] spezifizierten Operationen der AAS-API bereitstellen (s. Weblinks [28, S. 97]). Aufgrund der immer kürzeren Entwicklungszyklen, bietet die Internetplattform bereits Operationen an, die bisher nicht in der Spezifikation [28] dokumentiert sind. Die Verwendung solcher Operationen in dieser Arbeit wird an den entsprechenden Stellen erwähnt.

Um den Umfang dieser Arbeit einzugrenzen, werden Maßnahmen zur Fehlererkennung und -behandlung nicht explizit in den Lösungsansätzen aufgeführt. Diese wichtigen Punkte sind bei einer realen Implementierung zu berücksichtigen. Weiterhin werden die Abläufe in den Operationen, die intern in der Softwareapplikation stattfinden, nur grob beschrieben. Der Fokus der technologiespezifischen Sequenzdiagramme liegt auf den Interaktionen zwischen der Softwareapplikation und den AAS sowie den zusätzlichen Systemkomponenten wie Registry, Discovery und Repository. Die Aufgaben der Systemkomponenten werden laut [28, S. 22] folgendermaßen beschrieben:

Repository: Agiert als Server, zur Speicherung von AAS, Submodels und Concept Descriptions. Über das Repository-Interface ist ein Zugriff auf die verwalteten Elemente gewährleistet.

Discovery: Ermöglicht die schnelle Suche nach Identifiern durch die Filterung bestimmter Key-Value-Pairs. Die Discovery speichert lediglich Kopien grundlegender Daten, die für die Suche benötigt werden.

Registry: Die Registry arbeitet ähnlich einem Domain Name System (DNS). Sie speichert die Asset-IDs und die mit den Assets verbundenen AAS-IDs sowie Submodel-IDs ab. Zu jeder dieser AAS-IDs und Submodel-IDs kennt die Registry den Endpunkt. Mit Eingabe einer bekannten Asset-ID, AAS-ID, bzw. Submodel-ID werden die dazugehörigen Endpunkte durch das Registry-Interface bekanntgegeben.

Zusätzliche Informationen zu den Service-Infrastrukturen der AAS sind in [12] zu finden.

Hinweis: Die Bezeichnungen der AAS aus dem oben dargestellten Beispiel werden in den folgenden Sequenzdiagrammen teilweise übernommen. Für die AAS der auszutauschenden Geräte stehen stellvertretend die allgemeineren Bezeichnungen *newDevice* und *oldDevice*. Die AAS von *Asset_X* und *Asset_Z* werden mit *AAS_X* und *AAS_Z* bezeichnet.

Begonnen wird die Dokumentationspflege mit der Identifizierung der AAS aus Schritt eins. Als Hilfe dient das technologieneutrale Sequenzdiagramm aus Abbildung 3.6 (S. 68). Die Beschaffung des Endpunktes über den die AAS erreichbar und der Zugriff auf deren Daten gewährleistet ist, ist das Ziel dieses Schritts. Das technologiespezifische Sequenzdiagramm zur Erfüllung dieser Aufgabe ist in Abbildung 4.12 an dem Beispiel der AAS *oldDevice* zu sehen. Hierbei kann die AAS *oldDevice* die einzige oder eine von vielen zu dem Gerät existierenden AAS sein. Für die Identifizierung der AAS des neuen Geräts muss die Sequenz erneut ausgeführt werden.

Das Sequenzdiagramm besteht aus drei Optionen, die jede für sich allein die Aufgabe aus Schritt eins erfüllt. Mit Hilfe der genannten Systemkomponenten wird der Zugang zu einer AAS gelegt und deren Daten in Form einer JSON-Datei (s. [28, S. 101]) als Antwort erhalten.

In Option eins besteht die Möglichkeit, direkt mit der Asset-ID die Daten aller zu dem Asset gehörenden AAS von dem Repository zu erhalten (1.0). Dies stellt die einfachste Realisierung der Aufgabe dar. Mit etwas mehr Rechenaufwand verbunden sind hingegen die anderen beiden Optionen, bei denen zuerst die AAS-ID von der Discovery angefordert wird (2.0), (3.0). Durch die AAS-ID ist die direkte Kommunikation zum einen mit dem Repository (2.2) und zum anderen mit der AAS (3.5) gewährleistet. Die letztere Option benötigt vor der Kommunikation mit der AAS den über die Registry bereitgestellten AAS-Endpunkt (3.2), (3.4).

Rechts neben den Nachrichten ist die jeweilige Schnittstelle (engl. Interface) mit dem Namen der Operation aus der Spezifikation [28] als Kommentar aufgeführt. Die Merkmale zum Rechenaufwand sowie dem Informationsgehalt aller drei Optionen sind in Tabelle 4.3 kurz zusammengefasst.

Hinweis: Dadurch, dass jede der drei Optionen aus Schritt eins alle zu einem Asset dazugehörigen AAS zurückgibt, ist der Durchlauf des Sequenzdiagramms der entsprechenden Option nur einmal notwendig. Um dies zu ermöglichen ist eine Voraussetzung, dass sich in Option eins und zwei alle zu einem Asset gehörenden AAS in ein und demselben Repository befinden. Sollte diese Voraussetzung nicht vorhanden sein und die AAS in verschiedenen Repositories liegen, muss die Sequenz, je nach Anzahl der Repositories, wiederholt werden. Wird die dritte Option implementiert gilt laut dem Sequenzdiagramm die Annahme, dass alle zu einem Asset gehörenden AAS in einer einzigen Registry gemeldet sind. Ist dies nicht der Fall, weicht das Sequenzdiagramm, in Bezug auf die Kommunikation mit den Registries, ab.

Option	Merkmale	Anzahl notwendiger Operationen für ein Gerät ^a
1	<ul style="list-style-type: none"> Die AAS, die zu einem Asset existieren, werden als Array zurückgegeben. Es sind keine Informationen zu den Endpunkten der AAS enthalten. 	$y = f(x_1) = 1$ (4.1)
2	<ul style="list-style-type: none"> Die zu einem Asset gehörenden AAS-IDs werden als Array zurückgegeben. Die AAS werden iterativ aufgerufen, je nach der Anzahl der zum Asset gehörigen AAS. Es sind keine Informationen zu den Endpunkten der AAS enthalten. 	$y = f(x_1) = 1 + x_1$ (4.2)
3	<ul style="list-style-type: none"> Die AAS-IDs eines Assets werden als Array zurückgegeben. Durch den Descriptor der AAS sind sowohl deren Endpunkt als auch die Endpunkte der in der AAS enthaltenen Submodels bekannt. Diese Informationen müssen über eine interne Operation der Softwareapplikation extrahiert werden. 	$y = f(x_1) = 1 + 3x_1$ (4.3)

^a x_1 : Anzahl AAS, y : Anzahl Operationen

Tabelle 4.3: Merkmale und Rechenaufwand zur Identifizierung der AAS eines Geräts über HTTP/REST-API

Die folgenden Schritte zwei bis dreizehn müssen für jede AAS des alten bzw. neuen Geräts durchlaufen werden. Zur Senkung des Programmieraufwands wird empfohlen, die Sequenzen nicht parallel für alle AAS eines Geräts auszuführen, sondern nacheinander jede

AAS des Geräts anhand der zwölf Schritte fehlerfrei abzuarbeiten. Somit werden die zwölf Schritte in einer Schleife, für jede zum Gerät vorhandene AAS, durchlaufen. Dadurch wird auch erkannt, wenn das alte Gerät beispielsweise mehr AAS als das neue Gerät besitzt. Die Daten, die während der Ausführung dieser Schritte anfallen, sollten mit Hilfe eines objektorientierten Ansatzes gespeichert werden.

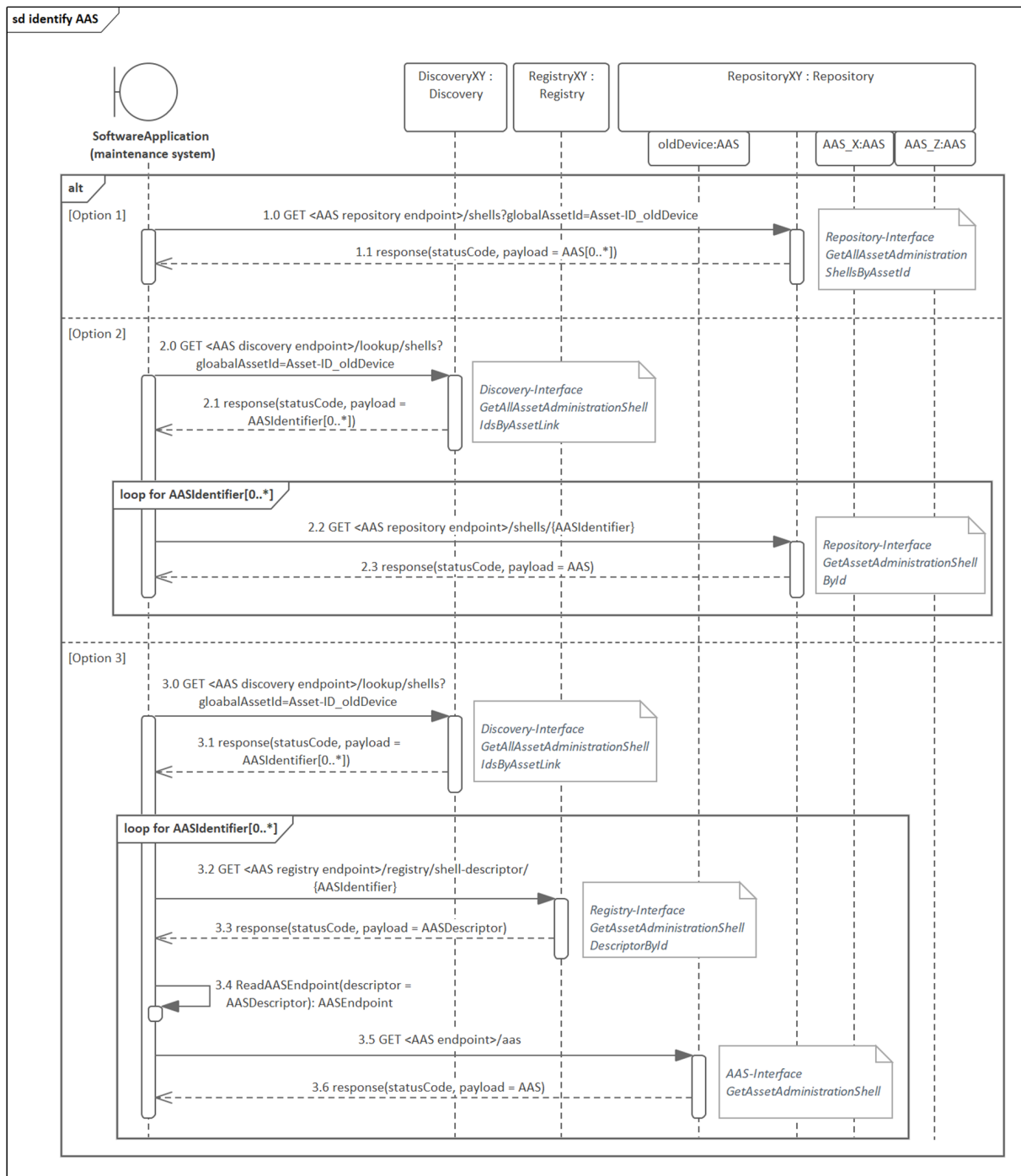


Abbildung 4.12: HTTP/REST-spezifische Zugriffe auf die AAS-API zur Identifizierung einer AAS

Hinweis: Existieren mehr als eine AAS zu einem Gerät, muss die Softwareapplikation wissen in welche AAS des neuen Geräts diese die Informationen übertragen soll. Die Konzepte, mit denen ein Vergleich zwischen den AAS des alten Geräts und denen des neuen Geräts realisierbar ist, sind nicht Teil dieser Arbeit.

Nach der Identifizierung der AAS des alten und neuen Geräts, befasst sich der zweite Schritt mit der Speicherung der Referables eines Views. Da für den View kein Interface existiert, müssen die entsprechenden Daten aus den JSON-Dateien von Schritt eins extrahiert werden. Dafür wird eine eigene Operation entwickelt (s. Abbildung 4.13 Nachricht (1.0)), die als Ergebnis die Referables nach Relationship- und Reference-Elements in zwei Arrays aufteilt. Die Informationen, auf welches Element das Referable verweist, sind in der serialisierten JSON-Datei enthalten. Die ordnungsgemäße Interpretation der JSON-Datei ist mit Hilfe der Spezifikation [26, S. 245–272] gewährleistet.

Ein Beispiel, wie das Reference-Element des AAS-Metamodells serialisiert wird, ist durch dessen *value* im unten eingerahmten Quellcode gezeigt. Der Typ sowie weitere Informationen werden in *keys*, aus denen sich der *value* zusammensetzt, gespeichert. Alle *keys* zusammen bilden die Referenz auf ein AAS-Element, das in diesem Beispiel das Relationship-Element

PickAndPlaceConsistsOfWorkstation1,

mit dem Typen-Pfad

AssetAdministrationShell/Submodel/SubmodelElementCollection/RelationshipElement,

ist.

Bei der Betrachtung des Referable ergibt sich für den *value* die gleiche Struktur wie bei dem Reference-Element. Damit ermöglicht die Auswertung des letzten *key* der Liste, hinsichtlich des angegebenen Typen des AAS-Elements, die Unterteilung der Referables in die zwei oben genannten Arrays. Infolge der Unterteilung wird jedem Array-Element der *value* des entsprechenden Referables zugewiesen.

```
"value": {
  "keys": [
    {
      "type": "AssetAdministrationShell",
      "value": "https://example.com/ids/aas/6210_4190_2022_5263",
      "index": 0,
      "idType": "IRI"
    },
    {
      "type": "Submodel",
```

```

    "value": "https://example.com/ids/sm/5020_0190_2022_7216",
    "index": 1,
    "idType": "IRI"
  },
  {
    "type": "SubmodelElementCollection",
    "value": "RelationshipsConstructive",
    "index": 2,
    "idType": "IdShort"
  },
  {
    "type": "RelationshipElement",
    "value": "PickAndPlaceConsistsOfWorkstation1",
    "index": 3,
    "idType": "IdShort"
  }
]
}

```

Mit den beiden Arrays aus Schritt zwei, in denen sich die Werte der Referables befinden, ist die Umsetzung der Aufgaben aus den Schritten drei und neun möglich. Das Sequenzdiagramm, das die Schritte zwei, drei und neun mit seinen notwendigen Operationen darstellt ist in Abbildung 4.13 zu sehen.

Zu Anfang der Sequenz werden wie oben beschrieben alle Referables eines Views gespeichert (1.0). Danach muss im Fall, dass die erste oder zweite Option zur Identifizierung der AAS genutzt wurde, der Descriptor der AAS mit Hilfe der Registry beschafft werden (1.1). Durch den AAS-Descriptor, der zusätzlich die Descriptoren aller in der AAS enthaltenen Submodels bereitstellt, sind die Endpunkte der Submodels für eine direkte Interaktion mit diesen bekannt (s. [28, S. 69 ff.]).

Die Endpunkte werden durch eine interne Operation der Softwareapplikation (1.3) aus dem AAS-Descriptor extrahiert. Dazu wird als zusätzliches Attribut die Submodel-ID aus dem jeweiligen Array-Element des Array *relationshipLinks[]* bzw. *referenceLinks[]* genutzt und mit den Submodel-IDs aus den gegebenen Submodel-Descriptoren auf Übereinstimmung geprüft. Bei einem positiven Ergebnis enthält die Ausgabe den Endpunkt des Submodels. Anschließend beginnt durch den gegebenen Endpunkt eine direkte Interaktion mit dem entsprechenden Submodel, in dem sich das Reference- bzw. Relationship-Element befindet. Dabei werden wie in Schritt drei und neun anfangs beschrieben, die Informationen zu z.B. semanticId und Qualifier aus dem Submodel (1.4) und der Submodel Element Collection (1.6) extrahiert. Der Parameter *trimmed* hinter den HTTP/REST-Befehlen teilt dem

Interface mit, das ausschließlich Metainformationen des mitgegebenen Objekts gewünscht sind (s. [28, S. 100 f.]). Diese Metainformationen werden durch den objektorientierten Ansatz zu dem jeweiligen Reference- bzw. Relationship-Element gespeichert und stehen für die noch folgenden Aufgaben zur Verfügung. Zum Schluss der Sequenz findet ein weiterer Zugriff auf das Submodel-Interface statt, um die Daten des Reference- bzw. Relationship-Elements zu erhalten und zu speichern.

Alle Operationen des Sequenzdiagramms aus Abbildung 4.13 werden für jeden View eines Beziehungstyps wiederholt. In dieser Schleife befindet sich eine zweite Schleife, die für jedes Reference- und Relationship-Element die benötigten Informationen beschafft.

