

# **Modellierung von Dienstleistungen, Capabilities und Skills zur Einbindung von Unternehmen in eine marktplatzbasierte Fertigung**

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

**Doktor-Ingenieur (Dr.-Ing.)**

genehmigte

**Dissertation**

von

Herrn

**Diplom-Ingenieur Magnus Volkmann**

aus Ludwigshafen am Rhein

Tag der mündlichen Prüfung: 13.12.2023

Dekan: Prof. Dr. rer. nat. Roland Ulber

Promotionskommission:

Vorsitzender: Prof. Dr.-Ing. Oliver Koch

1. Berichterstatter: Prof. Dr.-Ing. Martin Ruskowski

2. Berichterstatter: Prof. Dr.-Ing. Christian Diedrich

# Danksagung

Diese Dissertation entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter an der Rheinland-Pfälzischen Technischen Universität sowie der SmartFactory-KL.

Vor allem möchte ich mich bei Prof. Dr.-Ing. Martin Ruskowski bedanken, der mir kurz nach Beginn seiner Professur die Möglichkeit gab, als Mitarbeiter die Neuausrichtung des Lehrstuhls WSKL mitzugestalten. Die übertragene Verantwortung und Freiräume ermöglichten es, neue Ideen in Projekte umzusetzen und die Inhalte der Dissertation zu erarbeiten. Martin hatte dabei stets die Vision, wie Industrie 4.0 in der Realität einmal aussehen soll. Dies trieb uns an, technische Lösungen zu erarbeiten und diese zu testen.

Ebenso gilt mein Dank Prof. Dr.-Ing. Christian Diedrich für die 2. Begutachtung und das Anstoßen der Vernetzung von Wissenschaft und Industrie über die Plattform Industrie 4.0 im Bereich Capabilities, Skills und Services. So konnten Definitionen, Begrifflichkeiten und Zusammenhänge, die in der Dissertation diskutiert wurden, konsolidiert und in einem Whitepaper veröffentlicht werden. Des Weiteren danke an Prof. Dr.-Ing. Oliver Koch, der den Vorsitz der Prüfungskommission übernahm.

Vielen Dank an alle Kollegen des Lehrstuhls WSKL, der SmartFactory-KL und des Fachbereichs IFS am DFKI für die tolle Zusammenarbeit über die ganzen Jahre. Insbesondere danke an alle Kollegen aus dem Circle Skill-based Engineering, mit denen ich meine Ideen diskutieren konnte, die mir Feedback gaben und mit denen wir an einigen Demonstratoren unsere Idee der flexiblen Fertigung zeigen konnten. Namentlich noch einmal danke an Jesko, Andreas und Tatjana für die Unterstützung in der finalen fachlichen Diskussion und Korrektur der Dissertation. Danke an unser Technikteam, die uns beim Aufbau der Demonstratoren halfen und immer die richtigen Ideen bei Problemen haben. Danke auch an alle Studenten, die beispielsweise durch wissenschaftliche Arbeiten oder als Hiwi bei der Umsetzung der Implementierung unterstützt haben.

Vielen Dank an alle, mit denen ich mich im Rahmen der Erarbeitung des Capability-Skill-Service (CSS)-Modells institutsübergreifend fachlich austauschen konnte. Nur durch die intensive Zusammenarbeit mehrerer Forschungsinstitute und Industrieunternehmen konnte so projektübergreifend ein konsolidiertes Ergebnis über die Plattform Industrie 4.0 veröffentlicht werden.

Final möchte ich mich noch bei meiner Familie und meiner Partnerin bedanken, die mich stets unterstützten, motivierten und mir auch den Rücken während der Promotion frei hielten.



# Inhaltsverzeichnis

<b>Danksagung</b>	<b>III</b>
<b>Kurzfassung</b>	<b>VII</b>
<b>Abstract</b>	<b>VIII</b>
<b>1 Einleitung</b>	<b>1</b>
1.1 Motivation und Problemstellung . . . . .	2
1.2 Zielsetzung und Forschungsfragen . . . . .	3
1.3 Vorgehensweise . . . . .	5
<b>2 Grundlagen und Stand der Technik</b>	<b>7</b>
2.1 Produktions- und Steuerungssysteme . . . . .	7
2.1.1 Verwaltungsschale / I4.0-Komponente / I4.0-Sprache . . . . .	10
2.1.2 Cyber-physische Systeme . . . . .	12
2.1.3 IEC 61499 . . . . .	14
2.1.4 OPC UA . . . . .	15
2.1.5 Namur open Architecture (NoA) . . . . .	16
2.2 Aktuelle Forschung im Bereich der flexiblen Fertigung . . . . .	17
2.2.1 Multiagentensysteme und holonische Systeme . . . . .	21
2.2.2 Unternehmensübergreifende verteilte Systeme . . . . .	24
2.3 Automatisierte Prozessplanung in der spanenden Bearbeitung . . . . .	28
2.3.1 Programmierung von Werkzeugmaschinen . . . . .	29
2.3.2 Computer Aided Process Planning . . . . .	30
2.3.3 STEP . . . . .	33
2.4 Forschung zur automatisierten Fertigung mit Werkzeugmaschinen . . . . .	37
2.5 Marktplätze und digitale Angebotserstellung für Lohnfertiger . . . . .	39
2.6 Zusammenfassung und Analyse . . . . .	41
<b>3 Systemkonzept für eine marktplatzbasierte Fertigung mit Capabilities und Skills</b>	<b>44</b>
3.1 Definition des Fertigungsablaufs und der Systemarchitektur . . . . .	44
3.2 Featureextraktion zur Generierung einer Produkt-VWS . . . . .	56
3.2.1 Extrahierbare Features aus STEP-AP 242 . . . . .	57

3.2.2	Extrahierbare Features in Siemens NX . . . . .	60
3.2.3	Produkt-VWS . . . . .	62
3.3	Beschreibung von Dienstleistung, Capabilities und Skills . . . . .	65
3.3.1	Beschreibung von Dienstleistungen . . . . .	66
3.3.2	Beschreibung von Capabilities und Skills . . . . .	68
3.3.3	Matching von Capabilities und Features . . . . .	76
3.4	Skill-Schnittstelle des CPPM . . . . .	82
3.4.1	OPC UA-Skills . . . . .	85
3.4.2	VWS-Teilmodell für Skills . . . . .	91
<b>4</b>	<b>Implementierung und Validierung</b>	<b>94</b>
4.1	Extraktion von Features aus CAD/CAM-Software . . . . .	95
4.2	Teilmodelle für ein Mapping von Services, Capabilities und Skills . . . . .	99
4.3	Skillbasierte Ansteuerung eines Fräsroboters . . . . .	104
<b>5</b>	<b>Fazit und Ausblick</b>	<b>113</b>
5.1	Zusammenfassung . . . . .	113
5.2	Fazit . . . . .	113
5.3	Ausblick . . . . .	116
<b>Anhang</b>		<b>119</b>
A	Appendix A . . . . .	119
<b>Literaturverzeichnis</b>		<b>122</b>
	Monografien und Artikel . . . . .	140
	Normen und Richtlinien . . . . .	145
	Webseiten und Internetreferenzen . . . . .	149
	Betreute studentische Arbeiten mit thematischer Relevanz . . . . .	149

## Kurzfassung

Die Ausarbeitung erläutert die Umsetzung eines RAMI 4.0 konformen Marktplatz in der spannenden Bearbeitung. Ziel ist es einen Lösungsansatz zu definieren, in dem firmenübergreifende Prozessketten für kleine Losgrößen automatisiert identifiziert werden und die Fertigung eines individuellen Produktes realisiert wird. So sollen in Zukunft Prozessketten in einem Datenraum aus mehreren Unternehmen identifiziert werden. Nachdem notwendige Technologien analysiert und die wichtigen Schnittstellen des Datenraums definiert sind, wird der Fokus der Dissertation auf die folgenden drei Punkte gelegt. Zunächst wird eine Produktbeschreibung in Form einer Verwaltungsschale definiert, um benötigte Produktinformationen zwischen Herstellern in einem herstellernerneutralen Format auszutauschen. Hierzu werden Beschreibungsformen des Produktes untersucht sowie verwendbare Technologien zur automatisierten Überführung der Produktinformationen in eine Verwaltungsschale identifiziert. Im nächsten Schritt werden mögliche Beschreibungsformen für Dienstleistungen (Services) eines Unternehmens sowie für Capabilities von Produktionsressourcen eines Unternehmens untersucht. Dienstleistungen werden benötigt, um mögliche Unternehmen für einen Fertigungsschritt in der Prozesskette zu identifizieren. Capabilities enthalten alle notwendigen Informationen, um ein automatisiertes Matching zwischen Produkthanforderungen sowie den vorhandenen Produktionsressourcen in einem Unternehmen zu ermöglichen. Abschließend wird eine Skill-Schnittstelle für die Produktionsressourcen definiert. Diese ermöglicht, dass Capabilities auf einer Produktionsressource individualisiert gefertigt werden können. Die Extraktion von Produktinformationen, die Fertigung eines individualisierten Produktes sowie die Beschreibung der Informationen in den Verwaltungsschalen wird validiert. Vor allem stellt sich als Herausforderung für die Zukunft heraus, eine gemeinsame Semantik für die Beschreibung von Capabilities zu definieren. Diese würde ermöglichen, dass ein Matching zwischen proprietären Produktinformationen und Skills möglich wird.

# Abstract

The thesis explains the implementation of a RAMI 4.0 compliant marketplace in machining. The aim is to define an approach in which cross-company process chains for small batch sizes are identified automatically and the manufacture of an individual product is realized. In this way, process chains from several companies will be identified in a future data space. After the necessary technologies are analyzed and the important interfaces of the data space are defined, the focus of the thesis is on the following three points. First, a product description in the form of an asset administration shell is defined to exchange required product information between manufacturers in a manufacturer-neutral format. For this purpose, description forms of the product are examined, as well as usable technologies for the automated transfer of the product information into an asset administration shell. In the next step, possible description forms for the services of a company as well as for capabilities of its production resources are examined. Services are needed to identify possible companies for a manufacturing step in the process chain. Capabilities contain all necessary information to enable an automated match between product requirements and the available production resources in a company. Finally, a skill interface is defined for the production resources. This enables capabilities to be manufactured individually on a production resource. The production of an individualized product as well as the description of the information in the asset administration shells will be validated. Above all, it turns out to be a challenge for the future to define a common semantics for the description of capabilities. This would enable matching between proprietary product information and skills.

# 1 Einleitung

Volatile Märkte, individuelle Produkte und kürzere Lebenszyklen benötigen in Zukunft dynamischere Produktionsumgebungen [Hu13; LRZ06]. Die Corona-Pandemie hat gezeigt, wie eine plötzliche Veränderung der Nachfrage bestimmter Güter zu einem globalen Lieferproblem aufgrund starrer Produktions- und Lieferketten führt [Ma21]. Die Anpassungsfähigkeit an unvorhergesehene Änderungen im Produktionsablauf sowie die Fähigkeit individuellere und personalisierte Anfragen zu bedienen, werden in neuartigen Produktionssystemen verstärkt gefordert [SB16b]. Somit werden sich zukünftig digitale Geschäftsmodelle etablieren, die z. B. über Plattformen angeboten werden [SB16b]. Eine Möglichkeit, Unternehmen individuell zur Produktion eines Bauteils zu vernetzen, basiert auf dem Konzept Produktions-Dienstleistungen auf digitalen Marktplätzen anzubieten und wird als *Production as a Service* (PaaS) bezeichnet [He20]. Digitale Marktplätze sollen sich im Rahmen der Entwicklung der 4. Industriellen Revolution langfristig etablieren [Ci17; Ge19]. Im Zuge dieser Entwicklung wird es notwendig, die Automatisierung von Anlagen flexibler zu gestalten, um kundenindividuelle Anfragen bedienen zu können [He19]. Die Entwicklungen im Zuge der vierten Industriellen Revolution, auch als Industrie 4.0 (I4.0) bezeichnet, wird die Geschäftsmodelle durch die Konnektivität bzw. digitale Vernetzung von Produktionsumgebungen verändern [Ba14; SB16b].

Eines der großen Ziele, die durch die Weiterentwicklung erreicht werden sollen, ist die auf Kunden angepasste Verarbeitung von flexiblen Aufträgen in unterschiedlichen Losgrößen. Produktionslandschaften sind noch immer zum größten Teil sehr starr als Ablauflogik programmiert, wie auch von einem Industrievertreter in einer gemeinsamen Diskussionsrunde bestätigt [@SF-Live]. Aus der Diskussion geht hervor, dass für Industrieunternehmen die Notwendigkeit besteht, Produktionssysteme flexibler anzusteuern [@SF-Live]. Flexibilität wird bisher vor allem durch eine Variantenvielfalt von Produkten erreicht. Zur Fertigung der Variantenvielfalt werden in den meisten Fällen verschiedene vorher festgelegte Programme nacheinander ausgeführt.

Werden neue Produkte auf einer Produktionslinie gefertigt, ist hierzu häufig eine Umrüstung der Systeme sowohl auf Hardware- als auch auf der Softwareseite notwendig. Vor allem Produktionsprozesse in kleinen und mittleren Unternehmen (KMU), die typischerweise kleinere Losgrößen fertigen, sind häufiger mit längeren Umrüstzeiten konfrontiert [Di12; LSB16]. Hinzu kommt, dass zur Programmierung, der Simulation oder dem Betreiben von Fertigungsmaschinen Expertenwissen notwendig ist und je nach Maschine die Programmierung nur von geschultem und erfahrenem Personal im Unternehmen durchgeführt werden kann. Bei einem Wechsel zwischen unterschiedlichen Produkten, die in einer Produktionsumgebung gefertigt werden,

entstehen somit Umrüstzeiten und Aufwände für individuelle Aufträge, die es durch flexiblere Produktionsarchitekturen abzufangen gilt. Im Rahmen der Arbeit werden Lösungsansätze für digitale Schnittstellen für Produktbeschreibungen, Dienstleistungsbeschreibungen und Konzepte für Maschinenschnittstellen in der spanenden Bearbeitung betrachtet. Das Ziel ist, eine durchgängige Digitalisierung der Wertschöpfungskette zu ermöglichen und konkrete Lösungsansätze für eine marktplatzbasierte Fertigung zu liefern.

## 1.1 Motivation und Problemstellung

Die Zielgruppe der Dissertation sind Lohnfertiger in der spanenden Bearbeitung. Diese sind unter anderem spezialisiert, unterschiedliche Aufträge ab einer Losgröße der Stückzahl 1 zu bedienen (Kapitel 2.3). Die Losgrößen der einzelnen Aufträge und deren zeitliche Abwicklung kann aufgrund des oftmals stark schwankenden Bedarfs der Kunden sehr unterschiedlich und schwer vorhersehbar sein, somit ist die Flexibilität eines Lohnfertigers ein wichtiger Wettbewerbsvorteil [Wi08]. Im Angebotsprozess müssen aufgrund der Vielfalt unterschiedlicher Anfragen erfahrene Mitarbeiter eingebunden werden. Diese prüfen, ob ein Auftrag im vorhandenen Maschinenpark fertigbar ist und führen eine Kostenkalkulation und grobe Prozessplanung durch. Digitale Hilfsmittel, die den Angebotsprozess beschleunigen, sind inzwischen auf dem Markt verfügbar und erste digitale Marktplätze etablieren sich (Kapitel 2.5).

Die bisher verfügbaren Softwarelösungen berücksichtigen allerdings keine implementierungsspezifischen Faktoren der vorhandenen Produktionsressourcen. Die Komplexität in der Angebotserstellung und Produktionsplanung ist in einem flexiblen Produktionsumfeld so hoch, dass der Mensch diese bei vielen individuellen Anfragen nicht abdecken kann. In Zukunft wird diese Komplexität noch deutlich weiter steigen. So müssen z. B. Unternehmen in Zukunft Informationen über den ökologischen Fußabdruck ihres Produktes angeben [Eu19]. Daraus folgt die Anforderung, dass zusätzlich in der Planung Fertigungsmittel mit geringem Energiebedarf ausgewählt werden, genauer gesagt der CO<sub>2</sub>-Ausstoß eines Fertigungsschritts vorhergesagt wird. Individuelle Optimierungen nach ökologischen, ökonomischen oder zeitlichen Faktoren sind mit dem heutigen Stand der Technik in der Prozessplanung nicht möglich. Vor allem den KMU fehlen aufgrund mangelnder Digitalisierung relevante Informationen über die einzelnen Produktionsschritte.

Sind Informationen vorhanden, können diese aufgrund unterschiedlichster digitaler Schnittstellen oftmals nicht einheitlich abgerufen werden. Somit ist der Aufwand, jede Maschine an ein Datenbanksystem anzuschließen, sehr hoch. Herstellerübergreifende Schnittstellen ermöglichen es, Produktionsmodule als gekapselte Module in die Produktionsumgebung zu integrieren. Durch Spezifikationen können die Beschreibung und Datenpunkte semantisch definiert werden. Allerdings fehlt eine individuelle Ansteuerung und Prozessplanung in Abhängigkeit des gewünschten Produktes und verfügbaren Maschinenparks.

Innerhalb dieser Dissertation werden Möglichkeiten untersucht, die durch die Digitalisierung von Wertschöpfungsketten Prozesse von der Angebotserstellung bis zur Produktion beschleunigen. Des Weiteren wird die Möglichkeit untersucht, die Maschinenansteuerung zu flexibilisieren und somit die Resilienz von Lieferketten zu erhöhen. Hierzu werden im folgenden Kapitel Forschungsfragen definiert, die Lösungsansätze für verteilte, marktplatzbasierte Produktionsstrukturen sowie die automatisierte Fertigung von kleinen Losgrößen liefern. Zusammenfassend wird die Problemstellung in Abbildung 1.1 dargestellt.

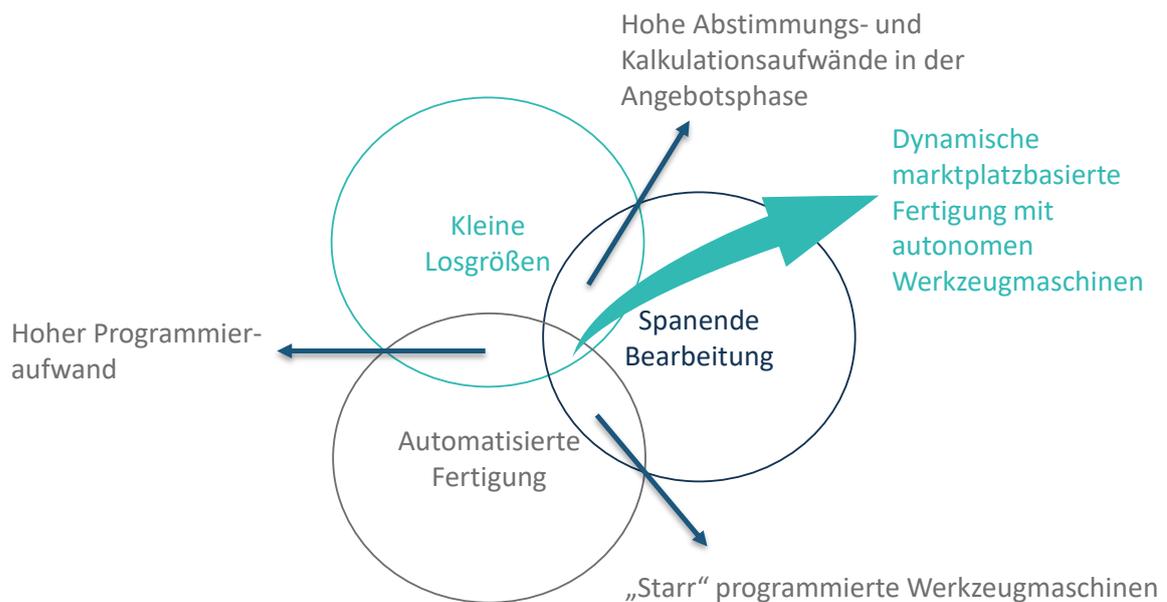


Abbildung 1.1: Darstellung existierender Probleme und eines Lösungsansatzes

## 1.2 Zielsetzung und Forschungsfragen

Die Zielsetzung ist die Definition von Schnittstellen, um in Zukunft eine weitgehend digitale Produktion zu ermöglichen. Somit werden individuelle Produktionsaufträge im Bereich der spanenden Bearbeitung durch die Schaffung von Schnittstellen (Standards), im Sinne der Entwicklungen von I4.0, möglichst digital abgewickelt. Es soll sowohl der Angebotsprozess als auch die Fertigung eines Produkts durch Lohnfertiger beschleunigt und automatisiert werden. Bei der Untersuchung der Lösungsmöglichkeiten werden die Anforderungen von I4.0 in der Produktionsarchitektur beachtet. So entstehen klar definierte Schnittstellen, die Unternehmen für eine verteilte Produktionsumgebung nutzen können. Hierzu wird sich an der DIN SPEC 91345:2016 orientiert, die das Referenzarchitekturmodell für Industrie 4.0 (RAMI 4.0) definiert. Es zeigt die Voraussetzung für eine Kooperation und Kollaboration zwischen Assets in einem Produktionssystem und definiert die Verwaltungsschale (siehe Kapitel 2.1.1). In RAMI 4.0 wird ein dreidimensionales Modell mit den wichtigsten Informationsebenen für Assets definiert (siehe Abbildung 1.2).

Eine **Referenzarchitektur** ist ein „Modell für eine Architekturbeschreibung (für I4.0), die allgemein genutzt wird und als zweckmäßig anerkannt ist (Referenzcharakter hat)“ [DIN SPEC 91345].

Ein **Asset** ist ein „Gegenstand, der einen Wert für eine Organisation hat“ [DIN SPEC 91345].

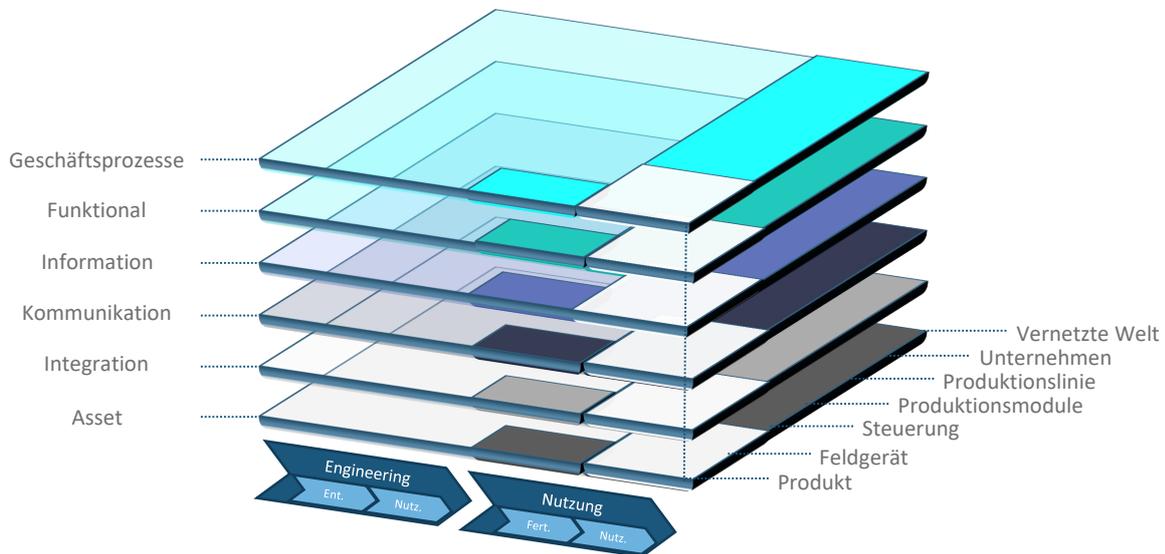


Abbildung 1.2: Betrachtung der fokussierten Schichten in RAMI 4.0 in Anlehnung an [DIN SPEC 91345; He21]

Bei einem Asset kann es sich z. B. um ein Gerät, ein Produkt, eine Maschine oder ein Anlagenmodul handeln, aber auch um Artefakte wie Konstruktionsdateien, Technische Zeichnungen oder Softwaremodule [Ba16]. Wie in Abbildung 1.2 zu erkennen, besteht RAMI 4.0 aus den folgenden drei Achsen [DIN SPEC 91345]:

- **Architektur-Achse (Layer):** Besteht aus 6 Schichten (Asset, Integration, Kommunikation, Information, Funktionalität und Business) und dient der Beschreibung von strukturellen Eigenschaften von einem oder mehreren verknüpften Assets.
- **Lebenszyklus-Achse (Life-Cycle & Value Stream):** Beschreibt in Bezug auf die IEC 62890 den Lebenszyklus und Wertschöpfungsprozess eines Assets.
- **Hierarchie-Achse (Hierarchy):** In Anlehnung auf die Normen DIN EN 62264-1 und DIN EN 61512-1 werden den Ebenen verschiedene funktionelle Modelle zugewiesen: Vernetzte Welt, Unternehmen, Technische Anlage, Station (hier als Produktionsmodul bezeichnet), Steuereinheit, Feldgerät, Produkt. Somit werden die hierarchischen Ebenen einer automatisierten Produktionsumgebungen beschrieben (siehe Kapitel 2.1) und ergänzt diese.

Für ein marktplatzbasiertes Produktionsverfahren werden im Kontext von RAMI 4.0 die Anforderungen zur Beschreibung von Produkten (*Product*), Produktionsmodulen (*Station*) und Unternehmen (*Enterprise*) betrachtet (siehe Abbildung 1.2). Hierbei ist es wichtig, die Schnittstellen zwischen den Hierarchie-Achsen zu berücksichtigen. So werden Möglichkeiten aufgezeigt, wie

Unternehmen über die vernetzte Welt miteinander agieren und Steuerungen zu Produktionsmodulen bzw. Produktionsmodule zu Produktionslinien hierarchisch verknüpft werden können. Einzelne Feldgeräte werden nicht einbezogen. Der Fokus liegt hinsichtlich der Lebenszyklusachse auf der Fertigung eines Produktes. Des Weiteren wird ein Einblick in die Möglichkeiten der Vernetzung von Unternehmen verschafft, um firmenübergreifend dynamisch Produktionsaufträge zu erzeugen.

Für die Ausarbeitung ergeben sich die folgenden drei Forschungsfragen und Ziele, die in Bezug auf die Lohnfertigung in der Fräsbearbeitung betrachtet werden. Der Fokus liegt auf der Definition von wichtigen Schnittstellen von der Business-Schicht bis zur Feldebene. Es werden die notwendigen Vorgaben definiert, um sich standardisiert und möglichst schnell an ein Produktionsnetzwerk anzubinden und flexibel Produktionsaufträge fertigen zu können.

#### **Forschungsfragen:**

1. Wie kann eine herstellerunabhängige, RAMI 4.0-konforme Produktbeschreibung umgesetzt werden, um marktplatzbasierte Verfahren in der spanenden Bearbeitung zu ermöglichen? Ziel ist die Erstellung eines Teilmodells einer Produktbeschreibung, die alle relevanten Produktinformationen enthält, um eine automatisierte Prozessplanung individueller Aufträge zu realisieren.
2. Wie können Dienstleistungen und Fähigkeiten von Unternehmen in der spanenden Bearbeitung herstellerübergreifend beschrieben werden, sodass die Identifikation eines Produktionsablaufes möglich wird? Ziel ist die Definition von Dienstleistungs- und Fähigkeitsbeschreibungen. Somit können potenzielle Unternehmen zur Fertigung kontaktiert werden und eine Prozessplanung anhand der Produktbeschreibung stattfinden.
3. Wie sieht eine herstellerunabhängige Maschinenschnittstelle und deren Beschreibung aus, um individuell geplante Produktionsabläufe umzusetzen? Ziel ist die Definition einer standardisierten Schnittstelle zur individuellen und herstellerunabhängigen Einbindung von Produktionsressourcen in die Produktionsplanung und deren flexibler Ansteuerung.

## **1.3 Vorgehensweise**

Die drei Forschungsfragen bilden die Zielformulierung der Arbeit und werden in den nächsten Kapiteln wie folgt diskutiert. Zunächst wird in Kapitel 2 der Stand der Technik beschrieben. Hierzu werden benötigte Definitionen ausgearbeitet und thematisch relevante Literatur untersucht. In Kapitel 3 werden Lösungsansätze für die aufgeführten Fragestellungen erarbeitet. Zunächst wird die benötigte Steuerungsarchitektur eines Unternehmens diskutiert, um den Anforderungen an eine flexible Produktion gerecht zu werden (siehe Kapitel 3.1). Kapitel 3.2 zeigt in Bezug auf Forschungsfrage 1 einen Lösungsansatz einer Produktbeschreibung für die spanende Be-

arbeitung. Kapitel 3.3 fokussiert Forschungsfrage 2 und definiert die Beschreibung von Dienstleistungen und Fähigkeiten eines Unternehmens. Kapitel 3.4 befasst sich mit der individuellen Ansteuerung von Maschinen und der Definition einer Schnittstelle (Forschungsfrage 3). Kapitel 4 dokumentiert eine Implementierung und Validierung der erarbeiteten Lösungsansätze.

## 2 Grundlagen und Stand der Technik

Zum Verständnis der entwickelten verteilten Fertigungsstruktur, werden vorab Grundlagen aus den Bereichen verteilter Systeme, automatisierter Prozessplanung, cloudbasierter Fertigung und existierende teilautomatisierte Lösungen diskutiert. Zunächst wird in Kapitel 2.1 der Stand der Technik von automatisierten Produktionsanlagen sowie verschiedener Ansätze verteilter Systeme analysiert. Es wird auf existierende Normen, verschiedene Systemarchitekturen sowie Konzepte aus der Entwicklung rund um Industrie 4.0 eingegangen. Kapitel 2.2 beschreibt eine Auswahl von Forschungsprojekten aus dem Themengebiet der flexiblen Fertigung. In Kapitel 2.3 werden vor allem Lösungen zur Automatisierung der spanenden Bearbeitung betrachtet, die in Fertigungsprozessen heute standardmäßig Verwendung finden. In Kapitel 2.4 werden Ansätze aus der Forschung zur autonomen spanenden Bearbeitung vorgestellt.

### 2.1 Produktions- und Steuerungssysteme

Die Automatisierungspyramide (siehe Abbildung 2.1) stellt in heutigen automatisierten Produktionsumgebungen die eingesetzte Steuerungsarchitektur dar. Hierbei wird eine vertikale hierarchische Ordnung von der Prozess-/ Feldebene bis zur Unternehmensebene abgebildet. Die Funktionalitäten der einzelnen Ebenen sind in [SG16] wie folgt beschrieben:

- **Ebene 0** Prozessebene: Auf dieser Ebene findet der Produktionsprozess bzw. die Veränderung eines Produktzustandes statt. Produkte können Informationen, bspw. die Produkt-ID zur eindeutigen Zuordnung z. B. über RFID-Chips an die nächsthöhere Ebene weitergeben.
- **Ebene 1** Feldebene (*Shopfloor*): In der Feldebene werden durch Sensoren produktionsrelevante Informationen erfasst und als Ausgangssignal an die Steuerungsebene weitergegeben. Aktoren führen nach der Interpretation der Eingangssignale in der Steuerungsebene entsprechende Funktionen in der Feldebene aus.
- **Ebene 2** Steuerungsebene: Speicherprogrammierbare Steuerungen (SPS) verarbeiten Eingangssignale, die von Sensoren in der Feldebene stammen und geben Ausgangssignale an die Aktoren weiter.
- **Ebene 3** (Prozess-) Leitebene: Mensch-Maschine-Schnittstellen (engl. *Human-Machine Interface* (HMI)), Prozessleit- oder SCADA- (*Supervisory control and data acquisition*) Sys-

teme visualisieren und speichern ggf. relevante Prozessvorgänge und -daten. Dies dient Anwendern vor allem als Bedien- und Visualisierungssystem.

- **Ebene 4** Betriebsebene: Das *Manufacturing Execution System* (MES) übernimmt die Kontrolle und Steuerung der Feldebene. Es verbindet die Unternehmensebene mit der Steuerungsebene und sorgt zwischen den Ebenen für einen Datentransfer. Neben der Steuerung der Feldebene dient das MES als Bindeglied zwischen erfassten Maschinendaten und der Unternehmensebene.
- **Ebene 5** Unternehmensebene (*Topfloor*): *Enterprise-Resource-Planning* (ERP)-Systeme übernehmen die Grobplanung und Bestellabwicklung der Fertigung. Im ERP-System werden alle wichtigen Produktionsressourcen abgebildet.

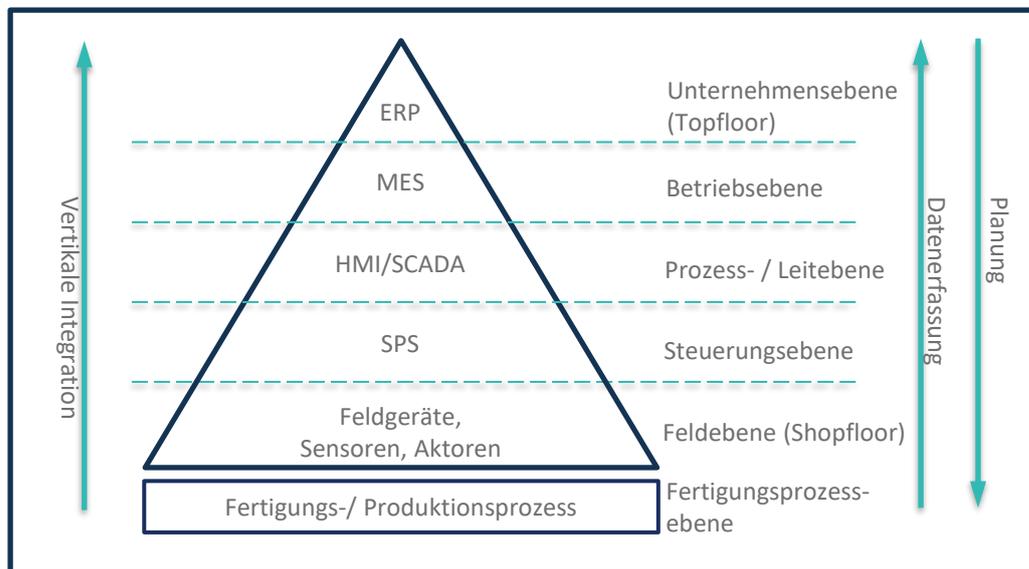


Abbildung 2.1: Automatisierungspyramide in Anlehnung an [SG16; VD13]

Um zukünftigen Trends wie der kundenindividuellen Produktion gerecht zu werden, müssen I4.0-fähige Produktionsansätze nach Siepmann im Gegensatz zu klassischen Automatisierungspyramiden die folgenden Paradigmen und Voraussetzungen erfüllen [SG16]:

- **Vertikale und horizontale Integration.** Durch einheitliche Standards wird eine durchgängige Kommunikation zwischen den Hierarchieebenen eines Unternehmens (vertikale Kommunikation) sowie die Vernetzung über die Unternehmensgrenzen hinweg (horizontale Kommunikation), bspw. mit Dienstleistern, Kunden und Lieferanten, erreicht.
- **Dezentrale Intelligenz.** Produktionsmittel und -anlagen können ortsunabhängig und individuell benötigte Informationen an ein dezentrales Steuerungssystem weitergeben, um basierend auf diesen Informationen Entscheidungen zu treffen.
- **Dezentrale Steuerung.** Verteilte Produktionssysteme ebnen den Weg zu einer flexiblen und ortsunabhängigen Produktion.