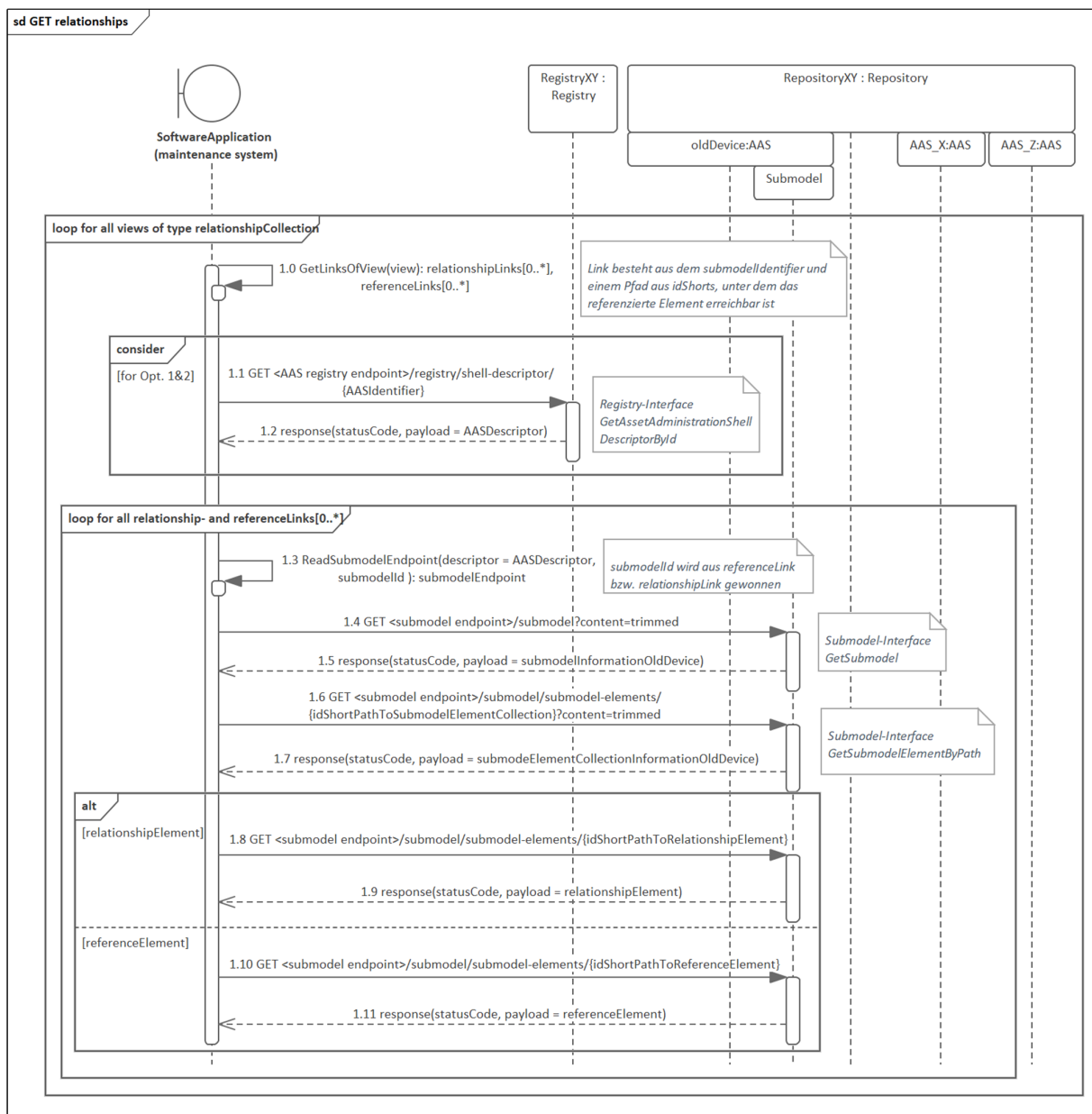


Abbildung 4.13: HTTP/REST-spezifisches Sequenzdiagramm zur Umsetzung der Interaktion „GET Beziehung“

Eine zweite Möglichkeit die Aufgaben aus Schritt drei und neun zu lösen, besteht in der Kombination des Repository-Interface mit dem AAS-Interface und dem Submodel-Interface wie in [28, S. 23 f.] beschrieben. Das heißt, es kann alternativ nur eine Kommunikation über das Repository, in dem sich die AAS mit dem Reference- bzw. Relationship-Element befindet, stattfinden. Die Nachrichten (1.4) bis (1.10) müssen dadurch folgendermaßen geändert und direkt an das Repository adressiert werden:

```
Nachricht 1.4
  GET <AAS repository endpoint>/shells/{AASIdentifizier}/aas/
    submodels/{submodelId}/submodel
Nachricht 1.6
  GET <AAS repository endpoint>/shells/{AASIdentifizier}/aas/
    submodels/{submodelId}/submodel/submodel-elements/
    {idShortPathToSubmodelElementCollection}
Nachricht 1.8 bzw. 1.10
  GET <AAS repository endpoint>/shells/{AASIdentifizier}/aas/
    submodels/{submodelId}/submodel/submodel-elements/
    {idShortPathTo<ELEMENT>}
```

Vorteil dieser Implementierung ist, dass die Endpunkte der Submodels nicht bekannt sein müssen und deshalb auch kein AAS-Descriptor vorliegen muss. Lediglich die AAS-ID aus Schritt eins und die Submodel-IDs sowie die idShort-Pfade aus den Arrays *referenceLinks[]* und *relationshipLinks[]* müssen bekannt sein.

Die Belastung des Repositories nimmt mit dieser Methode jedoch zu, da wahrscheinlich nicht nur die Softwareapplikation die für den Gerätetausch zuständig ist, sondern auch andere Applikationen auf das Repository zugreifen wollen und daraus längere Wartezeiten resultieren. Weiterhin muss bekannt sein, in welchem Repository sich die gesuchte AAS befindet. Ist dies nicht der Fall, ist eine Durchsuchung aller vorhandenen Repositories erforderlich.

Im vierten Schritt, der für die Aktualisierung des Relationship-Elements zuständig ist, müssen die Informationen aus der vom Relationship-Element referenzierten Entity gewonnen werden. Das passende Sequenzdiagramm dazu ist in Abbildung 4.14 dargestellt.

Zu Beginn der Sequenz erfolgt die Extraktion der Referenz auf die Entity aus dem Endpunkt *second* des Relationship-Element (1.0). Das Ergebnis wird in der Variablen *linkToEntity* gespeichert und ist wie die Array-Elemente aus Schritt zwei strukturiert. Die darauffolgenden Sequenzen (1.1) bis (1.7) aus Abbildung 4.14 werden von dem Sequenzdiagramm aus Abbildung 4.13 wiederverwendet. Auch hier ist es denkbar, direkt mit dem Repository anstatt dem jeweiligen Submodel zu interagieren.

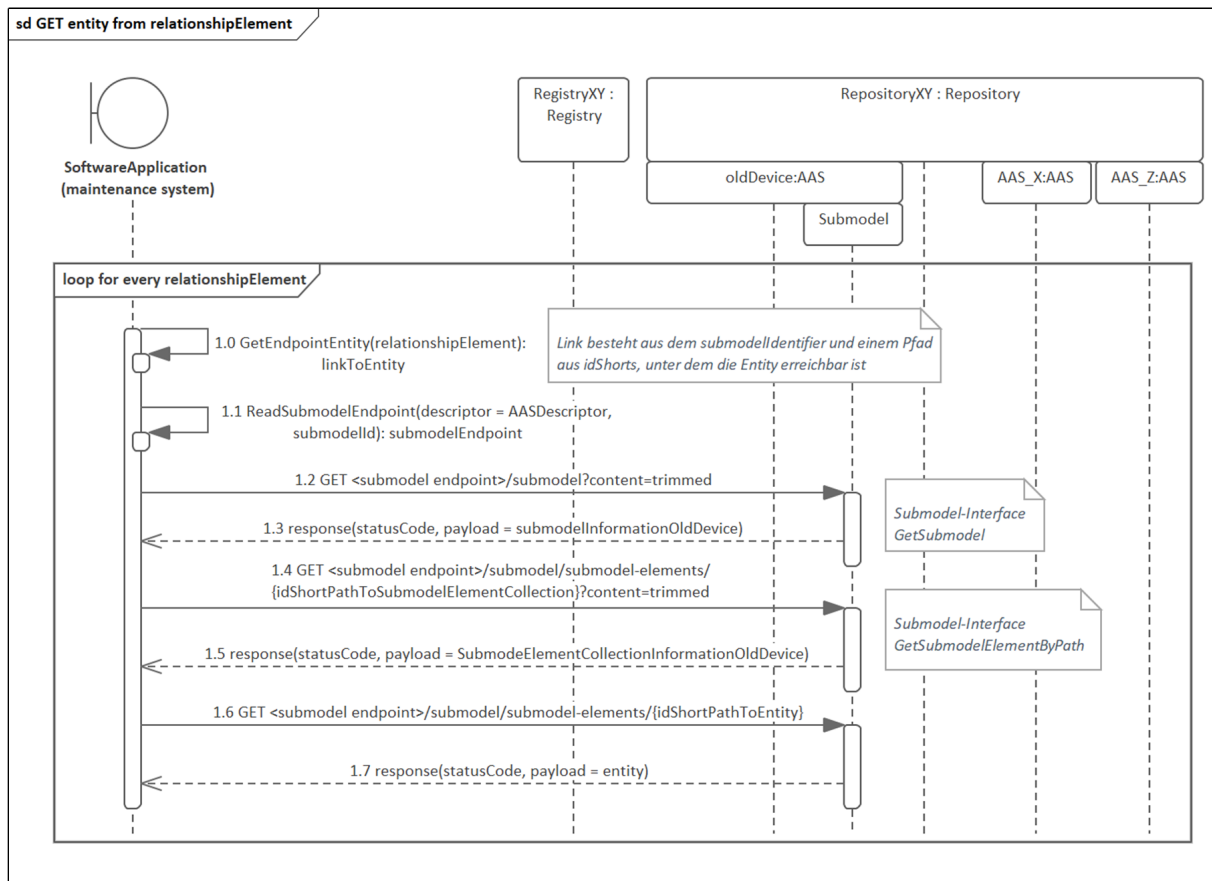


Abbildung 4.14: HTTP/REST-spezifisches Sequenzdiagramm für die Speicherung einer Entity

Nachdem die Informationen zu allen Beziehungen und Entities abgelegt sind, folgen die Schritte fünf und zehn, die das Einfügen der AAS-Elemente in die AAS des neuen Geräts vorsehen. Dafür sind mehrere Operationen und Prüfungshandlungen notwendig, die durch das Sequenzdiagramm in Abbildung 4.15 abgebildet sind.

Der zu der AAS *newDevice* gehörende AAS-Descriptor ist bereits durch Schritt eins, der Identifizierung der AAS, bekannt. Dieser Descriptor wird gelesen und dessen Submodel-Descriptor in einem Array ausgegeben (1.0). In der nachfolgenden Operation erfolgt der Vergleich dieser Submodel-Descriptor mit dem aktuellen Submodel-Descriptor der AAS *oldDevice* (2.0). Das Ergebnis aus dem Vergleich ist ein Wahrheitswert, der angibt, ob sich bereits ein Submodel mit den gespeicherten Submodel-Metainformationen in der AAS *newDevice* befindet.

Ist das Ergebnis der Operation (2.0) positiv und ein solches Submodel damit bereits in der AAS *newDevice* vorhanden, muss die Existenz der Submodel Element Collection auf die gleiche Weise überprüft werden (3.0), (4.0). Wenn auch dieses Ergebnis positiv ist bedeutet dies, dass das AAS-Element (Entity, Relationship- bzw. Reference-Element) an die entsprechende Stelle eingefügt werden kann (3.2). Ist das Ergebnis der zweiten Prüfung jedoch negativ, ist die Submodel Element Collection und im folgenden auch das

AAS-Element neu einzufügen (4.2), (4.4). Resultiert jedoch schon bei der Überprüfung des Submodels ein negatives Ergebnis, muss das komplette Submodel, mit der Submodel Element Collection (5.0) und dem AAS-Element (5.2) zu der AAS hinzugefügt werden.

Besonders hervorgehoben in dem Sequenzdiagramm ist das Einfügen eines neuen Submodels in die AAS, gekennzeichnet durch einen Interaktionsrahmen mit dem Namen *POST new submodel to AAS*. Für die Implementierung dieser Aktion existiert in der Spezifikation [28] bisher keine Operation. Schaut man jedoch auf die Entwicklungsplattform „Swagger“ ist eine Operation des Repository-Interface zu finden, die die Einfügung eines neuen Submodels erlaubt. Die Kombination des Repository-Interface mit dem AAS-Interface (s. [28, S. 23 f.]) lässt die Gestaltung der Nachricht in dem Interaktionsrahmen nach folgendem Schema zu:

```
POST <AAS repository endpoint>/shells/{AASIdentifier}/  
aas/submodels
```

Diese Nachricht wird direkt an das Repository, das zur Verwaltung der AAS dient, gesendet. Die gesamte oben beschriebene Sequenz wird für alle Nachrichten, ausgenommen von (1.0), für jedes Reference- bzw. Relationship-Element und für jede Entity wiederholt.

Mit dem Einfügen der drei AAS-Elementtypen (Reference, Relationship und Entity) in die AAS des neuen Geräts, sind anschließend die AAS-Elementtypen in den verknüpften AAS zu aktualisieren. Eine Aufgabe wurde in Schritt sechs beschrieben, die die Aktualisierung eines Reference-Elements aus der verknüpften AAS *AAS_X* zur Aufgabe hat. Das Sequenzdiagramm zur Erfüllung der Aufgabe ist in Abbildung 4.16 dargestellt.

Zu Beginn dieses Schritts muss die Asset-ID aus einer self-managed Entity extrahiert werden (1.0). Die Entity selbst ist bereits aus dem Sequenzdiagramm in Abbildung 4.14 bekannt. Mit Hilfe der Asset-ID werden die dazugehörigen AAS bestimmt. Der Ablauf dafür ist exakt der aus dem Sequenzdiagramm in Abbildung 4.12. Infolgedessen resultiert mindestens eine AAS aus der Asset-ID, in der die gesuchte Referenz für die Aktualisierung abgelegt sein kann.

In einer Schleife wird jede zu der Asset-ID bekannte AAS genommen und in dieser nach einem View gesucht, der die Referables zu einem vorgegebenen Beziehungstypen enthält (2.0). Den gesuchten Beziehungstyp gibt dabei das Relationship-Element durch seinen Qualifier vor. Aus allen Referables selektiert Operation (2.0) nur die, die auf ein Reference-Element verweisen, da nur diese für den Schritt sechs von Relevanz sind.

Mit den *referenceLinks* beginnt eine zweite Schleife, in der die Daten jedes vom Referable verlinkten Reference-Elements über das Submodel-Interface geholt werden (3.1). Der Endpunkt des Submodels wird über die bereits mehrfach verwendete Operation (3.0) bereitgestellt. Der *value* des Reference-Elements liefert die AAS-ID, die Submodel-ID und den idShort-Pfad des Gegenstücks der Beziehung, nämlich dem Relationship-Element aus

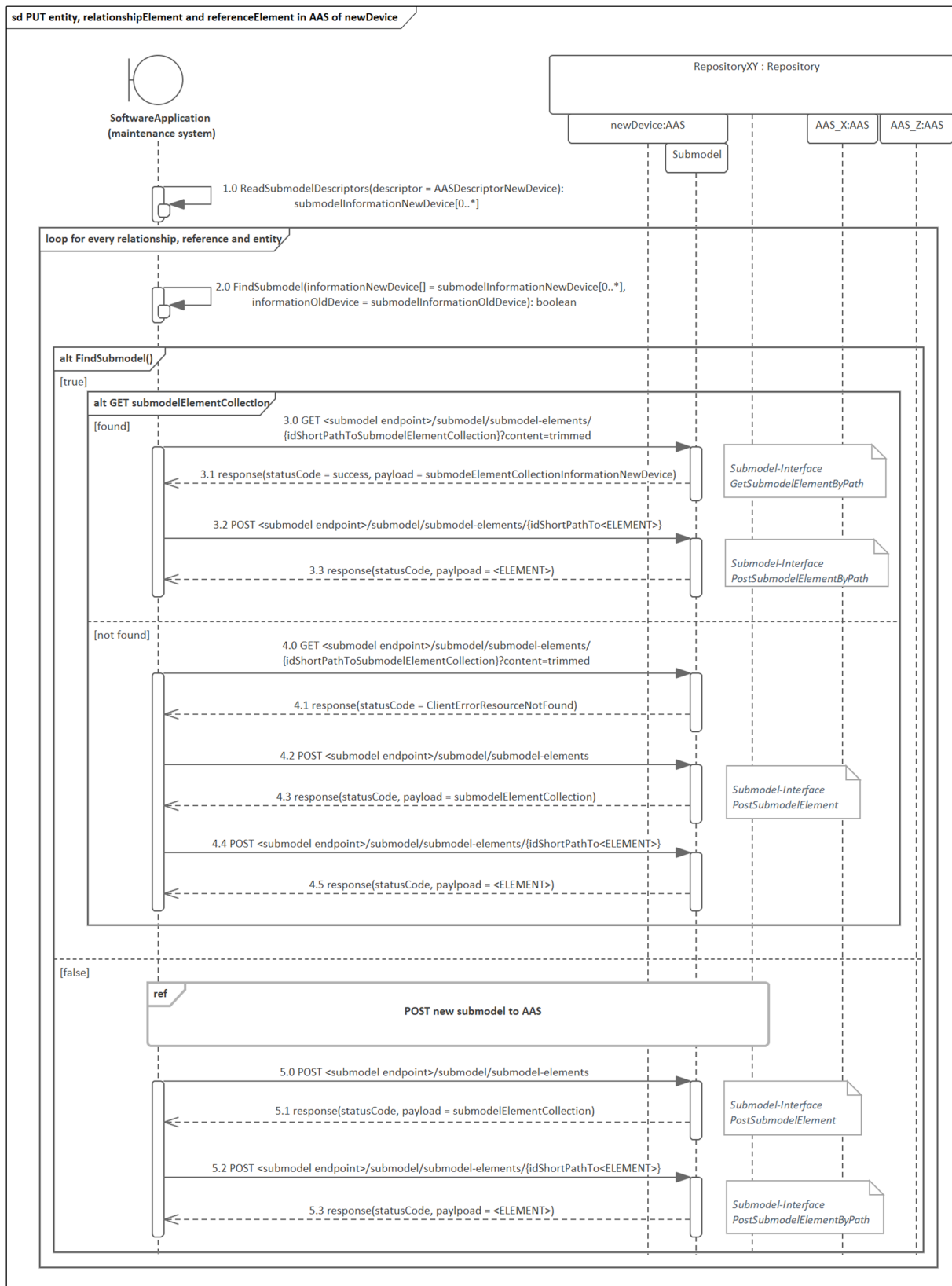


Abbildung 4.15: HTTP/REST-spezifisches Sequenzdiagramm für die Einbindung von Entity, Relationship- oder Reference-Element in eine AAS

der AAS *oldDevice*. Daraufhin ist es möglich, die Informationen des aktuell untersuchten Relationship-Elements (äußerste Schleife im Sequenzdiagramm) mit den Informationen aus dem *value* des Reference-Elements auf Übereinstimmung zu prüfen (3.3). Resultiert aus der Überprüfung ein positives Ergebnis, wird der *value* des Reference-Elements mit dem Verweis auf das Relationship-Element der AAS *newDevice* ersetzt. Ist das Ergebnis jedoch negativ, ist das nächste Reference-Element in der verknüpften AAS *AAS_X* für einen erneuten Vergleich aufzurufen.

Eine geringfügige Vereinfachung der Sequenz ist auch in diesem Schritt durch die Kombination des Repository-Interface mit dem Submodel-Interface gegeben. Dadurch entfällt die Nachricht (3.0) und die Nachrichten (3.1) und (4.0) sehen folgendermaßen aus.

Nachricht 3.1

```
GET <AAS repository endpoint>/shells/{AASIdentifier}/aas/
submodels/{submodelId}/submodel/submodel-elements/
{idShortPathToReferenceElement}
```

Nachricht 4.0

```
PUT <AAS repository endpoint>/shells/{AASIdentifier}/aas/
submodels/{submodelId}/submodel/submodel-elements/
{idShortPathToReferenceElement}
```

Bei der direkten Kommunikation mit dem Repository ist auch hier die möglicherweise entstehende längere Wartezeit infolge eines hohen Kommunikationsaufkommens zu beachten. Die beschriebene Sequenz ist für jedes Relationship-Element aus der AAS *oldDevice* zu wiederholen.

Hinweis: Die Existenz von mehr als einer AAS zu einem Asset sowie die nicht eindeutige Verbindung zwischen dem Relationship-Element und dem Reference-Element tragen zur Komplexität der aufgeführten Lösungsansätze bei. Maßnahmen zur Reduktion der Komplexität sind zu dem Zeitpunkt dieser Arbeit nicht bekannt.

Grundsätzlich würde nun die Umsetzung der Schritte sieben und acht für die Aktualisierung des Relationship-Elements folgen. Da diese jedoch zusammen mit den Schritten zwölf und dreizehn die Beendigung der Aktualisierung von Relationship- und Reference-Elementen darstellen, folgt als Einschub die technologiespezifische Entwicklung von Schritt elf. Das Sequenzdiagramm für Schritt elf ist in Abbildung 4.17 zu sehen.

Eine Teilaufgabe während dieses Schritts ist, den *value* jedes Reference-Elements aus der AAS des alten Geräts zu holen, da darin auf das dazugehörige Relationship-Element der verknüpften AAS verwiesen wird (1.0). Anschließend können mit Hilfe des Verweises die Daten des Relationship-Elements gewonnen werden (1.1). Voraussetzung für den fehlerfreien Ablauf des Sequenzdiagramms ist, dass jedes Reference-Element sowohl die AAS-ID, die Submodel-ID als auch den idShort-Pfad zur eindeutigen Referenzierung

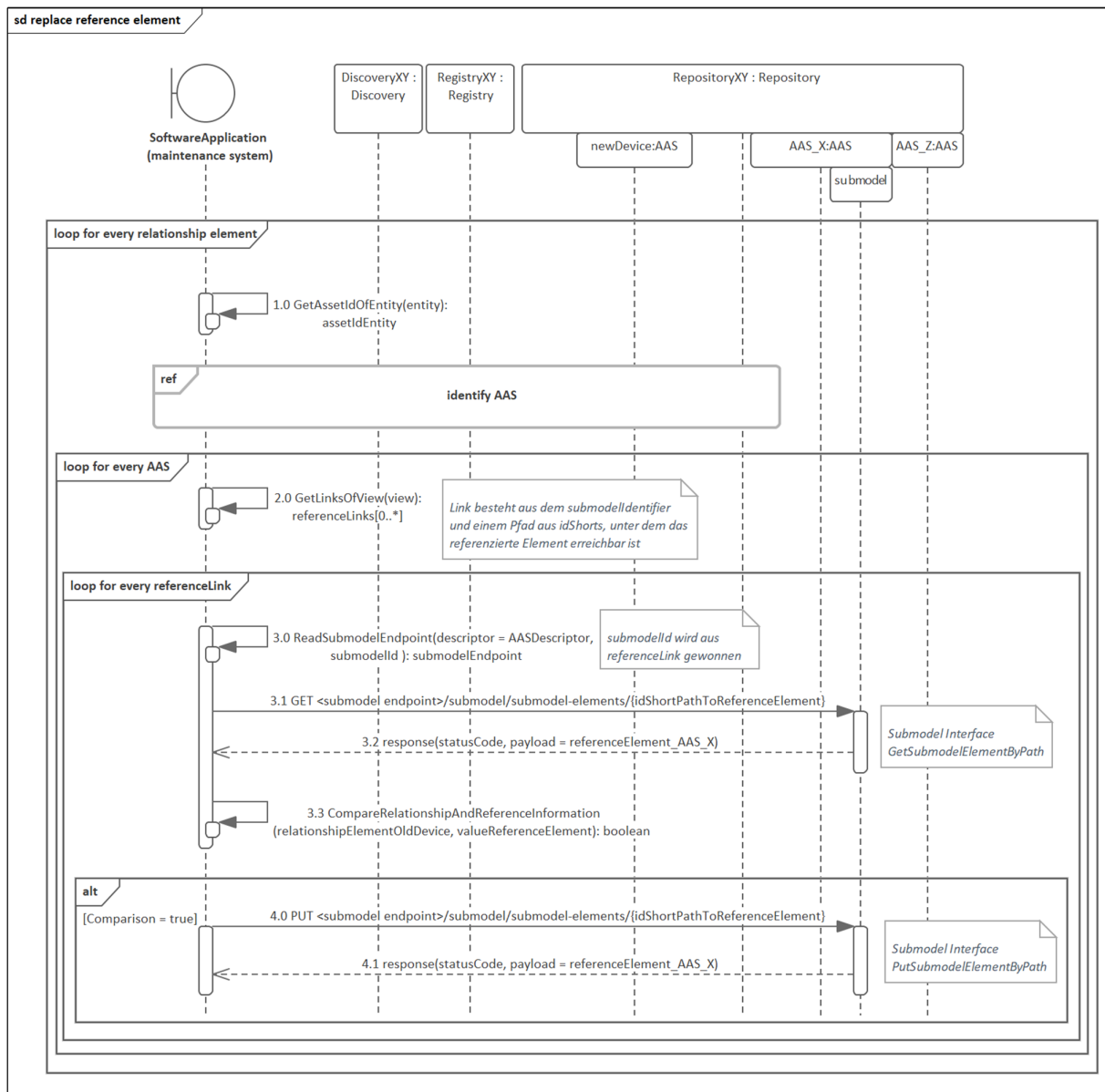


Abbildung 4.16: HTTP/REST-spezifisches Sequenzdiagramm für das Aktualisieren eines Reference-Elements

des Relationship-Elements enthält. Die Interaktion geschieht dieses mal direkt mit dem Repository-Interface und nicht mit dem Submodel-Interface, um auch diese Variante in einem Sequenzdiagramm dargestellt zu haben.

Über die Daten des Relationship-Elements ist der Zugriff auf die Entity der AAS AAS_Z gewährleistet. Die Speicherung der Entity erfolgt nach dem Sequenzdiagramm aus Abbildung 4.14. Aus dem wiederverwendeten Sequenzdiagramm ist der Endpunkt des Submodels, in dem sich die Entity befindet, bekannt. Infolgedessen ist nur noch die Aktualisierung der Asset-ID der Entity, die bisher auf das alte Gerät zeigt, durch die Asset-ID des neuen Geräts zu ersetzen (1.3).

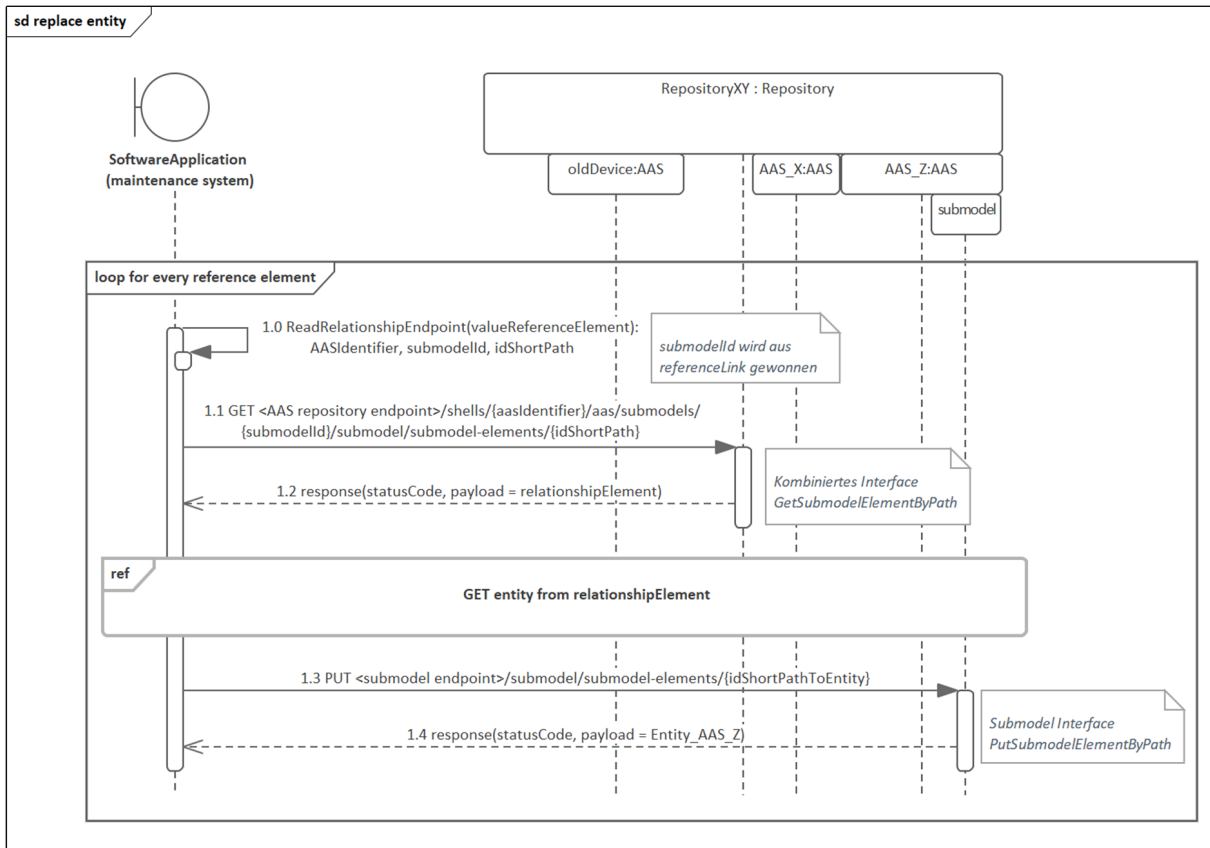


Abbildung 4.17: HTTP/REST-spezifisches Sequenzdiagramm für das Aktualisieren einer Entity

Nachdem alle Elemente aus der AAS des alten Geräts in die AAS des neuen Geräts kopiert wurden, müssen die Views der AAS *newDevice* ebenso aktualisiert werden. Schritt sieben und zwölf beschreiben diese Aufgabe für das Relationship- und Reference-Element. Die Entwicklung der Sequenz ist unabhängig von den beiden AAS-Elementen und wird durch den Platzhalter *<ELEMENT>* vereinheitlicht. In Abbildung 4.18 ist das entsprechende Sequenzdiagramm abgebildet.

Da kein Interface für den Zugriff auf einen View existiert, ist zuerst die Beschaffung der AAS *newDevice* in Interaktion mit dem AAS-Interface notwendig (1.0). Da die AAS als serialisierte JSON-Datei zurückgegeben wird, müssen intern in der Softwareapplikation alle nötigen Operationen, zur Aktualisierung des View, durchgeführt werden. Das beinhaltet zum einen das Aufsuchen des View, in dem das AAS-Element einzufügen ist (2.0). Zum anderen muss eine Übersetzung der Beziehungstypen stattfinden, da der View die *semanticId* besitzt und die AAS-Elemente hingegen den Qualifier zur Beschreibung des Beziehungstyps nutzen.

Ist der richtige View selektiert, kann dessen Ergänzung durch das entsprechende Referable erfolgen (3.0). Dazu wird lediglich die Submodel-ID und der *idShort*-Pfad des AAS-Elements benötigt. Ist der gesuchte View allerdings nicht in der AAS enthalten, muss dieser zusammen mit dem Referable neu hinzugefügt werden (4.0), (4.1). Mit Beendigung

der Bearbeitung der JSON-Datei kann die Aktualisierung der AAS über das AAS-Interface stattfinden (5.0).

Auch in dieser Sequenz, muss die Kommunikation nicht unbedingt mit dem AAS-Interface erfolgen. Die folgenden Nachrichten (1.0) und (5.0) zeigen, wie die Kommunikation mit dem Repository-Interface aussehen würde.

Nachricht 1.0

GET <AAS repository endpoint>/shells/{AASIdentifizier}

Nachricht 5.0

PUT <AAS repository endpoint>/shells/{AASIdentifizier}

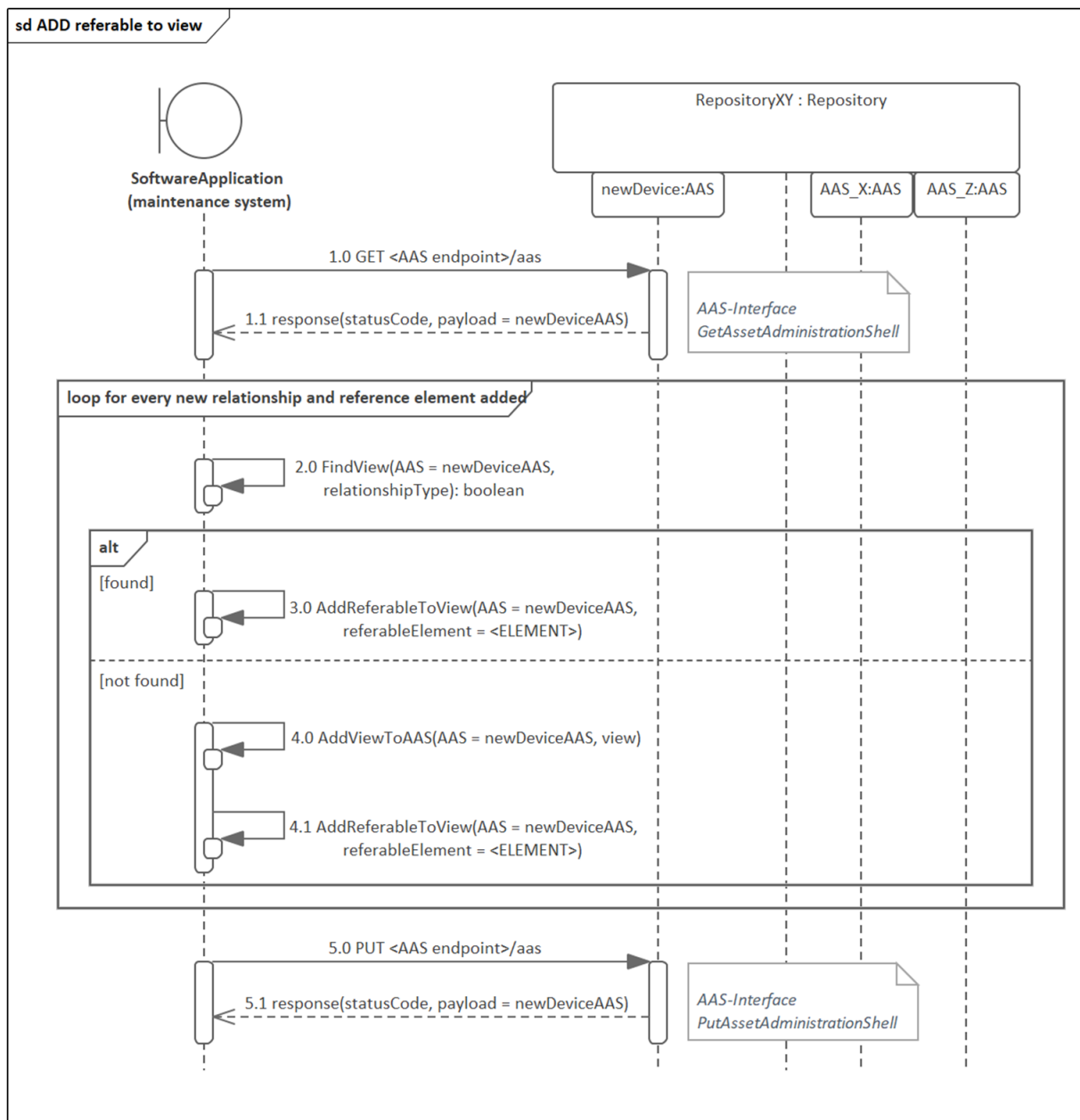


Abbildung 4.18: HTTP/REST-spezifisches Sequenzdiagramm für das Hinzufügen eines Elements in einen View

Mit Beendigung der Schritte eins bis sieben und neun bis zwölf, ist die AAS des neuen Geräts für den Produktionsbetrieb nahezu einsatzbereit. Allerdings fehlt der Schritt zur Löschung der AAS-Elemente in den AAS des alten Geräts, da dieses nun keine bzw. fast keine Beziehungen mehr zu den verknüpften AAS unterhält. Diese Aufgabe wurde bereits durch die Schritte acht und dreizehn beschrieben und ist im technologiespezifischen Sequenzdiagramm der Abbildung 4.19 dargestellt.