- **Durchgängiges, digitales Engineering.** Die vollständige digitale Abbildung aller verfügbaren Ressourcen ermöglicht das nahtlose Ineinandergreifen von physischen und virtuellen Prozessen.
- **Cyber-physische Produktionssysteme (CPPS).** Beschreiben die gesamte Industrie 4.0 Produktionsanlage, die eine individuelle Fertigung von Produkten ermöglicht (Definition in Kapitel 2.1.2).

Um den aufgelisteten Paradigmen gerecht zu werden, muss die klassische hierarchische Steuerungsarchitektur im Rahmen von Industrie 4.0 zu einer dezentralen Architektur transformiert werden. In dieser Weiterentwicklung wird sich die strikte Trennung der einzelnen Ebenen der Automatisierungspyramide sukzessiv auflösen [Ho14; SG16]. Die Systeme der Automatisierungsebenen werden nach Kleinemeier allerdings nicht verschwinden, sondern viel wichtiger wird die nahtlose Verbindung der Systeme miteinander [KI14]. Somit setzt das MES auch in Zukunft weiterhin den Fokus auf die lokale Maschinensteuerung, ERP-Systeme werden hingegen globale Standorte über Firmengrenzen hinweg vernetzen [KI14]. Damit dezentrale Strukturen und die Produktion kundenindividueller Bauteile möglich werden, müssen Produktionssysteme in Zukunft autonom handeln [KI17]. Scholz-Reiter und Freitag beschreiben, dass zum Erlangen einer Autonomie die Unabhängigkeit der Systeme von ihrer Umgebung sowie die Fähigkeit der eigenständigen Steuerung benötigt wird [SF07]. Als Lösung schlagen Sie vor, die Fertigungsebene in Subsysteme mit standardisierten Schnittstellen zu untergliedern sowie deren Steuerung aufzuteilen. Die Subsysteme erhalten gewisse Freiheiten auf Änderungen zu reagieren und sich anzupassen, somit wird der Autonomiegrad erhöht [SF07]. Entsprechend werden in Abbildung 2.2 die möglichen Steuerungsstrukturen skizziert. Diese bestehen aus den klassischen zentralen Steuerungsarchitekturen sowie dezentralen oder autonomen Architekturen.

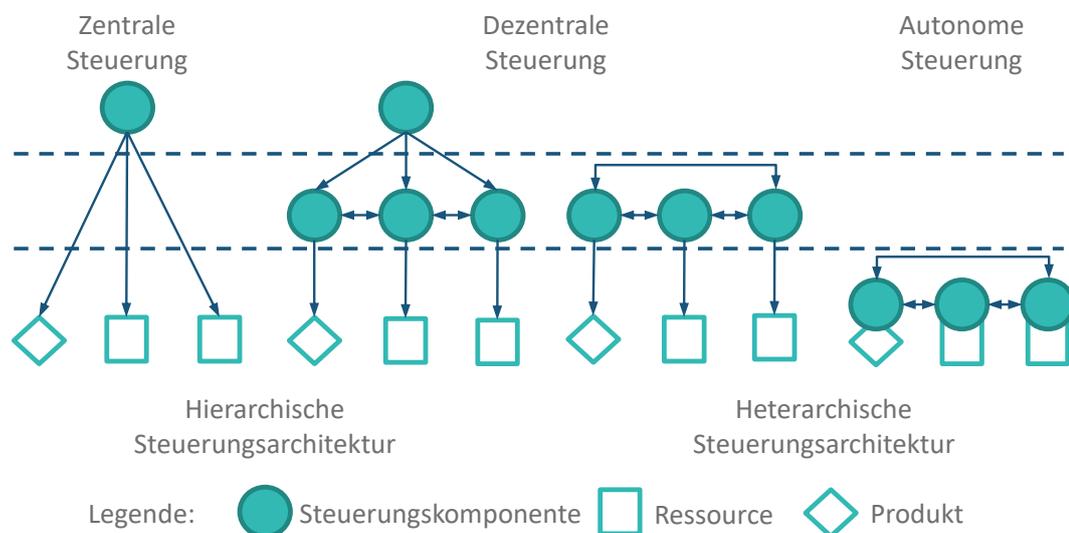


Abbildung 2.2: Konzepte zur Gestaltung von Steuerungsarchitekturen in Anlehnung an [Ko18; Lo13; SF07; Tr09]

In der zentralen hierarchischen Steuerungsarchitektur übernimmt eine Steuerung die Kontrolle über den Produktfluss und die Ressourcen der Produktion. In dezentralen Architekturen kommunizieren mehrere Steuerungseinheiten miteinander und geben Befehle an die Produktionsumgebung. Als autonome Steuerung beschreibt Scholz-Reiter und Freitag die Entwicklung im Rahmen des IoT [SF07]. Jedes Objekt erlangt die Fähigkeiten identifiziert und lokalisiert zu werden sowie mit anderen Teilnehmern zu kommunizieren. Erfüllt ein Asset diese Eigenschaften, wird von einer I4.0-Komponente gesprochen (siehe Kapitel 2.1.1). Bisher fehlt es an den standardisierten Schnittstellen, sodass sich Systeme autonom verbinden und Informationen austauschen können. Die autonome Steuerung ermöglicht, dass die teilnehmenden Systeme eigenständig oder im Verbund Entscheidungen treffen können, um ein gewünschtes Ziel zu erreichen. Bei der Implementierung von autonomen Steuerungen (z. B. durch Agentensysteme) hat sich gezeigt, dass komplett verteilte Steuerungsarchitekturen kaum umsetzbar sind und daher häufig eine hybride Form aus hierarchischen und heterarchischen Architekturen gewählt wird [Lo13; MVK06]. Somit sollte auch im Rahmen von I4.0 eine grundlegende Architektur vorgegeben werden, in den entsprechenden Ebenen treffen Agenten die benötigte Entscheidungen.

### 2.1.1 Verwaltungsschale / I4.0-Komponente / I4.0-Sprache

Innerhalb von RAMI 4.0 wird zur zuverlässigen Identifikation und Kommunikation die Industrie 4.0-Komponente (I4.0-Komponente) definiert [DIN SPEC 91345]. Die I4.0-Komponenten kommunizieren auf Basis einer serviceorientierten Architektur und einer gemeinsamen I4.0-konformen Semantik.

*„Die **Industrie 4.0-Komponente** (I4.0-Komponente, I4.0 component) ist ein weltweit eindeutig identifizierbarer, kommunikationsfähiger Teilnehmer bestehend aus Verwaltungsschale und Asset [...] mit digitaler Verbindung innerhalb eines I4.0-Systems [...], welcher dort Dienste mit definierten Quality of Service (QoS)-Eigenschaften anbietet“* [DIN SPEC 91345].

*Die **Verwaltungsschale (VWS)** umfasst „die relevanten Informationen zur Repräsentation des Assets durch Informationen, einschließlich seiner fachlichen Funktionalität. Sie stellt der Informationswelt die nach RAMI 4.0 strukturierten Informationen über das Asset bzw. über mehrere Assets zur Verfügung“* [DIN SPEC 91345].

*Ein **I4.0-System** setzt sich aus I4.0-Komponenten sowie Komponenten geringerer Informations- und Kommunikationsfähigkeit zusammen. Das I4.0-System dient einem bestimmten Zweck, weist definierte Eigenschaften auf und unterstützt standardisierte Dienste und Zustände, des Weiteren kann ein I4.0-System gekapselt als I4.0-Komponente auftreten.* [DIN SPEC 91345]

„Die **Semantik** beschreibt die Interpretation, d. h. den Bedeutungsgehalt einer Sprache“ [Fe89].

Abbildung 2.3 zeigt die grafische Darstellung eines Assets, das in Kombination mit einer zugehörigen VWS zu einer I4.0-Komponente wird.



Abbildung 2.3: Darstellung eines Assets, das mit einer VWS eine I4.0-Komponente bildet

Die Implementierung einer VWS beruht auf keiner spezifischen Technologie [Ba16], somit handelt es sich um eine generische Lösung, die auf unterschiedlichen Standards aufgesetzt werden kann. Die Informationen eines Assets werden in Teilmodellen (TM) oder im Englischen als *Sub-models (SM)* bezeichnet und durch verschiedene definierte Datentypen beschrieben [PI20a]. Es existieren nach aktueller Definition der Plattform Industrie 4.0 drei Typen von VWS [BD19; PI20b]:

- **Passive VWS (Typ 1):** Die VWS ist als Datei hinterlegt (im XML oder JSON Format) und enthält die Beschreibung des jeweiligen Assets anhand der standardisierten TM.
- **Reaktive VWS (Typ 2):** Auf eine I4.0-Komponente kann mittels *Application Programming Interface (API)* zugegriffen werden, um die standardisierten TM abzurufen. Somit wird die vertikale Integration des Assets ermöglicht.
- **Proaktive VWS (Typ 3):** Eine VWS, die zusätzlich über einen Interaktionsmanager verfügt, der die horizontale Integration sowie ein zielgerichtetes Verhalten durch Entscheidungs- und Optimierungsalgorithmen ermöglicht. Untereinander können I4.0-Komponenten über die I4.0-Sprache kommunizieren.

In Abbildung 2.4 wird die Einordnung der 3 genannten VWS-Typen in die Informationsebenen von RAMI 4.0 gezeigt. Die Definition der Verwaltungsschale kann unter [*@PlattformI4.0*] eingesehen werden. Aktuell existieren Spezifikationen für die VWS Typ 1 und 2 sowie die I4.0-Sprache zur Vernetzung von I4.0-Komponenten mit den Normen VDI/VDE 2193-1:2020 und VDI/VDE 2193-2:2020. Die I4.0-Sprache kann verwendet werden, um benötigte Bearbeitungsschritte zwischen I4.0-Komponenten zu evaluieren und so benötigte Ressourcen für die Fertigung zu identifizieren [PI18].

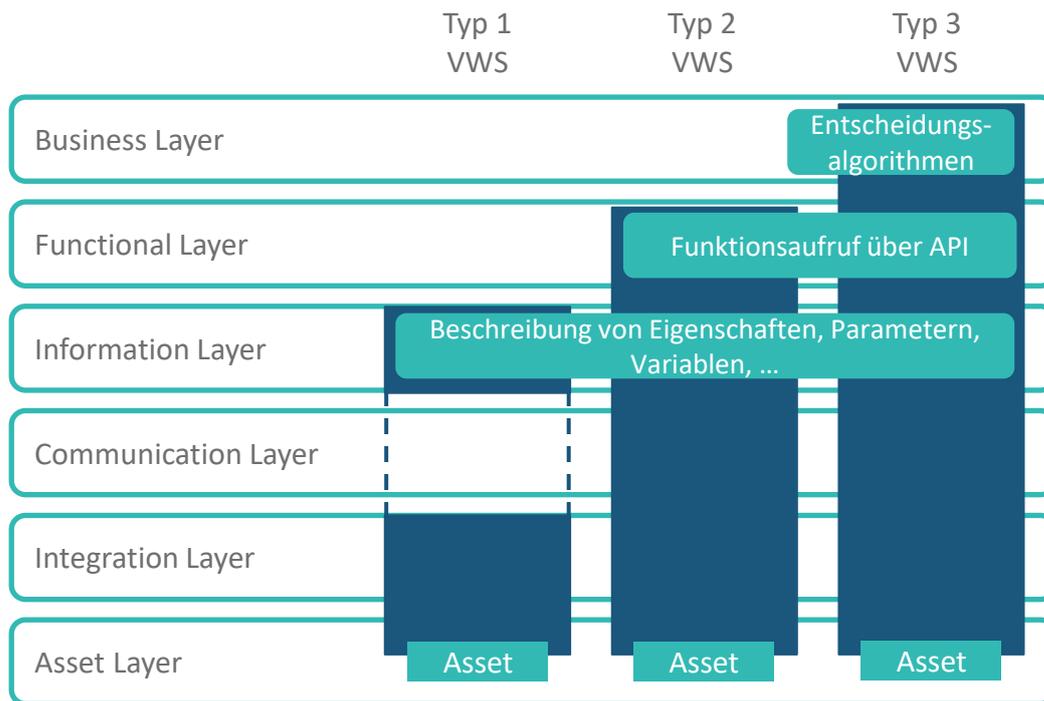


Abbildung 2.4: Verschiedene Typen der VWS eingeordnet in die strukturellen Schichten von RAMI 4.0 in Anlehnung an [BD19; PI20b]

Yli-Ojanperä analysiert Steuerungsarchitekturen, zwar bilden einige die Kernaspekte von Industrie 4.0 teilweise oder vollständig ab, allerdings wird selten ein Bezug zu RAMI 4.0 hergestellt [YI19]. Daher werden im Folgenden die Konzepte Cyber-physischer Systeme und verteilter Systeme betrachtet, die das Potenzial mitbringen, die Vorgaben von RAMI 4.0 sowie Anforderungen an I4.0-Systeme teilweise oder vollständig abzudecken.

### 2.1.2 Cyber-physische Systeme

Neben den in RAMI 4.0 erwähnten I4.0-Komponenten und I4.0-System wird im Kontext von Industrie 4.0 häufig der Begriff Cyber-physisches System (CPS) verwendet. CPS sind ein wesentlicher Treiber der modularen Produktion und bieten Potential, in Zukunft das kosteneffiziente Produzieren kleiner Losgrößen zu ermöglichen [We18]. Im Folgenden werden CPS in Anlehnung an [Br10; Co17; LS15] definiert:

*Ein **Cyber-physisches System (CPS)** ist ein eingebettetes System mit physischer Komponente, integrierter Recheneinheit und Logik, das über eine geeignete Systemschnittstelle zur Kommunikation mit dem globalen Netzwerk sowie über eine Benutzerschnittstelle verfügt.*

Assets auf dem RAMI 4.0-Hierarchielevel „Technischer Anlagen“ (*Work Centers*) können durch Konzepte von CPS umgesetzt werden. Realisierte, verteilte Produktionsanlagen werden als Cyber-physische Produktionssysteme (CPPS) bezeichnet. CPPS sind modular aufgebaut und set-

zen sich aus Cyber-physischen Produktionsmodulen (CPPM) zusammen (siehe Abbildung 2.5). CPPM sind somit ein Asset, das im Hierarchielevel „Produktionsmodul“ (*Station*) von RAMI 4.0 einzuordnen ist. CPPS und CPPM basieren grundsätzlich auf CPS. Die relevanten Begrifflichkeiten werden im Folgenden erläutert. In Anlehnung an [KHM18; RB18; RH18] wird ein CPPM definiert:

**Cyber-physische Produktionsmodule (CPPM)** setzen sich aus mehreren Komponenten (z. B. Aktoren, Sensoren, CPS), einer Steuerung und einer cyber-physischen Präsentation des Moduls zusammen. Das CPPM bietet cyber-physische Schnittstellen und Algorithmen, um mit menschlichen Ressourcen, weiteren CPPM und Subsystemen interagieren zu können. Die Fähigkeiten des Moduls werden in einer abstrahierten Form dargestellt, um diese von technischen Details zu entkoppeln.

In Anlehnung an [KI17; Mo14; RB18; RH18] wird ein CPPS wie folgt definiert:

Ein **Cyber-physisches Produktionssystem (CPPS)** ist die Vernetzung von CPPM, menschlichen Ressourcen, Subsystemen und Produkten über cyber-physische Interaktionsschnittstellen, die der Überwachung, dem Betrieb des CPPS und der Nutzung von generiertem Wissen über den gesamten Produktionszeitraum dient. Das interne Wissen dient der kontinuierlichen Verbesserung des CPPS.

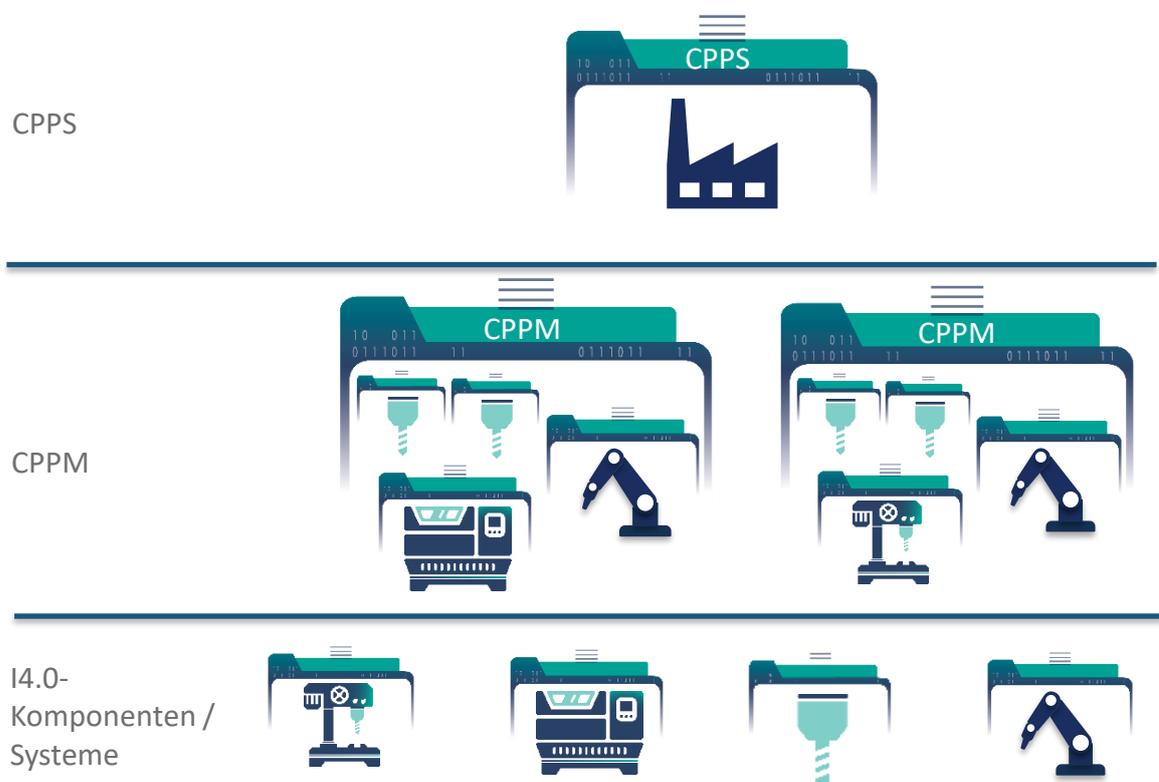


Abbildung 2.5: Hierarchische Gliederung einer Industrie 4.0 fähigen Produktion in CPPS, CPPM und I4.0-Komponenten in Anlehnung an [KHM18]

Der Hauptunterschied zwischen CPS und eingebetteten Systemen ist die offene Schnittstelle, die einen Austausch von Informationen zwischen CPS mit dem Menschen ermöglicht [Be20b]. Wird ein CPS durch eine VWS beschrieben, sind alle benötigten Informationen des CPS über die Informationsschicht von RAMI 4.0 für weitere Assets sowie den Menschen zugänglich. Ist also ein CPS mit einer VWS ausgestattet, wird es zur I4.0-Komponente, gleiches gilt für das CPPM. Eine I4.0-Komponente ist somit ein spezieller Typ eines CPS [VVZ15]. Des Weiteren kann ein CPPM als I4.0-System bezeichnet werden, wenn es sich aus mehreren Komponenten und I4.0-Komponenten zusammensetzt. Aufgrund der Regeln der Schachtelbarkeit und Kapselungsfähigkeit von I4.0-Komponenten in RAMI 4.0 [DIN SPEC 91345], kann ein I4.0-System zur I4.0-Komponente eines übergeordneten Systems werden. In einer I4.0-konformen Produktion ist das CPPM somit eine I4.0-Komponente des CPPS. Damit das CPS zu einer I4.0-Komponente wird, muss es eine VWS besitzen. Zur Einbindung in ein System ergeben sich somit drei Möglichkeiten, je nachdem welcher Typ der VWS eingesetzt wird. Das CPS sowie seine Schnittstellen zur Umwelt wird in einer VWS mindestens vom Typ 1 beschrieben. Wird die Schnittstelle in der VWS beschrieben, können z. B. Agentensysteme anhand der vorhandenen Informationen eine Kommunikation mit dem System aufbauen. Als direkte Schnittstelle zum CPS kann auch eine Typ 2 oder Typ 3 VWS eingesetzt werden. In diesem Fall ist ein Verbindungsaufbau zum CPS über eine Schnittstelle (z. B. basierend auf REST (*Representational State Transfer*)) oder über ein Interaktionsprotokoll (z. B. I4.0-Sprache) möglich (siehe Abbildung 2.6).

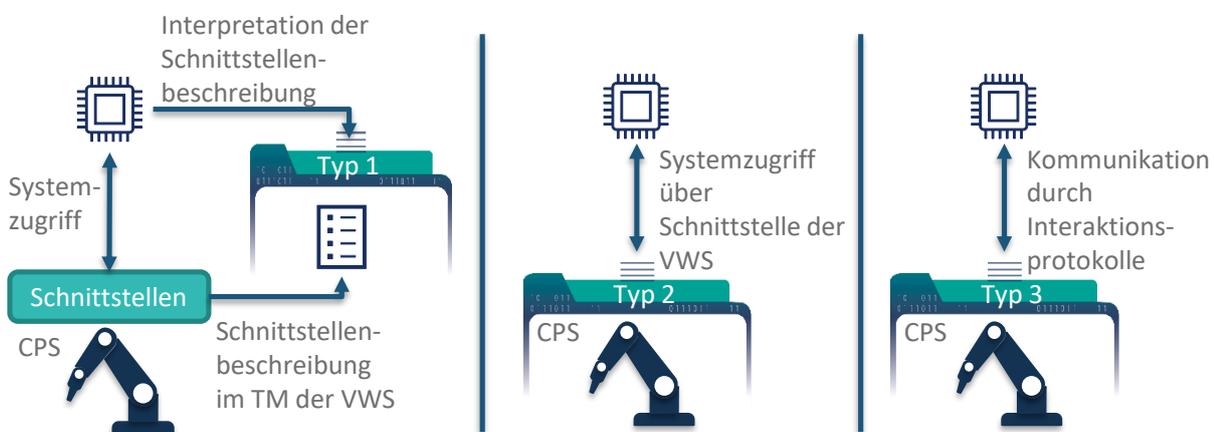


Abbildung 2.6: Zugriffsmöglichkeiten auf CPS als I4.0-Komponenten

### 2.1.3 IEC 61499

In der Norm IEC 61499 und DIN EN 61499 werden Leitlinien für ein Konzept eines verteilten industriellen Leitsystems auf Basis von Funktionsbausteinen definiert. Teil 1 der Normen beschäftigt sich mit der Architektur der industriellen Leitsysteme .

*Ein Funktionsbaustein ist eine „funktionale Software-Einheit, die eine mit einem Namen versehene Kopie einer Datenstruktur und zugeordnete Operationen umfasst, die durch einen entsprechenden Funktionsbausteintyp festgelegt“ ist [DIN EN 61499-1].*

Die Norm 61499-1:2012 bietet Referenzarchitekturen, Modelle und Leitlinien, um verteilte Systeme von der Prozess- bis zur Ressourcenebene auf Basis von Funktionsbausteinen zu entwickeln. Ein Funktionsbaustein hat definierte Ereignisseingänge und -ausgänge, die einen Ereignisfluss ermöglichen sowie definierte Dateneingänge und -ausgänge, die einen Datenfluss definieren. Des Weiteren kann ein Funktionsbaustein Zugriff auf die Fähigkeiten einer Ressource haben. Hierbei kann es sich bspw. um das Ausführen von Plänen oder die Abbildung von Prozessen handeln. Durch das Triggern der Ereignisseingänge wird ein Zustandsautomat im Inneren der Funktionsbausteine gesteuert. Mit Hilfe der übertragenen Daten wird durch implementierte Algorithmen die gewünschte Aktion umgesetzt. Somit kann auf physikalisch verteilter Steuerungshardware, wie SPS oder Mikrocontrollern sowie auf unterschiedlichen Ebenen eine ereignisorientierte statt einer zyklisch ablaufenden Prozesssteuerung umgesetzt werden [Lo13]. Die Funktionsblöcke sind zum einen in unterschiedliche Ebenen, zum anderen durch die Definition von Basisfunktionsbausteinen und zusammengesetzten Funktionsbausteinen schachtelbar. Die höchste Architekturebene ist die Geräteebene. Ein Gerät kapselt mehrere Ressourcen über eine definierte Schnittstelle. Innerhalb der Ressourcen sind die Funktionsblöcke zur Ausführung von Operationen zugeordnet. Die *Open-Source-Software* 4diac ermöglicht, Steuerungen auf Basis der Normung zu programmieren [@4diac].

#### 2.1.4 OPC UA

*Open Platform Communication Unified Architecture* (OPC UA) ist ein von der OPC Foundation [@OPC-Foundation] veröffentlichtes Kommunikationsframework, das den herstellerunabhängigen Datenaustausch vorantreibt. OPC UA wird das Potential zugesprochen, zu einem der De-facto-Standards der Industrie 4.0 Kommunikation zu werden [Bu19; Ho14; Sc16] und hat sich bei einigen Komponenten bzw. Maschinen als herstellerunabhängige Schnittstelle durchgesetzt. Wie aus Abbildung 2.7 hervorgeht, ist OPC UA als mehrschichtiges Framework aufgebaut. Dieses unterstützt die Kommunikation über Client-Server als auch über Publish-Subscribe (Pub-Sub) Architekturen und bietet verschiedene Kommunikationstechnologien [@OPC-Foundation; Bu19]. Security und Authentifizierungsaspekte werden im Framework bereits beachtet [@OPC-Foundation; Bu19]. Die Security von OPC UA wurde vom Bundesamt für Sicherheit in der Informationstechnik (BSI) überprüft und bestätigt [Bu20]. Einer der großen Vorteile, die durch OPC UA entstehen, ist die Möglichkeit, Informationsmodelle zu beschreiben. Standardisierte Informationsmodelle (*Companion Specifications*) werden innerhalb der OPC Foundation [@OPC-Foundation], teilweise auch bei oder mit Partnern z. B. im VDMA (Verband Deutscher Maschinen- und Anlagenbau e. V.) [@VDMA], in unternehmensübergreifenden Arbeitsgruppen definiert. Diese standardisieren bspw. die Beschreibung bestimmter Hardware oder die Über-

führung existierender Standards in OPC UA. So beschreibt z. B. die *Companion Specification* für Industrieroboter [OPC 40010-1], wie das OPC UA-Informationsmodell eines Industrieroboters zur vertikalen Integration in eine Produktionsumgebung modelliert werden muss. Alle spezifizierten Informationen eines Industrieroboters können mit Hilfe des OPC UA-Servers abgerufen werden. Im Bereich der Werkzeugmaschinen existieren Spezifikationen für CNC-Systeme [OPC 40502], Werkzeugmaschinen [OPC 40501-1] und Maschinen [OPC 40001-1]. Ein Überblick über existierenden Spezifikationen ist unter [ @OPC-References ] einsehbar.

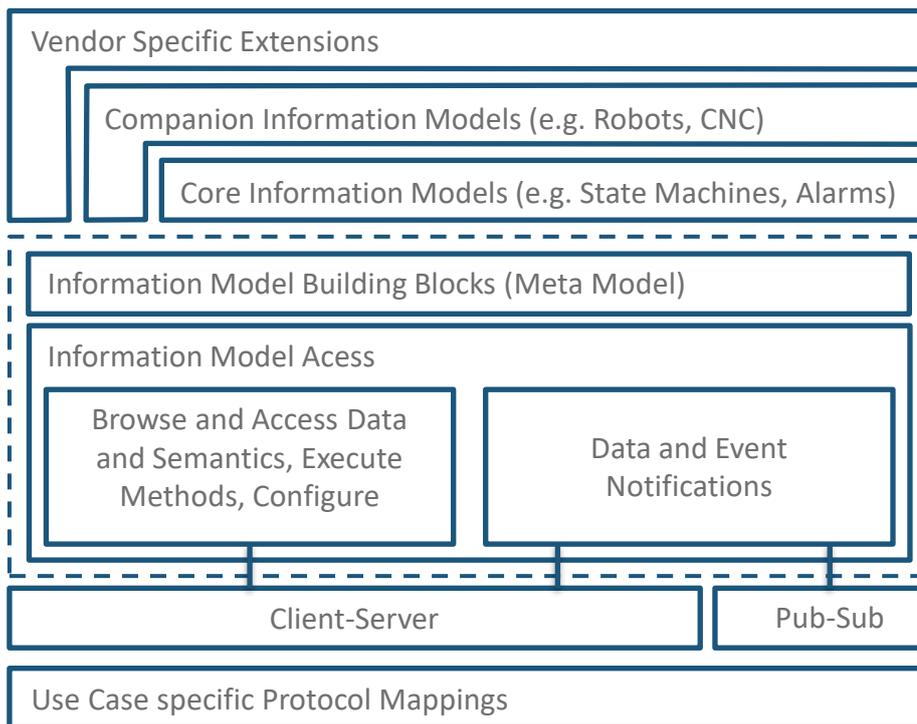


Abbildung 2.7: Architektur des OPC UA-Frameworks in Anlehnung an [ @OPC-Foundation ]

### 2.1.5 Namur open Architecture (NoA)

Die Namur Interessengemeinschaft Automatisierungstechnik der Prozessindustrie e. V. hat als Ziel, die Erfassung von Produktionsdaten in der Prozessindustrie zu vereinfachen [ @NAMUR ]. Besonders zu erwähnen ist die erarbeitete VDI/VDE/Namur 2658, die Schnittstellen modularer Anlagen in der Prozessindustrie standardisiert. Die Norm soll dazu beitragen, sowohl die Planungszeiten von Neuanlagen als auch Aufwände im Umbau bzw. der Sanierung von neuen Anlagen zu reduzieren. Hierzu werden *Module Type Packages* (MTP) definiert, die Schnittstellen und Funktionen der Moduleinheiten beschreiben. In VDI/VDE/Namur 2658 werden Dienste als Funktionsblöcke definiert, die zur Ansteuerung durch einen *Process Orchestration Layer* (POL) dienen. Die Norm ist spezifisch für die Prozessindustrie ausgelegt.

## 2.2 Aktuelle Forschung im Bereich der flexiblen Fertigung

Ein weiterer Ansatz zur verteilten Produktion ist die capability- oder skillbasierte Produktion. Die Softwarearchitekturen werden ebenfalls zur Kapselung von unternehmensinternen Prozessen verwendet und ähneln dem Konzept aus IEC 61499 (siehe Kapitel 2.1.3). Skillbasierte Ansätze werden häufig in der Forschung eingesetzt, um implementierte Fähigkeiten einer Ressource (bspw. eines CPPM) zu kapseln und diese, ähnlich zu einem Funktionsblock auf Ressourcenebene in IEC 61499, über eine definierte Schnittstelle anzubieten. Durch die Kapselung können gewünschte Funktionen auf der Feldebene umgesetzt werden. Bei skillbasierten Architekturen spielt vorwiegend die Zuordnung von vorhandenen Skills und Capabilities auf die vorhandene Prozess- oder Produkthanforderung eine große Rolle, um individuellere Prozessabläufe zu ermöglichen. Produktionsressourcen werden in diesem Ansatz entwickelt, um Skills und Capabilities in Teilschritten eines Prozesses bereitzustellen und umzusetzen, anstatt die Ressourcen für einen spezifischen Prozess zu entwickeln [PI20c]. Die Reihenfolge der Fertigungsschritte wird somit erst individuell in der Produktion festgelegt [PI20c]. Die in dieser Arbeit verwendeten Begrifflichkeiten werden wie folgt definiert:

*In Anlehnung an [Mo21; PI20c] ist ein **Skill (Fertigkeit)** die ressourcenabhängige Implementierung zur Realisierung einer Funktion mit einer definierten Schnittstelle und definierten Zuständen.*

*Eine **Funktion** ist „[...] eine Anforderung an eine Einheit, beschrieben durch die lösungsneutrale Beziehung zwischen Eingangs-, Ausgangs- und Zustandsgrößen, die von den Funktionalitäten dieser Einheit erbracht werden soll“ [He21].*

*In Anlehnung an [PI20c] wird eine **Capability (Fähigkeit)** definiert als die implementierungsunabhängige Beschreibung einer Funktion, einen bestimmten Effekt in der physischen oder virtuellen Welt zu bewirken.*

*Eine **Funktionalität** bezeichnet „[...] die Fähigkeit eines Produktes oder einer Komponente, eine bestimmte Funktion oder Gruppe von Funktionen zu erfüllen“ [DIN 40912; He21].*

Das Prozess-Produkt-Ressourcen (PPR)-Modell spielt in skillbasierten Ansätzen eine wesentliche Rolle. In [He20; He21] sind die relevanten Begrifflichkeiten skillbasierter Ansätze in das PPR-Modell eingeordnet (Abbildung 2.8). Diese Einordnung visualisiert den Einsatz von Skills, um Produkthanforderungen auf die vorhandenen Ressourcen zu matchen und einen resultierenden Prozess abzuleiten. Um den Vorgang eines automatisierten Matchings zu ermöglichen, müssen die Capabilities der CPPM eindeutig beschrieben und auf Skills referenziert werden. Implementierte Skills ermöglichen, aktiv eine benötigte Capability zu erfüllen und somit eine Funktionalität auf der Feldebene auszuführen. Bei skillbasierten Steuerungsarchitekturen handelt es sich grundsätzlich um ein verteiltes System, das jederzeit durch Komponenten oder Module mit neuen Capabilities und Skills erweitert werden kann. Eine Middleware (z. B. ein Multiagentensys-

tem [Ru20]) verknüpft die einzelnen CPPM des Gesamtsystems sowie registriert und koordiniert dessen Capabilities und Skills.