Es ist zu beachten, dass sowohl das AAS-Element aus der entsprechenden Submodel Element Collection als auch aus dem View, in Form des Referables, zu löschen ist. Für erst genanntes, wird in dem Sequenzdiagramm das Submodel, in dem sich das zu löschende Element befindet, angesprochen und der Befehl zum Löschen gegeben (1.0). Diese Operation wird für jedes Relationship- und Reference-Element wiederholt.

Danach erfolgt die Beschaffung der AAS mit den Views, die ähnlich dem Sequenzdiagramm aus Abbildung 4.18, über eine JSON-Datei von dem AAS-Interface zurückgegeben werden (2.0). Daran angeknüpft findet die Löschung des Referable, das auf eines der beiden AAS-Elemente zeigt, statt (3.0). Dafür wird der *referenceLink* bzw. *relationshipLink*, der zu Beginn aus Schritt zwei gewonnen wurde, genutzt. Diese Operation wird ebenfalls für alle Relationship- und Reference-Elemente wiederholt.

Der View an sich wird in diesem Use Case nicht gelöscht, da die Verwendung des alten Geräts nach einer erfolgreichen Instandsetzung auch an anderer Stelle im Produktionsbetrieb denkbar ist (Wiederverwendung). Sobald die AAS auf dem aktuellen Stand ist, kann die JSON-Datei entweder über das AAS-Interface (4.0) oder über das Repository-Interface (5.0) veröffentlicht werden.

Hinweis: Die gesamte Sequenz für die Dokumentationspflege bei einem Gerätetausch ist nur technisch umsetzbar, wenn alle Elemente des AAS-Metamodells wie in Kapitel 4.3 definiert verwendet werden. Zusätzlich müssen die Datenmodelle (Submodel Element Collections) in den Submodels in allen AAS nach dem Konzept aus Kapitel 4.3 strukturiert sein, um eine Interoperabilität für diesen spezifischen Use Case zu gewährleisten.

Darüber hinaus sind Änderungen in der AAS-Struktur von z.B. einem Produktionsmodul nur durch die Sperrung dessen möglich. Ansonsten besteht die Gefahr, dass während dem Austausch der betroffenen AAS in der AAS-Struktur ein Zugriff auf nicht konsistente Daten durch externe Systeme erfolgt.

Ein weiterer Gedanke gilt den Beziehungen, die auch nach dem Entfernen des alten Geräts aus einem Produktionsmodul in der AAS des Geräts verbleiben. Bisher wurde angenommen, dass alle Beziehungen aus den AAS des alten Geräts gelöscht werden. Beispielfhaft seien Beziehungen zu Instandhaltungssystemen wie einem MES genannt, die auch nach Entfernung eines Geräts aus dem Produktionsmodul weiterhin Bestand haben. In diesem Fall müssen Unterscheidungen zwischen den Beziehungen, die an die AAS des neuen Geräts übertragen und den Beziehungen die erhalten bleiben sollen, getroffen werden. Ein Lösungsansatz für diese Beziehungen wäre die Verlinkung durch Referables in einem

View auszusetzen. Dadurch hat die von extern zugreifende Softwareapplikation keine Kenntnis über diese Beziehungen und kann sie somit nicht verändern. Die Anpassungsfähigkeit der bisher entwickelten Konzepte ist in nachfolgenden Forschungsarbeiten zu untersuchen.

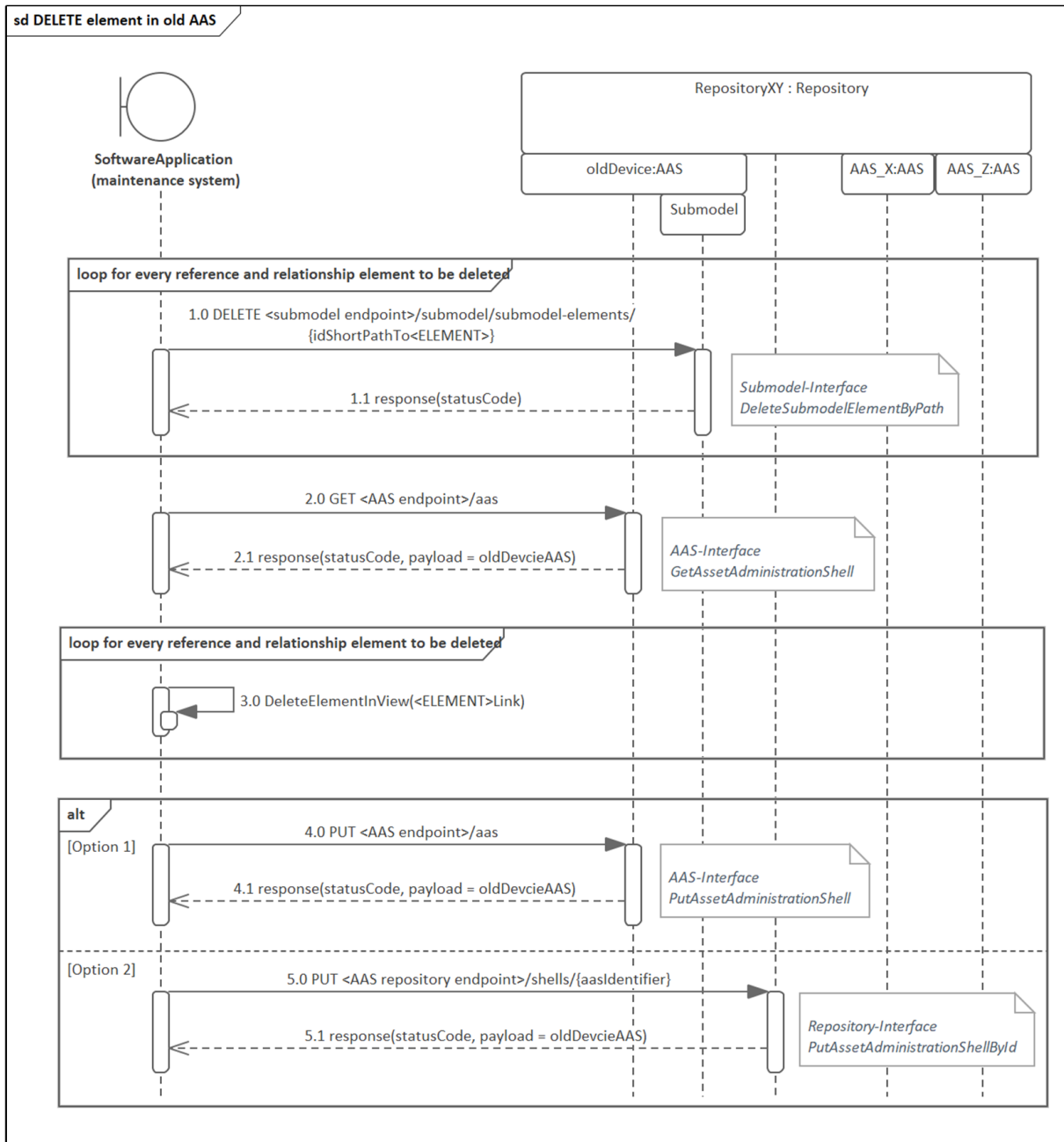


Abbildung 4.19: HTTP/REST-spezifisches Sequenzdiagramm für das Löschen eines Elements aus der AAS und dem dazugehörigen View

5 prinzipielle Umsetzung des Use Case „Dokumentationspflege bei Gerätetausch“ am LIA-I4.0-Demonstrator

Um die Allgemeingültigkeit der erstellten Konzepte für den Use Case „Dokumentationspflege bei Gerätetausch“ zu zeigen, werden im Folgenden die Konzepte aus Kapitel 4 in prinzipieller Art am LIA-I4.0-Demonstrator veranschaulicht. Das heißt, es wird nicht die direkte Implementierung beschrieben, sondern die theoretische Vorgehensweise bei einer Implementierung.

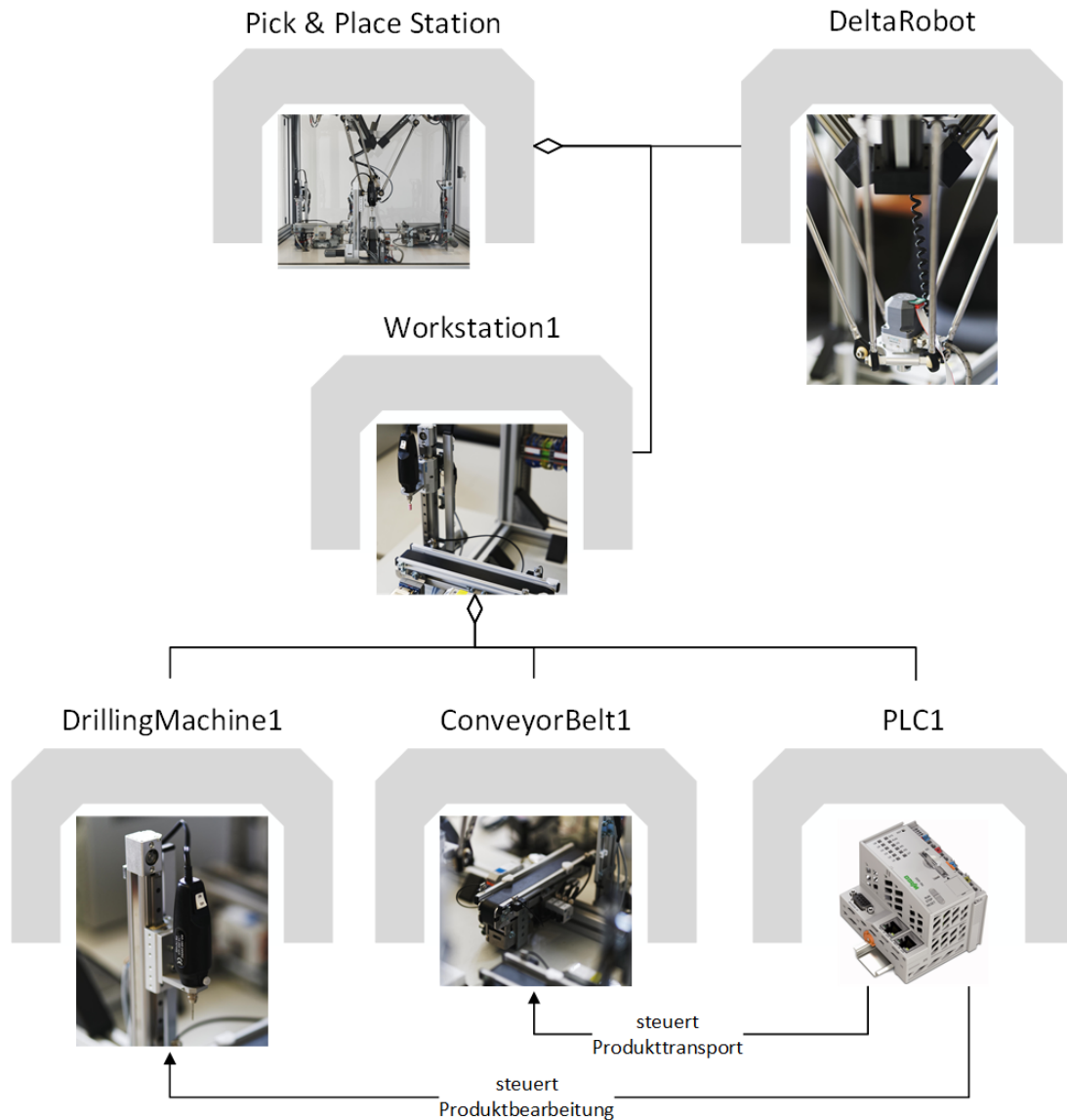
Zu Beginn dieses Kapitels wird der Ausgangspunkt beschrieben, auf dem alle weiteren Abschnitte basieren. Dazu zählt ein Modell, das die Produktionsressourcen des Demonstrators mit seinen AAS und den verwendeten Submodels abbildet. Danach folgt die Implementierung dieses Modells in dem AASX-Package-Explorer, durch eines der erarbeiteten Konzepte aus Kapitel 4.3. Abschließend wird am Beispiel eines technologiespezifischen Sequenzdiagramms aus Kapitel 4.4 ein spezifischer Gerätetausch am Demonstrator modelliert.

5.1 Ausgangssituation LIA-I4.0-Demonstrator

Der LIA-I4.0-Demonstrator realisiert eine Pick and Place Station, in der ein Produkt durch einen Deltaroboter zwischen drei Bearbeitungsstationen transportiert wird. In zwei der drei verbauten Bearbeitungsstationen wird das Produkt gebohrt und in der dritten Station gehont. Zu jeder Bearbeitungsstation gehört ein Bearbeitungswerkzeug (Bohrer bzw. Honahle) und ein Förderband, das den Transport des Produkts zu dem Bearbeitungswerkzeug ermöglicht.

Für die Verdeutlichung des Prinzips des Use Case „Dokumentationspflege bei Gerätetausch“ werden nicht alle verbauten Komponenten modelliert. Einzelne Baugruppen der Pick and Place Station werden herausgegriffen und in einem „Ausgangsmodell“ abgebildet. Die Darstellung der Beziehungen zwischen den Produktionsressourcen erfolgt unter Einbindung der Beziehungstypen aus Kapitel 4.1. Dargestellt ist das Ausgangsmodell in Abbildung 5.1. Zu sehen sind in der Grafik zum einen die Produktionsressourcen mit ihren AAS und zum anderen die Beziehungen zur Verknüpfung der Produktionsressourcen.

Insgesamt befinden sich zwei Beziehungstypen in dem Modell, und zwar der konstruktive und funktionale Beziehungstyp. Die konstruktiven Beziehungen beschreiben den Aufbau der Anlage und sind durch das Notationselement „Aggregation“ der UML dargestellt. Durch gerichtete Assoziationen gekennzeichnet sind die funktionalen Beziehungen, die zum besseren Verständnis eine Beschreibung der Funktion am Notationselement enthalten.



Bildquellen: Lehrstuhl für integrierte Automation der Otto-von-Guericke-Universität Magdeburg
www.wago.com/de

Abbildung 5.1: Ausgangsmodell des LIA-I4.0-Demonstrators

Das Ausgangsmodell kann auch durch eine Tabelle repräsentiert werden, in der jede Zeile eine Beziehung darstellt. In den Spalten der Tabelle sind Informationen über den Kontext der Beziehung, d.h. die Rolle der Beziehung im Gesamtmodell, den Beziehungstyp und die dazugehörigen Endpunkte hinterlegt. Die Endpunkte können hierbei sowohl durch Assets als auch AAS, Submodels, AAS-Elemente etc. beschrieben werden. Tabelle 5.1

zeigt die Realisierung am Beispiel des grafischen Ausgangsmodells aus Abbildung 5.1. Die Endpunkte werden in diesem Beispiel durch die zu den Assets dazugehörigen AAS gebildet, wobei der erste Endpunkt die Rolle des Subjekts und der zweite Endpunkt die Rolle des Objekts einnimmt.

Beziehungskontext	Beziehungstyp	Endpunkt 1	Endpunkt 2
Anlagenstruktur	Konstruktiv	Pick and Place Station	Workstation1
Anlagenstruktur	Konstruktiv	Pick and Place Station	DeltaRobot
Anlagenstruktur	Konstruktiv	Workstation1	DrillingMachine1
Anlagenstruktur	Konstruktiv	Workstation1	ConveyorBelt1
Anlagenstruktur	Konstruktiv	Workstation1	PLC1
Produkttransport	Funktional	PLC1	ConveyorBelt1
Produktbearbeitung	Funktional	PLC1	DrillingMachine1

Tabelle 5.1: Modellierung von Beziehungen in einer Tabelle (in Anlehnung an [27, S. 15])

Folgendes Szenario wird in den anschließenden Abschnitten prinzipiell umgesetzt:

Der Betreiber des LIA-I4.0-Demonstrators möchte die WAGO-Steuerung *PLC1* gegen ein neueres Modell mit mehr Funktionalitäten austauschen. Das neuere Modell stellt dabei alle Funktionalitäten bereit, die auch die jetzige *PLC1* besitzt. Nach dem physischen Gerätetausch, durch beispielsweise „Plug and Produce“, muss die Dokumentation auf den aktuellen Stand gebracht werden. Die AAS liegen, wie in Kapitel 3.1 gezeigt, in einem unternehmensinternen Repository. Die Dokumentationspflege übernimmt eine Softwareapplikation, die nach den Konzepten aus den vorherigen Kapiteln agiert.

Die Steuerung *PLC1* wird als Beispiel gewählt, da sie sowohl in funktionalen als auch konstruktiven Beziehungen zu anderen Assets steht und in diesen Beziehungen sowohl als Subjekt als auch als Objekt agiert. Dadurch werden die unterschiedlichen Schwerpunkte, die in den Konzepten aus Kapitel 4 versucht wurden zu vereinen, auf ihre Praktikabilität hin untersucht.

5.2 Beziehungsmodellierung im AASX-Package-Explorer

Zur Verdeutlichung des Prinzips, der Beziehungsmodellierung im AASX-Package-Explorer, werden nur die für das oben beschriebene Szenario benötigten AAS dargestellt. Alle anderen AASX-Dateien, die zur Vervollständigung des Ausgangsmodells (s. Abbildung 5.1) dienen, sind auf der beigelegten CD gespeichert. Weiterhin werden in den AAS keine weiteren Submodels wie „Technical Data“ oder „Handover Documentation“ hinterlegt, da diese für die entwickelten Konzepte keine Voraussetzung bilden und mit diesen nicht in direkter Verbindung stehen.

Die vorgegebene Struktur aus Abbildung 5.1 dient als Hilfe bei der Implementierung der Submodels und Submodel Element Collections mit ihren Elementen. In Abbildung 5.2 sind die für das Szenario erforderlichen AAS, die *Workstation1* (oberer Bildbereich), die *PLC1* (Bildmitte) sowie die *DrillingMachine1* und die AAS *ConveyorBelt1* (unterer Bildbereich), zu sehen.