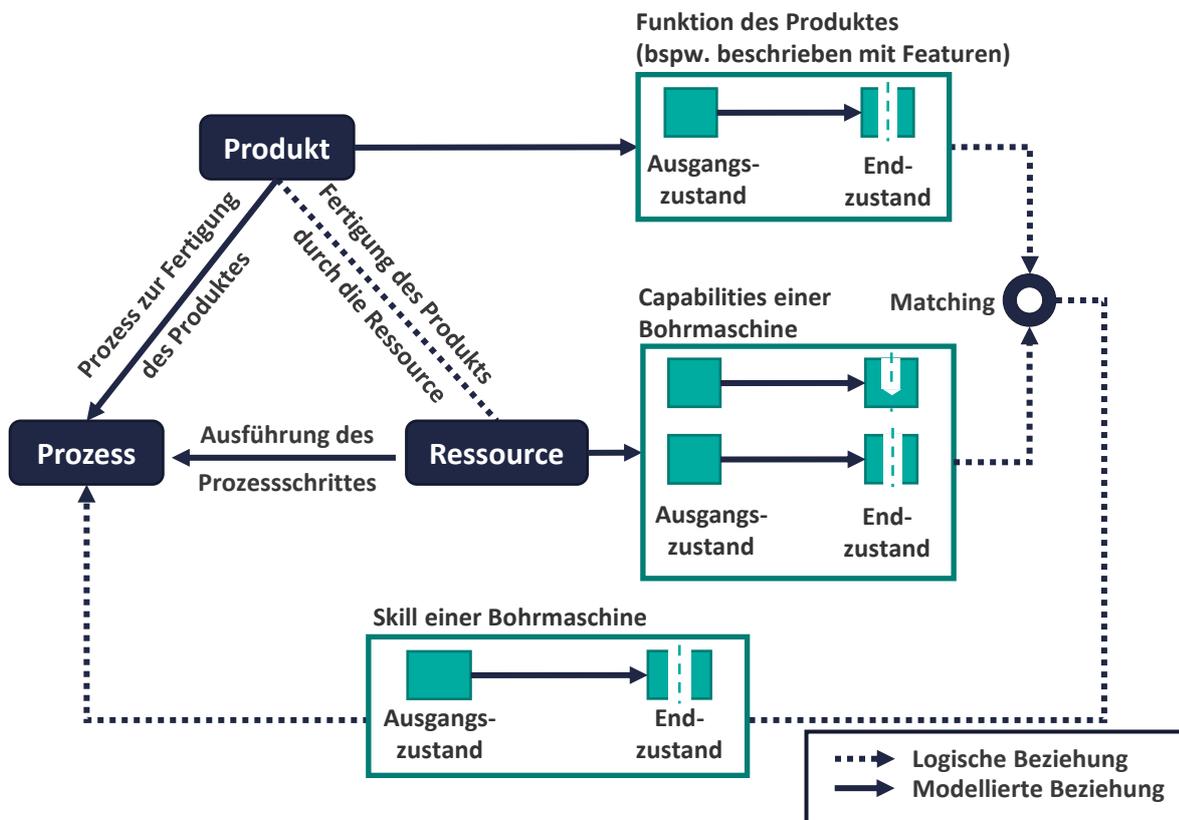


Abbildung 2.8: Zusammenhänge zwischen dem PPR-Modell sowie Capabilities, Skills und Features in Anlehnung an [He21]

In Anlehnung an die eigene Veröffentlichung [Vo20] ergeben sich die folgenden Vorteile durch die Kapselung von Skills:

- Produktionsressourcen können bei Bedarf über ein standardisiertes Interface flexibel angesteuert werden.
- Produktionsaufträge können an verändernde Funktionen und Anforderungen des Produktes angepasst werden.
- Die Capabilities einer Produktionsanlage werden standardisiert und somit „sichtbar“, sie können anschließend auf Skills gemappt werden.
- Maschinenbediener können in der Implementierung von Fertigungsaufträgen auf Skills zurückgreifen, so können Stillstandszeiten reduziert werden.
- Im Gegensatz zu Programmen wie z. B. G-Code ergibt sich eine bessere Übersicht auf durchzuführende Prozessschritte, da Maschinenprozesse mit Skills auf einer höheren Abstraktionsebene beschrieben sind.
- Durch den Einsatz eines einheitlichen Interfaces über alle Produktionsressourcen kann eine serviceorientierte Architektur (SOA) umgesetzt werden.

- Skills können auf ähnlichen Maschinen implementiert werden. Somit bietet sich eine einfachere Austauschbarkeit von Maschinen. Dies ermöglicht das Potential Ressourcen *Plug & Produce* in eine Produktionsumgebung einzubinden.

Bisher gibt es keine einheitlichen Standards, um Ressourcenfähigkeiten oder Skills zu beschreiben. Häufig werden zum Matching von Ressourcen und Fähigkeiten Ontologien verwendet, um Synonyme und Hierarchien von Fähigkeiten abzudecken sowie alternative Fähigkeiten zu finden, die das gleiche Ergebnis am Produkt erzielen können [PI20c]. Ontologien ermöglichen, in einer logischen Sprache Klassen, Instanzen, Eigenschaften, Beziehungen und Funktionen zu unterscheiden, sodass ein einheitliches Verständnis des Wissens entsteht [MVK06]. Ontologien können auch in unternehmensübergreifenden, verteilten Systemen eingesetzt werden, um die Fertigung von Produkten an unterschiedlichen Standorten zu ermöglichen. Diese Cloud-Architekturen werden im späteren Verlauf des Kapitels definiert.

Im Folgenden wird ein Überblick von bereits umgesetzten Softwarearchitekturen, Forschungsprojekten und Veröffentlichungen gegeben, die auf skillbasierten Konzepten beruhen. Der Begriff *Skill* wird teilweise in den Arbeiten auch als *Action*, *Service*, *Primitive*, *Elementary*, *Manufacturing Feature* oder *Capability* bezeichnet. Einen ausführlichen Überblick über die Entstehung und ersten Konzepte der Verwendung von Skills wird durch Backhaus aufgeführt in [Ba15]. Als erste Veröffentlichung, die eine Zerlegung von Steuerungscode in Aktions- bzw. Fertigungsprimitive vorschlägt, listet Backhaus die Arbeit von Reinisch [Re92]. Reinisch empfiehlt die Entwicklung von Primitiven in Anlehnung an die Funktionalitäten aus VDI 2860:1982 sowie DIN 8593-0:2003 zu entwickeln. Die VDI 2860:1982 ist die Richtlinie für Montage- und Handhabungsaufgaben (Norm ist zurückgezogen, eine Neuauflage ist geplant für 2023 [@VDI 2860 - Projekt]) und DIN 8593-0:2003 die noch existierende Norm für Fügeverfahren. Inzwischen gibt es viele verschiedene Ansätze in Forschungsprojekten, das Konzept von Skills aufzugreifen. Hiervon werden beispielhaft einige in Forschungsprojekten entwickelte Softwarearchitekturen sowie Veröffentlichungen aus einem Zeitraum der ca. letzten 5 Jahre aufgezeigt.

Im EU-Projekt LIAA (*Lean Intelligent Assembly Automation*) wurde eine Middleware auf Basis des *Robot Operating System* (ROS) implementiert. Durch die in ROS verfügbare SMACH Bibliothek [@SMACH] wurden Zustandsmaschinen als Skills hauptsächlich auf kollaborativen Robotern und Vision Systemen implementiert, um Montage- und Handhabungsprozesse durchzuführen [AN17; He17a]. Mit Hilfe von CAD (*Computer-Aided Design*)-Daten, wurden einige der benötigten Skill-Parameter automatisch gesetzt und somit wurde der Programmieraufwand reduziert [AN17].

Ähnlich hierzu ist der Aufbau der Architektur im EU-Projekt ReCaM, das ebenfalls eine Implementierung von Skills mit ROS und SMACH verwendet [He17b]. Dabei wird in [He17b] gezeigt, wie ein Zeitvorteil durch die Verwendung von skillbasierten Ansätzen in der Entwicklung von komplexen Aufgaben, dem Anpassen von Parametern oder dem Wechsel von Komponenten gegenüber herkömmlichen Programmiermethoden entsteht.

Adamson et al. definieren nach IEC 61499 Funktionsblöcke, die verschiedene Handhabungsaufgaben eines Roboters wie verschrauben, greifen oder einsetzen ermöglichen [AWM17]. Diese Operationen werden als *Manufacturing Features* bezeichnet und setzen sich aus Eigenschaften (*Properties*) der Maschinen zusammen. Mit Hilfe der definierten Funktionsblöcke können die Features parametrisiert und die Steuerung ausgeführt werden. In der Ausarbeitung wird zur Definition der Features weder ein Standard noch eine Norm gewählt. Der Fokus liegt auf der Möglichkeit, eigens programmierte Funktionsblöcke in das System zu integrieren. Das Hinzufügen neuer Funktionsblöcke reduziert sich in diesem Fall allerdings auf Roboter, die mit Hilfe der Programmiersprache Rapid (ABB Roboter) angesteuert werden, und bietet somit keine herstellerunabhängige Lösung. Um eine zuverlässige Zusammenführung und ein Matching der benötigten Features zu ermöglichen, wird die Entwicklung einer Ontologie vorgeschlagen.

Ein „Betriebssystem für Produktionsanlagen“ wird in Projekten rund um BaSys erforscht [@BaSys]. In den Projekten wurde die Open-Source verfügbare Software BaSyx entwickelt [@BaSyx]. Die Software ermöglicht I4.0-Komponenten über eine BaSys40-Dienstschnittstelle und Middleware miteinander zu verbinden und diese flexibel anzusteuern [@BaSyx; GWE18]. Die Komponenten werden ebenfalls über Zustandsautomaten gesteuert, zu deren Implementierung es in [GWE18] Empfehlungen und Festlegungen gibt. Die im BaSys 4.0 Metamodell [GWE18] definierte Festlegung beschreibt, dass die Betriebszustände einer Komponente auf den Standard PackML [ANSI/ISA-TR88.00.02] abbildbar sein müssen. Auch wenn der Begriff Skill bzw. Capability in der Ausarbeitung des Metamodells nur beiläufig fällt, veröffentlicht [Ma18a] im Rahmen des Projektes eine Ausarbeitung, wie Capabilities (in der Ausarbeitung als Skills bezeichnet) in das PPR-Modell eingebunden werden können. In [Ma18a; Pe19] wird die semantische Beschreibung von Capabilities und Skills sowie deren Matching auf die Produkthanforderungen diskutiert. Beide Veröffentlichungen weisen auf die Wichtigkeit der eindeutigen Beschreibung und den Bedarf einer Standardisierung hin. In [El19] wird die Möglichkeit aufgezeigt, Capabilities sowie Skills mit benötigten Produkthanforderungen gegeneinander zu matchen. [Mo21] zeigt die Möglichkeit eines Matchingverfahrens anhand der Beschreibung von Produkthanforderungen auf definierte Capabilities. Anschließend wird anhand des Matchings ein Produktionsplan als Petri-Netz modelliert, um die benötigten Skills auf der Feldebene auszuführen [Mo21].

Im Forschungsprojekt FabOS wird an einem echtzeitfähigen, flexiblem Produktionsbetriebssystem entwickelt, das die Austauschbarkeit von Maschinen und Softwareservices unterstützt [@FabOS]. So ist unter anderem BaSyx in das System integriert, um einen herstellerunabhängigen Datenaustausch zu ermöglichen [@FabOS-Components].

Das durch die Europäische Kommission geförderte Projekt openMOS [@openMOS; Da17; Da18; Do17; DW19; Mi18] beschäftigt sich mit einer offenen *Plug & Produce* Architektur von Automatisierungssystemen. Eine Middleware, im Projekt als *Manufacturing Service Bus* bezeichnet, koordiniert mit Hilfe eines *SkillRecipes* den Produktionsprozess. So kann jedem Produktionsschritt ein entsprechendes Produktionsmodul mit den benötigten Fähigkeiten zugewiesen werden. Ein

Adapter erstellt, sobald ein Modul in die Produktionsanlage eingebunden wird, mit Hilfe der Informationen eines hinterlegten *AutomationML*-Modells, automatisch einen entsprechenden OPC UA-Server. Über diesen können Modulinformationen abgerufen und das Modul aktiv gesteuert werden. Somit wird die Einbindung eines neuen Moduls automatisch mit den benötigten Skills im Gesamtsystem ermöglicht. Im Projekt wird ein Skill-Modell verwendet, das auf der OPC UA-Spezifikation 10 für OPC UA-Programme [OPC 10000-10] beruht und somit ein Zustandsmodell mit vier Zuständen abbildet sowie über OPC UA-Methoden gesteuert wird [DZ18]. Die OPC UA-Spezifikation 10 bietet somit eine Möglichkeit, Skills OPC UA-konform zu implementieren.

Ähnlich zu dem in openMos verwendeten Modell, wird in [Zi19] ein von der Arbeitsgruppe *Integrated Assembly Solutions* (IAS) des VDMA erarbeitetes *Skill Execution Metamodel* vorgestellt. Dieses besteht aus einer Zustandsmaschine mit den vier Zuständen *Idle*, *Executing*, *Suspended* und *Locked*. Für die Automatica 2018 wurde ein Demonstrator aufgebaut, der auf einem hierarchischen Skill Konzept beruht. Dabei werden die Skills einer höheren Ebene über ein OPC UA-Netzwerk geteilt und ermöglichen eine herstellerübergreifende Steuerung von Montageaufgaben [BD18; Zi19]. In [Do19] wird das vom IAS vorgestellte Skill-Modell in seiner Leistungsfähigkeit mit dem traditionellen hierarchischen Steuerungsansatz verglichen und aufgezeigt, dass durch den erhöhten Kommunikationsaufwand leichte Defizite in den Prozesszeiten entstehen. Positiv geht hervor, dass Skills Vorteile in der Flexibilität von Prozessen oder beim Austausch von Komponenten bieten [Do19].

Aus der Recherche geht hervor, dass Projekte den Begriff Skills meistens mit flexiblen Montage- und Handhabungsaufgaben assoziierten. Im Rahmen der aktuellen Entwicklung rund um Industrie 4.0 ist allerdings erkennbar, dass nicht nur Robotersysteme mit Skills ausgestattet werden, sondern Fertigungsressourcen durch Skills herstellerunabhängig ansteuerbar werden. Dabei wird in der spannenden Bearbeitung häufig der Begriff Fertigungs- oder Bearbeitungsfeature für die Skillbeschreibung gewählt (siehe Kapitel 2.3). Bisher fehlen Umsetzungen, die sowohl eine einheitliche Semantik, herstellerübergreifende Schnittstellen und eine gemeinsame Produktion über unterschiedliche Standorte ermöglichen. Der Ansatz, die mechanische Bearbeitung von Bauteilen flexibler zu gestalten, wird in Kapitel 2.3 genauer untersucht.

### 2.2.1 Multiagentensysteme und holonische Systeme

Einen Aufbau von verteilten Systemen, die in der Lage sind, Informationen untereinander auszutauschen, bieten agentenbasierte Ansätze. Agenten werden bereits seit mehr als zwei Jahrzehnten in Domänen wie z. B. der Produktionsautomatisierung, Logistik oder in der Energietechnik eingesetzt [Le16]. Im Produktionsumfeld werden Agentensysteme hauptsächlich auf der ERP- und MES-Ebene verwendet [Cr18; Le16; Vo18]. Meistens werden Agenten gewählt, um eine bessere Flexibilität, Rekonfigurierbarkeit, Zuverlässigkeit sowie Anpassbarkeit eines Systems zu erreichen [Vo18]. Durch den Einsatz von CPS im Kontext von I4.0 wird der Einsatz von Agenten

abwärts bis auf die Feldebene interessant [Vo18]. Ein Agent (siehe Abbildung 2.9) und Multiagentensystem wird in Anlehnung an [JW98; Ka20a; Lo13; MVK06] wie folgt definiert:

*Ein **Agent** ist ein Softwaresystem, das eigenständig (autonom) handelt, um ein vorgegebenes Ziel zu erreichen. Ist ein Agent nicht in der Lage ein Problem eigenständig zu lösen, kann er mit anderen Agenten kommunizieren, um eine Aufgabe kooperativ zu lösen.*

*Ein **Multiagentensystem (MAS)** ist ein Netzwerk aus verteilt vorliegenden Agenten, die in der Lage sind, miteinander zu kommunizieren und in Kooperation ein Problem zu lösen.*

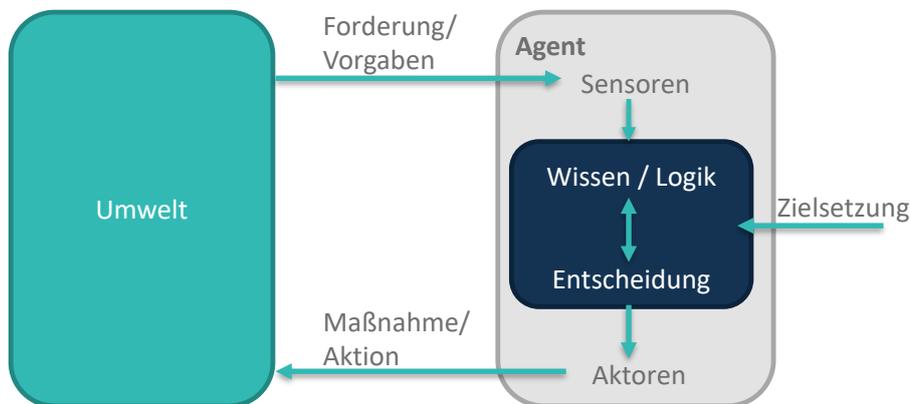


Abbildung 2.9: Interaktion eines Agenten mit seiner Umgebung in Anlehnung an [Ka20a; MVK06; RN21]

Nach [MVK06] bringen Agenten die folgenden Eigenschaften mit sich:

- Agenten agieren im Auftrag ihres Erstellers oder Nutzers und haben die Aufgabe, einen bestimmten Zweck oder ein bestimmtes Ziel zu erfüllen.
- Agenten sind so weit autonom, dass sie ihren eigenen Zustand und ihr Verhalten in der Umwelt kontrollieren können.
- Die Intelligenz von Agenten ermöglicht es, feste Regeln bis hin zu Denk-, Planungs- und Lernfähigkeiten umzusetzen.
- Agenten agieren mit ihrer Umwelt und mit anderen Agenten.
- Im Idealfall sind Agenten adaptiv, sodass sich ihr Verhalten an Veränderungen der Umwelt automatisch anpasst.

Generell verfolgen holonische und Multiagentensysteme ähnliche Ansätze. Dabei haben Agenten ihren Ursprung in der verteilten Künstlichen Intelligenz und holonische Systeme im *Computer Integrated Manufacturing* (CIM) [Ka20a; Lo13]. Die folgenden Definitionen werden in Anlehnung an [Ch94; GB04; Ka20a; Le09; Lo13] für Holone und Holonische Produktionssysteme vorgenommen:

Ein **Holon** ist ein autonomer, unabhängiger, kooperierender Baustein eines Fertigungssystems, der aus einer informationsverarbeitenden Komponente und optional aus einer physischen Komponente besteht. Ein Holon kann sich aus mehreren Holonen zusammensetzen.

Ein **Holonisches Produktionssystem (engl.: Holonic Manufacturing System (HMS))** ist ein Verbund von Holonen, die miteinander kooperieren, um ein Fertigungsziel zu erreichen, und bindet auch Menschen als Ressource in das System ein. Ziel des HMS ist es, den gesamten Fertigungsprozess von der Konstruktion über die Buchung bis hin zur Fertigung abzudecken.

Sowohl Agenten als auch Holone fördern Eigenschaften wie z. B. Autonomie, Kooperation, Reorganisation, Reaktivität und Lernfähigkeit [GB04]. Holone haben allerdings zusätzlich rekursive Eigenschaften, sodass eine hierarchische Gliederung von Holonen vorgesehen ist, um Funktionalitäten von Fertigungssystemen zu kapseln [Ka20a; Lo13]. Agenten werden hingegen typischerweise nicht zur Steuerung physischer Komponenten eingesetzt [Ka20a]. Gründe dafür, dass agentenbasierte Technologien wie MAS und HMS in der Produktion selten zur Anwendung kommen, sind eine mangelhafte Skalierbarkeit, fehlende Echtzeitfähigkeit, ein hoher Kommunikationsaufwand sowie fehlende Softwarewerkzeuge, Standards und Methodiken [Le09; Lo13; Wa03]. Aufgrund der Ähnlichkeit beider Konzepte wird in [Ka20a; MM05] ein Holon definiert, in dem ein Agent die Informationsverarbeitung in der Entscheidungsfindung übernimmt (siehe Abbildung 2.10).

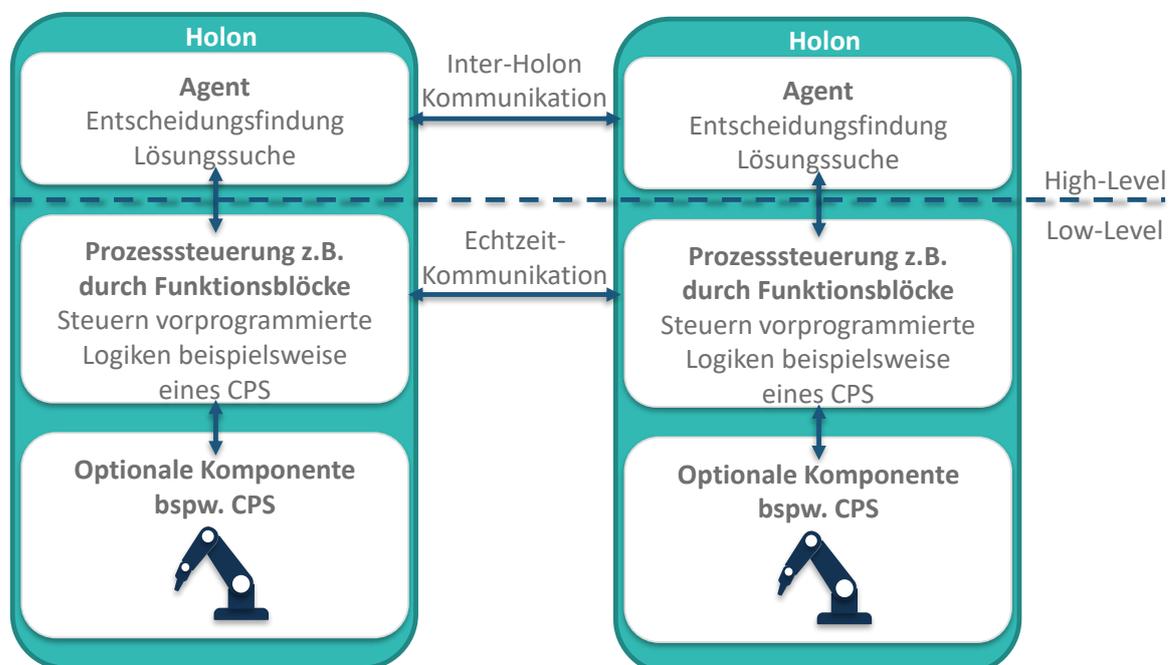


Abbildung 2.10: Agenten übernehmen die Nicht-Echtzeit Kommunikation zwischen Holonen in Anlehnung an [Ka20a; MM05]

Um autonome CPPM in der Produktionsumgebung umzusetzen, bietet sich das hybride Konzept von Holonen mit Agenten an. So können die CPPM als Holon definiert werden, die mit Hilfe von

implementierten Agenten eigenständig Entscheidungen treffen, wie und ob ein individueller Produktionsschritt umgesetzt werden kann. Werkzeugmaschinen, die als Holon eingesetzt werden, sind in Kapitel 2.4 beschrieben.

### 2.2.2 Unternehmensübergreifende verteilte Systeme

Eine Weiterentwicklung der dezentralen Fertigung sind cloudbasierte Fertigungsarchitekturen, die häufig als *Cloud Manufacturing* bezeichnet werden. Ähnlich zu den skillbasierten Architekturen, die sich zunächst auf unternehmensinterne Prozessabläufe spezifizieren, existieren im *Cloud Manufacturing* auch unternehmensübergreifende Ansätze. Diese ermöglichen eine Produktion von Bauteilen über die Unternehmensgrenzen hinweg, als Service anzubieten. Somit wird es in Zukunft ermöglicht, individuelle, dynamisch veränderbare Wertschöpfungsketten zu erstellen. Diese Architekturen werden als *Production as a Service* (PaaS) bezeichnet und basieren auf SOA. [OA06] definiert ein Referenzmodell für SOA. Services werden in diesem Referenzmodell in Bezug auf SOA wie folgt aus [OA06] ins Deutsche übersetzt und definiert:

*Ein **Service** ist ein Mechanismus, der den Zugriff auf eine oder mehrere Fähigkeiten ermöglicht, wobei der Zugriff über eine vorgeschriebene Schnittstelle erfolgt und in Übereinstimmung mit den in der Dienstbeschreibung angegebenen Einschränkungen und Richtlinien ausgeübt wird.*

Der Aufruf der Services erfolgt über eine standardisierte Schnittstelle, sodass die Implementierung programmiersprachenunabhängig ist und der *Service-Requester* keinen Einblick in die konkrete Implementierung bekommt [OA06]. SOA sind ein Architekturparadigma, das interoperable Softwarebausteine und -funktionen modular kapselt, sodass diese als Service wiederverwendet werden können [Ka20a; Lo13]. Aus [OA06] wird die SOA wie folgt übersetzt und definiert:

*Die **serviceorientierte Architektur (SOA)** ist ein Paradigma für die Organisation und Nutzung verteilter Fähigkeiten, die unter der Kontrolle verschiedener Besitzerdomänen stehen können.*

Während SOA vielfach zur Abbildung von Geschäftsprozessen verwendet werden [ZF09], existieren Ansätze, dieses Konzept auf die Produktion anzuwenden. Hierbei werden die Fähigkeiten eines produzierenden Unternehmens als Service gekapselt und einem Kunden auf einer (webbasierten) Plattform angeboten. In diesem Fall wird von PaaS gesprochen. Eine Verknüpfung der Ansätze von PaaS und SOA eröffnet die Möglichkeit einer dynamischen verteilten Produktion für individualisierte Produkte (siehe Abbildung 2.11) [HRR20]. In Anlehnung an [Ba17; Ba18] wird PaaS wie folgt definiert:

***Production as a Service (PaaS)** ist ein webbasiertes Framework, das auf einer SOA basiert und Produktionsaufträge in kleine Schritte (Services) untergliedert, um diese auf verschiedene Unternehmen mit den entsprechenden Fähigkeiten zu verteilen.*

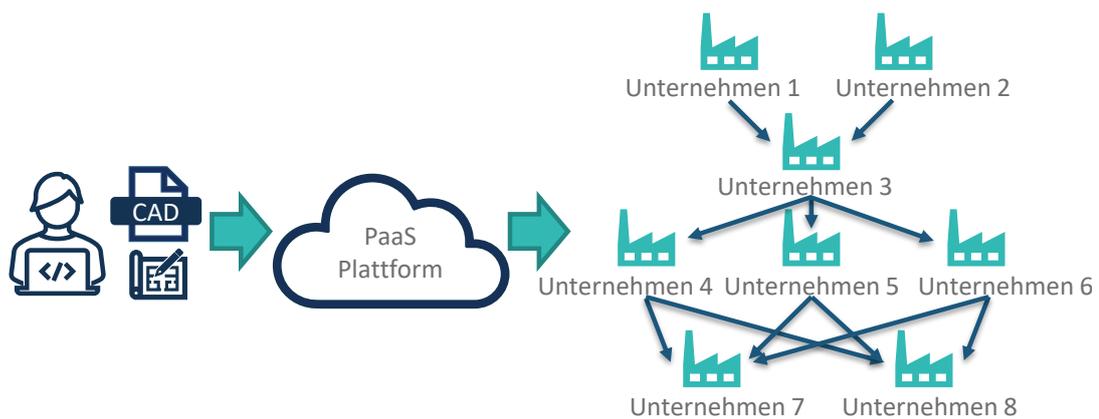


Abbildung 2.11: Bereitstellung von Wertschöpfungsketten durch PaaS-Plattformen in Anlehnung an [He21]

Die Bewahrung des geistigen Eigentums für die jeweiligen Unternehmen ist bei PaaS eine wichtige Eigenschaft der SOA [Ba17]. Vorteile, die PaaS mit sich bringt, sind die Flexibilisierung von Lieferketten, Fertigung individueller Produkte, die Planung nachhaltiger Lieferwege, eine effizientere Maschinennutzung sowie kürzere Lieferzeiten [He21]. Neben dem Begriff PaaS werden auch häufig die Begriffe *Manufacturing as a Service* oder *Hardware as a Service* verwendet [He21]. PaaS Architekturen lassen sich mit unter dem CM zusammenfassen. Dabei stellt eine Cloud die Verknüpfung zwischen Kunden und Servicedienstleistern dar. Die Cloud dient somit als Erweiterung des ERP-Systems, um auf Business-Ebene die Annahme von Produktionsaufträgen zwischen Unternehmen zu koordinieren. Die benötigten Services zur Abwicklung eines Auftrages werden über die Cloud auf angebundene Unternehmen verteilt. Cloud Manufacturing wird aus [Li19b] wie folgt definiert:

**Cloud Manufacturing (CM)** ist ein kundenorientiertes Fertigungsmodell, das den On-Demand-Zugriff auf eine gemeinsame Sammlung diversifizierter und verteilter Fertigungsressourcen (wie Software, Maschinen und Capabilities) bietet. So können Fertigungsservices mit minimalem Managementaufwand voneinander genutzt werden.

Die benötigten Produkthanforderungen müssen auch in diesem Konzept auf mögliche Unternehmensfähigkeiten und deren Produktionsressourcen gematcht werden. Ausführliche Zusammenfassungen über Forschungsansätze im CM sind unter anderem in [Ad15; ENV19; Ta15; Wu13] zusammengefasst. Folgende Vorteile ermöglicht CM in der Produktion:

- Die schnelle Fertigung von Produkten mit individuellen Kundenwünschen [Li19b; LWX19].
- Bisher unwirtschaftliche Fertigungsumgebungen werden durch die höhere Flexibilität und geteilten Ressourcen konkurrenzfähiger, so kann bspw. auch dem wirtschaftlichen Druck gegenüber Niedriglohnländern Stand gehalten werden. Vor allem KMU können durch diesen Ansatz profitieren. [Ad15; Le19; Sc14]

- Neue Technologien wie *Internet of Things* (IoT) und *Cloud Computing* werden eingesetzt [Li19b; Zh17].
- Es wird eine skalier- und rekonfigurierbare Produktionsumgebung ermöglicht [Ad15; Li19b; Wu13].
- Es entsteht die Möglichkeit, dass räumlich verteilte Unternehmen oder Standorte zusammenarbeiten, um einen Produktionsauftrag zu fertigen [Ad15; Wu13].

Ein Überblick ausgewählter Projekte, die sich thematisch mit CM auseinandersetzen, wird im Folgenden vorgestellt:

In [Yl19] werden mehrere CM-Lösungen betrachtet und festgestellt, dass herstellerübergreifende Standards wie RAMI 4.0 oder OPC UA ignoriert werden. In der Ausarbeitung wird eine mögliche Produktbeschreibung in OPC UA entwickelt. Anhand des abstrakten OPC UA-Modells kann ein herstellendes Unternehmen die Beschreibung in einen automatisierten Montageprozess überführen.

In dem durch die EU-Kommission geförderten Forschungsprojekt MANUCLOUD [@manucloud] wurde ein CM-System entwickelt. In [Ra14; RS12; RSS13] wird die SOA des Projektes definiert. Unternehmen melden sich in der Cloud mit ihren Dienstleistungen an. Über einen Cloud-Konnektor, der auf Webservices beruht, werden Fertigungsaufträge an die Unternehmen weitergeleitet. Einzelne Services bieten die Option, zusammengeführt zu werden und somit einen neuen Service zu aggregieren. Kunden haben die Möglichkeit, sich über ein Webinterface mit der Cloud zu verbinden und ihre Produktkonfiguration vorzugeben. Wird der Produktionsauftrag abgesendet, wird ein Prozessplan in der Cloud erstellt und Fertigungsaufträge über den Cloud-Konnektor an die jeweilige Unternehmens-IT (bspw. ERP- oder MES-Systeme) weitergeleitet. Dem Kunden wird ermöglicht, den aktuellen Status seines Auftrages über die Cloud zu verfolgen.

In [Ba17; Ba18] wird ein PaaS-Framework entwickelt, das sicherstellt, geistiges Eigentum eines Unternehmens im Angebotsprozess zu schützen. Über eine Nutzerschnittstelle können Konstrukteure ein Bauteil bereitstellen, das durch einen Featureextraktor analysiert wird. Die abstrahierten Features werden anschließend auf vordefinierte Prozesse mit einer Ontologie der „*Manufacturing Service Description Language*“ gemappt. Die Durchführbarkeit eines Schrittes wird dabei von jedem Unternehmen anhand eines Herstellerinterfaces geprüft und bestätigt.