Die Relationship- und Reference-Elements, mit denen die Beziehungen umgesetzt werden, sind grün eingerahmt und durch die gleichen Notationselemente wie aus Abbildung 5.1 verbunden. Die oberflächliche Unterscheidung zwischen den Relationship und Reference-Elements ist durch die am Zeilenanfang blau hinterlegten Abkürzungen *Rel* und *Ref* möglich.

Es ist zu erkennen, dass sich genau zwei Relationship-Elements in dem Submodel *controlledDevices* der AAS *PLC1* befinden. Diese realisieren die zwei funktionalen Beziehungen aus Tabelle 5.1, in denen die *PLC1* die Rolle des Subjekts einnimmt. Des Weiteren ist die konstruktive Beziehung zu der übergeordneten *Workstation1* mit Hilfe des Reference-Elements *Workstation1ConsistsOfPLC1* im Submodel *plantStructure* implementiert.

Zusätzlich sind in der AAS *PLC1* mit roten Pfeilen die internen Verweise zwischen den AAS-Elementen gekennzeichnet. Damit wird zum einen veranschaulicht, auf welche AAS-Elemente die Referables der Views zeigen. Zum anderen verdeutlichen die Pfeile die Verbindung zwischen der Entity und dem auf sie zeigenden Relationship-Element.

Einen detaillierteren Einblick in die Daten der Relationship- und Reference-Elemente sowie der Entity und den Referables der Views geben die folgenden Abbildungen. Nummernkreise in den Abbildungen erleichtern die Nachvollziehbarkeit der Erläuterungen und werden im Text (kursiv und in Klammern gesetzt) hervorgehoben.

Begonnen wird mit den Informationen die das Relationship-Element und seine damit zusammenhängenden AAS-Elemente bereitstellen. Das Relationship-Element *PLC1ControlsDrillingMachine1 (1)* aus Abbildung 5.3 dient als Beispiel. Es besitzt eine *idShort* sowie eine *semanticId* und einen *Qualifier*. Die *semanticId Relationship* sagt aus, dass es sich bei dem Relationship-Element um die Implementierung einer Beziehung handelt. Das ermöglicht die Verwendung der Relationship-Elemente in Submodels auch in anderen Kontexten, die außerhalb dieses Use Case liegen. Des Weiteren wird durch die Verwendung des Relationship-Elements verdeutlicht, dass das Asset in der Beziehung die Rolle des Subjekts einnimmt. Die Einordnung des Beziehungskontextes erfolgt durch den Wert des *Qualifiers*, in diesem Fall *productProcessing*.

Danach folgt die Angabe der Endpunkte des Relationship-Elements. Der Wert des ersten Endpunkts wird durch die Konzepte aus Kapitel 4.3 mit der Asset-ID des zur AAS gehörenden Assets beschrieben. In dem Beispiel ist der erste Endpunkt die Asset-ID vom Asset *PLC1* (gelb markiert).

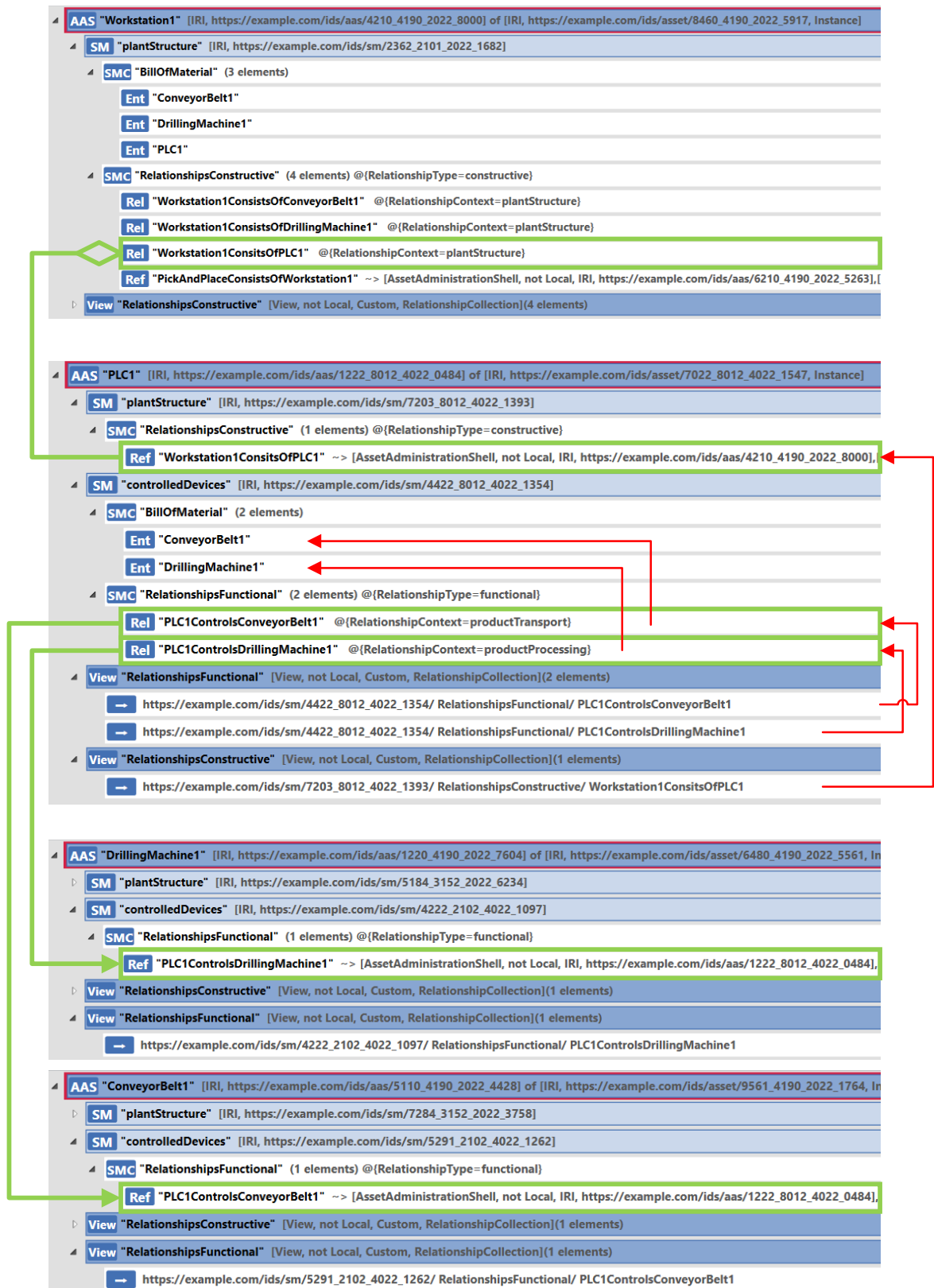


Abbildung 5.2: Übersicht über die Implementierung des konzeptionellen AAS-Aufbaus im AASX-Package-Explorer

The image shows a tree view of AAS elements on the left and three detailed panels on the right. Red circles with numbers 1, 2, and 3 highlight specific elements in both views.

Tree View Elements:

- AAS "PLC1" [IRI, https://example.com/ids/aas/1222_8012_4022_0484] of [IRI, https://example.com/ids/asset/7022_8012_4022_1547, Instance]
 - SM "plantStructure" [IRI, https://example.com/ids/sm/7203_8012_4022_1393]
 - SMC "RelationshipsConstructive" (1 elements) @({RelationshipType=constructive})
 - Ref "Workstation1ConsistsOfPLC1" ~-> [AssetAdministrationShell, not Local, IRI, https://example.com/ids/aas/4210_4190_2022_8000]
 - SM "controlledDevices" [IRI, https://example.com/ids/sm/4422_8012_4022_1354]
 - SMC "BillOfMaterial" (2 elements)
 - Ent "ConveyorBelt1"
 - Ent "DrillingMachine1" **2**
 - SMC "RelationshipsFunctional" (2 elements) @({RelationshipType=functional})
 - Rel "PLC1ControlsConveyorBelt1" @({RelationshipContext=productTransport})
 - Rel "PLC1ControlsDrillingMachine1" @({RelationshipContext=productProcessing}) **1**
 - View "RelationshipsFunctional" [View, not Local, Custom, RelationshipCollection](2 elements)
 - https://example.com/ids/sm/4422_8012_4022_1354/ RelationshipsFunctional/ PLC1ControlsConveyorBelt1
 - https://example.com/ids/sm/4422_8012_4022_1354/ RelationshipsFunctional/ PLC1ControlsDrillingMachine1 **3**
 - View "RelationshipsConstructive" [View, not Local, Custom, RelationshipCollection](1 elements)
 - https://example.com/ids/sm/7203_8012_4022_1393/ RelationshipsConstructive/ Workstation1ConsistsOfPLC1

Submodel Element (RelationshipElement) 1	
Referable:	
idShort:	PLC1ControlsDrillingMachine1
Semantic ID:	
semanticId:	(RelationshipElement) (no-local) [Custom] Relationship
Qualifiable:	
Qualifier 1	
type:	RelationshipContext
value:	productProcessing
RelationshipElement	
first:	(Asset) (local) [IRI] https://example.com/ids/asset/7022_8012_4022_1547
second:	(Submodel) (local) [IRI] https://example.com/ids/sm/4422_8012_4022_1354 (SubmodelElementCollection) (local) [IdShort] BillOfMaterial (Entity) (local) [IdShort] DrillingMachine1

Submodel Element (Entity) 2	
Referable:	
idShort:	DrillingMachine1
Semantic ID:	
semanticId:	(ConceptDescription) (no-local) [IRI] http://admin-shell.io/sandbox/CompositeComponent/General/IsPartC
Qualifiable:	
Entity	
# of statements:	0
entityType:	SelfManagedEntity
Asset:	(Asset) (no-local) [IRI] https://example.com/ids/asset/6480_4190_2022_5561

Reference (containedElement) of View 3	
containedElement:	(Submodel) (local) [IRI] https://example.com/ids/sm/4422_8012_4022_1354 (SubmodelElementCollection) (local) [IdShort] RelationshipsFunctional (RelationshipElement) (local) [IdShort] PLC1ControlsDrillingMachine1

Abbildung 5.3: Detaillierte Darstellung der Daten für ein Relationship-Element und der mit diesem in Verbindung stehenden AAS-Elemente

Der zweite Endpunkt verweist auf die Entity, die das Objekt in dieser Beziehung indirekt verlinkt. Die Entity befindet sich in demselben Submodel wie das Relationship-Element. Die drei untergeordneten Key-Value-Pairs des zweiten Endpunkts (s. Abbildung 5.3) geben den Speicherort der Entity an. Über die Submodel-ID (gelb markiert), dem ersten Key-Value-Pair, ist die eindeutige Identifizierung des dazugehörigen Submodels gewährleistet (im oberen Bildbereich gelb markiert). Mit den zusätzlichen idShort's wird ein relativer Pfad generiert, der in Verknüpfung mit der Submodel-ID die Adressierung der Entity ermöglicht.

Die Entity selbst, in diesem Beispiel *DrillingMachine1*, enthält ebenfalls eine idShort und eine semanticId (2). In diesem Fall hängt die semanticId *isPartOfForBOM* mit der übergeordneten Submodel Element Collection *BillOfMaterial* zusammen und sagt aus, dass die Entity Teil einer Stückliste ist (vgl. [29, S. 30–34]). Ein Qualifier wird nicht definiert, da die Entity keine Beziehung darstellt und die Qualifier für diesen Use Case nur zur Beschreibung von Beziehungstypen bzw. -kontexten dienen. Zusätzlich ist im unteren Bereich der Entity zu sehen, dass es sich bei der *DrillingMachine1* um eine self-managed Entity mit der darunter angegebenen Asset-ID handelt.

Abschließend wird für das Relationship-Element ein Referable im View *RelationshipsFunctional* (3) hinzugefügt. Über die drei untergeordneten Key-Value-Pairs im Referable ist die eindeutige Referenzierung des Relationship-Elements gewährleistet. Durch die Suche der Submodel-ID

`https://example.com/ids/sm/4422_8012_4022_1354`

im AAS-Aufbau (oberer Bildbereich), kann das dazugehörige Submodel *controlledDevices* gefunden werden (gelb markiert). Mit der Verfolgung des idShort-Pfads

`RelationshipFunctional/PLC1ControlsDrillingMachine1`

gelangt man am Ende zu dem Relationship-Element aus diesem Beispiel.

Hinweis: Wie bereits in Kapitel 4.3 erwähnt, werden die semanticId und der Qualifier mit abstrakten Begriffen beschrieben, für die bisher keine semantische Definition vorliegt.

Im Folgenden wird das Reference-Element betrachtet. Als Beispiel wird das in Abbildung 5.4 abgebildete Reference-Element *Workstation1ConsistsOfPLC1* (1) der AAS *PLC1* gewählt. Das Reference-Element ist ähnlich dem Relationship-Element aufgebaut. Die semanticId *Reference* beschreibt, dass das Reference-Element auf eine bestehende Beziehung referenziert und in diesem Fall die AAS bzw. das dazugehörige Asset die Rolle des Objekts in der Beziehung einnimmt. Dadurch ist die Verwendung des Reference-Elements, wie bereits bei dem Relationship-Element erwähnt, in anderen Kontexten gewährleistet. Der *value* des Reference-Elements wird durch Key-Value-Pairs gebildet, die auf die AAS und

das Submodel, in der das zur Beziehung zugehörige Relationship-Element liegt, verweisen. Ein zusätzlicher View zur Filterung der konstruktiven Beziehungstypen enthält das entsprechende Referable (2).

The screenshot shows a tree view of AAS elements. The root element is 'AAS "PLC1"'. It contains several submodel elements (SM) and submodel collections (SMC). A red circle '1' highlights the 'Ref "Workstation1ConsitsOfPLC1"' element. Another red circle '2' highlights the 'View "RelationshipsConstructive"' element.

Submodel Element (ReferenceElement)

Referable:
 idShort: Workstation1ConsitsOfPLC1
 description: [en] Reference to Relationship of Workstation1

Semantic ID:
 semanticId: (ReferenceElement) (no-local) [Custom] Reference

Qualifiable:

Qualifier 1
 type: RelationshipContext
 value: plantStructure

ReferenceElement
 value:
 (AssetAdministrationShell) (no-local) [IRI] https://example.com/ids/aas/4210_4190_2022_8000
 (Submodel) (no-local) [IRI] https://example.com/ids/sm/2362_2101_2022_1682
 (SubmodelElementCollection) (local) [IdShort] RelationshipsConstructive
 (RelationshipElement) (local) [IdShort] Workstation1ConsitsOfPLC1

Reference (containedElement) of View
 containedElement: (Submodel) (local) [IRI] https://example.com/ids/sm/7203_8012_4022_1393
 (SubmodelElementCollection) (local) [IdShort] RelationshipsConstructive
 (ReferenceElement) (local) [IdShort] Workstation1ConsitsOfPLC1

Abbildung 5.4: Detaillierte Darstellung der Daten für ein Reference-Element und der mit diesem in Verbindung stehenden AAS-Elemente

Als drittes und letztes wird ein Einblick in die Metainformationen der erstellten Submodels, Submodel Element Collections und Views gegeben. Die Daten dazu sind in Abbildung 5.5 zu sehen. Zur Erklärung werden einzelne Exemplare der genannten AAS-Modellelemente herangezogen.

The image shows a hierarchical tree view of a model structure on the left, with three detailed data panels on the right. Red circles with numbers 1, 2, and 3 indicate the correspondence between the tree nodes and the data panels.

Tree View:

- AAS "PLC1" [IRI, https://example.com/ids/aas/1222_8012_4022_0484] of [IRI, https://example.com/ids/asset/7022_8012_4022_1547, Instance]
 - SM "plantStructure" [IRI, https://example.com/ids/sm/7203_8012_4022_1393]
 - SMC "RelationshipsConstructive" (1 elements) @({RelationshipType=constructive})
 - Ref "Workstation1ConsistsOfPLC1" ~-> [AssetAdministrationShell, not Local, IRI, https://example.com/ids/aas/4210_4190_2022_8000]
 - SM "controlledDevices" [IRI, https://example.com/ids/sm/4422_8012_4022_1354] **1**
 - SMC "BillOfMaterial" (2 elements)
 - Ent "ConveyorBelt1"
 - Ent "DrillingMachine1"
 - SMC "RelationshipsFunctional" (2 elements) @({RelationshipType=functional}) **2**
 - Rel "PLC1ControlsConveyorBelt1" @({RelationshipContext=productTransport})
 - Rel "PLC1ControlsDrillingMachine1" @({RelationshipContext=productProcessing})
 - View "RelationshipsFunctional" [View, not Local, Custom, RelationshipCollection](2 elements)
 - https://example.com/ids/sm/4422_8012_4022_1354/ RelationshipsFunctional/ PLC1ControlsConveyorBelt1
 - https://example.com/ids/sm/4422_8012_4022_1354/ RelationshipsFunctional/ PLC1ControlsDrillingMachine1
 - View "RelationshipsConstructive" [View, not Local, Custom, RelationshipCollection](1 elements)
 - https://example.com/ids/sm/7203_8012_4022_1393/ RelationshipsConstructive/ Workstation1ConsistsOfPLC1 **3**

SubmodelReference (1)

SubmodelReference
submodelRef: (Submodel) (local) [IRI] https://example.com/ids/sm/4422_8012_4022_1354
Submodel
Referable:
idShort: controlledDevices
Identifiable:
idType: IRI
id: https://example.com/ids/sm/4422_8012_4022_1354
Semantic ID:
semanticId: (Submodel) (no-local) [Custom] SM_controlledDevices
Qualifiable:

Submodel Element (SubmodelElementCollection) (2)

Submodel Element (SubmodelElementCollection)
Referable:
idShort: RelationshipsFunctional
Semantic ID:
semanticId: (SubmodelElementCollection) (no-local) [Custom] RelationshipCollection
Qualifiable:
Qualifier 1
type: RelationshipType
value: functional

View (3)

View
of containedElements: 2
Referable:
idShort: RelationshipsFunctional
description: [en] Contains the references to all relationships of type functional.
Semantic ID:
semanticId: (View) (no-local) [Custom] RelationshipCollection

Abbildung 5.5: Detaillierte Darstellung der Daten für Submodel, Submodel Element Collection und View

Das Submodel *controlledDevices* (1) enthält neben der *idShort* und einer global eindeutigen ID auch eine *semanticId*. Die *semanticId* ist wieder abstrakt gewählt und folgt dem Schema

SM_<idShortOfSubmodel>.

Durch die Verwendung der Submodel-*idShort* in der *semanticId* muss für jedes Submodel, das die Datenstruktur zur Beziehungsmodellierung enthält, eine eigene semantische Beschreibung existieren. Der Grund für die Namenslegung liegt darin, dass in dem beispielhaften Submodel *controlledDevices* nicht nur die zwei Submodel Element Collections *BillOfMaterial* und *RelationshipsFunctional* vorkommen müssen. Vielmehr können weitere Submodel Element Collections neben den bereits genannten existieren, die nicht im Zusammenhang mit der Beziehungsmodellierung stehen aber kontextuell in das Submodel gehören. Mit der Option, solche Erweiterungen zuzulassen, steigt der Implementierungsaufwand. Tritt ein solches Submodel jedoch in mehreren AAS auf, kann es mit seiner semantischen Beschreibung als Template fungieren und somit den Aufwand mindern.

Sollen hingegen die Submodels für die Beziehungsmodellierung abgegrenzt von weiteren Informationen sein und rein für die Implementierung der Beziehungen bereitstehen, ist die Verwendung einer allgemeinen *semanticId* denkbar. So eine *semanticId* könnte *SM_Beziehungsmodellierung* heißen und einer von extern darauf zugreifenden Softwareapplikation vermitteln, dass sich in dem Submodel alle Informationen zur Modellierung der darin vorhandenen Beziehungen befinden.

Die dem Submodel untergeordneten Submodel Element Collections (2), zur Verwaltung der Relationship- und Reference-Elemente, erhalten die allgemeine *semanticId* *RelationshipCollection*. Diese *semanticId* kann für jede Submodel Element Collection Anwendung finden, sofern sie für die Implementierung der in den Konzepten beschriebenen Beziehungen vorgesehen ist. Um allerdings die Unterscheidung zwischen den verschiedenen Beziehungstypen für eine Softwareapplikation zu ermöglichen, wird der Submodel Element Collection ein Qualifier hinzugefügt. Dieser Qualifier beschreibt den Beziehungstyp, der aus einer vorgegebenen Menge an Werten zu wählen ist. In diesem Beispiel ist der Wert des Beziehungstyps *functional*. Die Menge an Beziehungstypen gilt es in zukünftigen Arbeiten zu definieren und durch ein standardisiertes Vokabular bereitzustellen. Für die Submodel Element Collection *BillOfMaterial* existiert bereits eine semantische Beschreibung (vgl. [29, S. 30–34]).

Der View, der die Elemente aus den Submodel Element Collections referenziert, enthält weniger Metainformationen als die oben beschriebenen AAS-Modellelemente. Da ein Qualifier als Attribut für den View im AAS-Metamodell nicht definiert ist, muss eine andere Lösung für die eindeutige semantische Unterscheidung der Views für die verschiedenen Beziehungstypen entwickelt werden. Ein Lösungsansatz in dieser Arbeit ist die Kombination von *semanticId* und *idShort* zur eindeutigen semantischen Bestimmung. Die *semanticId*, mit

dem Wert *RelationshipCollection*, wird von der Submodel Element Collection übernommen, da der View ähnlich der Submodel Element Collection Beziehungen zusammenfasst. Die *idShort* setzt sich nach dem Schema

Relationships<*relationshipType*>

zusammen. Durch die Verknüpfung der *semanticId* mit der *idShort* nach dem vorgegebenen Schema, ist ein eindeutiges Suchkriterium für eine externe Softwareapplikation gegeben. Da die Konzepte nur einen View je Beziehungstyp in einer AAS vorsehen, wird eine Dopplung und somit eine mögliche Fehlerquelle für das Suchkriterium ausgeschlossen.

Eine nicht zu unterschätzende Fehlerquelle, die in dieser Realisierung nicht auszuschließen ist, ist der menschliche Aspekt. Solange die *idShort*'s von Hand eingetragen werden, kann die Softwareapplikation bei Tippfehlern die volle Funktionalität nicht mehr gewährleisten. Weiterhin ist diese Lösung sehr starr in ihrer Struktur, da in jeder AAS die *idShort* des jeweiligen View einheitlich bezeichnet werden muss.

Eine andere Möglichkeit ist die Entwicklung einer *semanticId* für jeden View vom Beziehungstyp *xy*, die in der Abbildung nicht dargestellt ist. Das würde in der Organisation, die diese *semanticIds* spezifizieren müsste, zu einem höheren Arbeitsaufwand führen. Der Aufwand wäre jedoch einmalig und würde die Bereitstellung der *semanticIds* im Anwendungsbereich für jeden View ermöglichen. Damit wird eine weitaus weniger fehleranfällige und flexiblere Methode realisiert, da nur eine begrenzte und definierte Menge an *semanticIds* zur Auswahl stünden.

Während der Umsetzung des Ausgangsmodells aus Kapitel 5.1 im AASX-Package-Explorer zeigt sich der immense Arbeitsaufwand hinter den in Kapitel 4 entwickelten Konzepten. Für das Anlegen einer Beziehung sind ungefähr drei bis acht Implementierungsschritte je nach Ausgangszustand notwendig, die mit hoher Konzentration ausgeführt werden müssen. In Bezug auf eine kleinere Anlage mit mehreren hundert Beziehungen ist der Arbeitsaufwand für die AAS-Entwickler nicht rentabel. Ist nur ein Endpunkt falsch definiert oder wird eine bereits referenzierte Submodel-ID in einer AAS geändert führt das zu schwerwiegenden Fehlern, die eine aufwendige Fehlersuche und -behebung nach sich ziehen.

Eine Lösung für die Umgehung der direkten Implementierung über den AASX-Package-Explorers wäre der Einsatz einer graphischen Benutzeroberfläche. Diese würde dem Anwender die graphische Modellierung ähnlich dem in Abbildung 5.1 dargestellten Ausgangsmodells ermöglichen. Daraufhin übersetzt eine Software, die sich hinter der Benutzeroberfläche befindet, die Informationen mit Hilfe von AAS-Spezifikationen automatisch in eine AASX-Datei. Diese Vorstellung ist allerdings zukunftsorientiert und findet möglicherweise in nachfolgenden Forschungsarbeiten ihren Platz.

5.3 Prinzipieller Datenaustausch im LIA-I4.0-Demonstrator

Mit den in Kapitel 5.2 entwickelten AAS können die verallgemeinerten technologiespezifischen Sequenzdiagramme aus Kapitel 4.4 auf den Demonstrator zugeschnitten werden. Zur Darstellung des Prinzips genügt die Umsetzung eines der Sequenzdiagramme, da die Vorgehensweise bei der Anpassung der Sequenzdiagramme die selbe bleibt. Das Sequenzdiagramm, das in diesem Kapitel näher betrachtet wird, ist die Aktualisierung eines Reference-Elements in einer der verknüpften AAS (s. Abbildung 4.16 auf Seite 113). Im Fall des Gerätetausches von der Steuerung *PLC1* heißt das, dass entweder das Reference-Element in der AAS *DrillingMachine1* oder in der AAS *ConveyorBelt1* mit Hilfe des dazugehörigen Relationship-Elements aus der AAS *PLC1* aktualisiert wird (s. Abbildung 5.2).

Als Fallbeispiel fällt die Wahl auf das Reference-Element *PLC1ControlsDrillingMachine1*. Unter dieser Vorgabe ist zur Erfüllung des Sequenzdiagramms zuerst die Betrachtung des dazugehörigen Relationship-Elements *PLC1ControlsDrillingMachine1* aus der AAS *PLC1* erforderlich. Davor ist wiederum die erfolgreiche Abarbeitung der vorhergehenden Sequenzen aus Kapitel 4.4 festzustellen. Mit dieser Voraussetzung können folgende Annahmen für die Abarbeitung der gewählten Sequenz getroffen werden:

- Alle Entities, Relationship- und Reference-Elemente wurden bereits in die AAS *PLC1.1* der neuen Steuerung kopiert.
- Im Speicher der Softwareapplikation befinden sich bereits das Relationship-Element *PLC1ControlsDrillingMachine1* sowie die Entity *DrillingMachine1* aus der AAS *PLC1*.
- Der Softwareapplikation sind die Submodel-ID und der idShort-Pfad, in dem sich das Relationship-Element befindet, bekannt.
- An der Interaktion sind neben der Softwareapplikation die AAS *DrillingMachine1* sowie eine Registry beteiligt.

Hinweis: Die AAS der neuen Steuerung wird mit der idShort *PLC1.1* zur besseren Unterscheidung gekennzeichnet. In der Praxis sollte die idShort der AAS des alten Geräts vorzugsweise mit der des neuen Geräts übereinstimmen. Das ist vor allem möglich, wenn man als idShort die Betriebsmittelkennzeichen (BMK) aus den vorhandenen Stromlaufplänen (bzw. Mess- und Regelstellen aus R&I-Fließschemata) übernimmt. Das BMK eines Gerätetypen ist beständig, lediglich die Informationen die mit dem BMK verbunden sind ändern sich.

Mit den oben genannten Informationen ist die Anpassung der Sequenz, zur Aktualisierung des Reference-Elements, auf den LIA-I4.0-Demonstrator möglich. Die mit den Parametern des LIA-I4.0-Demonstrators angepassten Sequenzdiagramme aus Kapitel 4.4 sind nochmals abgebildet. Weiterhin sind die Codezeilen in JSON aufgeführt, die als Informationsquelle zur Abarbeitung der Sequenzen dienen. Für die einfachere Zuordnung der Parameter aus den JSON-Dateien zu den Sequenzdiagrammen sind diese farblich hervorgehoben. Des Weiteren werden zur Verfolgung der Erläuterungen die Nachrichtennummern aus den Sequenzdiagrammen im Text (kursiv und in Klammern) einbezogen.

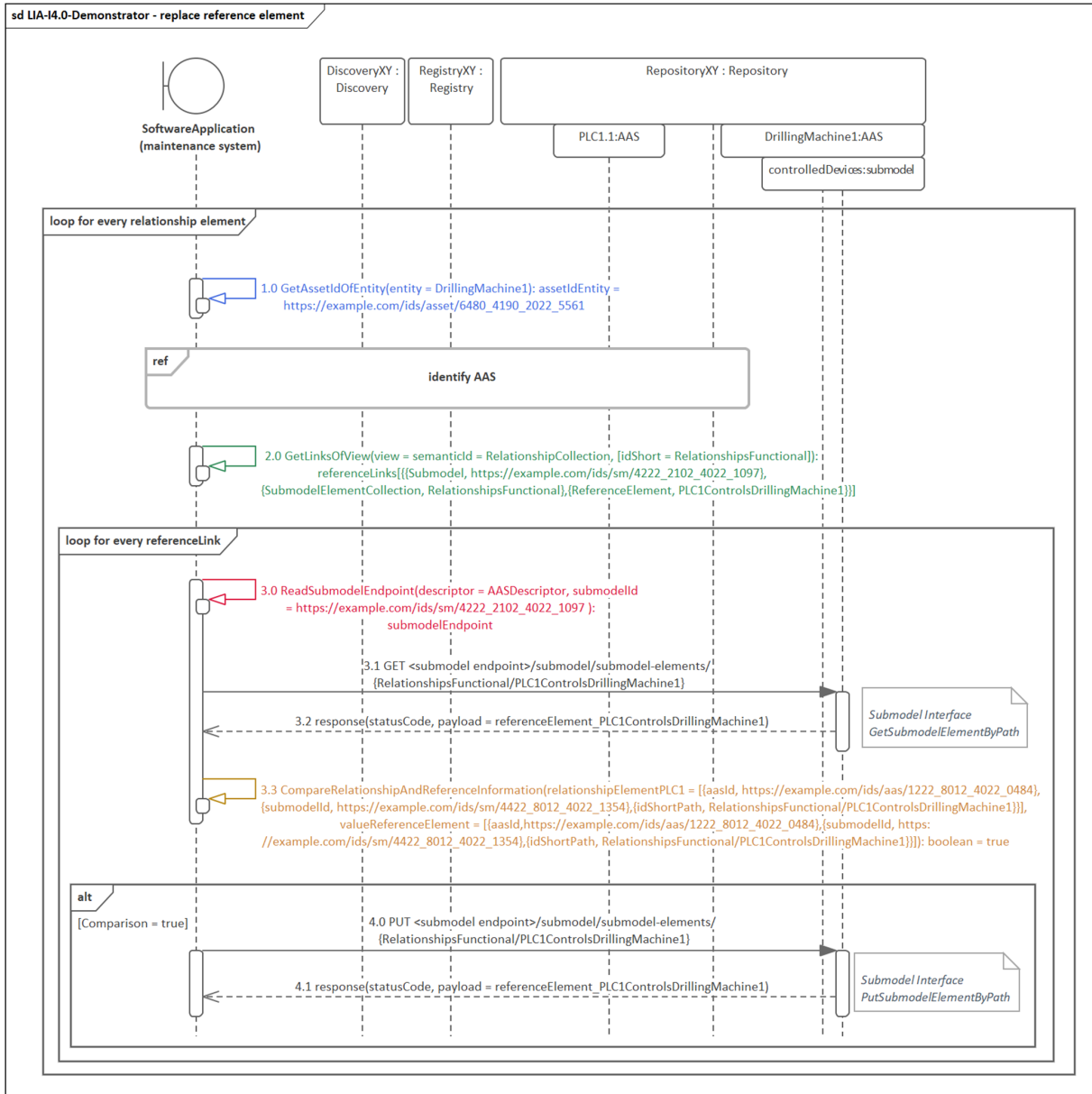


Abbildung 5.6: Technologiespezifisches Sequenzdiagramm für das Aktualisieren eines vorgegebenen Reference-Elements am Beispiel des LIA-I4.0-Demonstrators

In Abbildung 5.6 ist das angepasste Sequenzdiagramm, zur Aktualisierung des Reference-Elements *PLC1ControlsDrilling-Machine1*, zu sehen. In der äußersten Schleife der Sequenz wird das gleichnamige Relationship-Element, das sich in der AAS *PLC1* befindet, abgearbeitet. Die zum Relationship-Element gespeicherten Informationen, wie die Metainformationen der AAS *PLC1*, des Submodels *controlledDevices* und der Submodel Element Collection *RelationshipsFunctional* sind in Abbildung 5.7 dargestellt. Mit diesem JSON-Code sind bereits die Informationen (in ocker hervorgehoben) bekannt, die für Nachricht (3.3) notwendig sind. Bevor auf diese Nachricht jedoch eingegangen wird, folgt zuvor die Abarbeitung der Nachrichten (1.0) bis (3.2).

Für Nachricht (1.0) muss die Softwareapplikation aus der Entity *DrillingMachine1* die Asset-ID extrahieren. Dazu erhält die Applikation die Entity als JSON-Objekt. Codeauschnitte des JSON-Objekts sind in Abbildung 5.8 aufgeführt, wobei die zu extrahierende Asset-ID blau hervorgehoben ist.

<pre> JSON-Codeausschnitt der Metainformationen des Relationship-Elements PLC1ControlsDrillingMachine1 aus der AAS PLC1 { "assetAdministrationShells": [{ "identification": { "idType": "IRI", "id": "https://example.com/ids/aas/.. ..1222_8012_4022_0484" }, "idShort": "PLC1", "modelType": { "name": "AssetAdministrationShell" }, "submodels": [{ "semanticId": { "keys": [{ "type": "Submodel", "value": "SM_controlledDevices", "idType": "Custom" }] }, "qualifiers": [], "identification": { "idType": "IRI", "id": "https://example.com/ids/sm/.. ..4422_8012_4022_1354" }, "idShort": "controlledDevices", "modelType": { "name": "Submodel" } }] }] } </pre>	<pre> .. "submodelElements": [{ "semanticId": { "keys": [{ "type": "SubmodelElementCollection", "value": "RelationshipCollection", "idType": "Custom" }] }, "constraints": [{ "type": "RelationshipType", "value": "functional", "modelType": { "name": "Qualifier" } }], "idShort": "RelationshipsFunctional", "modelType": { "name": "SubmodelElementCollection" }, "value": [{ "idShort": "PLC1ControlsDrillingMachine1", "modelType": { "name": "RelationshipElement" } }] }] } </pre>
---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Abbildung 5.7: JSON-Codeausschnitt der Metainformationen zum Relationship-Element *PLC1ControlsDrillingMachine1* aus der AAS *PLC1*


```
JSON-Codeausschnitt der Entity DrillingMachine1 aus der
AAS PLC1

{
  "entityType": "SelfManagedEntity",
  "asset": {
    "keys": [
      {
        "type": "Asset",
        "value": "https://example.com/ids/asset/..
                ..6480_4190_2022_5561",
        "idType": "IRI"
      }
    ]
  },
  "idShort": "DrillingMachine1"
}
```

Abbildung 5.8: JSON-Codeausschnitt der Entity *DrillingMachine1* aus der AAS *PLC1*

Mit der Asset-ID ist im nächsten Schritt die Identifizierung der dazugehörigen AAS realisierbar, bei der unter anderem drei Optionen zur Verfügung stehen (s. Abbildung 4.12 S. 104). Die Wahl fällt hierbei auf Option drei, da zur Aktualisierung des Reference-Elements der AAS Descriptor der AAS *DrillingMachine1* notwendig ist (s. (3.0) aus Abbildung 5.6). Die an das Fallbeispiel angepasste Sequenz, zur Identifizierung der AAS, ist in Abbildung 5.9 zu sehen. Darin wird zu Beginn mit Hilfe der Asset-ID der Entity *DrillingMachine1* die dazugehörige AAS-ID von der Discovery angefordert (1.0). Die Informationen, die von der Discovery bereitgestellt werden (1.1), stammen aus den in Abbildung 5.10 aufgeführten Metainformationen der AAS *DrillingMachine1*. Mit der in Abbildung 5.10 orange hervorgehoben AAS-ID ist die anschließende Aushändigung des AAS Descriptors durch die Registry möglich (1.2), (1.3). Der JSON-Codeausschnitt aus dem AAS-Descriptor ist ebenfalls in Abbildung 5.10 dargestellt. Aus diesem AAS Descriptor kann die Softwareapplikation den notwendigen Endpunkt der AAS, zur direkten Kommunikation mit dieser, lesen (1.4). Zuletzt fordert die Softwareapplikation die AAS als JSON-Objekt über das AAS-Interface an. Danach folgt der Wechsel zurück zu der Sequenz aus Abbildung 5.6.