Hermann et al. beschreiben die Möglichkeit, Produktfeatures auf vorhandene Fertigungsskills zu matchen [He19; He20; He21; HRR20]. Dabei wird das PPR-Modell genutzt, um anhand der vorhandenen Ressourcenfähigkeiten einen möglichen Produktionsweg zu ermitteln, der die Realisierung aller benötigten Produktfeatures erfüllt. Der beschriebene Weg besteht aus 8 Schritten und untergliedert das vorliegende Produkt zunächst in Produktfeatures. Anschließend werden die Features auf die vorhandenen Technologien unterschiedlicher Unternehmen gemappt

und mit dem vorhandenen Equipment in den Produktionsumgebungen abgeglichen. Die Anfrage an ein Unternehmen und der Abgleich mit den dort vorhandenen Produktionsressourcen geschieht über Microservices. Beim Matching werden eventuell nicht wertschöpfende Tätigkeiten, wie Transportvorgänge, nicht betrachtet. Alle analysierten Produktionsketten werden ausgegeben und mit den gewünschten Produktionsvorgaben abgeglichen. So wird am Ende die optimale Route zur Bauteilproduktion generiert. Nach Prüfung, ob das benötigte Equipment für den ersten Produktionsschritt bereitsteht, startet die Fertigung. Während der Laufzeit wird der Prozess ständig überwacht, um im Störfall eine alternative Route bestimmen zu können. Das Konzept ist sehr generisch gehalten und beschreibt generell die Möglichkeit verschiedener Fertigungsarten. Hierbei wurden sowohl additive als auch subtraktive Fertigungsprozesse beachtet.

Das *Industrial Internet Consortium* (IIC) betreibt als Testbed die Plattform *smartfactoryweb* [@SFweb], die sich zum Ziel setzt, Produktionsaufträge zwischen verschiedenen Standorten je nach verfügbaren Ressourcen zu verteilen. Somit stellt dieses Testbed auch eine CM-Lösung dar. Die verwendeten Technologien und Standards sind in einem Whitepaper des IIC [Be20a] veröffentlicht. Auf der Feldebene werden Produktionsmodule durch den in IEC 62714 definierten AutomationML (AML)-Standard beschrieben. Mit Hilfe des AML-Modells wird ein entsprechender OPC UA-Server auf dem Produktionsmodul generiert. In der übergeordneten Plattformebene analysiert ein Aggregationsserver alle relevanten Informationen der vorhandenen Module anhand ihrer AML-Modelle und fügt gegebenenfalls weitere Semantiken hinzu. Zwischen der Plattformebene und der Modulebene wird über OPC UA kommuniziert. Über einen weiteren *Layer* den *Factory cloud coupler* können Daten mit verschiedenen Clouddiensten ausgetauscht werden. Zwischen diesen Ebenen kann die Kommunikation über eine REST-Schnittstelle oder OPC UA geschehen.

Gaia-X soll die zukünftige vernetzte europäische Dateninfrastruktur definieren [@Gaia-X]. Die Architektur basiert auf einem dezentralen Konzept und gewährleistet den Nutzern die Datensouveränität [@Gaia-X]. Ein großer Fokus liegt auf sogenannten *Federation Services*, die in einem Datenraum angeboten werden können [Ga21]. Nutzer können in *Federated Catalogues* benötigte Datenservices finden und nutzen, während die Datenhoheit gewährt wird. Dabei entsteht kein großer Datenraum, sondern es können viele getrennte individuelle Ökosysteme aufgesetzt werden [Ga21] z. B. Datenräume nur mit vertrauten Partnerunternehmen. Um den sicheren Datenaustausch zu gewährleisten, werden speziell entwickelte Konnektoren benötigt. Zurzeit wird z. B. am DSC (*Data Space Connector*) [@DSC] für die *International Data Spaces* (IDS) [In19] und dessen Nachfolger dem EDC (*Eclipse Data Space Connector*) [@EDC] entwickelt. Diese Konnektoren könnten sich in Zukunft als Gaia-X kompatible Konnektoren etablieren, und aufgrund der verteilten herstellerunabhängigen Services in Gaia-X kann auf der Plattform auch die Idee von verteilten Produktionsdienstleistungen in einem Netzwerk aus Partnern verankert werden.

## 2.3 Automatisierte Prozessplanung in der spanenden Bearbeitung

Durch die Auslagerung von Fertigungsaufträgen an spezialisierte Zulieferunternehmen können Kosten- und Leistungsvorteile genutzt werden. Diese müssen in einer *Make-or-Buy-Analyse* allerdings genau abgeschätzt werden [Wa14]. Laut [Sp13] wird die Fertigungstiefe von produzierenden Unternehmen aufgrund immer komplexer werdender Produkte weiter abnehmen und die Auslagerung von Fertigungsprozessen und die Schaffung von Lieferantennetzwerken weiter fortgeführt. Die flexible Bestellung „*Production-on-Demand*“ wird in Zukunft eine wichtige Rolle spielen [Sp13]. Die Beschreibung von typischen Lieferanten und Abnehmer-Verhältnissen ist in Abbildung 2.12 dargestellt.

Leistungsumfang Kompetenz	Durch Abnehmer vordefinierte Produkte und Verfahren	System- und Problem- lösungskapazität
Produktions-Know-how	Teilefertiger Lohnfertiger	Produktionsspezialist
Produktions- und Produkt-Know-how	Entwicklungspartner	Wertschöpfungspartner

Abbildung 2.12: Beziehungen zwischen Abnehmern und Lieferanten in Anlehnung an [Fr14; Wi08]

Teilefertiger und Produktionsspezialisten werden im Fall der Produktionsauslagerung erst spät oder gar nicht in den Entwicklungsprozess des Abnehmers eingebunden. Diese Firmen werden in Anlehnung an [Fr14] als Lohnfertiger bezeichnet. Teilefertiger sind dabei vorwiegend auf das Produzieren von einfachen und standardisierten Komponenten spezialisiert und müssen sich hauptsächlich durch einen günstigen Preis von der Konkurrenz absetzen. Produktionsspezialisten besitzen durch Prozessinnovationen ein Alleinstellungsmerkmal und können spezifische Fertigungsaufgaben durchführen [Fr14; VDA 4961/2; Wi08].

Die spanende Fertigung von Bauteilen ist ein Fertigungsbereich, in dem Aufträge häufig an externe Unternehmen vergeben werden. Allein in Deutschland existieren im Jahr 2020 mehr als 8000 Unternehmen in der Metallverarbeitung [Ge21]. Die Metallverarbeitung ist die Branche mit den meisten Unternehmen in der Metall- und Elektro-Industrie [Ge21]. Der Wandel zu einer Fremdfertigung, auch besonders im Ausland, erhöht den Druck für inländische Firmen sich z. B. durch Innovationen, Flexibilität, Nähe zum Produzenten und Qualität abzugrenzen [Ma18b; Wa14]. Im Sinne der Entwicklungen rund um I4.0 stellt sich die Frage, wie durch eine stärkere digitale Konnektivität Vorteile für Unternehmen entstehen können. Vor allem sollen interne Pro-

zesse verschlankt werden, sodass die Auftragsfertigung flexibler wird. Dies bringt zum einen die Möglichkeit mit sich den Angebotsprozess zu optimieren, zum anderen aber auch die automatisierte Planung und Ausführung eines individuellen Fertigungsprozesses. Vor allem Lohnfertiger können hier profitieren, da diese teilweise mit einem wechselnden Kundenkreis sowie kleinen heterogenen Fertigungsaufträgen konfrontiert sind. So entstehen hohe Varianzen in den Geometrien und Bearbeitungsaufgaben, die realisiert werden müssen.

### 2.3.1 Programmierung von Werkzeugmaschinen

Klassisch werden Werkzeugmaschinen in der spanenden Bearbeitung mit dem sogenannten „G-Code“ (auch als RS-274 bezeichnet) programmiert, der in unterschiedlichen Normen standardisiert ist, z. B. ISO 6983 oder DIN 66025. Dieser kann von den gängigsten CNC (*Computerized Numerical Control*)-Maschinen interpretiert werden und enthält Bearbeitungsinformationen wie z. B. Werkzeugbewegungen, Vorschubgeschwindigkeiten, Positionsvorgaben oder Instruktionen zum Aktivieren von Zusatzfunktionen, wie einem Werkzeugwechsel oder dem Aktivieren der Kühlschmiermittelpumpe [BW21]. Der zu erzeugende CNC-Code wird in der Auftragsfertigung typischerweise über zwei verschiedene Wege generiert. Zu Beginn steht häufig ein konstruiertes CAD-Modell des zu fertigenden Bauteils. Aus diesem kann anschließend eine Technische Zeichnung erstellt oder das 3D-Modell in ein CAM (*Computer-Aided Manufacturing*)-System überführt werden. Eine Technische Zeichnung wird typischerweise verwendet, um die werkstatorientierte Programmierung vorzunehmen. Im Falle der werkstatorientierten Fertigung wird die Bedieneroberfläche einer Werkzeugmaschine genutzt, um den Code für die Bearbeitung zu erzeugen [BW21]. Das CAM-System hingegen ermöglicht, am Computer den Bearbeitungsvorgang zu erstellen und anschließend in G-Code zu überführen. Die möglichen Vorgehensweisen zur Fertigung eines Fräsauftrages sind in Abbildung 2.13 abgebildet.

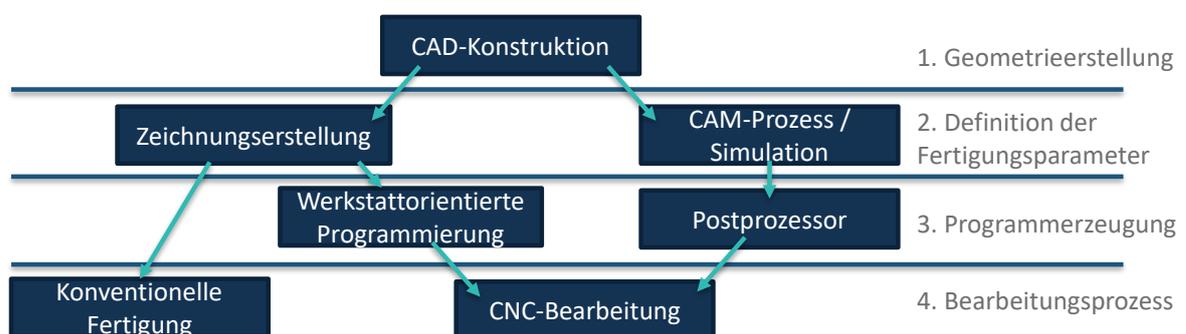


Abbildung 2.13: Mögliche Vorgehensweisen zur spanenden Bearbeitung eines Bauteils an CNC-Maschinen

Die Wirtschaftlichkeit einer Werkzeugmaschine ist dabei direkt abhängig vom Programmieraufwand bzw. der Programmierdauer, um ein Teil zu produzieren [BW21]. Ob ein CAM-Programm oder die werkstatorientierte Programmierung zu bevorzugen ist, ist hauptsächlich abhängig von der Komplexität des Bauteils [BW21]. Für komplexe Fertigungsschritte eignet sich generell

ein CAM-System, weniger komplexe Fertigungsaufgaben können schnell an der Maschine programmiert werden [BW21]. Eines der größten Anwendungsprobleme des G-Codes ist, dass je nach Steuerungshersteller zusätzliche eigens definierte G-Funktionen existieren [Po14]. So sind je nach Maschinentyp unterschiedliche Postprozessoren notwendig. Der G-Code ist somit kein Standard, der herstellerübergreifend problemlos eingesetzt werden kann.

Aufgrund der teilweise aufwendigen Programmierung, speziell für die werkstatorientierte Programmierung bei kleinen Stückzahlen [BW21], sowie der mangelnden herstellerübergreifenden Kompatibilität zwischen unterschiedlichen Maschinen, ist die Wiederverwendung von Code kaum möglich. Gleiches gilt auch für CAD- und CAM-Systeme, hier gibt es eine Vielzahl verschiedener Hersteller am Markt, die einen eigenen proprietären Dateistandard pflegen. Somit ist ein einfacher Datenaustausch über die gesamte digitale Prozesskette zwischen unterschiedlichen Softwareherstellern teilweise nicht möglich.

### 2.3.2 Computer Aided Process Planning

Um die Fertigungsprozesse möglichst zu digitalisieren, sollten die gewünschten Bauteile hinsichtlich geometrischer Maße und Anforderungen an Passgenauigkeiten und Toleranzen sehr genau beschrieben werden. Hierzu eignen sich 3D-CAD-Formate, in denen die Bauteile heutzutage üblicherweise konstruiert werden. Häufig wird z. B. im Angebotsprozess immer noch eine Technische Zeichnung als Grundlage von Angeboten und Aufträgen verwendet. Heutige CAD-Systeme besitzen allerdings alle notwendigen technischen Möglichkeiten, ein Bauteil in einer dreidimensionalen Umgebung entsprechend zu konstruieren und alle notwendigen Fertigungsinformationen am digitalen Modell zu repräsentieren. Des Weiteren stellt sich die Frage, wie die Informationen sinnvoll über die gesamte Fertigungsprozesskette weiterverwendet werden können.

Das *Computer-Aided Process Planning* (CAPP) zählt zu einer der Technologien, die in der Zeit des *Computer-Integrated Manufacturing* (CIM) entstanden ist. CAPP-Technologien sollen Prozessplaner bei ihren Arbeiten unterstützen und sind häufig Schlüsseltechnologien in der Verbindung von CAD- und CAM-Software [YL14]. CAPP unterstützt den Anwender, einen Fertigungsauftrag schneller zu bearbeiten und zu finalisieren. Die Einordnung von CAPP in den Fertigungsprozess ist in Abbildung 2.14 dargestellt. Alle Computer Aided-Technologien werden auch unter dem Kürzel CAX zusammengefasst [SB16a].

Mit Hilfe der im CAD-Modell vorhandenen Informationen kann unter anderem die Prozessplanung zur Fertigung eines Bauteils vereinfacht oder der Konstrukteur durch CAPP bereits in der Entwicklungsphase eines Bauteils mit vorgefertigten Objekten unterstützt werden. Somit sollen primär Zeitaufwände verkürzt werden, die in der Konstruktion von Bauteilen, in der Offline-Programmierung von Maschinen oder der Prozessplanung entstehen.

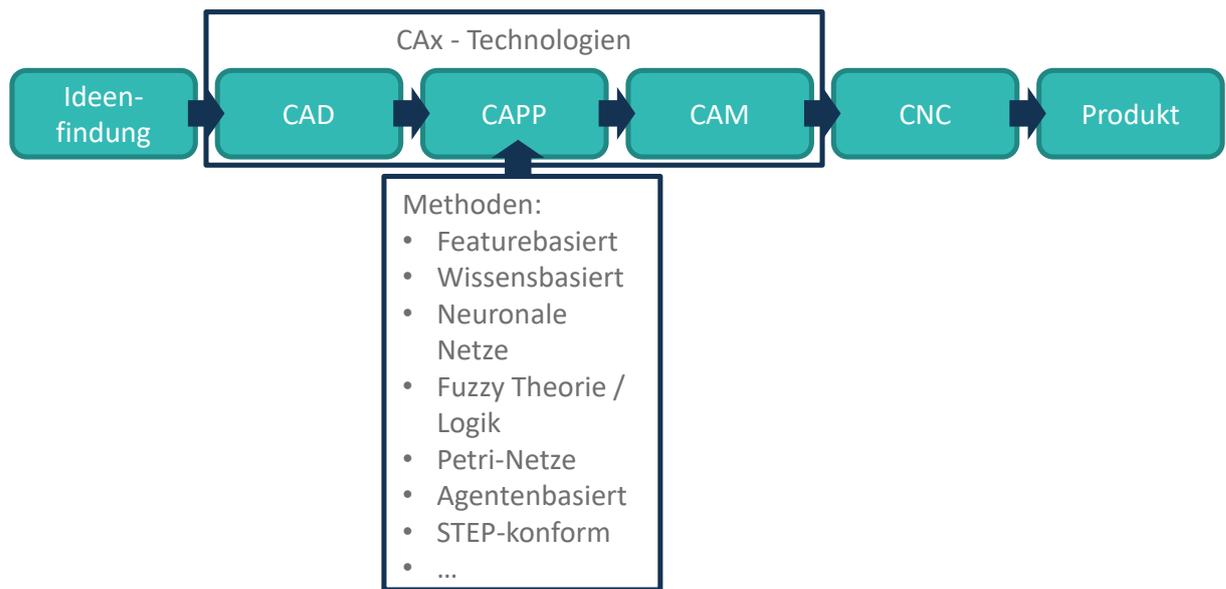


Abbildung 2.14: Gesamte Fertigungskette von der Ideenfindung bis zur Fertigung mit Hilfe von CAD-, CAPP- und CAM-Systemen in Anlehnung an [YL14]

[XWN11] unterteilt CAPP in zwei generell unterschiedliche Ansätze, den Varianten-Ansatz und den generativen Ansatz. Im ersten Ansatz klassifiziert ein Mitarbeiter das vorhandene Produkt und erhält aus einer Datenbank auf Basis von ähnlichen Varianten einen Prozessplan, der anschließend modifiziert wird [XWN11]. Im generativen Ansatz werden Prozesspläne mit sehr geringer menschlicher Interaktion auf Basis von wissensbasierten Entscheidungsalgorithmen getroffen [XWN11]. Für die Entwicklung von CAPP Technologien werden verschiedenste Methoden wie objektorientierte Ansätze, Neuronale Netze, Petri-Netze, fuzzybasierte Prozessplanung, Featureerkennung oder featurebasierte Ansätze, wissensbasierte Ansätze oder agenten- und internetbasierte Ansätze verwendet [Sa20; XWN11; YL14]. Im Speziellen sind für diese Arbeit die featurebasierten Methoden und die Featureerkennung von hoher Relevanz. Ein Feature ist nach VDI 2218:2003 <sup>1</sup> wie folgt definiert:

*„Features sind informationstechnische Elemente, die Bereiche von besonderem (technischen) Interesse von einzelnen oder mehreren Produkten darstellen. Ein Feature wird durch eine Aggregation von Eigenschaften eines Produkts beschrieben. Die Beschreibung beinhaltet die relevanten Eigenschaften selbst, deren Werte sowie deren Relationen und Zwangsbedingungen (Constraints).“* [VDI 2218; We96]

Je nach Einsatzgebiet und Zweck kann es mehrere verschiedene Typen von Features geben, z. B. Form-Feature, Fertigungs-Feature oder Geometrische-Feature [SB16a].

*„Form-Features sind Struktur-orientierte Gruppierungen geometrischer Elemente, also eine formalisierte Aggregationen [sic!] von Gestalteigenschaften innerhalb eines Produkt-*

<sup>1</sup>Die VDI 2218:2003 bezieht sich auf die Definition von [We96]

modells [...], die unter einen gemeinsamen Namen erzeugt, gespeichert, geändert und gelöscht werden können.“ [VDI 2218]

**Fertigungs-Features** „werden hinsichtlich Begriff, Gestalt und Topologie in erster Linie durch die betreffenden Bearbeitungsverfahren bestimmt [...]“, es „repräsentiert neben der fertigungsrelevanten Geometrie sämtliche fertigungsspezifischen Toleranzen, Oberflächenbeschreibungen sowie ergänzende Angaben.“ [VDI 2218]

Form-Features finden somit ihre Anwendung im CAD-System im Konstruktionsprozess, während Fertigungs-Features im CAM eingesetzt werden, um z. B. zu entfernendes Volumen zuzuordnen [MDN16]. Abbildung 2.15 zeigt einen Auszug möglicher Fertigungs-Feature in der Bohr- und Fräsbearbeitung.

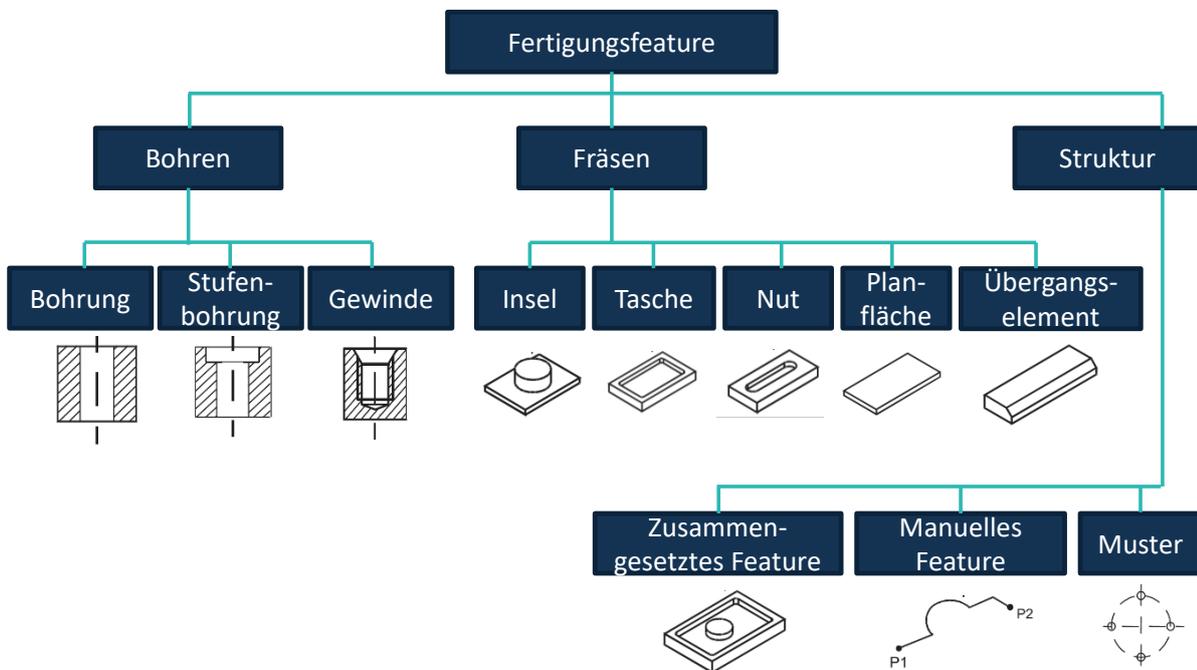


Abbildung 2.15: Möglichkeit der Definition von Fertigungs-Features in der Fräs- und Bohrbearbeitung in Anlehnung an [VDI 2218]

Da sich der Begriff des Form-Feature nach VDI 2218:2003 rein auf geometrische Elemente bezieht, jedoch im Rahmen der Ausarbeitung auch weitere an ein Feature gebundene Informationen wie Toleranzen oder Qualitätsinformationen verarbeitet werden, wird zusätzlich der Begriff des semantischen Features definiert. Die zusätzlichen Informationen erhalten je nach Phase des Produktlebenszyklus eine interpretierbare Bedeutung. Im Rahmen dieser Arbeit werden Features vordergründig in der Konstruktionsphase, der Fertigungsplanung und der Fertigung betrachtet.

Unter einem **semantischen Feature** wird „[...] die Verbindung eines Form-Features mit weiteren Eigenschaften aus einer anderen Eigenschaftsklasse verstanden. Die Semantik ist da-

*bei die in der jeweiligen Phase des Produktlebenszyklus interpretierbare Bedeutung“ [VDI 2218]*

Beim Einsatz von Feature-basierten Verfahren werden drei unterschiedliche Prinzipien in der Arbeit mit Features unterschieden [Ku17; Le17]. Die existierenden Arbeitsweisen werden mit Fokus auf den Bereich Konstruktion und Fertigung im Folgenden erläutert:

- **Design by feature:** Mit Hilfe vorgegebener Features wird ein Bauteil konstruiert [SSM91]. Die Features enthalten alle benötigten Fertigungsinformationen, um diese in der automatisierten Fertigungsplanung zu berücksichtigen [Kr11]. Der Konstrukteur benötigt somit ein erhöhtes Wissen im Bereich der Fertigung und muss dies während der Konstruktion beachten.
- **Featuremapping:** Konstruktionsfeatures werden automatisch auf Fertigungs-Features gemappt [LBT10]. Dabei muss allerdings im Konstruktionsprozess darauf geachtet werden, dass bestimmte Features verwendet werden oder alle notwendigen Featureinformationen vorhanden sind [Ku17].
- **Featureerkennung:** Nach [Ku17] lässt sich diese in die folgenden Unterpunkte gliedern:
  - **Interaktive Featureerkennung:** Der Anwender kann bspw. mehrere Features auswählen und diese zu einem Fertigungs-Feature zusammenfassen sowie erweiterte Informationen hinterlegen. Derartige Varianten der Featureerkennung sind in heutiger kommerzieller CAM-Software häufig zu finden.
  - **Automatische Featureerkennung:** Die Software findet automatisch vorhandene Features im Bauteil und ordnet diese zu.

Um Anwender bei der Fertigung eines Bauteiles zu unterstützen, bietet kommerzielle CAM-Software hauptsächlich eine interaktive Featureerkennung. Teilweise unterstützt ein integriertes Featuremapping, indem Softwareanwender bspw. zu Form-Features ein entsprechendes Fertigungs-Feature vorgeschlagen bekommen. So entstehen hybride Ansätze, die die Fertigungsplanung beschleunigen. Damit dies in einer möglichst durchgängigen digitalen Prozesskette funktioniert, ist der Einsatz von Standards sinnvoll. Hierzu wird im folgenden Kapitel STEP erläutert.

### 2.3.3 STEP

Um eine durchgängige digitale Prozesskette von der Konstruktion bis zur Fertigung eines Bauteiles zu garantieren, existieren bereits Normen, die eine einheitliche und durchgängige Dateistruktur garantieren. Einige der bekanntesten Standards im Bereich Konstruktion, Fertigungsplanung, Inspektionsplanung, Fertigung und Inspektion sind in Abbildung 2.16 aufgezeigt.

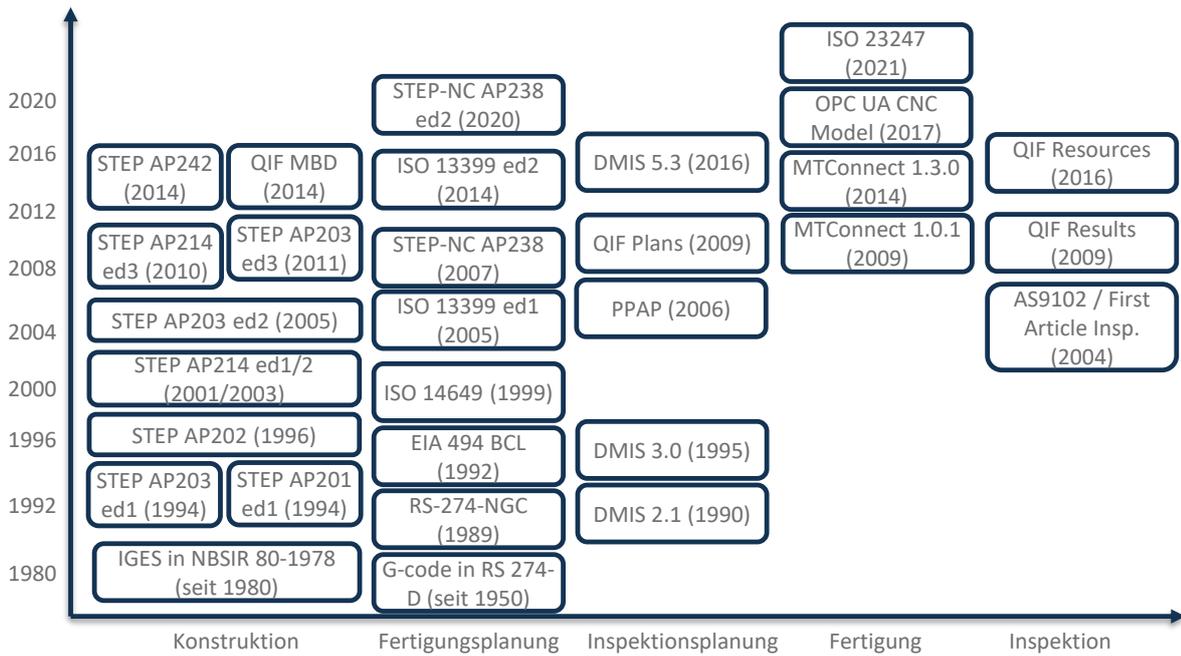


Abbildung 2.16: Wichtige Standards für verschiedene Produktlebenszyklen in Anlehnung an [Be18; Lu20; LXW20]

Wie aus Abbildung 2.16 hervorgeht, hat sich zum bekanntesten Austauschformat im Konstruktionsbereich für CAD-Dateien das Format STEP (*Standard for the exchange of product model data*) entwickelt. STEP wird in der Norm ISO 10303 definiert. In ISO 10303 werden mehrere verschiedene *Application Protocols* (AP) entwickelt, die den Austausch von CAD-Daten in bestimmten Einsatzbereichen definieren. Aus Abbildung 2.16 wird ersichtlich, dass bereits mehrere STEP-AP existieren. Neben den AP werden noch weitere Dokumente in der ISO 10303 beschrieben, einen Überblick verschafft Tabelle 2.1:

Tabelle 2.1: Übersicht der Bestandteile von ISO 10303 nach [ISO 10303-1]

Art der STEP Norm	Nummerierung ISO 10303-
<i>Overview and fundamental principles</i>	1
<i>Description methods</i>	1x
<i>Implementation methods</i>	2x
<i>Conformance testing methodologies and frameworks</i>	3x
<i>Integrated generic resources</i>	6x
<i>Integrated application resources</i>	1xx
<i>Application Protocols (AP)</i>	2xx
<i>Abstract test suites</i>	3xx
<i>Application modules (AM)</i>	4xx, 1xxx
<i>Application Interpreted Constructs (AIC)</i>	5xx
<i>Business Object Model (BO-Modell)</i>	3001

<i>Domain Model</i>	4442
<i>Guidance of usage of ISO 10303-214</i>	5001

Die relevantesten STEP-AP im Konstruktionsbereich sind AP203 , AP214 und AP242 . AP203 definiert neben einer Struktur für ein Konfigurationsmanagement vor allem die Darstellung von Geometrien sowie das Hinzufügen von Toleranzinformationen . AP214 wurde an die Bedürfnisse der Automobilindustrie angepasst und beinhaltet AP203 komplett [Kr01]. Weiterhin referenziert AP214 auf STEP *machining features*, die identisch zu AP224 sind [KX09].

Für den Austausch von vollumfänglichen Produktdaten im Konstruktionsbereich wurde das AP242 entwickelt, es unterstützt nicht nur die Darstellung des 3D-Modells, sondern bspw. auch die Möglichkeit, geometrische Toleranzen, Fertigungsplan- und Fertigungsinformationen oder kinematische Simulationsdaten zu beschreiben . STEP-AP 242 hat somit nicht nur die zwei existierenden AP203 und AP213 zusammengeführt, sondern fokussiert den Ansatz, die gesamte Abbildung des *Modell Based Engineering* (MBE) in einem Dateiformat zu unterstützen. Somit wird der STEP-Standard mit Funktionalitäten erweitert [LL15; MDN16] (siehe Abbildung 2.17).

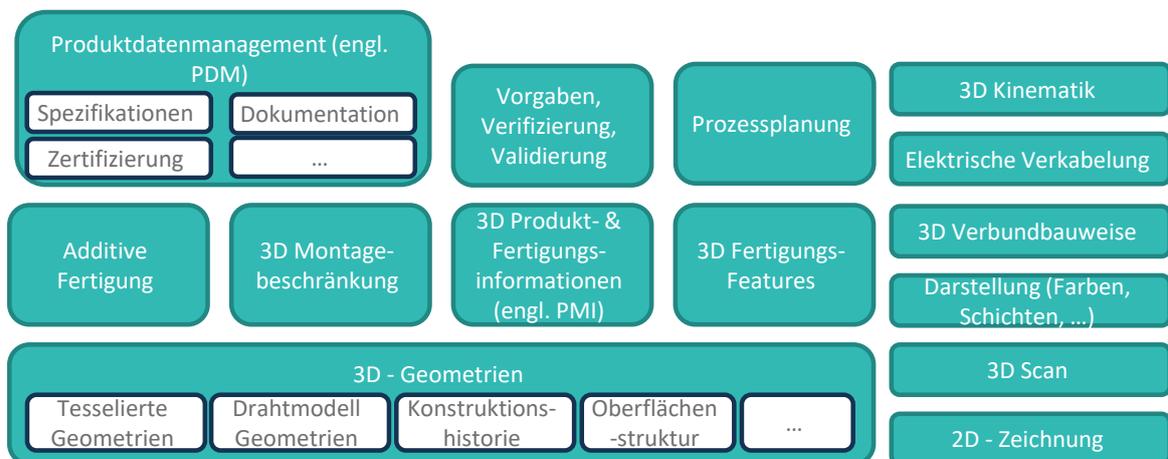


Abbildung 2.17: Auswahl von Funktionalitäten, die durch AP242 im Sinne von MBE bereitgestellt werden in Anlehnung an [STEP242; LXW20] und Informationen aus [ISO 10303-242]

Wie die in Abbildung 2.17 dargestellten Fertigungs-Features zeigen, wurde innerhalb von STEP an einem Konzept gearbeitet, das eine herstellerübergreifende Programmierung von Werkzeugmaschinen ermöglichen soll. Der als STEP-NC bezeichnete Standard, definiert in ISO 10303-238:2022 (AP238) und ISO 14649 , wurde entwickelt, um den bis heute noch verwendeten G-Code aus ISO 6983 bzw. DIN 66025 zu ersetzen. Während G-Code hauptsächlich die Werkzeugbewegung vorgibt, liefert STEP-NC auch höherwertige Informationen wie die Art des zu fertigenden Features, Werkzeuginformationen, Bearbeitungsstrategien oder den Bearbeitungsablauf [Li18; LRH08; SZM19]. STEP-NC basiert auf dem Konzept, dass Werkzeugmaschinen anhand der lokalen Gegebenheiten Bearbeitungsinformationen interpretieren [LXW20]. Die benötigten

zusätzlichen Attribute zur Bearbeitung wurden zunächst in der ISO 14649 definiert und anschließend in das entsprechende AP238 in ISO 10303-238 transferiert [LRH08]. Die Bearbeitungsfeatures, die in AP238 und ISO 14649 definiert sind, besitzen zwar Ähnlichkeiten, weisen allerdings an einigen Stellen signifikante Unterschiede auf [KX09]. Obwohl an dem Konzept von STEP-NC seit den 1990ern geforscht wird [Ga19], wird der Standard bisher kaum eingesetzt. Hauptsächlich mangelt es an kommerziell verfügbaren Steuerungen, die die Fähigkeit besitzen, STEP-NC zu interpretieren [LXW20]. Somit wird festgehalten, dass sich der Standard immer noch in der Entwicklung befindet und es bislang nicht klar ist, ob sich dieser Standard in der industriellen Fertigung durchsetzen wird. Die objektorientierte Struktur von STEP-NC bietet einige Vorteile gegenüber dem klassischen G-Code:

- Die echtzeitfähige Optimierung von Maschinenparametern, basierend auf den zu fertigenden Features und Maschinenbedingungen, wird ermöglicht [Ku18; LXW20; Ra12].
- Die Möglichkeit eine Kollisionsüberwachung einzusetzen und Rückzugsstrategien zu definieren [BW21].
- Der Austausch und die Rückverfolgung von Informationen über die gesamte CAD-, CAPP-, CAM- und CNC-Prozesskette ist gegeben [Ga19; Li18].
- Ein Feedback von der Fertigung zum Designprozess ist realisierbar [XN06].
- Eine hohe Flexibilität und die Möglichkeit zum Austausch von Code zwischen Maschinen ist gegeben, da das Format herstellerunabhängig ist [SZM19]. Im G-Code nutzen die Steuerungshersteller zusätzliche eigens definierte G-Funktionen [Po14]. Somit ist für jeden Maschinentyp ein spezieller Postprozessor notwendig. Durch STEP-NC sind Postprozessor-Mechanismen nicht mehr notwendig [XN06]. Des Weiteren können bereits existierende Programmabläufe leichter wiederverwendet werden [BW21].
- Das unabhängige Format kann mit verschiedenen Softwarelösungen bearbeitet bzw. verändert werden [SZM19].
- Die Bearbeitungszeit kleiner Losgrößen kann reduziert werden, da eine „intelligenter“ Optimierung von Prozessen auf den STEP-NC-Steuerungen möglich ist [XN06].
- XML-Dateien können als Informationsträger genutzt werden, um eine webbasierte, verteilte Fertigung zu ermöglichen [XN06].

Eine Fertigung basierend auf STEP-AP 242 und AP 238 bringt somit einige Vorteile beim Aufbau einer verteilten Produktion mit sich. Geometriefeatures können zuverlässig auf Fertigungs-Features gemappt werden, sowie Daten zur Qualitätskontrolle direkt digital verarbeitet werden. Auftretende Fehler können im digitalen Modell hinterlegt werden, dies vereinfacht die Prozessoptimierung. Der digitale Prozess ist in Abbildung 2.18 dargestellt. In Realität existieren allerdings Systeme, die die ISO 10303 gar nicht oder nicht im vollen Umfang unterstützen. Des Weiteren mangelt es wie bereits erwähnt an kommerziell verfügbaren Maschinensteuerungen, die

in der Lage sind, STEP-NC zu interpretieren. Zu beachten ist, dass sich die Interoperabilität in der Fertigungsprogrammierung stark verbessert und eine vollständig digitale Prozesskette entsteht. Allerdings ist der Prozess weiterhin von einem CAD- und CAM-System abhängig. Die Autonomie des Fertigungsprozesses wird durch STEP-NC zunächst nicht erhöht, bietet durch den objektorientierten Aufbau allerdings das Potential, Optimierungen durchzuführen (siehe Kapitel 2.4).

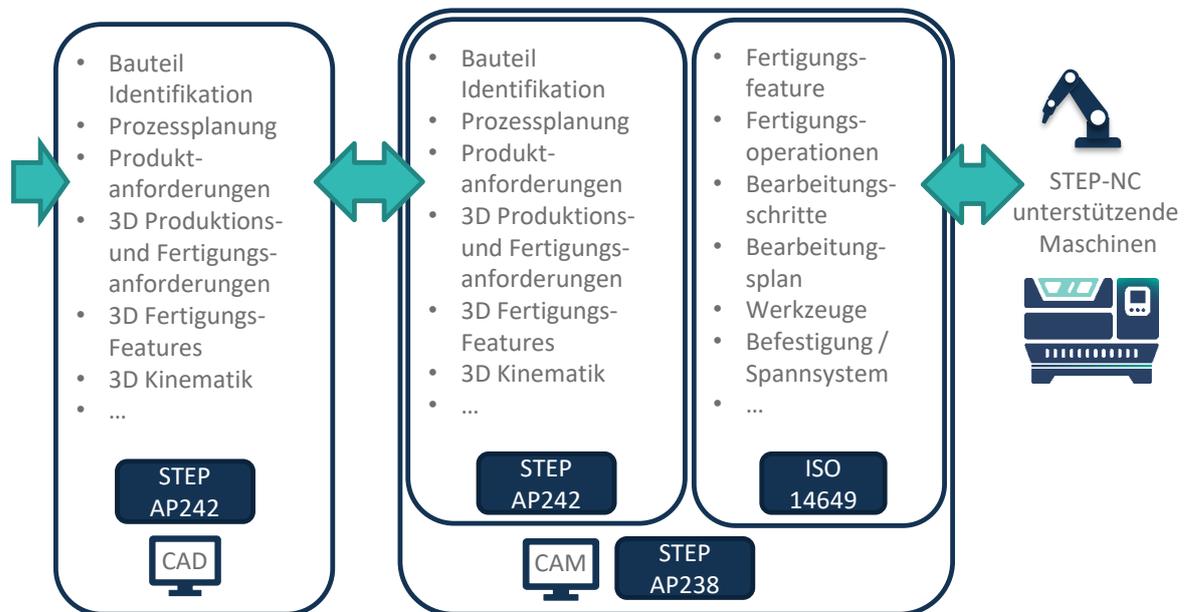


Abbildung 2.18: STEP-AP 242 und AP 238 Prozesskette zur Ansteuerung unterstützender Maschinen in Anlehnung an [LXW20]

## 2.4 Forschung zur automatisierten Fertigung mit Werkzeugmaschinen

In der aktuellen Forschungsliteratur sind verschiedene Ansätze einer dezentralen automatisierten Fertigung im Werkzeugmaschinenbereich zu finden. Das Kapitel beschreibt eine Auswahl an Projekten, die den gesamten Fertigungsprozess vom Design bis zur vollständigen Fertigung auf einer Maschine verfolgen. Des Weiteren werden Projekte zur Erhöhung der Autonomie von Werkzeugmaschinen betrachtet. Häufig werden die Vorteile von STEP-NC als objektorientierte Modelldarstellung genutzt:

[OJM20] gibt einen Überblick über existierende Ansätze der verteilten Fertigung, die sich auf IEC 61499 Funktionsblöcke beziehen. Der umfangreichste Ansatz wird in den Veröffentlichungen von Wang et al. beschrieben. In [Wa01] wurde die Vision eines Konzeptes zur Bauteilfertigung durch ein holonisches Produktionssystem vorgestellt, in dem Menschen und Maschine dynamisch miteinander arbeiten. Die Vision beschreibt autonome und kooperative Einheiten, die als *holons* bezeichnet werden. Diese können Aufgaben eigenständig planen, ausführen und über-

wachen. In der Idee wird beschrieben, dass der Fertigungsprozess auf einem featurebasierten Konstruktionsprozess aufbaut. Die Prozessplanung übernimmt ein zentrales Prozessplanungs- und Kontrollsystem. Ein MAS entscheidet über benötigte Fertigungs-Features und gleicht diese mit in einer Datenbank vorhandenen Maschinenfähigkeiten, Werkzeugen und Fertigungstechnologien ab. Das MAS plant die benötigten Maschinen und Fertigungsschritte zur Bauteilfertigung. Das Konzept basiert auf Werkzeugmaschinen und das System erzeugt anhand der vorhandenen Informationen automatisiert den benötigten NC-Code, der auf den entsprechenden Maschinen ausgeführt werden soll. Somit gibt es ein zentrales „intelligentes“ Steuerungssystem, das sowohl die Planung, Codeerzeugung und Koordination der Maschinen übernimmt. Des Weiteren wird die Idee beschrieben, dass einzelne Maschinenprozesse in Funktionsblöcke nach IEC 61499 gekapselt werden können und somit auf verschiedenen Maschinen ausführbar sind. Das Konzept wird im Laufe der Zeit genauer ausgearbeitet, sodass die vollständige Prozessplanung der Fertigung automatisiert abläuft und beispielhaft an Bauteilen validiert wird [Wa13; Wa15; WJF06; WLM13; WXD07]. Es wird die Möglichkeit betrachtet, die Prozessplanung in die Cloud zu verlagern, sodass ein Datenaustausch in einem verteilten Fertigungsnetzwerk realisierbar ist.

In [XN06] werden Projekte analysiert, mit dem Ziel STEP-NC für eine optimierte verteilte Produktion einzusetzen, und untergliedert in der Ausarbeitung die Maschinenansteuerung mit STEP-NC in drei Typen. Typ 1 sind konventionelle CNC-Maschinen, die mit Hilfe eines STEP-NC zu G-Code Interpreters angesteuert werden. In Typ 2 fallen Projekte, die durch eine STEP-NC-Steuerung einen Bearbeitungsauftrag starten können. Typ 3 sind Projekte, die eine „intelligente“ STEP-NC-Steuerung besitzen. Aus allen Bereichen wird beispielhaft ein Projekt vorgestellt.

In [SZM19; To19; ZSM18] wurde gezeigt, dass eine im CAM erstellte STEP-NC-Datei in der Lage ist, zwei unterschiedliche Robotertypen anzusteuern. Dies zeigt, dass z. B. individuelle Programmieraufwände mit Hilfe von STEP-NC leichter ausgelagert werden können und für mehrere unterschiedliche Werkzeugmaschinen wiederverwendet werden können. Die Roboter besitzen allerdings keinen direkten STEP-NC-Interpreter, sondern die Datei wurde vor der Ansteuerung der Roboter in den G-Code übersetzt (Typ 1).

In [Xu06] wird eine CNC-Maschine geretrofittet und direkt über einen STEP-NC-Interpreter angesteuert (Typ 2). Mit Hilfe der Konstruktionsinformationen auf den Maschinenebenen kann eine Kollisionsprüfung ausgeführt werden. Außerdem können Fertigungsdaten von der Maschine an die Konstruktion zurücktransferiert werden.

Mit Typ 3 wird das Ziel verfolgt, dass die Werkzeugmaschinen „intelligente“ Aktivitäten umsetzen können. So entscheidet die Werkzeugmaschine z. B. über eine optimale Werkzeugweggenerierung, Kollisionsprüfung oder auch über die Reihenfolge der Bearbeitungsschritte [XN06]. In [SC02; SCH02; Su02; Su03] wird ein autonomes featurebasiertes Steuerungssystem vorgestellt. Die konzeptionierte und zum Teil entwickelte Steuerung analysiert die Geometrie-, Technologie- und Werkzeugbeschreibungen und entscheidet anhand eines Prozess-Sequenz-Graphens über den Ablauf der Fertigungsschritte. Ein Werkzeugwegplaner und -simulator übernimmt die Pla-

nung einer Bearbeitungsstrategie. In [Ra12] wird ein System entwickelt, das eigenständig den Werkzeugpfad generiert und in der Lage ist, Maschinenparameter auf Basis der Rückführung von Fertigungsparametern zu optimieren. In [Ga19] wird ein „intelligentes“ CNC-Steuerungs-Framework betrachtet, das mehrere STEP-Bearbeitungsschritte für eine CNC-Maschine anfragt. In der Cloud wird mit Hilfe einer Wissensdatenbank von bereits umgesetzten Fertigungsschritten ein Prozessplan identifiziert. Durch das Feedback von Mitarbeitern wird die Wissensdatenbank ständig aktualisiert. Wird die Maschine zur Bearbeitung der vorgegebenen Bearbeitungsfeatures ausgewählt, übernimmt eine Maschinensteuerung die automatisierte Werkzeugwegplanung.

Die Möglichkeit der objektorientierten Programmierung von STEP-NC wird in [Wo06] genutzt, um Echtzeit-Maschinendaten anhand des derzeitigen Fertigungs-Features zu erfassen und für einen Fehler-/ Toleranzabgleich zu nutzen. Mögliche online- oder offline-Optimierungen der Achsansteuerung können somit featurebasiert eingesetzt werden. Ebenso wird in [Ku18] ein Framework zur Echtzeitdatenerfassung vorgestellt, um spätere Optimierungen durchzuführen.

Bei den bisher betrachteten Fertigungs-Features handelt es sich in der Regel um weniger komplexe Features wie z. B. Bohrungen oder Taschen (siehe Abbildung 2.15). In [LL19] wird ein Konzept vorgestellt, das Freiformflächen als Feature analysiert. Ein entsprechender Funktionsblock ermöglicht die Fertigung dieses Features.

[NNA06] entwickelt ein globales Fertigungssystem auf der Basis von STEP-NC. Es wird ein Fertigungsmodell definiert, das Informationen aus den Produkthanforderungen, Prozessanforderungen und Ressourcenbeschreibungen enthält. Wird dem existierenden STEP-NC-Code eine Ressource zugeordnet, wird ein ressourcenspezifischer Code zur Bearbeitung des Auftrages generiert.

[Sa20] setzt für ein CM-System ein agentenbasiertes Prozessplanungs-Framework ein, das in der Lage ist, CAD-Features mit Hilfe einer komplexen Ontologie auf mögliche Fertigungsressourcen zu mappen. Hierbei werden in den Ontologien auch beispielsweise Qualitäts- und Toleranzparameter beachtet.

## **2.5 Marktplätze und digitale Angebotserstellung für Lohnfertiger**

In Kapitel 2.2.2 wird die Motivation erkennbar, verteilte Produktionsnetzwerke zu schaffen. Dieses Kapitel gibt einen Überblick über verfügbare Fertigungsplattformen am Markt. Diese koppeln Lohnfertiger an einen Marktplatz oder vereinfachen Lohnfertigern die Angebotsbearbeitung. Im Folgenden ist eine Übersicht einiger am Markt zu findender Produkte gelistet. Die Software kann in 3 unterschiedliche Typen gegliedert werden.

- Typ 1 ist eine automatisierte Angebotserstellung mit Identifikation von Fertigungsressourcen. Die Software ermöglicht Lohnfertigern, CAD-Konstruktionen oder Technische Zeichnungen in die Software zu laden. Die zu fertigenden Features werden analysiert. Der Nutzer hat die Möglichkeit, seinen Maschinenpark in der Software mit Kostenparametern zu hinterlegen. Der Nutzer erhält auf Basis des CAD-Objekts und der hinterlegten Werkzeugmaschinen ein automatisch erzeugtes Angebot (siehe Beispiel up2parts).
- Typ 2 ist ein Marktplatz. Hier können Einkäufer eine Anfrage inkl. CAD oder Technischer Zeichnung in einen Angebotspool laden. Lohnfertiger können mit Angeboten oder zur Klärung fertigungsspezifischer Fragen Kontakt zu den Einkäufern aufnehmen. Lohnfertiger bekommen Aufträge, die zum Portfolio eines Unternehmens passen, bevorzugt angezeigt (siehe Beispiel Orderfox).
- Typ 3 ist eine Kombination aus Typ 1 und 2. Es können anhand von 3D-Modellen oder Technischen Zeichnungen das gewünschte Bauteil online hochgeladen werden. Am Marktplatz sind mehrere Unternehmen mit ihren Fertigungsressourcen hinterlegt und können individuelle Kostenparameter für Fertigungszeiten definieren. Neue Bauteile werden möglichen Lohnfertigern auf der Plattform zugeordnet und können durch diese begutachtet werden. Mit Hilfe der automatisierten Kalkulation wird eine nahezu sofortige Angebotserstellung ermöglicht (siehe Beispiel Spanflug).

Up2parts [[@up2parts](#)] bietet Lohnfertigern die Möglichkeit in einer Datenbank oder Cloudumgebung die vorhandenen Fertigungsressourcen zu hinterlegen und individuell zu konfigurieren. Bauteile können als CAD-Datei (STEP, STL) oder Technische Zeichnung (bspw. PDF) in das System geladen werden. Anschließend wird ein automatischer Maschinenvorschlag inklusive einer Preiskalkulation innerhalb weniger Sekunden erzeugt. Durch Nutzereingaben (z. B. Angabe von realen Fertigungszeiten) wird eine firmenspezifische Künstliche Intelligenz trainiert, um das System zu optimieren und an die realen Fertigungsbedingungen anzupassen. Up2parts unterstützt die Verfahren Fräsen, Drehen, Blechbearbeitung und Laserschneiden.

Orderfox [[@Orderfox](#)] ist ein Marktplatz, der zwischen Lohnfertigern und Herstellern Aufträge im Bereich der CNC-, Blechbearbeitung, Spritzguss, Schweiß und Guss vermittelt. Als Lohnfertiger können Dienstleistungen des Unternehmens auf der Plattform angeboten werden. Als Einkäufer kann eine Produkthanfrage z. B. auf Basis einer CAD-Konstruktion gestellt werden. Ein Matching-Algorithmus sucht nach passenden Lohnfertigern, um ein Angebot anzufragen. Die Lohnfertiger haben die Möglichkeit, auf ein angefragtes Bauteil oder ein Bauteil im globalen Anfragenpool ein Angebot abzugeben. Die Angebotsdauer wird hier mit ca. 48 Stunden angegeben.

Spanflug [[@Spanflug](#)] bearbeitet individuelle Anfragen von der Einzelteilmontage bis hin zur Serienfertigung. Auf Basis von CAD-Modellen (STEP, Autodesk, Inventor, Siemens NX, PTC ProE/Creo, Solidworks, Solidedge) oder Technischen Zeichnungen (bspw. PDF), die auf einer

Website hochgeladen werden, wird innerhalb von Sekunden ein Preis kalkuliert. Die Fertigung des Bauteils wird bei einem von rund 150 registrierten Lohnfertigern vorgeschlagen. Mögliche zugeordnete Unternehmen können per Mausklick einen Auftrag annehmen. Auf der Plattform werden teilweise Wärme- und Oberflächenbehandlungen berücksichtigt. Die Plattform vermittelt Aufträge für Dreh-, Fräs- und Dreh-Frästeile. Für Einkäufer wird die Möglichkeit einer Anbindung in das ERP-System geboten. Lohnfertiger können die Software ebenfalls intern für die eigene Angebotserstellung nutzen. Zur Kalkulation kann ein eigener Maschinenpark in einer Datenbank angelegt werden. Für die Maschinen hinterlegte Rüstkosten und Vorschübe sind die Basis der automatisierten Preiskalkulation. So kann der Lohnfertiger seinen eigenen Angebotsprozess optimieren und beschleunigen.

In die drei unterschiedlichen Typen lassen sich auch weitere Unternehmen wie xometry [@Xometry], easy2parts [@easy2parts] oder Frästeile-Marktplatz [@Fraesteile] einordnen. Untereinander unterscheiden sich die Anbieter entweder in der Handhabung oder in den Typen an angebotenen Fertigungsprozessen.

## 2.6 Zusammenfassung und Analyse

Die bisherigen Ansätze von verteilten Produktionssystemen und CM-Systemen geben bereits einen Eindruck, wie individuelle Fertigungsaufträge über unterschiedliche Firmen identifizierbar sind. Die erforschten Konzepte konnten sich im industriellen Umfeld bisher nicht durchsetzen und heutige Produktionsarchitekturen sind häufig starr an einen Prozess gebunden. Vor allem mangelt es an der hochautomatisierten Fertigung von kleinen Stückzahlen. Die IEC 61499 standardisierte schon die Möglichkeit, Fertigungsschritte modular zu untergliedern und gibt ein Konzept für eine verteilte Steuerung mit Hilfe von Funktionsblöcken vor. Daraus entwickelten sich mehrheitlich dynamischere Lösungen mit Hilfe von skillbasierten Ansätzen. Skills wurden in der Literatur in der Vergangenheit vor allem im Handhabungsbereich eingesetzt. Im Rahmen der Entwicklungen rund um Industrie 4.0 ist zu erkennen, dass Skills als Implementierung gekapselter Funktionen von Fertigungsressourcen genutzt werden. Aus der untersuchten Literatur geht im Bereich der Skills und Capabilities hervor, dass es an Beschreibungsstandards fehlt. So sind die in Forschungsprojekten dynamischen Prozessketten bisher spezifisch entwickelte Lösungen, die nur in einer bestimmten Produktionsumgebungen genutzt werden können. Zur Beantwortung von Forschungsfrage 2 und 3 werden somit Möglichkeiten untersucht, Capabilities und Skills zu beschreiben und diese in einem Netzwerk als Dienstleistung zur Verfügung zu stellen.

In der untersuchten Literatur zu spanender Bearbeitung wurde in der Fertigung bisher nicht der Begriff Skill, sondern der Begriff Fertigungs-Feature gewählt. Es existieren Ansätze, Produktionsaufträge anhand von Features zu untergliedern. Pfadplaner können den Werkzeugpfad für ein Feature und eine individuelle Ressource automatisch generieren. Trotz existierender Standards wie STEP zur Beschreibung von Features oder OPC UA als herstellerunabhängiges Kommunikati-

onsframework, konnte sich keine herstellerunabhängige und individuelle Prozessplanungen und -steuerungen im industriellen Umfeld durchsetzen. In dem Kapitel 2.3 wurde gezeigt, wie Standards dazu beitragen, die digitale Kette im Lebenszyklus eines Produktes zu optimieren. So können herstellerübergreifende und automatisierte Fertigungsplanungen sowie -optimierungen ermöglicht werden. Viele der Ansätze basieren auf der Entwicklung von CAPP Systemen, die auf STEP und STEP-NC basieren. Die Optimierungsansätze wurden häufig in CAM-Systeme oder auf Maschinen integriert und übernehmen Aufgaben, die heutzutage ein Mitarbeiter händisch am Computer vollführt. Weiterhin ist es üblich, dass die Programmierung der Werkzeugmaschine für ein individuelles Bauteil im CAM oder durch die werkstatorientierte Programmierung vorgenommen wird. Lösungen für eine schnellere Programmierung mit Hilfe von Features oder die Möglichkeit, Qualitätsinformationen nach einer Prüfung in das CAD-System zu übertragen, bieten auch die proprietären CAD- und CAM-Systeme. Die CAD/CAM-Software Siemens NX enthält z. B. den *Machine Knowledge Editor* [Si11], um individuell eine featurebasierte Fertigungsstrategie zu konfigurieren. Ebenfalls sind in handelsüblichen CAD/CAM-Systemen wie NX die Rückführung von Qualitätsdaten zur Überprüfung des gefertigten Bauteils möglich [Si14]. Bei der Überführung in die Software eines anderen Herstellers kann es allerdings zu Problemen kommen. In einem verteilten Fertigungsnetzwerk würde dies bedeuten, dass alle Teilnehmer ein Softwaretool verwenden müssen, um eine durchgängige digitale Aktualisierung eines Produktes zu gewährleisten. In der Fertigung geht mit der Erstellung des G-Codes der Zusammenhang zu den zu fertigenden Features (wie Taschen oder Bohrungen) verloren. Rückschlüsse aus der Fertigung zu ziehen und auf ein Konstruktions-Feature zu schließen, ist somit nur mit erhöhtem Aufwand möglich. Optimierungspotentiale werden somit verhindert. Standards wie STEP-NC könnten Abhilfe schaffen, da diese objektorientiert aufgebaut sind. Genauso kann die Verwendung von Skills helfen. Gründe, dass sich neue Standards wie STEP-NC nicht etablieren, wurden in Kapitel 2.3.3 beleuchtet. So liegt es im Rahmen von STEP-NC vorwiegend an fehlenden Maschinensteuerungen auf dem Markt, die STEP-NC interpretieren können. Genauso benötigt es CAD/CAM-Systeme, die den STEP-Standard vollständig und durchgängig unterstützen müssen. Unternehmen, die CAD/CAM-Systeme entwickeln, bieten meist viel umfangreichere Funktionen als STEP und STEP-NC im Standard an. Somit würde es bei einer reinen Verwendung von STEP evtl. zu Problemen in der Anwendung kommen. Ein weiterer Punkt ist, dass es für ein Unternehmen sehr kostspielig sein kann, eine CAD/CAM-Software auszutauschen und alle Mitarbeiter auf ein neues System zu schulen. Ein wichtiger Ansatz für die Produktbeschreibung und der Beantwortung von Forschungsfrage 1 ist somit, wie eine herstellerunabhängige Produktbeschreibung aussehen muss. Existierende Systeme können somit in das Konzept eingebunden werden.

Im Bereich des CM wurden bereits Konzepte in der Forschung erarbeitet, um eine digitale Verknüpfung mehrerer Unternehmen zu bewerkstelligen. Es fehlt an praktischen Umsetzungen und so wurden derartig komplett vernetzte Fertigungsstrukturen nicht industriell realisiert. Gaia-X ermöglicht, das Themengebiet der vernetzten Produktion mit einem unternehmensübergreifenden, kompatiblen Ansatz zu entwickeln. Identifizierte Problematiken sind, dass an Unter-

nehmensschnittstellen möglichst herstellerunabhängige Standards benötigt werden sowie Systeme, die die Datensouveränität bewerkstelligen. Häufig sind die betrachteten CM-Lösungen an ein zentrales System gekoppelt, mit dem sich teilnehmende Unternehmen verbinden, während Gaia-X z. B. einen dezentralen Ansatz bevorzugt. Als negativ wurde in der Analyse von CM-Umsetzungen angesehen, dass die bisherigen Systeme häufig I4.0-Standards wie RAMI 4.0, die Verwaltungsschale oder herstellerübergreifende Kommunikationsmöglichkeiten wie OPC UA außer Acht lassen. Zur Beantwortung der Forschungsfragen wird beachtet, herstellerübergreifende Technologien einzusetzen.

Auf dem Markt ist bereits Software verfügbar, die Teilaufgaben der digitalen Vernetzung übernehmen. So können die in Kapitel 2.5 diskutierten Systeme genutzt werden, um die Angebotsprozesse zu beschleunigen und zu vereinfachen. Anhand eines CAD-Modells oder einer Technischen Zeichnung kann eine Ressourcenauswahl und eine Angebotserstellung durchgeführt werden. Wird die Lösung eines komplett automatisierten Angebotsprozesses betrachtet, wird vor allem das Fertigungs Know-how eines Dienstleisters vernachlässigt. Für alle Anwender gilt im Hintergrund die gleiche Registrierung von Maschinen, in denen vor allem nur Preis- und Konfigurationsparameter verändert werden können. Qualitätsbezogene, erfahrungsbasierte oder implementierungstechnische Faktoren einer spezifischen Produktionsumgebung werden vernachlässigt. Für die Aufstellung einer firmenübergreifenden Produktionsarchitektur, soll somit beachtet werden, dass Lohnfertiger intensiver am Angebotsprozess beteiligt werden. So können die Angebote eines automatisierten Marktplatzes individueller ausfallen.

Wichtige Faktoren, in Bezug auf die in Kapitel 1.2 gestellten Forschungsfragen, wurden analysiert. In der weiteren Ausarbeitung sind die folgenden Punkte in Bezug auf die Forschungsfragen zu beachten:

- Zu Forschungsfrage 1: Die Produktbeschreibung muss aufgrund der Vielzahl von unterschiedlichen CAD-Systemen am Markt mehrere Beschreibungsarten von Features ermöglichen.
- Zu Forschungsfrage 2: Existieren bereits Standards, die zur Beschreibung von Capabilities oder Skills genutzt werden können? Es gilt Capabilities und Skills in einer Art und Weise zu beschreiben, sodass Prozessketten möglichst einfach identifiziert werden können. Des Weiteren müssen Capabilities auf einem Marktplatz für Unternehmen erreichbar werden.
- Zu Forschungsfrage 3: Spezifische Anforderungen an einen Skill im Bereich der Werkzeugmaschinen sind zu evaluieren. Die Randbedingungen für eine autonome Prozessdurchführung sind zu analysieren. Die Stärke von Lohnfertigern ist das spezifische Prozesswissen, das verstärkten Einfluss auf die Angebotserstellung nehmen soll. Spezifischen Implementierungen und Werkzeugmaschinen müssen somit Einfluss auf einen automatisierten Angebotsprozess sowie die Prozessplanung nehmen. Dies kann nur durch eine Ende-zu-Ende-Betrachtung bis zur Ressourcenebene ermöglicht werden.

## 3 Systemkonzept für eine marktplatzbasierte Fertigung mit Capabilities und Skills

In diesem Kapitel wird ein marktplatzbasierter Angebotsprozess für die spanende Bearbeitung erarbeitet. Die Definition der wichtigsten Schnittstellen zur Beschreibung des Produktes, von Dienstleistungen, Capabilities und Skills wird fokussiert. Zur Umsetzung ist die digitale Vernetzung aller Produktionsebenen eines Unternehmens sowie ein firmenübergreifendes Produktionsnetzwerk notwendig. Als Lösung wird eine RAMI 4.0-konforme Fertigungsarchitektur zur Umsetzung des CM-Ansatzes angestrebt. Die Vernetzung der Unternehmen und der Ablauf des Angebotsprozesses wird in Kapitel 3.1 beschrieben. Zwischen den Unternehmen müssen Produktbeschreibungen ausgetauscht werden. Mögliche Beschreibungen des Produktes auf Basis heutiger CAD Systeme und die Möglichkeit einer Produktbeschreibung wird in Kapitel 3.2 beschrieben. Durch die Beschreibung von Dienstleistungen im Netzwerk und Capabilities wird ein möglicher Fertigungsprozess für ein individuelles Produkt definiert. Kapitel 3.3 untersucht hierzu die möglichen Beschreibungsformen von Dienstleistungen und Capabilities. Kapitel 3.4 beschreibt die notwendige Skill-Schnittstelle, um die Fertigung letztlich auf einer Maschine auszuführen. Das so entstehende Netzwerk ermöglicht, das Prinzip von PaaS umzusetzen.

### 3.1 Definition des Fertigungsablaufs und der Systemarchitektur

Gaia-X besitzt das Potential nach RAMI4.0 den Business-Layer in der vernetzten Welt abzudecken. Über einen Cloud-Konnektor können mehrere Unternehmen untereinander Dienstleistungen und Produktionsinformationen austauschen. Cloud-Konnektoren wurden schon in diversen Forschungsprojekten angewendet und in Kapitel 2.2.2 dargestellt. Ein einsetzbarer Konnektor, der im Zuge von Gaia-X entwickelt wird, ist z. B. der EDC oder DSC (siehe Kapitel 2.2.2). Der Konnektor trägt die Aufgabe, auf der Business-Schicht die Kommunikation zwischen Unternehmen und der vernetzten Welt zu bewerkstelligen. Die Business-Schicht eines Unternehmens soll Dienstleistungen in der vernetzten Welt anbieten. Das Bereitstellen von Fertigungsdienstleistungen wird auch als *Manufacturing as a Service* (MaaS) bezeichnet [Wu13]. [Vo21b] diskutiert, wie eine Servicebeschreibung und Kommunikation zwischen Unternehmen mit Hilfe der VWS und dem DSC im *International Data Space* [In19] umgesetzt werden kann. Nach RAMI 4.0 sollte

als Werkzeug zur generischen Beschreibung die VWS eingesetzt werden. Die Dienstleistungen eines Unternehmens werden somit in einer VWS beschreiben. Hierzu sind standardisierte TM der VWS zu entwickeln. Die Aufgabe der Unternehmens-VWS wird wie folgt definiert:

*Die **Unternehmens-VWS** beschreibt ein spezifisches Unternehmen, Unternehmensstandort oder die Abteilung innerhalb eines Unternehmens und dessen angebotenen Dienstleistungen in einem Netzwerk mehrerer Partnerunternehmen.*

Die Definition eines Service in Kapitel 2.2.2 bezieht sich auf die SOA. Die Definition einer Dienstleistung (engl. *Service*), die ein Unternehmen anbietet, wird in Bezug auf die allgemeingültige Definition in Kapitel 2.2.2 spezifiziert:

*Die **Dienstleistung (engl. Service)** ist das „Ergebnis einer Organisation mit mindestens einer Tätigkeit, die notwendigerweise zwischen der Organisation und dem Kunden ausgeführt wird“ [DIN EN ISO 9000]. Die Dienstleistung fasst die Capabilities eines Unternehmens zusammen, reichert diese mit kaufmännischen Informationen an, und wird möglichen Kunden angeboten.*

Damit mehrere Partner im Netzwerk untereinander Informationen austauschen können, besteht der Bedarf einer einheitlichen Kommunikation zwischen den Partnern. Ein möglicher Standard zur herstellerunabhängigen Kommunikation ist die in Kapitel 2.2.2 vorgestellte I4.0-Sprache, in der nach VDI/VDE 2193-2 ein Ausschreibungsverfahren spezifiziert wird (siehe Abbildung 3.1). So kann sichergestellt werden, dass die Nachrichten über Cloud-Konnektoren ausgetauscht und aufgrund einer einheitlichen Semantik im verteilten Produktionsnetzwerk interpretiert werden können.