Hinweis: Der Endpunkt im AAS Descriptor der Abbildung 5.10 enthält in diesem Beispiel keinen definierten Wert, da er je nach Implementierung variieren kann. Stattdessen wird der Platzhalter „string“ beibehalten.

Aus dem gewonnenen JSON-Objekt der AAS *DrillingMachine1* ist das Lesen der Referables aus den Views möglich (2.0). Es wird der View übergeben, dessen idShort *Relationships-Functional* ist. Da das Relationship- als auch das dazugehörige Reference-Element beide vom gleichen Beziehungstyp sein müssen, ist das Auffinden des Referable im View der AAS *DrillingMachine1* garantiert.

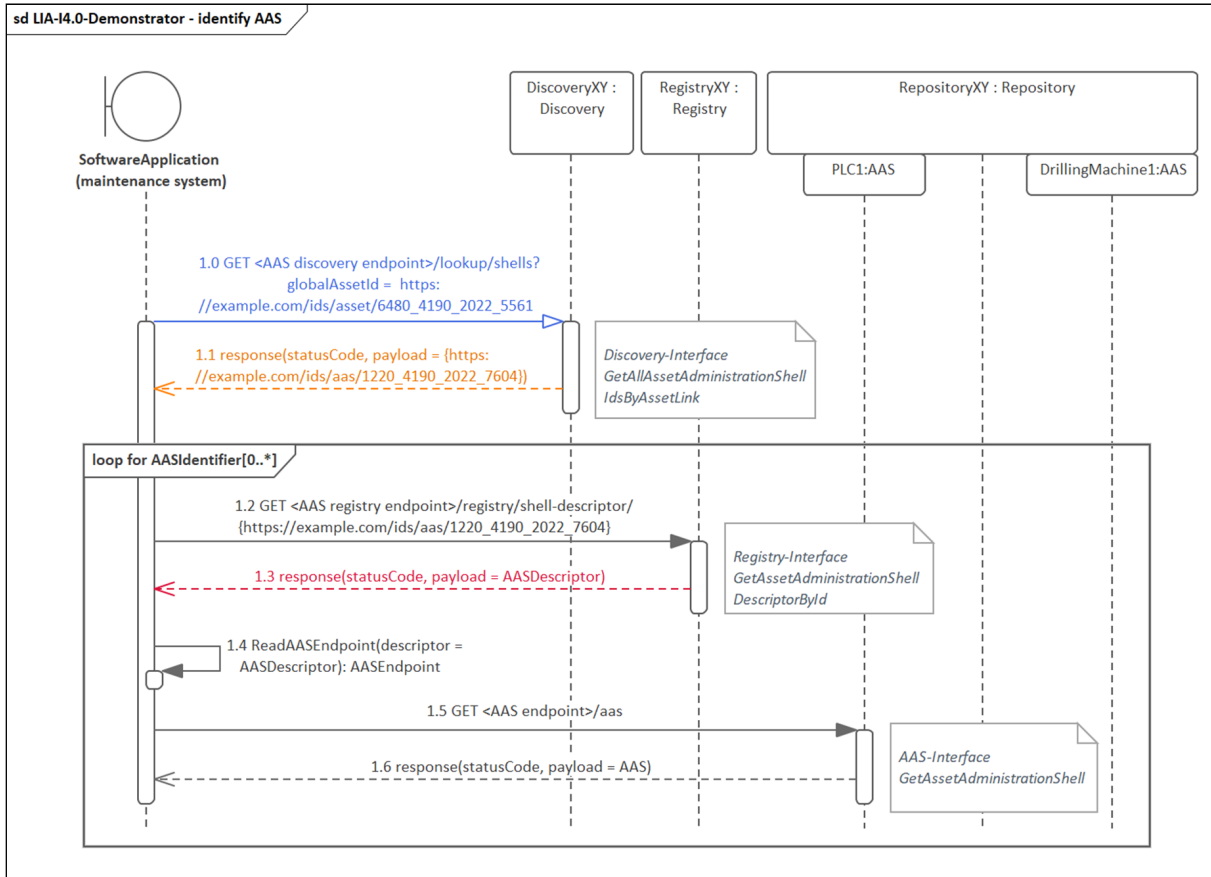


Abbildung 5.9: Technologiespezifisches Sequenzdiagramm zur Identifizierung der AAS *DrillingMachine1* des LIA-I4.0-Demonstrators



Abbildung 5.10: JSON-Codeausschnitt der Metainformationen von der AAS *DrillingMachine1* (links) und deren AAS Descriptor (rechts)

Bei der Übergabe des View erhält die Softwareapplikation eine Liste mit allen Referables des View, die sowohl auf Relationship- als auch auf Reference-Elemente zeigen. Die Operation (2.0) sucht hierbei nur die Reference-Elemente heraus und speichert diese in einem dreidimensionalen Array ab. In dem Array wird jedes Reference-Element durch einen Block abgebildet, der aus jeweils drei Zeilen und zwei Spalten besteht. Die Spalten definieren den Typ des AAS-Elements und seinen dazugehörigen Wert, während die Zeilen jeweils ein Key-Value-Pair abbilden. Diese Informationen entnimmt die Softwareapplikation aus dem Referable des View.

<pre> JSON-Codeausschnitt des View <i>RelationshipsFunctional</i> der AAS <i>DrillingMachine1</i> "views": [{ "semanticId": { "keys": [{ "type": "View", "value": "RelationshipCollection", "idType": "Custom" }] }, "idShort": "RelationshipsFunctional", "modelType": { "name": "View" }, "containedElements": [{ "keys": [{ "type": "Submodel", "local": true, "value": "https://example.com/ids/sm/.. ..4222_2102_4022_1097", "index": 0, "idType": "IRI" }] }] }] </pre>	<pre> .. { "type": "SubmodelElementCollection", "local": true, "value": "RelationshipsFunctional", "index": 1, "idType": "IdShort" }, { "type": "ReferenceElement", "local": true, "value": "PLC1ControlsDrillingMachine1", "index": 2, "idType": "IdShort" }] </pre>
------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------	--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------

Abbildung 5.11: JSON-Codeausschnitt des View *RelationshipsFunctional* der AAS *DrillingMachine1*

In Abbildung 5.11 ist das Referable aus dem View *RelationshipsFunctional*, das auf das Reference-Element *PLC1ControlsDrillingMachine1* zeigt, dargestellt. Grün hervorgehoben sind die Informationen, die für Nachricht (2.0) notwendig sind. Da nur ein Referable in dem View existiert, wird von der Nachricht ein Array mit nur einen Eintrag ausgegeben. Die sich anschließende Schleife in der Sequenz aus Abbildung 5.6 wird dadurch nur einmalig durchlaufen.

In dieser eben genannten Schleife wird den Links zu den Reference-Elementen gefolgt. Dafür werden zu Beginn der AAS Descriptor sowie die Submodel-ID aus dem Array der Nachricht (2.0) an die Softwareapplikation übergeben. Damit ist die Softwareapplikation in der Lage, den AAS Descriptor nach der übergebenen Submodel-ID zu durchsuchen

und dem zum Submodel gehörenden Endpunkt auszugeben (3.0). Mit dem Endpunkt und der aus dem Array von Nachricht (2.0) gegebenen idShorts, die zu einem idShort-Pfad umgewandelt werden, ist das im Submodel *controlledDevices* befindliche Reference-Element erreichbar (3.1). Das Submodel-Interface gibt, der Operation *GetSubmodelElementByPath* entsprechend, das Reference-Element als JSON-Objekt zurück (3.2). Der Code-Ausschnitt des Reference-Elements aus der AAS *DrillingMachine1* ist in Abbildung 5.12, mit den wichtigsten Informationen in ocker hervorgehoben, festgehalten.

JSON-Codeausschnitt des Reference-Element <i>PLC1ControlsDrillingMachine1</i> der AAS <i>DrillingMachine1</i>	
<pre> { "value": { "keys": [{ "type": "AssetAdministrationShell", "local": false, "value": "https://example.com/ids/aas/.. ..1222_8012_4022_0484", "index": 0, "idType": "IRI" }, { "type": "Submodel", "local": false, "value": "https://example.com/ids/sm/.. ..4422_8012_4022_1354", "index": 1, "idType": "IRI" }, { "type": "SubmodelElementCollection", "local": true, "value": "RelationshipsFunctional", "index": 2, "idType": "IdShort" }, { "type": "RelationshipElement", "local": true, "value": "PLC1ControlsDrillingMachine1", "index": 3, "idType": "IdShort" }] }, .. </pre>	<pre> .. "semanticId": { "keys": [{ "type": "ReferenceElement", "value": "Reference", "idType": "Custom" }] }, "constraints": [{ "type": "RelationshipContext", "value": "productProcessing", "modelType": { "name": "Qualifier" } }], "idShort": "PLC1ControlsDrillingMachine1", "modelType": { "name": "ReferenceElement" } </pre>

Abbildung 5.12: JSON-Codeausschnitt des Reference-Element *PLC1ControlsDrillingMachine1* der AAS *DrillingMachine1*

Mit den zuvor bekannten Informationen zu den Metainformationen des Relationship-Elements aus Abbildung 5.7 und dem jetzt hinzugewonnenen *value* des Reference-Elements aus Abbildung 5.12 folgt der Vergleich dieser Daten (3.3). Die in Arrays übergebenen Parameter sind ausführlich dargestellt, um den Vergleich zwischen dem Relationship- und dem Reference-Element anhand der identifizierenden Informationen nachvollziehen zu können. Die Operation liefert für dieses Beispiel als Ergebnis des Vergleichs den

Wahrheitswert *true* und bestätigt damit die Feststellung des gesuchten Reference-Elements. Im Anschluss muss das Reference-Element durch ein neues oder durch das von Nachricht (3.2) abgeänderte JSON-Objekt ersetzt werden. Eine Abänderung des JSON-Objekts ist möglich, da der Unterschied zwischen dem alten und dem neuen JSON-Objekt lediglich in einer veränderten AAS-ID und Submodel-ID im *value* des Reference-Elements besteht.

Durch den Austausch des Reference-Elements ist das Sequenzdiagramm aus Abbildung 5.6 abgeschlossen und die darauf folgenden Sequenzdiagramme zur Vervollständigung der Dokumentationspflege können bearbeitet werden.

Das Verfahren hat aufgezeigt, dass bereits bei diesem minimalen Fallbeispiel viele Informationen von der Softwareapplikation zwischengespeichert werden müssen. Maßnahmen für eine zusätzliche ressourcenschonende Optimierung, unter ausschließlicher Verwendung der Mittel des AAS-Metamodells, sind derzeit nicht bekannt.

6 Zusammenfassung

Ziel dieser Arbeit war es Use Cases zu ermitteln, die mit dem Einsatz von VWS einen nutzbringenden Mehrwert im industriellen Produktionssektor schaffen. Aus den gesammelten Use Cases wurde ein Use Case herausgegriffen und Konzepte für dessen konzeptionelle Umsetzung entwickelt. Dabei wurden vor allem die Mittel des VWS-Metamodells sowie die VWS-API einbezogen. Die prinzipielle Umsetzung der Konzepte am LIA-I4.0-Demonstrator der Otto-von-Guericke-Universität Magdeburg hat deren Praxistauglichkeit gezeigt.

Es lässt sich festhalten, dass es eine nicht geringe Menge an Use Cases gibt, in der der Einsatz der VWS nicht nur nutzbringend ist. Vielmehr ist auch die Umsetzung der Use Cases für die Unternehmen profitabel. Mit produktspezifischen value-based Services, einer sich flexibel anpassenden Fabrik oder dem Fokus auf den ressourcenschonenden Umgang bei der Herstellung, Verwendung und Wiederverwertung von Produkten, sind Unternehmen in der Lage sich von der Konkurrenz abzusetzen. Vor allem für das Assetmanagement ist der Einsatz von VWS vorrangig in Betracht zu ziehen.

Die Umsetzung des Use Case „Dokumentationspflege bei Gerätetausch“ hat gezeigt, welches Potential in einem einzelnen Use Case steckt. Jedes produzierende Unternehmen muss sich mit der Dokumentationspflege beschäftigen. Vielerorts geschieht das noch häufig manuell und erhöht damit das Risiko von Fehlern. Mit der Umsetzung der beschriebenen Konzepte können veraltete Dokumentationsstände durch eine teilweise Automatisierung zum großen Teil ausgeschlossen werden. Den kleineren Teil, der nicht automatisiert gepflegt werden kann, bilden übergreifende Planungsunterlagen, wie z.B. Netzpläne, Fließschemata, Schaltkreisdokumente etc., da diese unterschiedliche Assets und große Teile ihrer Informationen in einem Dokument vereinen.

Weiterhin hat die konzeptionelle Umsetzung des Use Case gezeigt, dass die Implementierung in der Praxis mit einem anfänglich hohen manuellen Aufwand verbunden ist. Daraus können Fehler entstehen, die sich durch Verknüpfungen mit weiteren Fehlern in ihrer Schwere potenzieren. Andere Technologien zur Umsetzung der Beziehungsmodellierung, die in die VWS integrierbar sind und die manuellen Tätigkeiten vereinfachen, sind in weiteren Forschungsarbeiten zu untersuchen.

Zusätzlich hat sich herausgestellt, dass die VWS-API eine Bandbreite an Operationen bietet, die eine automatisierte Dokumentationspflege nach dem Tausch eines Gerätes ermöglichen. Für die Erfüllung von nahezu jedem notwendigen Datentransfer bei der Dokumentationspflege und somit womöglich auch für die Umsetzung anderer Use Cases

stehen entsprechende Operationen bereit. Einzig die Möglichkeit und der Nutzen, ein bisher nicht existentes Interface für den View zu entwickeln, sollte von der Plattform Industrie 4.0 analysiert werden.

Als Voraussetzung für die unternehmensübergreifende Interoperabilität des Use Case müssen zuständige Gremien ein standardisiertes Vokabular zur Definition von Beziehungstypen entwickeln. Zusätzlich ist die semantische Definition der in den Konzepten verwendeten VWS-Elemente in Form ihrer semanticIds notwendig, um diese Elemente auch in anderen Kontexten einsetzen zu können.

Für die unternehmensinterne Interoperabilität, z.B. in einem Produktionsmodul, muss das umsetzende Unternehmen die Integration der Konzepte in allen beteiligten VWS laut den entwickelten Datenmodellen sicherstellen. Dazu zählt unter anderem die Verwendung der ausgewählten VWS-Elemente und deren Attributierung.

Bisher ungeklärt bleibt die Frage nach der Sicherstellung der fehlerfreien Übertragung der Elemente aus der VWS des alten Geräts in die VWS des neuen Geräts, wenn mehr als eine VWS zu den Geräten existiert. Hierfür muss zukünftig eine Lösung erarbeitet werden, die einen Vergleich einer VWS des alten und neuen Geräts aufgrund semantischer und identifizierender Informationen gewährleistet.

Obgleich diese Arbeit den Schwerpunkt auf die Fertigungstechnik legt, können viele der beschriebenen Use Cases ebenfalls in der Verfahrenstechnik Anwendung finden. Vor allem der ausgewählte Use Case „Dokumentationspflege bei Gerätetausch“ ist sowohl in der Fertigungstechnik als auch in der Verfahrenstechnik ohne erheblichen Mehraufwand einsetzbar.

Bis die VWS einen breiten Einzug in die industriellen Unternehmen hält, sind weitere Entwicklungszeiten notwendig, wie die konzeptionelle Umsetzung des Use Case „Dokumentationspflege bei Gerätetausch“ gezeigt hat. Viele Teilmodell-Templates zur Strukturierung der Daten befinden sich derzeit noch in der Entwicklung oder Planung und die Festlegung einer einzigen richtigen Datenstruktur ist umhin nicht möglich. Die IDTA und andere Unternehmen wie WITTENSTEIN SE oder der Verein Smart Factory^{KL} u.a. sind Wegweiser für den nutzbringenden Einsatz von VWS in vielen Anwendungsgebieten der Industrie, des Handwerks, der Logistik etc.

7 Ausblick

Die Ergebnisse dieser Arbeit, insbesondere das Konzept der Beziehungsmodellierung, können nicht nur für die Dokumentationspflege genutzt werden. Vielmehr sieht die AG1 „Joined Task Force Interaktionen“ die Beziehungsmodellierung im Kontext einer automatischen Verknüpfung von Assets. Das heißt, es existiert ein Modell, in dem jedes Asset weiß mit wem es interagieren muss, um seine Aufgaben vollumfänglich zu erfüllen. Diese Sichtweise geht über die Dokumentationspflege hinaus. Damit eröffnet das Konzept der Beziehungsmodellierung zusätzlichen Interpretationsspielraum und regt damit zur Ideenfindung für weitere Anwendungsmöglichkeiten, auch in anderen Use Cases, an. Im Folgenden sind einige der in dieser Arbeit beschriebenen Use Cases genannt, in denen die Umsetzung der Beziehungsmodellierung in einem anderen Kontext zu untersuchen ist.

Im Use Case Cluster der auftragsgesteuerten Produktion ist die Verknüpfung von Assets beispielsweise bei der Herstellung individualisierter Produkte durch Beziehungen denkbar. Diese Beziehungen entstehen mit der Konfiguration des Produkts und spiegeln den konstruktiven Aufbau dessen wieder. Des Weiteren können sie unterstützend für den nachfolgenden Lebenszyklus sein.

Ein weiterer Gedanke gilt dem Use Case „Value-Based-Services“, der im Kontext der Beziehungsmodellierung beispielsweise eine bedarfsgerechte Interaktion zwischen dem Hersteller eines Produkts und seinem Kunden zum Austausch neuer Informationen gewährleistet. Dazu werden unternehmensübergreifende Beziehungen zwischen den VWS gelegt. Sobald der Hersteller Änderungen in der Konfiguration des Produkts o.ä. durchgeführt hat, werden diese automatisch dem Kunden in Form von Updates mitgeteilt. Stimmt der Kunde dem angebotenen Update zu, wird die zu dem gekauften Produkt zugehörige VWS dahingehend aktualisiert. Umgekehrt kann der Kunde eine Anfrage an seinen Hersteller stellen, in der er sich nach neuen Informationen bzw. Updates erkundigt.

Ein letzter Use Case, der in dieser Arbeit auch eng mit der Beziehungsmodellierung verbunden ist, ist das Human-Machine-Interface. Die Präsentationslogik des Use Case benötigt eine Struktur, in der die einzelnen Bedien- und Visualisierungselemente des HMI miteinander in Verbindung gebracht werden. Die Beziehungsmodellierung kann hierfür eine hierarchische Struktur liefern, die diese Informationen bereitstellt.

Für die Realisierung der Beziehungsmodellierung bedarf es allerdings, wie bereits in der Zusammenfassung beschrieben, einer Definition eines einheitlichen Vokabulars sowie semantischer Beschreibungen durch eine öffentlich anerkannte Organisation. Dadurch ist ein

Ausschluss von Fehlern aufgrund von falschen Interpretationen seitens des VWS-Entwicklers oder der von extern zugreifenden Softwareapplikation erreichbar.

Die in der Zusammenfassung genannte Schwierigkeit der fehlenden Aktualisierung von übergeordneten Planungsunterlagen wird im Folgenden erneut aufgegriffen. Sie kann durch zwei Lösungsansätze gelöst werden, die in zukünftigen Forschungsarbeiten auf ihre Praktikabilität hin zu untersuchen sind. Eine mögliche Methode sieht die übergeordneten Planungsunterlagen als eigenständige Assets an und stellt Beziehungen zwischen diesen Dokumenten und den darin verwiesenen physisch verbauten Assets her. Eine andere Methode bietet die automatisierte und nahezu echtzeitorientierte digitale Abbildung solcher Dokumente in Teilmodelle der VWS. Durch diese digitale Abbildung wird für den Nutzer erst zum Zeitpunkt des Zugriffs ein für ihn zweckdienliches Dokument erstellt, das den Ist-Zustand der Dokumentation des Produktionsmoduls widerspiegelt.

Literatur

- [1] DIN: *Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)*. DIN SPEC 91345:2016-04. DIN ICS 03.100.01; 25.040.01; 35.240.50. April 2016.
- [2] Plattform Industrie 4.0: *Umsetzungsstrategie Industrie 4.0 - Ergebnisbericht der Plattform Industrie 4.0*. Ergebnisbericht. BITKOM e.V., VDMA e.V. und ZVEI e.V., April 2015. URL: <https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/umsetzungsstrategie-2015.html> (besucht am 08.02.2022).
- [3] Plattform Industrie 4.0: *Ergebnispapier - Aspekte der Forschungsroadmap in den Anwendungsprozessen*. Ergebnispapier. BMWi, April 2016. URL: <https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/anwendungsszenarien-auf-forschungsroadmap.html?nn=1105090#navigation> (besucht am 08.02.2022).
- [4] TC 65: *IEC TR 63283-2 ED1 - Industrial-process measurement, control and automation - Smart Manufacturing - Part 2: Use cases 65/833/CD*. Techn. Ber. Oktober 2020. URL: https://www.iec.ch/dyn/www/f?p=103:38:634909442124934::::FSP_ORG_ID,FSP_LANG_ID:1250,25 (besucht am 08.02.2022).
- [5] VDI/VDE: *Durchgängiges Engineering in Industrie 4.0 Wertschöpfungsketten*. Statusreport. VDI/VDE, Januar 2016. URL: <https://www.vdi.de/ueber-uns/presse/publikationen/details/durchgaengiges-engineering-in-industrie-40-wertschoepfungsketten> (besucht am 08.02.2022).
- [6] Standardization Council Industrie 4.0: *Deutsche Normungsroadmap Industrie 4.0*. Techn. Ber. Version 4. DIN e.V. und DKE, März 2020. URL: <https://www.din.de/de/forschung-und-innovation/themen/industrie4-0/roadmap-industrie40-62178> (besucht am 08.02.2022).
- [7] SAP Deutschland SE & Co. KG: *Was ist ERP?* o.J. URL: <https://www.sap.com/germany/insights/what-is-erp.html> (besucht am 09.02.2022).
- [8] Siemens AG: *Manufacturing Execution System (MES)*. o.J. URL: <https://www.plm.automation.siemens.com/global/de/our-story/glossary/manufacturing-execution-systems-mes/38072> (besucht am 09.02.2022).
- [9] V. S. Chakravarthi: *Internet of Things and M2M Communication Technologies - Architecture and Practical Design Approach to IoT in Industry 4.0*. 1. Aufl. Springer eBook Collection. Cham: Springer Nature Switzerland AG, 2021.

- [10] GEDYS Intraware GmbH: *Was ist ein CRM-System?* o.J. URL: <https://www.gedys-intraware.de/infothek/crm-wiki/was-ist-ein-crm-system/> (besucht am 10.02.2022).
- [11] A. M. Oberländer, J. Übelhör und B. Häckel: „IIoT-basierte Geschäftsmodellinnovation im Industrie-Kontext: Archetypen und praktische Einblicke“. In: *HMD Praxis der Wirtschaftsinformatik* 56.6 (2019), S. 1113–1125.
- [12] Plattform Industrie 4.0: *Functional View of the Asset Administration Shell in an Industrie 4.0 System Environment*. Diskussionspapier. Plattform Industrie 4.0, April 2021. URL: <https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Functional-View.html> (besucht am 16.04.2022).
- [13] Plattform Industrie 4.0: *Submodel Templates of the Asset Administration Shell - ZVEI Digital Nameplate for industrial equipment*. Spezifikation. Version 1.0. BMWi, November 2020. URL: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Submodel_templates-Asset_Administration_Shell-digital_nameplate.html (besucht am 08.02.2022).
- [14] Plattform Industrie 4.0: *Submodel Templates of the Asset Administration Shell - Generic Frame for Technical Data for Industrial Equipment in Manufacturing*. Spezifikation. Version 1.1. BMWi, November 2020. URL: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/Submodel_templates-Asset_Administration_Shell-Technical_Data.html?nn=1109556 (besucht am 08.02.2022).
- [15] Wikipedia: *Industrial Digital Twin Association*. Juni 2021. URL: https://de.wikipedia.org/wiki/Industrial_Digital_Twin_Association (besucht am 17.05.2022).
- [16] IDTA: *AAS Submodel Templates*. o.J. URL: <https://industrialdigitaltwin.org/content-hub/teilmodelle> (besucht am 10.03.2022).
- [17] F. Mayer u. a.: „Deutschlandweiter I4.0-Demonstrator - Technisches Konzept und Implementierung“. Forschungsbericht. München: Technische Universität München - Lehrstuhl für Automatisierung und Informationssysteme, 2013. URL: <https://mediatum.ub.tum.de/doc/1178726/1178726.pdf> (besucht am 14.02.2022).
- [18] Plattform Industrie 4.0: *Anwendungsszenario trifft Praxis: Auftragsgesteuerte Produktion eines individuellen Fahrradlenkers*. Ergebnispapier. BMWi, April 2017. URL: https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/anwendungsszenario-trifft-praxis.pdf?__blob=publicationFile&v=8 (besucht am 08.02.2022).
- [19] VDI/VDE: *Sprache für I4.0-Komponenten - Blatt 2 - Interaktionsprotokoll für Ausschreibungsverfahren*. VDI/VDE 2193. ICS 35.240.50. Januar 2020.

- [20] Smart Factory KL: *Exemplarische Übertragung der RAMI 4.0-Verwaltungsschale auf die SmartFactorywKL - Systemarchitektur für Industrie 4.0-Produktionsanlagen*. Whitepaper. April 2017. URL: https://smartfactory.de/wp-content/uploads/2017/11/SF_WhitePaper_2-1_DE-1.pdf (besucht am 08.02.2022).
- [21] Plattform Industrie 4.0: *Industrie-4.0 - Plug and Produce for Adaptable Factories: Example Use Case Definition, Models, and Implementation*. Workingpaper. BMWi, Juni 2017. URL: https://www.zvei.org/fileadmin/user_upload/Presse_und_Medien/Publikationen/2017/Juni/Industrie_4.0_Plug_and_produce/Industrie-4.0-Plug-and-Produce-zvei.pdf (besucht am 08.02.2022).
- [22] IDTA: *Factsheet Asset Management System*. o.J. URL: <https://industrialdigitaltwin.org/use-cases/asset-management-system> (besucht am 24.02.2022).
- [23] IDTA: *Product Carbon Footprint*. o.J. URL: <https://industrialdigitaltwin.org/use-cases/schaltschrank-mit-product-carbon-footprint> (besucht am 27.01.2022).
- [24] ETZ Redaktion: *Ranking der 25 besten IoT-Plattformen*. 1.08.2019. URL: <https://www.smart-production.de/etz/news-detailansicht/nsctrl/detail/News/ranking-der-25-besten-iot-plattformen/np/2> (besucht am 22.02.2022).
- [25] IDTA: *Wittenstein Service Portal*. o.J. URL: <https://industrialdigitaltwin.org/use-cases/wittenstein> (besucht am 27.01.2022).
- [26] Plattform Industrie 4.0: *Details of the Asset Administration Shell - Part 1 - The exchange of information between partners in the value chain of Industrie 4.0 (Version 3.0RC01 Review)*. Spezifikation. Version 3.0RC01 in Review. BMWi, o.J.
- [27] Plattform Industrie 4.0: *Relationships between I4.0 Components – Composite Components and Smart Production*. Working Paper. BMWi, Juni 2017. URL: https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/relationships-i40-components.pdf?__blob=publicationFile&v=5 (besucht am 20.04.2022).
- [28] Plattform Industrie 4.0: *Details of the Asset Administration Shell - Part 2 Interoperability at Runtime – Exchanging Information via Application Programming Interfaces*. Spezifikation. Version 1.0RC02. BMWi, November 2021. URL: <https://industrialdigitaltwin.org/content-hub/downloads> (besucht am 20.04.2022).
- [29] *VWS-Referenzmodellierung - Exemplarische Modellierung einer fertigungstechnischen Anlage mit AASX Package Explorer auf Basis des VWS-Metamodells*. Diskussionspapier. Plattform Industrie 4.0, April 2021. URL: <https://www.plattform-i40.de/IP/Redaktion/DE/Downloads/Publikation/VWS-Referenzmodellierung.html> (besucht am 15.03.2022).

A Anhang A - VWS-Teilmodell-Templates der IDTA

Tabelle A.1 zeigt alle Teilmodell-Templates, der bisher durch die IDTA entwickelten, in der Entwicklungsphase befindlichen oder noch zu entwickelnde Teilmodelle. Eine kurze Beschreibung zu jedem Teilmodell-Template ermöglicht eine Einschätzung über deren Einsatzmöglichkeiten in der Praxis. Die Langfassung der Beschreibungen sind auf der Website der IDTA zu finden [16].

Tabelle A.1: Auflistung der durch die IDTA entwickelten bzw. noch zu entwickelnde Teilmodelle-Templates [16], Stand vom 10.03.2022

Teilmodell-Template	IDTA-Nr.	Version-Nr.	Beschreibung	Status
Inclusion of Module Type Package (MTP) Data	2001	1.0	Das Teilmodell beschreibt die Integration von PEA (Instanz) und MTP (Typ) Informationen.	in Überprüfung
Contact Information	2002	1.0	Das Teilmodell soll die interoperable Bereitstellung von Kontaktinformationen in Bezug auf das Asset ermöglichen.	in Überprüfung
Generic Frame for Technical Data for Industrial Equipment in Manufacturing	2003	1.2	Das Teilmodell sorgt für die interoperable Bereitstellung von techn. Daten. Das Verständnis über die Daten ist für alle Akteure, wie Marktteilnehmer, Systemintegratoren, Betreiber von Industrieanlagen etc. eindeutig.	in Überprüfung

Teilmodell- Template	IDTA- Nr.	Version- Nr.	Beschreibung	Status
Handover Docu- mentation	2004	1.2	Das Teilmodell stellt alle Do- kumentationen bereit, die ein Hersteller an den Betreiber von Industrieanlagen nach In- betriebnahme übergibt. Dazu zählen das Bedienhandbuch, CE-Konformitätserklärungen, ATEX-Bescheinigungen etc.	in Überprü- fung
Simulation	2005	1.0	Das Teilmodell soll Simulati- onsmodelle beschreiben, die herstellerübergreifend genutzt werden können.	in Entwick- lung
Maintenance		tbd.	Das Teilmodell dient zur Übergabe aller wichtigen Instandhaltungsinforma- tionen an übergeordnete Instandhaltungssoftware/- systeme. Dazu gehören z.B. Wartungsintervalle und Instand- haltungsanweisungen.	in Planung
ZVEI Digital Nameplate for Industrial Equipment	extern	1.0	Dieses Teilmodell bildet das physisch vorhandene Typen- schild in der digitalen Welt ab. Es dient dem interoperablen Informationsaustausch zwischen Partnern eines Wertschöpfungs- netzwerks.	Veröffentlicht
Generic Frame for Technical Data for Indus- trial Equipment in Manufactu- ring	extern	1.1	Das Teilmodell sorgt für die interoperable Bereitstellung von techn. Daten. Das Verständ- nis über die Daten ist für alle Akteure, wie Marktteilnehmer, Systemintegratoren, Betrei- ber von Industrieanlagen etc. eindeutig.	Veröffentlicht

Teilmodell-Template	IDTA-Nr.	Version-Nr.	Beschreibung	Status
Digital Nameplate for Industrial Equipment		1.1	Dieses Teilmodell bildet das physisch vorhandene Typenschild in der digitalen Welt ab. Es dient dem interoperablen Informationsaustausch zwischen Partnern eines Wertschöpfungsnetzwerks.	in Entwicklung
OPC UA Server Data Sheet		1.0	Das Teilmodell enthält Informationen über einen OPC UA Server, d.h Informationen zum Auffinden des Servers und über dessen implementierten Inhalt.	in Entwicklung
Time Series Data		1.0	Das Teilmodell soll Daten von Zeitreihen darstellen, die sich über den gesamten Anlagenlebenszyklus ziehen. Schwerpunkt ist die Beschreibung semantischer Informationen über die Zeitreihen und deren Speicherung in Anlagen-VWS.	in Entwicklung
Service Order Creation		1.0	Mit diesem Teilmodell soll die Möglichkeit bestehen, Serviceaufträge in einem ERP-System zu erstellen.	in Entwicklung
Bill of Material		1.0	Das Teilmodell stellt eine Art Stückliste für Assets dar, die wiederum aus mehreren Assets bestehen. Die Assets werden in der Bill of Material durch Entitäten dargestellt, die Platzhalter für die Assets darstellen.	in Entwicklung

Teilmodell- Template	IDTA- Nr.	Version- Nr.	Beschreibung	Status
Platform Re- sources		tbd.	Das Teilmodell bildet eine Mo- dellstruktur zur (Laufzeit-) Darstellung von Geräteres- ourcen wie Speicher- und CPU-Kapazitäten. Es kann zur Analyse der Echtzeitfähigkeit dienen.	in Planung
Plant Asset Management		tbd.	Das Teilmodell stellt Informa- tionen über den Zustand von Assets in Produktionsprozes- sen sowie dessen Funktionen und Methoden für den Asset- Manager bereit.	in Planung
Functional Safety and Reliability		tbd.	Das Teilmodell spezifiziert Beschreibungen von Datenmo- dellen für funktionale Sicher- heit und Zuverlässigkeit von Geräten, die von Entwicklungs- werkzeugen für den Entwurf sicherheitsbezogener Steue- rungssysteme genutzt werden sollen.	in Planung
Control- ComponentType & ControlCom- ponentInstance		tbd.	Das Teilmodell besitzt eine einheitliche Beschreibung von Steuerungskomponenten für den Aufbau einer Prozesssteuerung. Dazu zählen z.B. verfügbare Schnittstellen, Betriebsarten und mögliche Fehler in jedem Betriebsmodus.	in Planung

Teilmodell- Template	IDTA- Nr.	Version- Nr.	Beschreibung	Status
Software Name- plate		tbd.	Das Teilmodell ähnelt dem Digital Nameplate, nur das in diesem Fall Informationen über eine Software bereitgestellt werden. Ein solches Typenschild kann Updates, Lizenzmanagement und Audits unterstützen.	in Planung
DEXPI		1.0	Vorschlag für ein VWS-Teilmodell zur Paketierung und Übermittlung von nach dem DEXPI-Standard verpackten P&I-Diagrammen.	in Entwicklung
Engineering of Power Drive Trains		tbd.	-	in Planung
Asset Interface Description		tbd.	Das Teilmodell dient zur Standardisierung der Struktur und Semantik einer Interaktion, die mit Anlagen oder Diensten stattfinden. Zu den Kommunikationsprotokollen gehören unter anderem OPC UA, HTTP, Modbus und MQTT.	in Planung