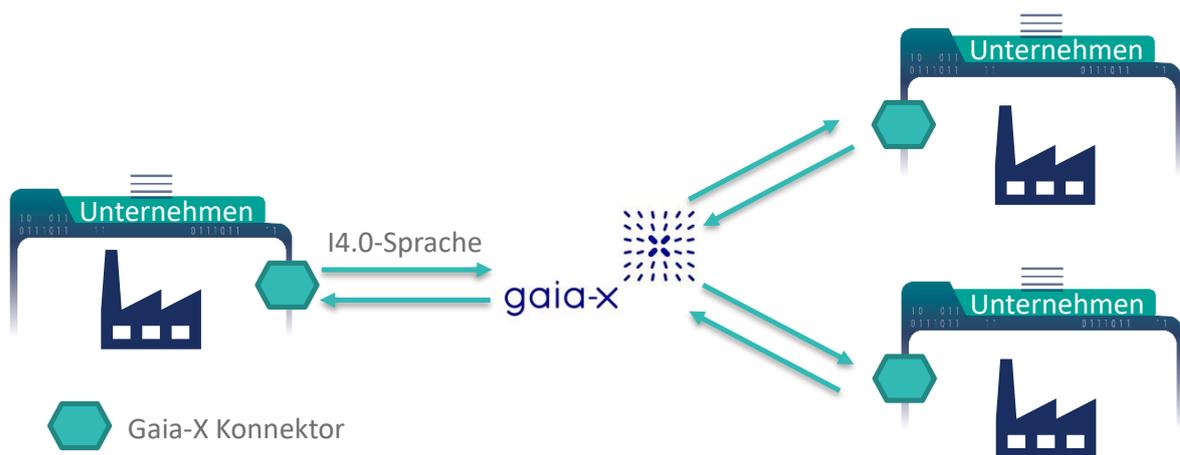


Abbildung 3.1: Vernetzung von Unternehmen mit Cloud-Konnektoren und der I4.0-Sprache

Damit die Fertigung I4.0-konform umgesetzt werden kann, besitzen in [PI20c] die Prozesse, die Produkte und die vorhandenen Ressourcen eine VWS. Somit können diese als I4.0-Komponenten in die Produktionsabläufe eingebunden werden. Das zu fertigende Produkt bzw. die durchzuführenden Produktionsschritte müssen ebenfalls standardisiert beschrieben wer-

den. Nur durch eine einheitliche Beschreibung von Produkthanforderungen ist ein automatisierter Angebotsprozess über mehrere Unternehmen möglich. Hierzu müssen Teilmodelle und Vorgehensweisen erarbeitet werden, eine Produkt-VWS zu erstellen.

Die Produkt-VWS soll Konstruktionsdaten und fertigungsrelevante Anforderungen enthalten, sodass ein Partner im Netzwerk das Produkt herstellen kann. Die in der Produkt-VWS enthaltenen Informationen spezifizieren die Features, Materialien und Stückzahlen des Produktes sowie organisatorische Kriterien, die an den Produktionsprozess gestellt werden. Die Suche nach möglichen Dienstleistern, die benötigte Fertigungsprozesse anbieten, kann auf Basis einer Produkt-VWS umgesetzt werden. Notwendige Qualitätsprüfungen werden in der Produkt-VWS hinterlegt, um eine vollständige Dokumentation des Fertigungsprozesses abzubilden. Die Produkt-VWS soll eine Vorlage bilden, die in jedem Schritt des Produktlebenszyklus mit zusätzlichen Informationen angereichert werden kann. Es entsteht eine digitale Lebenszyklusakte, wie in [PI22] beschrieben. Der Fokus der Arbeit liegt auf dem Angebotsprozess und der Fertigung eines Produktes. Die erarbeiteten Teilmodelle decken nach RAMI 4.0 daher den Beginn der Produktionsphase nach IEC 62890 ab (siehe Abbildung 3.2).

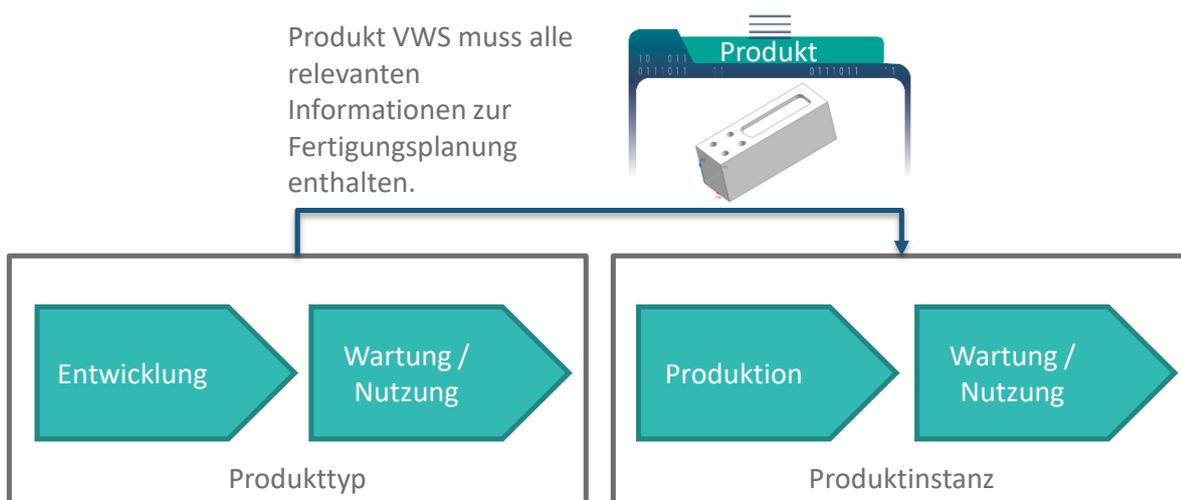


Abbildung 3.2: Fokus der entwickelten TM für die Produkt-VWS liegt in der Schnittstelle von der Entwicklung in die Produktion (Lebenszyklus in Anlehnung an RAMI 4.0 [DIN SPEC 91345])

In Anlehnung an die digitale Lebenszyklusakte aus [PI22], die auch als VWS umgesetzt wurde, wird die Produkt-VWS folgendermaßen definiert:

*Die **Produkt-VWS** enthält alle relevanten Informationen eines Produktes über den gesamten Produktlebenszyklus. Die relevanten Informationen werden in mehreren Teilmodellen abgebildet, die über den Lebenszyklus eines Produktes entstehen und ergänzt werden. Je nach Zugriffsberechtigung werden Teilmodelle im Partnernetzwerk freigegeben.*

Über die Produkt-VWS können individuelle Produktionsanfragen an produzierende Unternehmen gesendet werden. Um eine automatisierte Identifikation eines Fertigungsprozesses zu ermöglichen, müssen die beschriebenen Anforderungen der Produkt-VWS den möglichen Dienst-

leistungen und Fertigungsressourcen zugeordnet werden. Dieser Prozess wird im weiteren als Matching beschrieben [He21; Ma18a].

*Im **Matching** werden angeforderte Dienstleistungen und Capabilities mit vorhandenen Dienstleistungen und Capabilities abgeglichen sowie geprüft, ob die angeforderten Eigenschaften zugesichert werden können.*

Die angeforderten Dienstleistungen und Capabilities müssen durch Features und beschriebene Eigenschaften in der Produkt-VWS identifiziert werden. Abbildung 3.3 zeigt die notwendigen Matchings zur Identifikation eines individuellen Fertigungsablaufs.

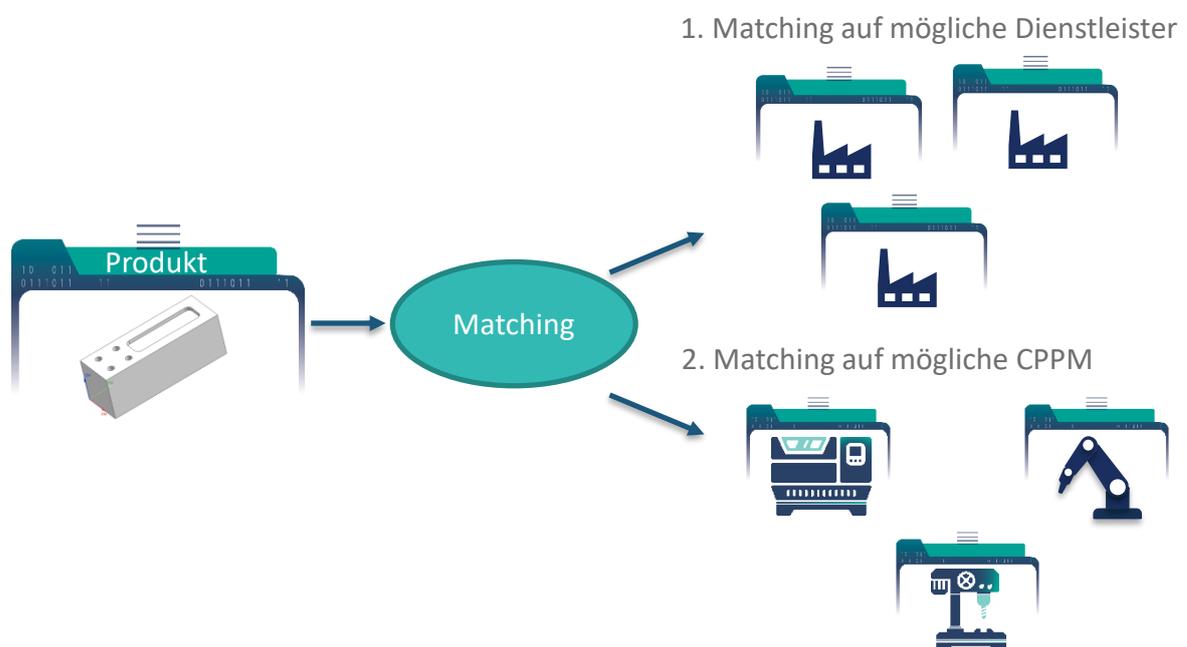


Abbildung 3.3: Zum individuellen Erzeugen von Produktionsprozessen ist das Matching der Produkthanforderungen auf mögliche Dienstleister und Produktionsressourcen notwendig

Um ein Matching zwischen Produkthanforderungen und Produktionsressourcen zu erlauben, ist eine vollständige Beschreibung der Produktionsressourcen notwendig. Eine Produktionsressource ist in diesem Fall ein CPPM und ist in Kapitel 2.1.2 definiert. In der VWS des CPPM werden unter anderem die Capabilities eines CPPM beschrieben. Ein CPPM kann sich aus mehreren I4.0-Komponenten zusammensetzen, z. B. eine Maschine inklusive Werkzeuge, die wiederum als I4.0-Komponenten eingebunden werden. Capabilities referenzieren auf Skills, sodass eine Capability durch ein CPPM realisiert werden kann (siehe Abbildung 3.4). In der VWS eines CPPM muss die Schnittstelle zu den angebotenen Skills beschrieben oder durch die VWS erreichbar werden (siehe Abbildung 2.6 in Kapitel 2.1.2). So können Planungs- und Steuerungskomponenten die Schnittstelle interpretieren und nutzen. Die CPPM bilden autonome Einheiten, die z. B. eigenständig ihre Capabilities erweitern, wenn ein zusätzliches Werkzeug zum CPPM hinzugefügt wird.

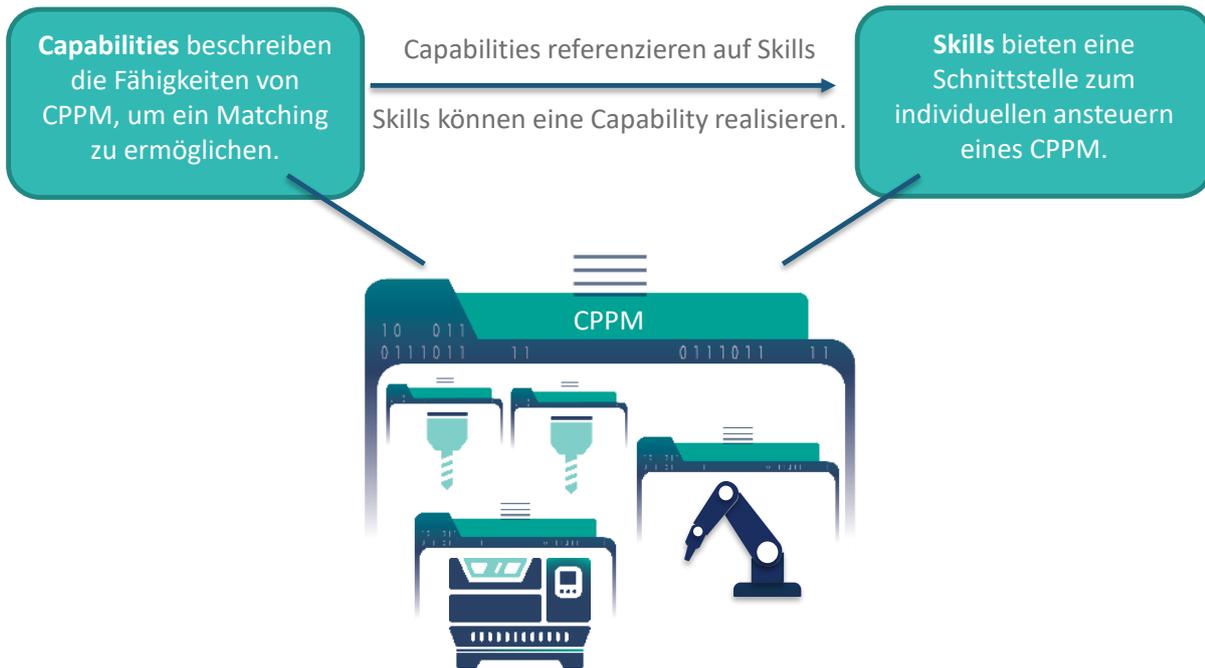


Abbildung 3.4: Beschreibung von Capabilities und Skills in der VWS eines CPPM

Um die Aufgaben des Matchings und der Fertigungsausführung auf Unternehmensebene zu erfüllen, wird eine Software benötigt, die Funktionalitäten eines MES- und ERP-Systems stark verzahnt. Es wird der Zugriff auf alle fertigungsrelevante Informationen wie vorhandene Rohmaterialien oder Capabilities und Skills der CPPM benötigt. Die bisherigen MES-Funktionalitäten werden hauptsächlich zur Kommunikation mit den vorhandenen CPPM benötigt, z. B. zur Prüfung der Ausführbarkeit und Ausführung von Skills. Die Funktionalitäten des ERP-Systems werden zur Planung der Ressourcen benötigt. Diese Ebene, die die Aufgaben von ERP- und MES-Systemen vereint, wird im weiteren als Middleware bezeichnet und verbindet die in der Produktionsumgebung vorhandenen Ressourcen sowie Steuerungsinformationen, Planungsalgorithmen und den Materialbedarf miteinander. Als Middleware können z. B. MAS eingesetzt werden, sodass verschiedene Agenten spezifische Aufgaben übernehmen.

Jungbluth et al. beschreiben z. B. den Einsatz von MAS in der Fertigungsausführung [Ju22]. Damit eine vollständige Fertigungsplanung und -ausführung umgesetzt werden kann, müssen alle notwendigen Assets wie Ressourcen oder Capabilities digital abrufbar sein. Die Assets sind, wie in Abbildung 3.5 dargestellt, als VWS modelliert. Durchzuführende Prozesse werden in die Auftragsplanung und die Auftragsdurchführung untergliedert. Die Auftragsplanung stellt das intern im Unternehmen stattfindende Matching dar, um ein Angebot und eine mögliche Prozesskette zu erstellen. Die Auftragsdurchführung ist das anschließende Aufrufen der benötigten Skills von CPPM, um ein Produkt zu fertigen. Weitere Ressourcen sind Logistikeinheiten. Die Beschreibung der Logistikeinheiten definiert, welche CPPM diese beliefern und welche Art von Produkten diese handhaben können.

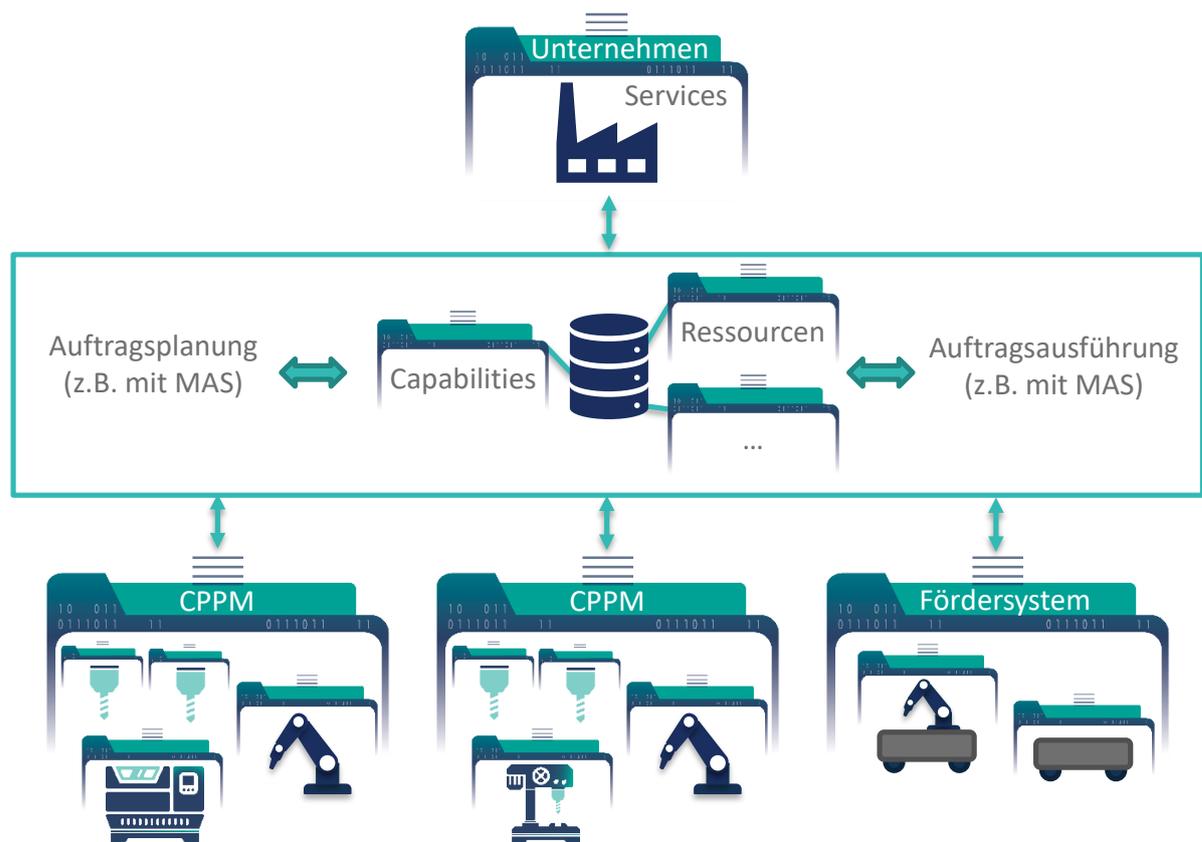


Abbildung 3.5: Darstellung der RAMI 4.0-konforme Modellierung eines Unternehmens

Mit Blick auf Lohnfertiger in der Fräsbearbeitung wird beispielhaft ein Fertigungsprozess in einer CM-Umgebung beschrieben. Das Ziel ist das automatisierte Bereitstellen möglicher Lohnfertiger. Der Prozess startet mit der Konstruktion und der Freigabe der Konstruktionsdateien im Netzwerk. Die Produktinformationen werden analysiert und in einer VWS aufbereitet. Benötigte Dienstleistungen zur Fertigung des Produktes sowie bereitstellende Teilnehmer werden im Netzwerk analysiert. Unternehmen, die eine benötigte Dienstleistung bereitstellen, wird die Produktbeschreibung zugesendet. In den jeweiligen Unternehmen werden die Produktanfragen auf die vorhandenen Capabilities gematcht. Kann das gewünschte Produkt gefertigt werden, gibt der Lohnfertiger ein Angebot ab. Anhand der erhaltenen Angebote können unterschiedliche Lieferketten identifiziert werden. Der Kunde kann eine der möglichen Lieferketten auswählen. Dabei können verschiedene Optimierungsfaktoren, wie eine besonders schnelle oder kostengünstige Fertigung, eine Rolle spielen. Der Prozess, in dem Informationen zwischen Unternehmen und einer Fertigungsplattform herstellerunabhängig ausgetauscht werden, ist in 6 Schritte untergliedert. Der Prozess wird in Abbildung 3.6 skizziert. Es wird in der Beschreibung ein Prozess fokussiert, in dem die vorhandenen CPPM eingebunden werden, um eine automatisierte Angebotserstellung zu ermöglichen. So sollen auch fertigungsspezifische Faktoren und Prozesswissen einen höheren Stellenwert im Angebotsprozess erhalten.

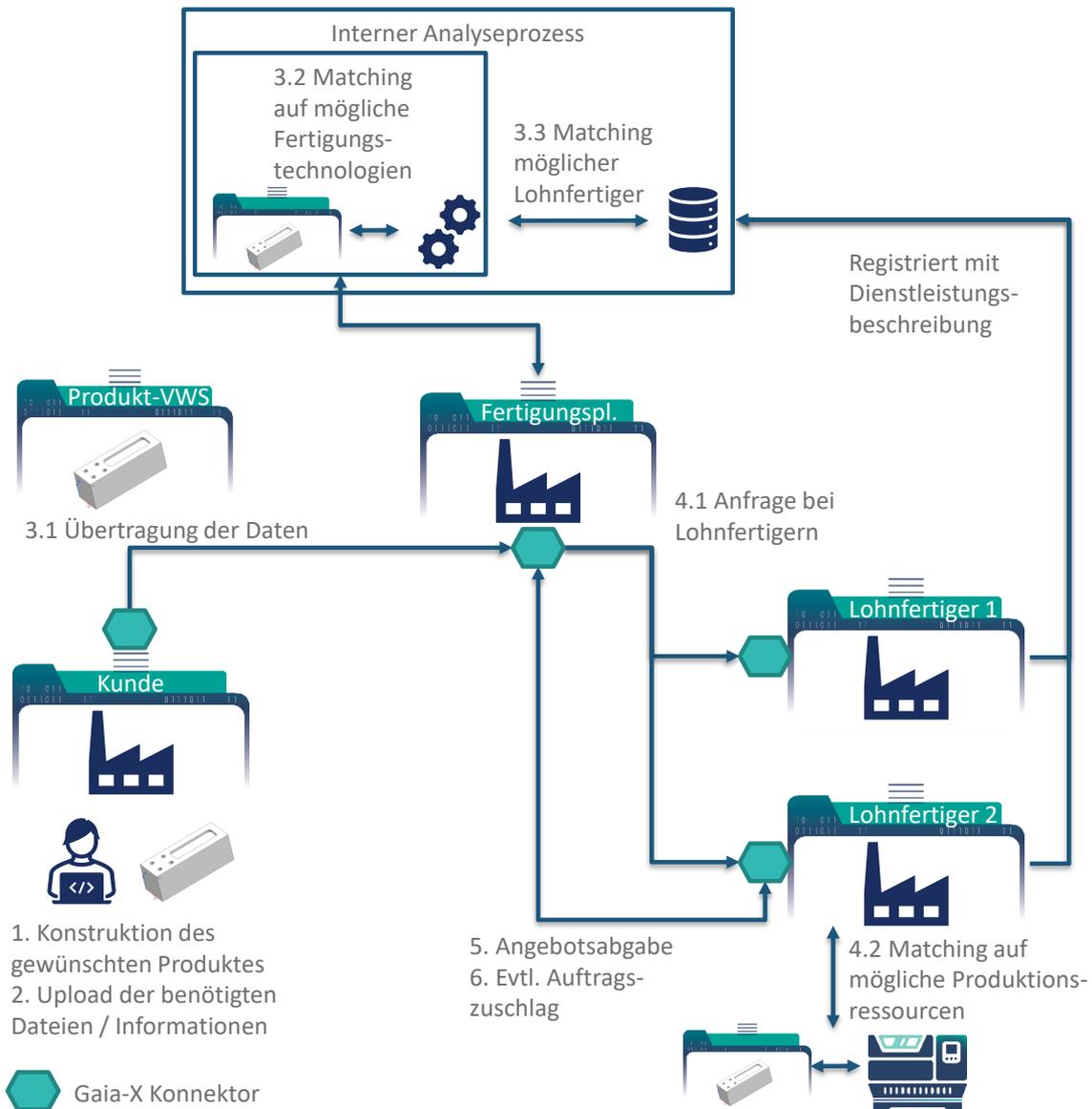


Abbildung 3.6: Prozessschritte zur Fertigung eines individuellen Bauteils über eine Fertigungsplattform

Die zu durchlaufenden Schritte des CM-Prozesses werden im Folgenden beschrieben:

### 1. Konstruktion und Upload eines CAD Files

Der Fertigungsauftrag startet mit dem Konstruktionsprozess in einer CAD-Software. Zum einen kann ein Kunde nach dem Konstruktionsprozess sein Modell inklusive einer optionalen Technischen Zeichnung über eine webbasierte Oberfläche auf eine Plattform hochladen. Zum anderen kann eine über das Netzwerk bereitgestellte Software (*Software as a Service (SaaS)*) genutzt werden, um die Konstruktion des Bauteiles z. B. direkt in einer Cloudumgebung vorzunehmen. Im Idealfall hinterlegt der Konstrukteur im CAD-Modell alle relevanten *Product Manufacturing Informations (PMI)* z. B. Toleranzen, Oberflächengüte, Passungen, Gewinde und das benötigte Material. Optional können diese auch in einer beigelegten Technischen Zeichnung definiert

werden. Die Verantwortung des Konstrukteurs, das Bauteil auf mögliche Fertigbarkeit zu prüfen, steigt in einem automatisierten Fertigungsprozess. In einer Eingabemaske gibt der Kunde zwingende sowie optionale Ergänzungen zum Auftrag an. Zusätzliche Parameter, die ein Nutzer im Angebotsprozess angibt, sind im Folgenden aufgelistet. Die Parameter werden für die Erstellung der Produkt-VWS in Kapitel 3.2.3 weiterverwendet. Benötigte Daten zum Starten eines Auftrages sind:

- Kontaktdaten
- Stückzahl
- Zahlungsbedingungen/-vorgaben
- Konstruiertes Bauteil bzw. Technische Zeichnung

Weitere evtl. benötigte Fertigungsparameter sind:

- Materialvorgaben, wenn diese nicht im CAD oder der Technischen Zeichnung hinterlegt sind
- Zusätzliche Wünsche bezüglich Oberflächenlegierungen, Nachbehandlungen oder die Vorgabe physikalischer Bauteilparameter (Härte, Zugfestigkeit)
- Gewünschter Liefertermin oder Etappenlieferungen von Losgrößen
- Vorgabe des Fertigungsprozesses (z. B. Fräsen, Drehen, Additive Fertigung)
- Allgemeintoleranzen für Längen- und Winkelmaße (DIN ISO 2768-1 ) sowie Form und Lage (DIN ISO 2768-2 ), wenn diese nicht im CAD oder der Technischen Zeichnung hinterlegt sind.
- Optimierung des Fertigungsprozesses nach möglichst ökologischen (ressourcenschonend), ökonomischen (kostengünstig) oder zeitlichen (schnelle Lieferung) Faktoren
- Gewährleistung bzw. Zertifizierung der Produktion nach existierenden Normen

## **2. Erstellung der Produkt-VWS**

Die gesamten Anforderungen und Dateien (CAD, Technische Zeichnung) werden in eine Produkt-VWS extrahiert. Hierzu müssen alle notwendigen Informationen über den Kunden und genaue Informationen über das Produkt hinterlegt werden. Die Erstellung der VWS kann sowohl direkt aus dem CAD-System geschehen, als auch bei einem dafür spezialisierten Dienstleister (z. B. einer Fertigungsplattform). Für die Anfrage individueller Bauteile in der spanenden Bearbeitung ist ein Dienstleister als Mittelsmann sinnvoll und wird in der weiteren Betrachtung in den Prozess einbezogen. Der Kunde übergibt seine CAD-Daten inklusive der Informationen aus Schritt 1 an einen Dienstleister. Dieser analysiert automatisiert das mitgelieferte CAD-Modell bzw. die optionale Technische Zeichnung und generiert eine Produkt-VWS. Anhand der Produkt-VWS werden mögliche Fertigungsfolgen und mögliche Fertigungsunternehmen bestimmt. Die Fertigungsun-

ternehmen sind anhand der Informationen ihrer Unternehmens-VWS bei dem Dienstleister in einer *Registry* (bzw. nach Gaia-X Begrifflichkeiten in einem *Federated Catalogue*) hinterlegt. Identifiziert die Fertigungsplattform z. B. einen individuellen Fräsauftrag für ein gewisses Material einer bestimmten Größe, beginnt dieser potenzielle Auftragsfertiger zu filtern.

### 3. Übermittlung der Daten an potenzielle Auftragsfertiger

Unternehmen bieten im Produktionsnetzwerk Dienstleistungen auf Basis ihrer Unternehmens-VWS an. Eine Dienstleistung kann auf einer sehr allgemeingültigen Beschreibung beruhen, z. B. auf verfügbaren Fertigungstechnologien, die durch limitierende Faktoren wie Stückzahlen oder Bauteilgrößen eingeschränkt werden. Ein Lohnfertiger könnte z. B. als Dienstleistung die Fräsbearbeitung eines individuellen Bauteils in dem Material Aluminium und einer Größe von 50x50x10 mm bis zu max. 400x400x300 mm anbieten. In [He21] wird ein Matching-Konzept erläutert, welches mögliche Technologieketten aus Produkthanfragen erzeugt (siehe Abbildung 3.7).

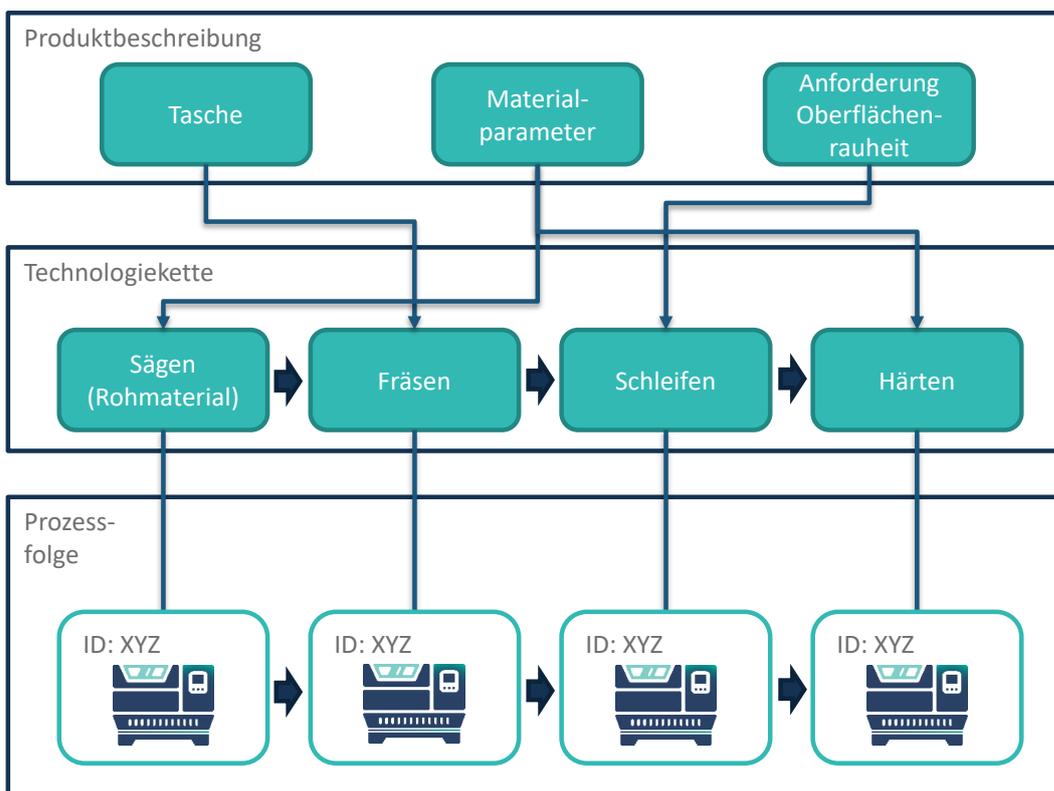


Abbildung 3.7: Matching einer Produktbeschreibung auf eine mögliche Technologiekette und Prozessfolge in Anlehnung an [He21]

Auf Basis der Technologieketten werden entsprechende Prozessschritte identifiziert [He21]. Eine Technologie wäre in diesem Fall das Fräsverfahren, ein Prozessschritt ein Fertigungs-Feature wie z. B. Tasche fräsen. Das Ergebnis kann genutzt werden, um benötigte Fertigungstechnologien oder Prozessschritte als Dienstleistung eines Unternehmens zu identifizieren. Sowohl die Features als auch Fertigungstechnologien als auch spezifische Fertigungsschritte sollten sich möglichst an einem Standard orientieren, sodass eine semantische Interpretation der Daten

möglich wird. Die Informationen aus der Analyse des Bauteils erweitern die Produkt-VWS und beschreiben die benötigten Technologien (z. B. Fräsen oder spezifischere Fertigungs-Features). Lohnfertiger, die die angeforderten Technologien oder Prozessschritte sowie erste Filterkriterien wie z. B. Materialien und Stückzahlen erfüllen, erhalten eine Angebotsanfrage für das gewünschte Produkt. So können die Informationen im Matching spezifischen Ressourcen zugeordnet werden. Für die Verhandlungen zwischen Unternehmen über einen Cloud-Konnektor kann das in der VDI/VDE 2193.2 standardisierte Ausschreibungsverfahren zum Informationsaustausch angewandt werden.

Sowohl auf Ebene der Fertigungstechnologie als auch eines Prozessschrittes ist eine Angebotsanfrage denkbar. Dienstleistungen können auf verschiedenen Abstraktionsebenen beschrieben werden. Genauso könnte eine Dienstleistung die Lieferung eines bereits spezifisch definierten Bauteils sein, das z. B. unter einer standardisierten ID angeboten wird. Für individuelle Anfragen ist das Versenden der gesamten Produkt-VWS notwendig, sodass ein Unternehmen Einblick in das zu fertigende CAD-Modell oder die Technische Zeichnung erhält. Eindeutige Produktanfragen, für standardisierte oder bekannte Produkte, können direkt als Nachricht zwischen zwei Unternehmen ausgetauscht werden. Zu der Dienstleistungsbeschreibung eines Unternehmens zählen nicht nur die ausführbaren Fertigungstechnologien und Prozessschritte, sondern auch Normen, Zertifikate, Zahlungsbedingungen und Qualitätsstandards, die ein Unternehmen anbieten kann. Die Konnektoren im Netzwerk sollten die Notwendigkeiten an den Datenschutz und die Sicherung der Nutzungsrechte im Angebotsprozess abdecken. Die Prüfung einer möglichen Fertigung wird anschließend von den angefragten Lohnfertigern durchgeführt. In Kapitel 3.3.1 wird genauer auf die Dienstleistungsbeschreibung als TM der VWS eingegangen.

#### **4. Verarbeiten einer Angebotsanfrage**

Die Angebotsanfrage kann auf drei unterschiedlich stark automatisierten Stufen von einem Lohnfertiger verarbeitet werden:

- Stufe 1: Der Angebotsprozess wird „per Hand“ durchgeführt. Ein Mitarbeiter erstellt anhand seiner Erfahrungen einen groben Fertigungsplan sowie eine Kostenkalkulation, die mit evtl. intern existierenden Kostentabellen erstellt wird.
- Stufe 2: Der Mitarbeiter wird bei der Auswahl der Fertigungsressourcen und Kostenkalkulation durch eine Software (wie in Kapitel 2.5 beschrieben) unterstützt. Hierzu wurden relevante Informationen, wie die Fertigungsressourcen, Rohmaterialien und Kostenparameter, durch den Lohnfertiger in einer Datenbank hinterlegt.
- Stufe 3: Die Erstellung eines Fertigungsplanes und der Kostenkalkulation ist automatisiert. Hierzu werden CPPM in den Planungsprozess einbezogen, um eine Fertigungsprüfung und Kostenanalyse durchzuführen.

Im Folgenden wird Stufe 3 und damit eine vollständig automatisierte Erstellung eines Angebotes beschrieben. Im Matchingprozess wird zunächst das in der Produkt-VWS definierte Material und die hinterlegte Bauteilgröße mit dem vorhandenen Rohmaterial abgeglichen. Ist das benötigte Rohmaterial nicht vorhanden, müsste zunächst die Bestellung des Materials geprüft werden. Die Materialbestellung kann ebenfalls über die Cloud-Konnektoren getriggert werden. Mögliche Verzögerung und zusätzliche Kosten fließen in den Angebotsprozess mit ein. Im nächsten Schritt wird geprüft, ob die im CAD-Modell vorhandenen Features von den vorhandenen Produktionsressourcen gefertigt werden können. Hierzu werden die geometrischen Features auf mögliche Capabilities und Fertigungsskills gematcht. Die Middleware identifiziert CPPM mit den benötigten Capabilities und Skills und führt eine Prüfung durch, ob die Featureinformationen von einem entsprechenden CPPM umgesetzt werden können. Trifft dies zu, wird die voraussichtliche Bearbeitungsdauer, Kostenscheinschätzung und Energieverbrauch für eine Skillausführung berechnet. Die für die Fertigung infrage kommenden CPPM werden mit ihrem jeweiligen Fertigungsschritt (Skill) und den zugehörigen Fertigungsinformationen (Dauer, Kosten, Energieverbrauch) gesammelt. Ein Schedulingalgorithmus kann anhand der bisher geplanten Maschinenauslastung eine grobe Planung der Fertigungsschritte und vsl. Kosten vornehmen. Somit wird ein Angebot automatisiert erstellt. Ist der Fertigungsprozess nicht möglich, wird die Angebotsanfrage storniert. Durch die automatisierte Angebotserstellung entsteht sowohl für den Kunden als auch für die Lohnfertiger eine Zeitersparnis.

## **5. Angebotsentscheidung**

Die Angebote können zum einen auf der Fertigungsplattform gesammelt und dem Kunden bereitgestellt oder direkt an den Kunden gesendet werden. Der Kunde kann anhand der Angebotsübersicht ein für ihn passendes Angebot auswählen. Wird der Auftrag ausgelöst, erhält der jeweilige Auftragsfertiger die Freigabe zur Fertigung. An die Mitbewerber kann über die Cloud-Konnektoren eine Absage des Angebotes erteilt werden. Die Konnektoren sollten dabei z. B. auch Konstruktionsdaten bei den Unternehmen löschen, die keine Angebotszusage erhalten haben.

## **6. Detailplanung und Ausführung des Auftrages**

Ein oder mehrere hintereinander geschaltete Lohnfertiger erhalten den Auftrag zur Produktion eines Bauteils. Die zeitliche Detailplanung und Verwendung der spezifischen Ressourcen werden festgelegt. Ein Mitarbeiter sollte noch die Möglichkeit erhalten, die Bearbeitungsreihenfolge zu prüfen und zu verändern. So ist die letzte Entscheidungsinstanz durch den Menschen gegeben. Sind mehrere Lohnfertiger am Produktionsprozess beteiligt, müssen mögliche zeitliche Abhängigkeiten untereinander in Betracht gezogen werden. In den Unternehmen werden die entsprechenden Skills der CPPM und Transportsysteme getriggert, um den Produktionsablauf umzusetzen. Informationen des Fertigungsprozesses können in der Produkt-VWS gesichert werden. Bei Verzögerungen sollten beteiligte Lohnfertiger durch eine automatisierte Nachricht rechtzeitig informiert werden. So können Prozessschritte evtl. rechtzeitig verschoben werden.

Dem Kunden sollten Informationen über die Teilschritte der Fertigung bereitgestellt werden, sodass dieser den Fertigungsablauf mitverfolgen kann.

In den folgenden Unterkapiteln werden, wie in Abbildung 3.8 skizziert, in Kapitel 3.2 die Erstellung einer Produkt-VWS diskutiert. Kapitel 3.3 analysiert die Beschreibungen von Capabilities und Services sowie Möglichkeiten des Matchings. Kapitel 3.4 beschäftigt sich mit der Schnittstelle, um CPPM anzusteuern.

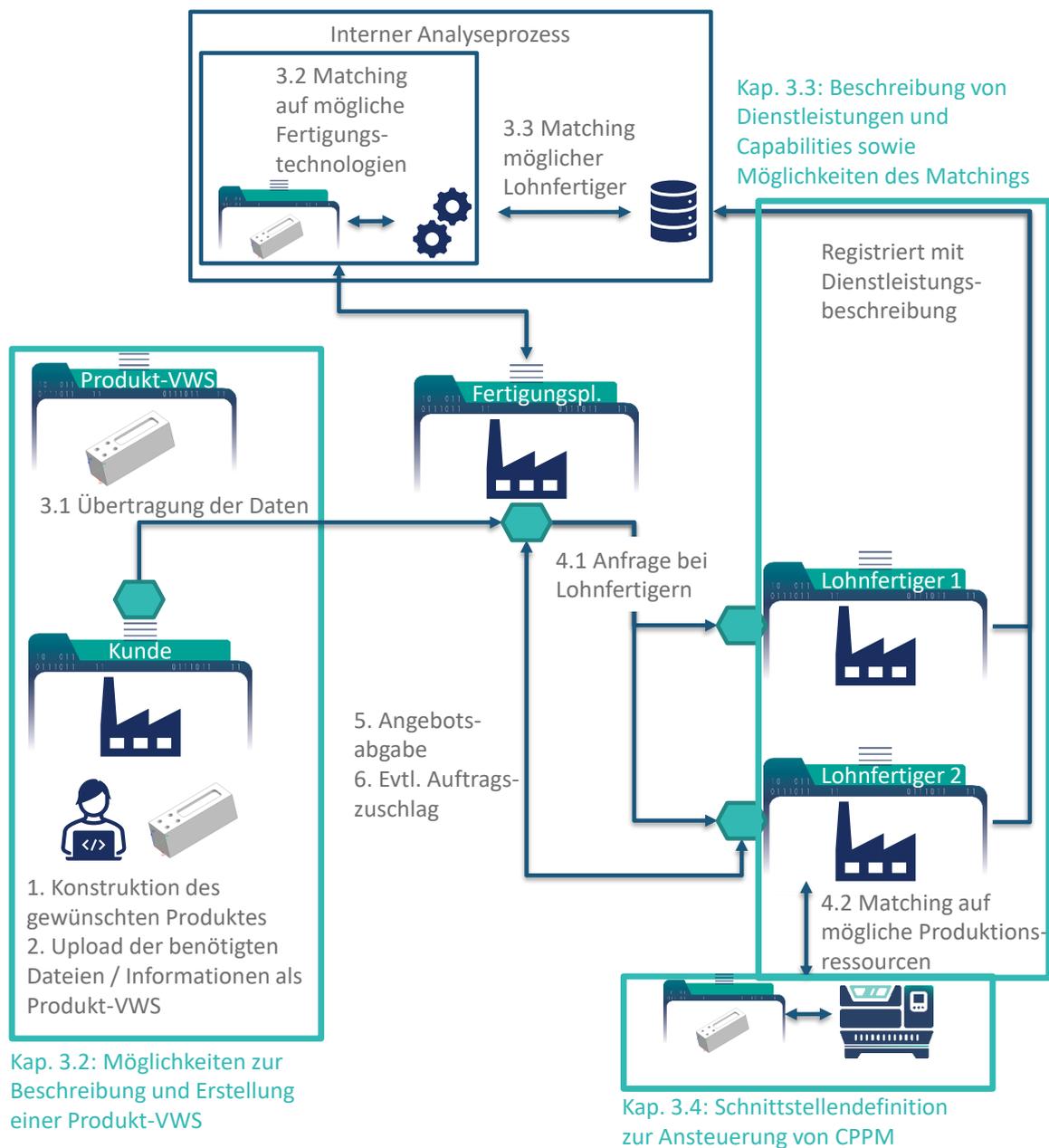


Abbildung 3.8: Übersicht Inhalte Kapitel 3

### 3.2 Featureextraktion zur Generierung einer Produkt-VWS

Zu Beginn der Fertigung werden Modelle des benötigten Bauteils erstellt. Einem Konstrukteur stehen hierzu verschiedene CAD-Softwarelösungen mit einer Vielzahl von Konstruktionsmöglichkeiten bereit. CAD-Programme haben zum größten Teil die manuelle Zeichnungserstellung abgelöst [Au17; Bö15]. Um Anforderungen des Produktes direkt aus einem 3D-Modell identifizieren zu können, ist das Ziel, semantische Feature mit den benötigten Parametern zu extrahieren. Die zu fertigenden Feature sollen im Matching möglichen Dienstleistungen und im späteren Produktionsverlauf, einer Fertigungsmaschine zugeordnet werden. Zur Featureerkennung können die im Kapitel 2.3.2 beschriebenen Möglichkeiten *Design by Feature*, Featureerkennung (interaktiv oder automatisch) und das Featuremapping eingesetzt werden. Das Ziel ist es eine automatische Featureerkennung umzusetzen, um nach dem Konstruktionsprozess automatisiert eine Liste der zu fertigenden Feature zu erzeugen. Der Konstrukteur ist durch die automatische Featureerkennung im Designprozess an keine fest vorgegebenen Konstruktionsfeature gebunden, sondern kann seine gewohnte Konstruktionsmethodik beibehalten und anwenden (siehe Abbildung 3.9).



Abbildung 3.9: Erstellung eines VWS-TM zur Produktbeschreibung von zu fertigenden Bauteilen

In bestehenden CAM-Softwareprodukten ist häufig eine interaktive Featureerkennung mit Featuremapping implementiert. Wählt der Anwender beispielsweise in einem CAD-Bauteil eine konstruierte Rechtecktasche aus, werden ihm verfügbare Fertigungsoperationen zum Fertigen einer Tasche vorgeschlagen. Jeder Anbieter von CAM-Software verwendet in der Regel eigens definierte Featuretypen. Ebenfalls existieren Anbieter für kombinierte CAD-/ CAM-Lösungen, z. B. Siemens NX oder Autodesk Fusion 360, die sowohl CAD- als auch CAM-Funktionen unterstützen. Die Hersteller verwenden in der Software eigens entwickelte, proprietäre Dateiformate. Siemens NX nutzt beispielsweise zur Speicherung von 3D-Modellen standardmäßig das Dateiformat *.prt* und Fusion 360 *.f3d*. Innerhalb der entsprechenden Softwareumgebung werden die nativen Dateiformate am besten unterstützt. Werden Bauteile in ein fremdes Format exportiert, kann es zu einem Informationsverlust kommen. Das bekannteste herstellernerneutrale Austauschformat ist STEP (siehe Kapitel 2.3.3). Somit ist es sinnvoll, die Featureerkennung auf Basis von STEP vorzunehmen. Die extrahierten STEP-Features können in der Produkt-VWS gesichert und von jedem Lohnfertiger eindeutig zugeordnet werden.

Mit Hilfe der Features kann anschließend ein Prozessplan erzeugt werden [KP09]. Im nächsten Kapitel wird die Extraktion von Features, standardisiert durch STEP-AP 242, betrachtet.

### 3.2.1 Extrahierbare Features aus STEP-AP 242

In Kapitel 2.2.3 wird STEP-AP 242 [ISO 10303-242] beschrieben. Als herstellerunabhängiger CAD-Standard eignet sich das Format ideal zur Definition von geometrischen Features. Zur besseren Verständlichkeit werden folgende Begrifflichkeiten zu STEP eingeführt [ISO 10303; KX09].

- *Application Module (AM)*: Ein AM bildet Fiktionalitäten ab und ähnelt einem AP (siehe Kapitel 2.3.3). AM sind kompakte Einheiten, die zu größeren Informationsmodellen zusammengesetzt werden können z. B. zu AP.
- *Application Reference Model (ARM)*: Ein ARM modelliert Daten spezifischer Applikationen und definiert die Begrifflichkeiten der jeweiligen Anwendungsgebiete.
- *Application Interpreted Construct (AIC)*: Ein AIC ist die Übersetzung einer *Unit of Functionality (UoF)* in die STEP Sprache.
- *Unit of Functionalty (UoF)*: Ist ein Ansatz zur Modularisierung von STEP und enthält einzelne Entitäten, die Teilbereiche von ARM abdecken.
- *Module Interpreted Model (MIM)*: Übersetzt ein AM mit Hilfe von *Integrated Resources*, die zur Übersetzung von domänenspezifischen Modellen eingesetzt werden.

Da STEP modular aufgebaut ist, verweisen die wichtigsten Module zur Beschreibung von Fertigungs-Features wie folgt aufeinander: AP 242 verweist auf das AM 442, welches Bezug auf die *Machining\_Feature* aus AM 1814 nimmt. Im AM 1814 findet die hauptsächliche Definition aller Features statt. In der ISO 10303-1814 ARM-Entitäten werden die Basisformelemente wie Bohrungen oder Taschen als *Machining\_Feature* definiert. Featureübergänge wie Fasen und Rundungen werden als *Transition\_Feature* und Muster als *Replicate\_Feature* bezeichnet. In der zugehörigen MIM in Clause 5 wird ein Mapping der Features auf Standard-Entitäten verwiesen, die wiederum in der AIC 522 definiert sind (siehe Abbildung 3.10) .

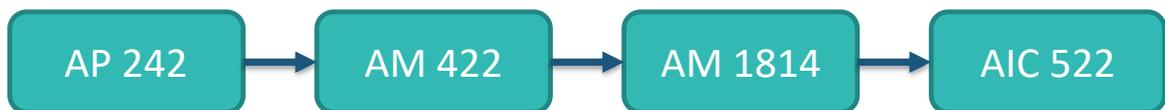


Abbildung 3.10: Referenzen auf *Machining Feature* in STEP-AP 242

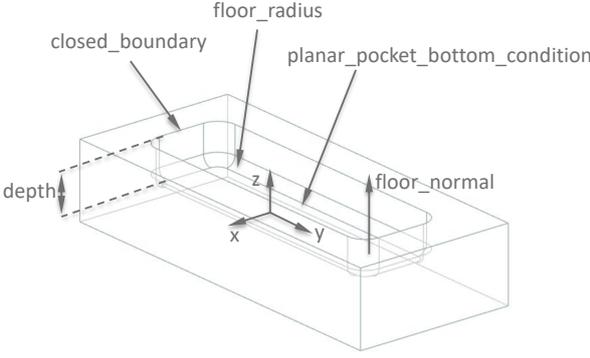
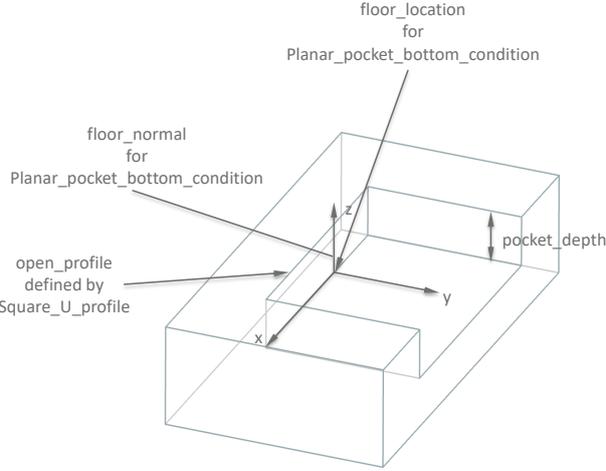
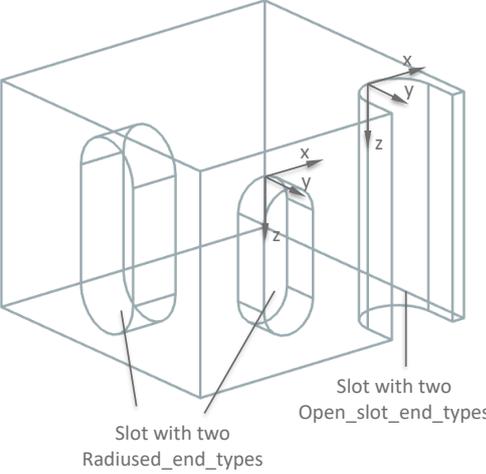
*Machining\_Feature* sind in STEP eine Unterkategorie vom *Manufacturing\_Feature*. Die wichtigsten Normen zur Beschreibung der *Machining\_Feature* sind:

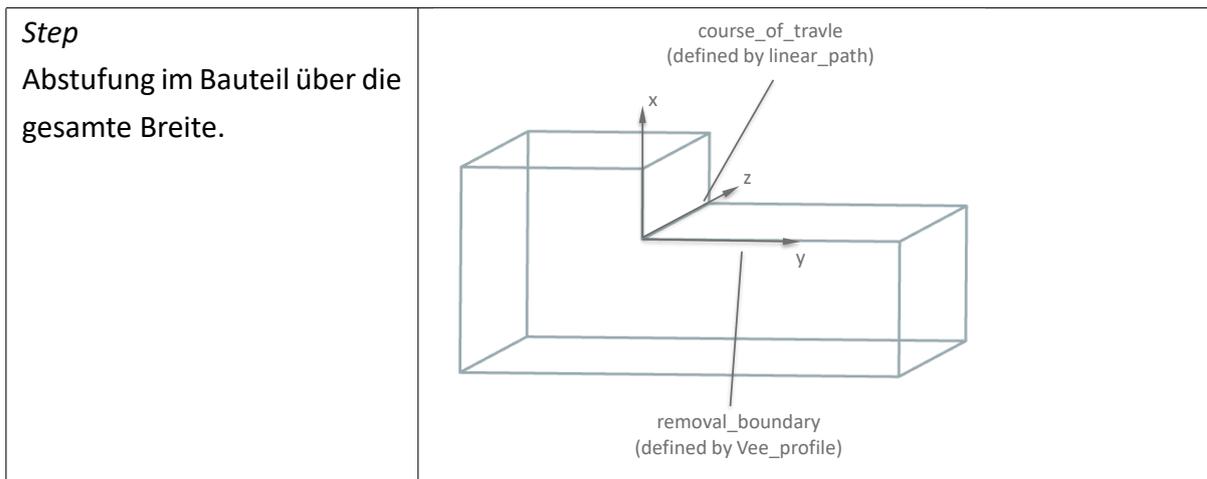
- ISO 10303-422: *Application module: AP242 managed model based 3D engineering*
- ISO 10303-1814: *Application Module: Machining Features*
- ISO 10303-522: *Application Interpreted Construct: Machining Features*

STEP-NC (AP 238 ) sieht vor, dass *Machining\_Feature* im CAM-Programm mit weiteren Informationsinhalten wie der Werkzeugbeschreibung, dem Werkzeugpfad und Fertigungsparametern

angereichert werden (*Machining\_Operations*). Eine STEP-NC-Steuerung wäre somit in der Lage, den erzeugten Code mit Bezug zum *Machining\_Feature* abzuarbeiten. AP 238 verweist auf mehrere Normen aus ISO 14649, in denen die *Machining\_Operations* genauer definiert werden. Eine Auswahl von *Machining\_Feature* ist in Tabelle 3.1 nach ISO 10303-1814 dargestellt, in Anhang 1 ist eine Gesamtübersicht der *Machining\_Feature* abgebildet.

Tabelle 3.1: Auswahl von Features nach [ISO 10303-1814]

STEP-AP 242 – Feature und Beschreibung	Abbildung in Anlehnung an [ISO 10303-1814]
<p><i>Rectangular_closed_pocket</i>                      Eine geschlossene Tasche (keine offenen Seitenwände). Die gegenüberliegenden Kanten sind gleichlang. Die Seitenwände sind rechtwinklig zueinander.</p>	 <p>The diagram shows a 3D perspective of a rectangular closed pocket. A dashed line indicates the 'closed_boundary' at the top. The 'depth' is shown as a vertical dimension. A coordinate system with x, y, and z axes is centered at the bottom. Labels include 'floor_radius' pointing to the rounded bottom corners, 'planar_pocket_bottom_condition' pointing to the flat bottom surface, and 'floor_normal' pointing to the normal vector of the bottom surface.</p>
<p><i>Rectangular_open_pocket</i>                      Eine offene rechteckige Tasche in der die gegenüberliegenden Seitenlängen gleichlang sind, jedoch eine der Seitenwände nicht vorhanden ist.</p>	 <p>The diagram shows a 3D perspective of a rectangular open pocket. The 'pocket_depth' is indicated by a vertical double-headed arrow. A coordinate system with x, y, and z axes is shown. Labels include 'floor_location for Planar_pocket_bottom_condition' pointing to the top surface, 'floor_normal for Planar_pocket_bottom_condition' pointing to the normal vector of the top surface, and 'open_profile defined by Square_U_profile' pointing to the open side of the pocket.</p>
<p><i>Closed_Slot</i>                      Eine geschlossene Nut.  <i>Opened_Slot</i>                      Eine offene Nut.</p>	 <p>The diagram shows three different slot configurations in a 3D perspective. The first is a 'Slot with two Radiused_end_types' (a rounded slot). The second is a 'Slot with two Open_slot_end_types' (a slot with open ends). A coordinate system with x, y, and z axes is shown for each slot.</p>



Die genauere Untersuchung vorhandener CAD-/ CAM-Systeme (z. B. Fusion 360, Siemens NX 1872, Mastercam) zeigte, dass ein Export oder Import von oder in das AP 242 Format teilweise mit Informationsverlusten behaftet ist und zu Problemen führte. CAD-/ CAM-Programme setzen STEP-Translatoren ein, um aus dem herstellereigenen Datenformat die CAD-Dateien in das STEP-Format zu überführen. In STEP werden entsprechende unterschiedliche *Conformance Classes* (CC) definiert. Eine CC besagt, dass bestimmte Funktionen eines STEP-AP unterstützt werden. In AP 242 wird nur die CC1 definiert, die auch eine Feature-Definition (AM 1814) voraussetzt. Verschiedene CAD-Softwares wurden dabei nach den CC häufig verfügbarer APs (bspw. AP224, 238, 242) untersucht.

Tabelle 3.2: Unterstützte STEP-AP in bekannten CAD Frameworks

CAD-Framework	Unterstützte STEP-AP		Quelle
	Import	Export	
OnShape	AP203, AP242 (geometry only)	AP214, AP242 E2	[@OnShape]
NX 1872	AP203, AP214, AP242		[@SiemensPLM3]
AutoDesk Fusion 360	AP203, AP242	AP214	durch Im-/Export getestet
Mastercam	keine eindeutige Angabe	AP203, AP214, AP242	MasterCam Help
Catia V5 (R2019x)	AP203 (CC2,3,4,5), AP214(CC1,2), AP242		[@Catia]
OpenCASCADE	AP214 (CC2), AP203, AP242 (teilweise)	AP203, AP214	[@OpenCascade]

Des Weiteren wurden die drei verfügbaren CAD-/ CAM-Systeme (Fusion 360, NX 1872 und Mastercam) nach einer Möglichkeit zur automatisierten STEP-Featureextraktion untersucht. Es ist

aufgefallen, dass automatisierte Featureextraktionen in den jeweiligen Softwares auf herstellerspezifischen Featuredefinitionen basiert. Keines der untersuchten Systeme erlaubte, Featureparameter konform zu AP 242 zu extrahieren. Dies deckt sich mit der Literaturrecherche [TDM21], in dieser wurden ebenfalls keine detaillierten Featureextraktoren bezogen auf STEP-AP 242 auffindig gemacht. Da der Fokus der Dissertation nicht auf der Entwicklung eines Featureextraktors liegt, wird im nächsten Kapitel die Möglichkeit der Extraktion von fertigungsrelevanten Features am Beispiel einer herstellerspezifischen Software untersucht.

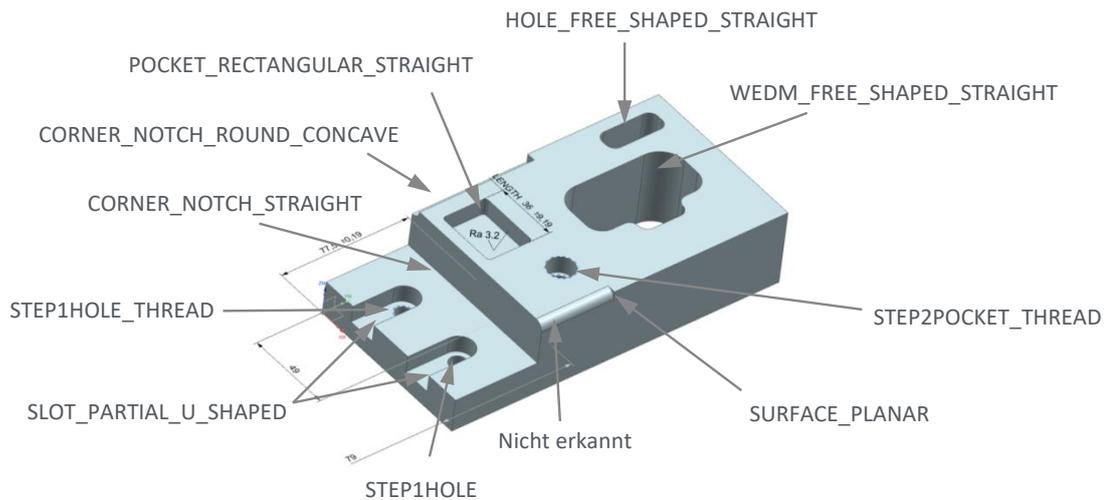
### 3.2.2 Extrahierbare Features in Siemens NX

Nach einer Analyse der verfügbaren CAD-/ CAM-Software (Siemens NX, Mastercam und Fusion 360) wurde unter anderem Siemens NX als System identifiziert, das die Möglichkeiten einer Featureerkennung anbietet. Um herstellerspezifische Features sowie STEP-Features vergleichen zu können, wird die Featuredefinition von Siemens NX in diesem Kapitel genauer betrachtet. Die Software bietet die Möglichkeit, über eine API das CAD-/ CAM-System individuell mit Softwarefunktionalitäten zu erweitern [@SiemensPLM1]. Mit Hilfe der API kann eine Featureliste mit allen verfügbaren Parametern exportiert werden. Daher werden die Siemens NX-Features genauer betrachtet. Die Analyse der konstruierten Bauteile wurde mit der Softwareversion 1872 von Siemens NX durchgeführt. Die Featureextraktion kann ebenfalls in der NX-Fertigungsumgebung per Mausklick verwendet werden. Mit Hilfe des Reiters „Formelement suchen“ kann die Featureerkennung gestartet werden. Es existieren drei unterschiedliche Modi der Featureerkennung [@SiemensPLM2]:

- Formelement Identifikation: Erkennt Features, die bereits featurebasiert konstruiert wurden.
- Manuelle Erkennung: Der Nutzer wählt Formelemente anhand von Flächen eigenständig aus.
- Parametrische Erkennung: Features werden automatisch anhand von Basisfeatures erkannt.

Wird eine Erkennung der Features durchgeführt, werden die identifizierten Features mit ihren Parametern angezeigt und können bei Bedarf auch als XML-Datei exportiert werden. Ein beispielhafter Auszug von häufig verwendeten Features in der Fertigung ist in Abbildung 3.11 dargestellt. Es wurde ein Musterteil konstruiert und eine parametrische Erkennung der Features angewendet. Bis auf die konvexe Rundung an der vorderen oberen Kante wurden alle zuvor konstruierten Features zuverlässig erkannt.

Ein Feature wird aus einer Anzahl von Parametern genauer beschrieben. Um einen Überblick über die umfangreiche Parameterdefinition zu geben, ist in Tabelle 3.3 ein Auszug der Parameter eines *POCKET\_RECTANGULAR\_STRAIGHT* Features gelistet. Neben Informationen wie der Län-



\*Die sechs Außenflächen des Rohteils werden als SURFACE\_PLANAR\_RECTANGULAR erkannt

Abbildung 3.11: Registrierte Feature nach einer parametrischen Erkennung in Siemens NX 1872

ge, Breite und der Tiefe der Tasche können auch Toleranz- und Oberflächenanforderungen im Feature hinterlegt werden. Aus Tabelle 3.3 wird ersichtlich, dass mit Hilfe der Feature-Parameter die Abfrage des Mittelpunktes einer Tasche sowie die Ausrichtung des Koordinatensystems auslesbar ist. In Abhängigkeit des globalen Koordinatensystems eines Bauteils ist die Position somit eindeutig definiert. Die Länge, Breite und Tiefe der Tasche sind inkl. Toleranzangaben (oberes und unteres Limit) vorhanden. Relevante Qualitätsanforderungen, wie die Oberflächenrauheiten vom Boden oder Seitenwänden, können in der Konstruktion hinterlegt werden. Des Weiteren sind für die Fertigung wichtige Parameter wie Eck- und Bodenradien definiert. Somit wird festgehalten, dass die Parameter zur Fertigung eines Features ausreichen.

Tabelle 3.3: Auszug der Parameter aus einem Siemens NX *POCKET\_RECTANGULAR\_STRAIGHT* Feature

Bottom- /Side_Roughness	Bottom /Side_Roughness_Standard	Bottom- /Side_Roughness_Waviness
Depth	Depth_Lower	Depth_Upper
Length	Length_Lower	Length_Upper
X_Position	X_Orientation_L	X_Orientation_D
Y_Position	Y_Orientation_L	Y_Orientation_D
Z_Position	Z_Orientation_L	Z_Orientation_D
Radius	Radius_Lower	Radius_Upper
Depth_Top_Chamfer	Angle_Top_Chamfer	Machining_Rule
...	...	...

NX bietet des Weiteren die Möglichkeit bereits vorgefertigte Bearbeitungsfeature unter dem Punkt *Machining\_Rule* zu hinterlegen. In der NX-Fertigungsumgebung steht hierzu der *Machine Knowledge Editor* (MKE) zur Verfügung [Si17]. Mit diesem können innerhalb der NX-Umgebung

unter anderem Feature auf einen anderen Featuretyp gemappt werden, um z. B. Fertigungsstrategien für bestimmte Feature und Feature-Parameter direkt im CAD-/CAM-System zu hinterlegen [Si17]. Mit Hilfe des MKE wird somit eine featurebasierte Programmierung eines Fertigungsprozesses innerhalb von NX ermöglicht, um den CAM-Prozess zu beschleunigen.

Die Parameter der Features wurden in Abhängigkeit von Konstruktionsvorgaben genauer betrachtet und durch zusätzliche Bauteilinformationen verändert. Hierzu wurden *Product Manufacturing Informations* (PMI) an der Rechtecktasche vorgesehen. Die Vorgabe einer gewünschten Rauheit für die Bodenfläche der Tasche ist in Abbildung 3.11 ersichtlich und wurde in den Parameter *Bottom\_Roughness* des *Pocket\_Rectangular\_Straight* Features übernommen. Toleranzangaben für Längen und Breiten der Taschen stellten sich problematisch dar. Die nachträglich eingefügten Toleranzen für eine Länge (siehe Tabelle 3.3) veränderten nicht die entsprechenden Parameter *Length\_Lower* oder *Length\_Upper* im Feature. NX orientiert sich bei der Vorgabe der Toleranzparameter an der DIN 2768 [DIN ISO 2768-1; DIN ISO 2768-2] für Allgemeintoleranzen und trägt diese je nach Größe eines geometrischen Parameters ein. Spezifische Nutzereingaben werden nicht übernommen. Dieses Problem wird in der späteren Implementierung gelöst und in Kapitel 4.1 genauer beschrieben. Mit NXOpen [@SiemensPLM1] wird ein *Application Programming Interface* (API) angeboten, über das die Features und Featureinformationen direkt ausgelesen werden können. Dieses kann genutzt werden, um eine Beschreibung der Produktinformationen automatisiert zu generieren. Die Überführung der extrahierbaren Produktinformationen in eine Produkt-VWS wird im nächsten Kapitel diskutiert.

### 3.2.3 Produkt-VWS

Eine VWS wird als RAMI 4.0-konformes Übergabemedium der relevanten Dateninformationen von einem Kunden an einen möglichen Lohnfertiger gewählt. [He21] ordnet der Klasse des „Hersteller-Produktmodells“, grundlegende Informationen, wie die Produkt-ID, den Produktnamen, Abmaße, das Gesamtgewicht, die Attribute Zieldimensionen, Funktionen und organisatorische Informationen zu. Nach [He21] definieren die Funktionen Informationen, „die für die Herstellung des Produkts notwendig sind“. Zieldimensionen definieren einzuhaltende Vorgaben im Bereich der Wirtschaftlichkeit, Geschwindigkeit, Qualität und Ressourceneffizienz [He21]. Organisatorische Informationen enthalten z. B. Liefer- oder Zahlungsbedingungen. Die Funktionen eines Bauteiles können durch semantische Features vom Typ Form-Feature und deren Eigenschaften genauer beschrieben werden [He21].

Die in Kapitel 3.2.2 beschriebenen NX-Feature werden genutzt, um eine Produkt-VWS zu erstellen. In Kapitel 3 wurde die Produkt-VWS als Mittel zum Informationsaustausch vorgestellt. Basierend auf Abbildung 3.6, werden zu den Teilschritten des Fertigungsablaufes Teilmodelle für die Produkt-VWS definiert. Hierzu sind zum einen die extrahierten Featureinformationen notwendig, zum anderen auch weitere organisatorische Informationen, die in Ka-

pitel 3 bereits spezifischer erläutert wurden. Als Produktbeispiel dient ein Anhänger Aufbau für einen Klemmbaustein-Lkw, der in einem *Shared-Production* Use-Case gemeinsam mit der Technologie-Initiative Smart-Factory KL e. V. gefertigt wird [@SFKL] (siehe Abbildung 3.12). In der folgenden Betrachtung werden die im Anhänger verwendeten Bohrungs-Feature und das Rechtecktaschen-Feature genauer betrachtet.

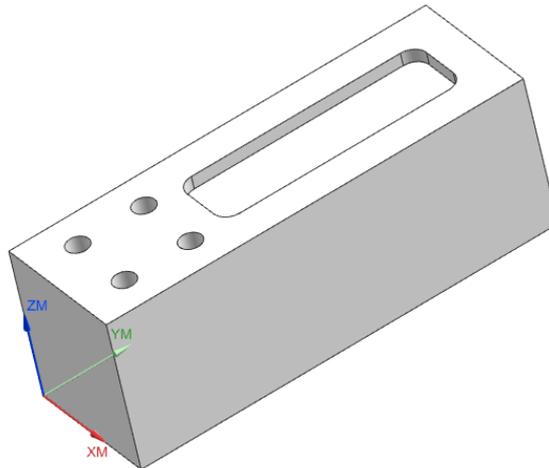


Abbildung 3.12: Anhänger als Beispiel zum Erzeugen der Produkt-VWS

Über den gesamten Produktlebenszyklus wird die Produkt-VWS mit weiteren TM ausgestattet bzw. vorhandene TM mit Informationen ergänzt. So liegen alle relevanten Informationen eines Produktes in digitaler Form vor. Der Fokus dieses Kapitels liegt auf der Beschreibung eines TM, um ein Matching durchführen zu können. Zur späteren Weiterverwendung der VWS-TM sind alle Bezeichnungen in den TM in Englisch beschrieben. Im Folgenden werden die relevanten TM zur Speicherung von Produktinformationen beschrieben, die zur Prozessgenerierung wichtig sind. Ist nach Abbildung 3.6 der Konstruktionsprozess abgeschlossen, beginnt Schritt zwei. Alle relevanten Daten, des CAD-Modells werden in eine VWS überführt. Zur Erstellung der VWS-TM wurde der *AASX-Package Explorer* [@AASX] genutzt. In Abbildung 3.13 wird die Produkt-VWS dargestellt. Das erstellte TM *ProductIdentification* enthält allgemeine Produktinformationen wie eine ID, den Produktnamen und -typ sowie eine Referenz zum Kunden. Die Firma des Kunden wird im *ContactInformation* TM beschrieben. Der Aufbau des *ContactInformation* TM kann unter [@IDTA] eingesehen werden. Das TM *DesignOfProduct* enthält die Versionierung des Designprozesses und speichert die Konstrukteure, Modellversionen und benötigten Dateien des gewünschten Produktes. In dem TM *CommercialProperties* werden alle organisatorischen Fertigungsbedingungen des Kunden beschrieben, wie einzuhaltende Normung, notwendige *Non Disclosure Agreements* (NDA) und Lieferkriterien. Unter den Lieferkriterien können genauere Angaben zum gewünschten Lieferdatum, das CO<sub>2</sub>-Limit zur Produktion des Bauteils, die Stückzahl, gewünschte Auslieferung der Losgrößen oder auch ein gewünschter Preisbereich für das Produkt vorgeben werden. Alle relevanten Informationen der Produkt-VWS sind für die Weitergabe an die Fertigungsplattform enthalten. Über die I4.0-Sprache kann ein „*call for proposal*“ [VDI/VDE 2193-2] an die Fertigungsplattform gesendet werden (Schritt 3.1 in Abbildung 3.6).

In Schritt 3.2 (Abbildung 3.6) analysiert die Fertigungsplattform die übermittelte Produkt-VWS, das vorhandene CAD-Teil sowie evtl. generierte Technische Zeichnungen. Hierzu kann eine automatisierte Featureextraktion eingesetzt werden.

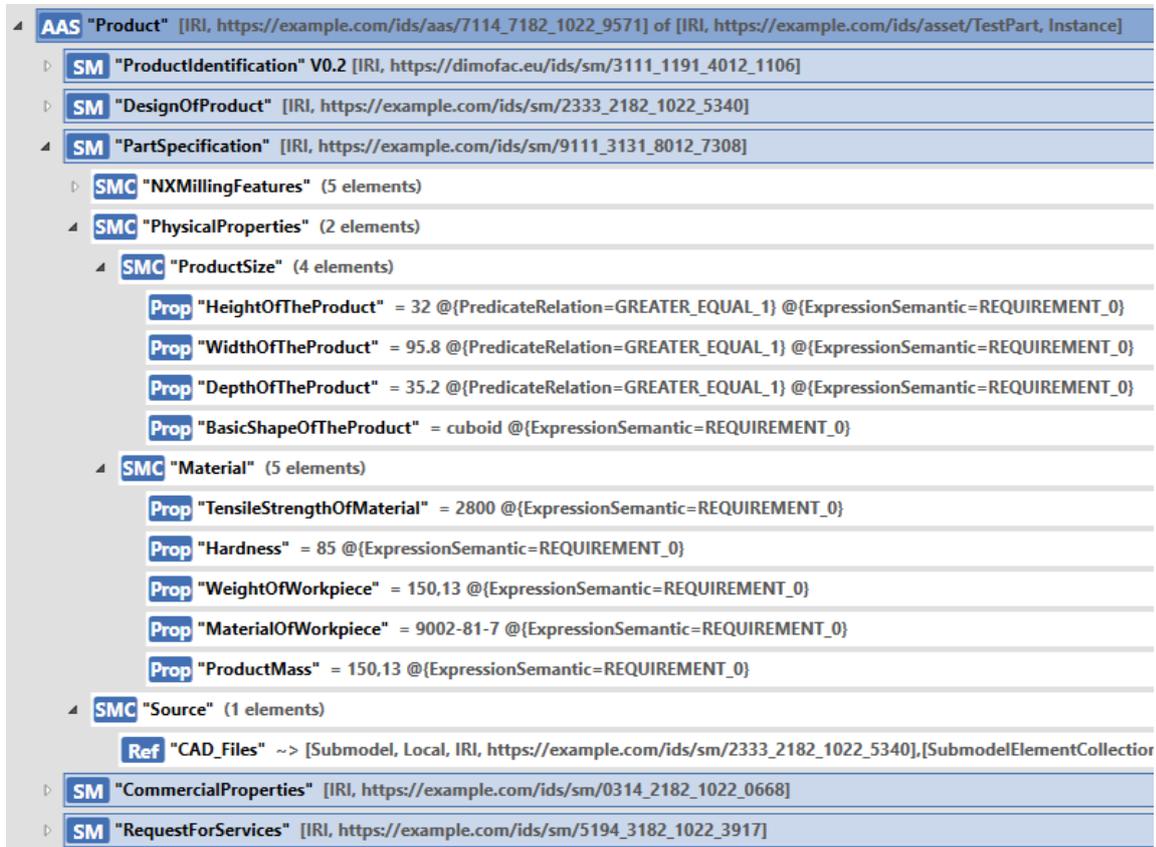


Abbildung 3.13: Übersicht der Produkt-VWS nach Identifikation der prozessrelevanten Informationen

Die bauteilspezifischen Informationen werden im TM *PartSpecification* gespeichert. Das TM enthält eine Sammlung mehrerer *SubElementCollections* (SMC). Die Quelldatei (CAD-Datei), die zur Generierung des TM *PartSpecification* verwendet wird, wird unter *Source* referenziert. Die Dateien können in der VWS abgespeichert werden. Zur Analyse der Fertigungsanforderungen werden zunächst die Außenmaße des Bauteils sowie die Geometrie aus dem bereitgestellten CAD-Bauteil identifiziert. Diese Daten werden in der SMC *PhysicalProperties* unter *ProductSize* gespeichert. Im nächsten Schritt wird das Material, Gewicht und weitere physikalische angegebene Parameter in der *Material* SMC gesichert. Alle Ergebnisse der Featureextraktion werden in der SMC *Features* gesichert. Zur Extraktion beispielhafter Feature wurde eine parametrische Featureerkennung mit NX händisch durchgeführt. Die 5 Feature (siehe Abbildung 3.12) im Produkt sind 4 Bohrungen und eine Tasche und wurden in die SMC übernommen. Alle Parameter der Features, inklusive der Werte, werden in der SMC eines Features gesichert. Je nach verwendeter Software können unterschiedliche Definitionen von Features existieren. Diese können sich sowohl im Typ, der Bezeichnung und in den beschreibenden Parametern unterscheiden. Um zu hinterlegen, auf welchen Standard sich die abgespeicherten Features beziehen, wurde die *Property FeatureStandard* angelegt. Werden die Features beispielsweise bei einem Kunden direkt

extrahiert oder ist ein Lohnfertiger mit unterschiedlichen Fertigungsplattformen verbunden, besteht die Möglichkeit, dass sich die Beschreibung und Parameter von *Features* unterscheiden.

Innerhalb der *Properties* eines NX-Features können *Qualifier* gesetzt werden. Die Boolesche Variable *QualityAssurance* definiert, ob eine Qualitätskontrolle für einen bestimmten Parameter notwendig ist (siehe Abbildung 3.14). In der Fertigungsplanung kann somit später berücksichtigt werden, ob gewisse Parameter eines Features noch durch eine zusätzliche Qualitätskontrolle geprüft werden müssen.

```
Prop "BottomRoughness" = 6.3 @{QualityAssurance=true} @{Certification=IATF 16949} @{Expre:
```

Abbildung 3.14: *Qualifier QualityAssurance* zur Vorgabe von Parametern, die durch eine zusätzliche Qualitätskontrolle geprüft werden.

Die VWS enthält im TM *PartSpecification* alle relevanten Informationen, um in Schritt 3 nach Abbildung 3.6, geeignete Fertigungstechnologien zu identifizieren. In diesem Beispiel kann aufgrund der rechteckigen Form des Bauteils, dessen Größe, des Materials, den Taschen sowie Bohrungsfeatures darauf geschlossen werden, dass ein Bohrprozess sowie ein Fräsprozess zur Fertigung notwendig ist. Des Weiteren wird die Bereitstellung des benötigten Rohmaterials als Service hinterlegt. Im verteilten Produktionsnetzwerk wird anschließend nach den Dienstleistungen gesucht. Die benötigten Dienstleistungen sind im TM *RequestForServices* hinterlegt. Die gesuchten Dienstleistungen können z. B. durch DIN 8580 [DIN 8580] bzw. die spezifischeren Normen DIN 8589-2 [DIN 8589-2] und DIN 8589-3 [DIN 8589-3] definiert werden. So werden die benötigten Technologieketten abgebildet. Die Beschreibungsmöglichkeiten von Dienstleistungen werden in Kapitel 3.3.1 diskutiert.

In der Produkt-VWS sind alle relevanten Informationen zur Fertigung inklusive einer möglichen Technologiekette gesammelt. In Schritt 3.3 identifiziert die Fertigungsplattform die benötigten Dienstleistungen. Ein Lohnfertiger, der die angeforderten Dienstleistungen anbietet, kann gematcht werden und eine Angebotsanfrage (Schritt 4.1) erhalten. Hierzu wird über den Gaia-X-Konnektor die Produkt-VWS freigegeben oder übermittelt. Der Lohnfertiger muss nun analysieren, ob und zu welchen Bedingungen das Produkt im vorhandenen Maschinenpark umgesetzt werden kann.

### 3.3 Beschreibung von Dienstleistung, Capabilities und Skills

Die Informationen der Produkt-VWS werden verwendet, um mögliche Lohnfertiger zu identifizieren. Für das Matching eines Produktfeatures bis zur Identifikation eines möglichen Fertigungsprozesses, müssen die in Abbildung 3.15 dargestellten Schritte durchgeführt werden. In dem Matchingprozess werden anhand der Produktbeschreibung benötigte Dienstleistungen erfasst und ein entsprechendes Unternehmen ausgewählt. Erhält ein Unternehmen die Produkt-VWS, werden die enthaltenen Features möglichen Capabilities zugeordnet. Die Capabilities re-

ferenzieren auf Skills, die diese fertigen können. Ob ein Skill ein Feature mit den gewünschten Qualitätsanforderungen fertigen kann, bleibt bei individuellen Produkthanfragen zu prüfen.

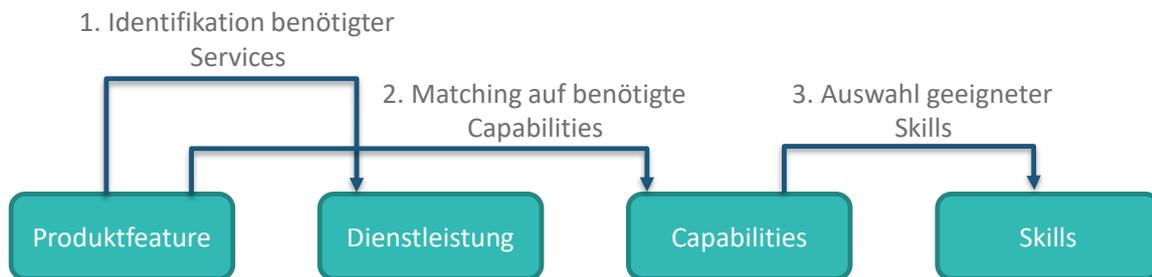


Abbildung 3.15: Benötigte Schritte zum Matching eines geometrischen Features auf einen Skill

### 3.3.1 Beschreibung von Dienstleistungen

Jedes im Produktionsnetzwerk vorhandene Unternehmen wird durch eine Unternehmens-VWS repräsentiert. In der Unternehmens-VWS werden die Dienstleistungen und mögliche Kriterien für die Auswahl eines Unternehmens gelistet. In diesem Fall handelt es sich um Lohnfertiger im Bereich der spanenden Bearbeitung. Die Dienstleistungen, die ein Lohnfertiger anbietet, können sehr abstrakt oder auch sehr spezifisch beschrieben werden. Eine abstraktere Beschreibung einer Dienstleistung wäre z. B. die Beschreibung einer Fertigungstechnologie, nach DIN 8580 [DIN 8580]: *Trennen → Spanen mit geometrisch bestimmter Schneide → Fräsen*. Des Weiteren bietet ECLASS in der Gruppe 25 eine Möglichkeit Dienstleistungen zu beschreiben: *Allgemeine Dienstleistungen → 25-09 Lohnfertigung → 25-09-07 Zerspanung (Lohnfertigung)* [ECLASS]. Noch spezifischer wäre Fertigungs-Feature wie das Fräsen einer Tasche oder das Fertigen eines spezifischen bereits definierten Produktes anzubieten (siehe Abbildung 3.16). Je spezifischer die Beschreibung einer Dienstleistung, desto kleiner wird der Aufwand beim Lohnfertiger, in der nachfolgenden Prozessplanung einen geeigneten Prozess zu identifizieren.

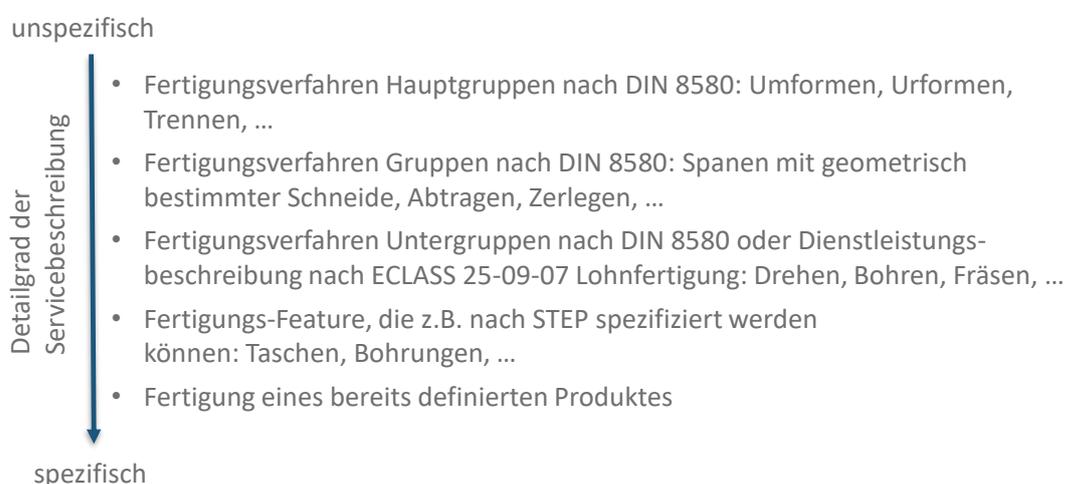


Abbildung 3.16: Verschiedene Detailgrade zur Beschreibung einer Dienstleistung

Die Unternehmens-VWS werden an einem zentralen Punkt z. B. Datenbank, *Repository* oder *Federated Catalogue* gesammelt, sodass die Fertigungsplattform Zugriff auf die Informationen hat. Je nach Anfrage sucht die Fertigungsplattform nach den benötigten Dienstleistungen und identifiziert potenzielle Fertiger. Um eine gewisse Vorfilterung von Unternehmen zu ermöglichen, müssen die Dienstleistungen mehr Informationen als z. B. nur die Fertigungstechnologie enthalten. Bei der Sammlung der Angaben, die in dem TM für eine Dienstleistungsbeschreibung enthalten sein müssen, wurde sich an der PAS 1018 (*Publicly Available Specification*) orientiert [PAS 1018]. Obwohl die PAS 1018 bereits zurückgezogen wurde, bietet diese eine gute Übersicht, um Anforderungen und Struktur von Servicebeschreibungen zu erstellen. Die Informationen wurden in die gelisteten Grundstruktur zur Beschreibung von Dienstleistungen in PAS 1018 eingeordnet. In Abbildung 3.17 wurde hierfür ein VWS-TM für Dienstleistungen angelegt und die Strukturierung der PAS 1018 in das TM übernommen. Die PAS 1018 ist wie folgt strukturiert:

- Informationen über Fertigungstechnologien sowie Fertigungs-Features, die umgesetzt werden können (z. B. orientiert an DIN 8580 [DIN 8580], ECLASS oder bearbeitbaren Materialien). Nach [PAS 1018] handelt es sich um die Klassifikation einer Dienstleistung. In Abbildung 3.17 wurde entsprechend die *Property ServiceOfferingDescription* angelegt. Die weitere Spezifizierung von möglichen Materialien und Größen wird in *GeneralConditions* hinterlegt.
- Angabe zu Stückzahlen und Bauteilgrößen, die ein Auftrag ca. umfassen darf, können den quantitativen Angaben einer Leistung nach [PAS 1018] zugeordnet werden. In der Service-VWS wurde die SMC *Quantity* vorgesehen.
- Auf den Ort der Durchführung einer Dienstleistung [PAS 1018] wird in der VWS durch die *Reference LocationOfProvider* verwiesen.
- Benötigte Informationen, bspw. eine CAD-Datei oder Technische Zeichnung, gehören nach [PAS 1018] zu den Rahmenbedingungen einer Dienstleistung. Auch die Optimierung des Prozesses nach ökologischen, ökonomischen oder zeitlichen Faktoren, kann unter Rahmenbedingungen fallen. Für die Beschreibung von Daten, die zur Ausführung einer Dienstleistung notwendig sind, ist die SMC *InputFactors* angelegt.
- Verweis auf mögliche Normen oder Gewährleistungen, die eingehalten werden müssen, zählen nach [PAS 1018] unter sonstige Vereinbarungen und können in dem SMC *OtherAgreements* spezifiziert werden.
- Qualitative Angaben einer Dienstleistung [PAS 1018], können auf eine Norm verweisen, wie z. B. Allgmeintoleranzen. Im TM wurde entsprechend eine SMC *QualitativeData* angelegt.
- Informationen über Zahlungs- und Lieferbedingungen [PAS 1018] sind im VWS-TM unter *TermsOfDeliveryAndPayment* zu finden.

- Vorgabepreis des Auftraggebers [PAS 1018], z. B. eine Preisvorstellung oder ein Schätzpreis, würde bei einer individuellen Fertigungsanfrage in einem entsprechenden Angebot hinterlegt werden.

Im Zuge einer derartigen Plattform kann über Bewertungskriterien nachgedacht werden, die öffentlich einsehbar sind, z. B. Einhaltung von Fristen, die Vollständigkeit von Bestellungen, Nachhaltigkeitsaspekte oder das Einhalten von Lieferkettengesetzen. Zusätzlich existiert eine SMC *CapabilityReferences*. Die SMC verweist auf alle Capabilities eines Unternehmens, die ein Service beinhaltet.

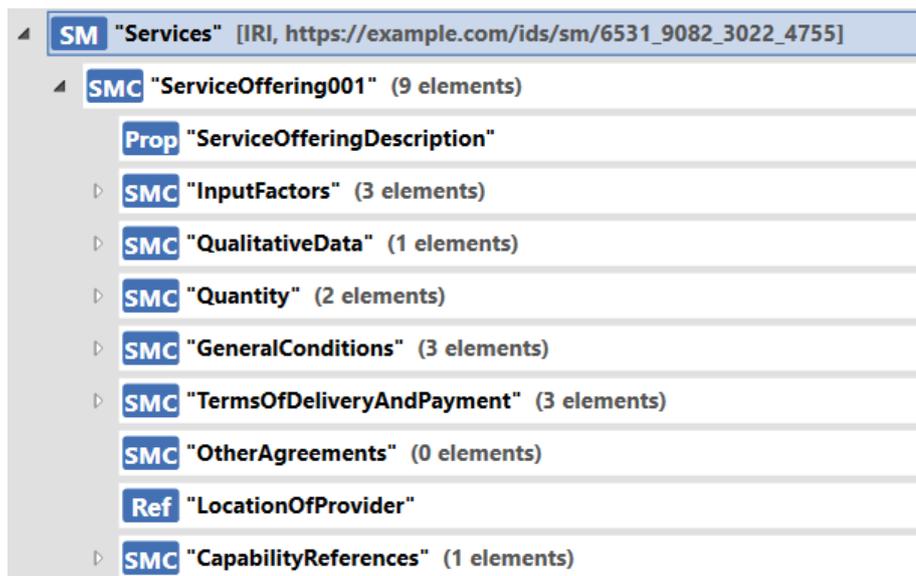


Abbildung 3.17: Darstellung eines VWS-TM zur Beschreibung von Dienstleistungen in einer Unternehmens-VWS

Die Dienstleistungsbeschreibung dient der Beschreibung von geschäftsrelevanten Informationen. So ermöglicht das TM Zahlungsinformationen, Lieferbedingungen und Informationen über mögliche Verträge abzubilden. Auch Vereinbarungen, wie Rückabsprachen vor oder nach der Fertigung eines Bauteils, können spezifiziert werden. Neben den geschäftlichen Informationen dient die Dienstleistungsbeschreibung als ein erster Filter. Nicht jedes Unternehmen im Produktionsnetzwerk wird somit eine Angebotsanfrage erhalten.

### 3.3.2 Beschreibung von Capabilities und Skills

Erhält ein Lohnfertiger eine Angebotsanfrage von der Fertigungsplattform, wird die Produkt-VWS inkl. dem CAD-Modell übertragen. Die enthaltenen Fertigungsinformationen müssen auf die in der Produktionsumgebung verfügbaren Capabilities gematcht werden. Capabilities und Skills eines CPPM müssen in Fertigungsumgebungen eindeutig beschrieben bzw. referenziert werden. Nur so kann sichergestellt werden, dass benötigte CPPM für einen gewünschten Fertigungsschritt anhand der definierten CAD-Features gematcht werden können. Ein Vorteil, den die eindeutige Beschreibung aller Capabilities in einem Unternehmen bietet, ist das Schaffen

eines Bewusstseins über mögliche auszuführende Fertigungsschritte. Für die Definition von Capabilities und Skills können unterschiedliche Beschreibungsformen gewählt werden. Ein Unternehmen hat generell die Möglichkeit eine prozessspezifische oder produktspezifische Beschreibungen von Capabilities oder Skills zu wählen. Für die spanende Bearbeitung können bereits existierende Normen wie z. B. STEP-AP 242, STEP-NC, DIN 8580 oder herstellerspezifische Featuredefinitionen aus CAD-/ CAM-Systemen zur Beschreibung verwendet werden. Des Weiteren könnte die Definition von G-Code Zyklen für Werkzeugmaschinen genutzt werden. G-Code Zyklen kapseln den Code von Fertigungs-Features auf CNC-Maschinen. So ist die Fertigung von bspw. Taschen oder Bohrungen anhand eines Zyklus, der einige Inputparameter erhält, möglich. Ein Unternehmen kann Bezeichnungen für Capabilities und Skills auch eigenständig, anhand von Funktionen und Bezeichnungen kreieren, die im täglichen Arbeitsalltag benötigt werden. Je nach gewählter Vorgehensweise ergeben sich verschiedene Vor- und Nachteile in der Bezeichnung von Capabilities und Skills, die in den folgenden Tabellen aufgelistet sind. Dabei wurde die Möglichkeit betrachtet, den Bezug zu einer Norm herzustellen (Tabelle 3.4), herstellerspezifische Formate zu nutzen (Tabelle 3.5) oder eine eigene Definition zu entwickeln (Tabelle 3.6).

Tabelle 3.4: Vor- und Nachteile in der Verwendung einer Norm zur Definition von Capabilities und Skills

<b>Verwendung von Normungen zur Definition von Capabilities / Skills</b>	
Vorteile	Nachteile
Einführung von herstellerübergreifenden Bezeichnungen für eingesetzte Feature und Bearbeitungsprozesse. Somit besteht die Möglichkeit, mehrere CAD-Systeme an einen allgemeingültigen Angebotsprozess anzubinden.	Die Bezeichnungen unterscheiden sich zum Teil stark von Bezeichnungen, die in der alltäglichen Sprache für Fertigungsschritte eingesetzt werden. Die Bezeichnungen setzen sich in den Normen teilweise aus langen Verkettungen von Featurebezeichnungen zusammen. Diese werden sich im alltäglichen Sprachgebrauch vermutlich nicht durchsetzen.
Eine Softwareanpassung für das spätere Matching von CAD-Features auf Capabilities und Skills ist nicht notwendig, da konsistent die identische Bezeichnung weiterverwendet wird.	In der Recherche wurde kein etabliertes System zur Featureextraktion gefunden, das sich an einer herstellerübergreifenden Norm wie STEP-AP 242 orientiert. Somit existiert ein offener Entwicklungsbedarf.
Es entsteht kein Zeitaufwand durch die Entwicklung eigener Definitionen.	

Tabelle 3.5: Vor- und Nachteile in der Verwendung herstellerspezifischer Bezeichnungen zur Definition von Capabilities und Skills

<b>Herstellerspezifische Bezeichnungen auf Basis von Features / G-Code Zyklen zur Definition von Capabilities / Skills</b>	
Vorteile	Nachteile
Etablierte CAD-/ CAM-Systeme mit zuverlässigen Schnittstellen existieren bereits, um Features aus einem CAD-Modell zu extrahieren. Capabilities und Skills können anhand der bestehenden Features entwickelt und auf diese zugeschnitten werden.	Es wird sich an einen Herstellerstandard gebunden. Parallel könnten sich somit verschiedene Beschreibungsarten für ähnliche oder gleiche Features durchsetzen.
Verwendung von G-Code Zyklen zur Beschreibung ermöglichen ein einfaches Mapping von Skill-Parametern auf den auszuführenden Code in der Steuerung. Der Vorteil ergibt sich allerdings nur für einen Steuerungshersteller, da G-Code Zyklen nicht herstellerübergreifend standardisiert sind.	Bei der Verwendung von G-Code Zyklen für die Beschreibung von Skills und Fähigkeiten wird ein Mapping der Feature-Parameter auf die Capabilities oder Skills notwendig.

Tabelle 3.6: Vor- und Nachteile in der Verwendung eigener Begrifflichkeiten zur Definition von Capabilities und Skills

<b>Verwendung eigener Begrifflichkeiten zur Definition von Capabilities / Skills</b>	
Vorteile	Nachteile
Im Arbeitsalltag verwendete Bezeichnungen können für Fertigungsschritte genutzt werden. Eventuell existiert bereits ein generischer Code für bestimmte Fertigungsschritte, der verwendet werden kann.	Hoher Zeitaufwand zur Erstellung von Definitionen.
Alle für ein Unternehmen relevanten Fertigungsoperationen können abgedeckt werden. Vor allem auch Fertigungsoperationen, die in keinem Standard hinterlegt sind.	Nutzerspezifisches Mapping von Featurebezeichnungen auf die Bezeichnung von Capabilities/ Skills wird notwendig.

Die Capabilities einer Produktionsressource sollten in dessen VWS beschrieben werden. Das TM für Capabilities kann verwendet werden, um auf die implementierten Skills zu referenzieren. Ist eine Capability eindeutig beschrieben, ist über die Referenz auf einen Skill abgesichert, dass diese gefertigt werden kann. Besteht die Möglichkeit ein Feature aus der Produkt-VWS auf eine Capability zu mappen, sind somit auch mögliche Skills zur Fertigung identifizierbar. Eine

Capability kann wiederum eine Komposition mehrerer Capabilities sein. Somit kann es je nach Definition der Fertigungs-Features, Capabilities und Skills vorkommen, dass eine Capability sich aus mehreren Capabilities zusammensetzt. Folgendes Beispiel ist in Abbildung 3.18 dargestellt, eine Capability Gewindebohrung könnte eine Komposition der Capabilities Anbohren, Bohren und Gewinde schneiden sein. Sowohl die einzelnen Capabilities als auch die Komposition können einer oder mehreren Ressourcen in der Produktionsumgebung zugeordnet werden. Ob es notwendig ist eine Capability in der zusammengesetzten Form zu beschreiben, ist je nach Produktionsumgebung und Auftragsform zu entscheiden.

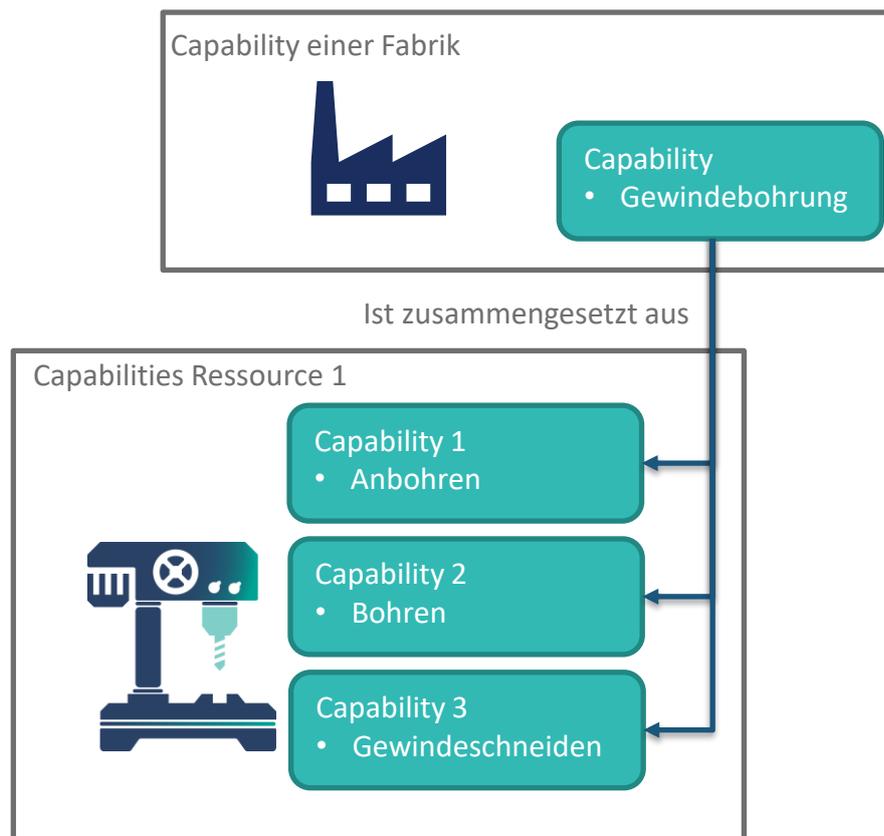


Abbildung 3.18: Komposition von Capabilities

In Abbildung 3.19 ist dargestellt, dass eine Komposition einer Capability durch einen einzelnen Skill oder durch mehrere Skills ausgeführt werden kann. Eine Capability kann somit auf die nacheinander auszuführenden Skills Anbohren, Bohren und Gewindeschneiden verweisen oder auf eine Maschine, die einen einzelnen Skill Gewindebohren anbietet. Nicht nur eine Capability könnte sich aus einer Komposition mehrerer Capabilities ergeben, sondern auch ein Skill eine Komposition mehrerer Skills sein (reflexive Assoziation). Somit ergibt sich mit der Definition von Capabilities die Möglichkeit, dass mehrere Ressourcen zur Umsetzung einer einzelnen Capability in Frage kommen. Es unter Umständen auch eine einzelne Maschine geben kann, die die Capability mit einem einzelnen Skill oder durch mehrere Skills in Folge realisiert. In der Modellierung eines VWS-TM muss somit beachtet werden, dass eine Capability mehr als nur einen Skill referenzieren kann.

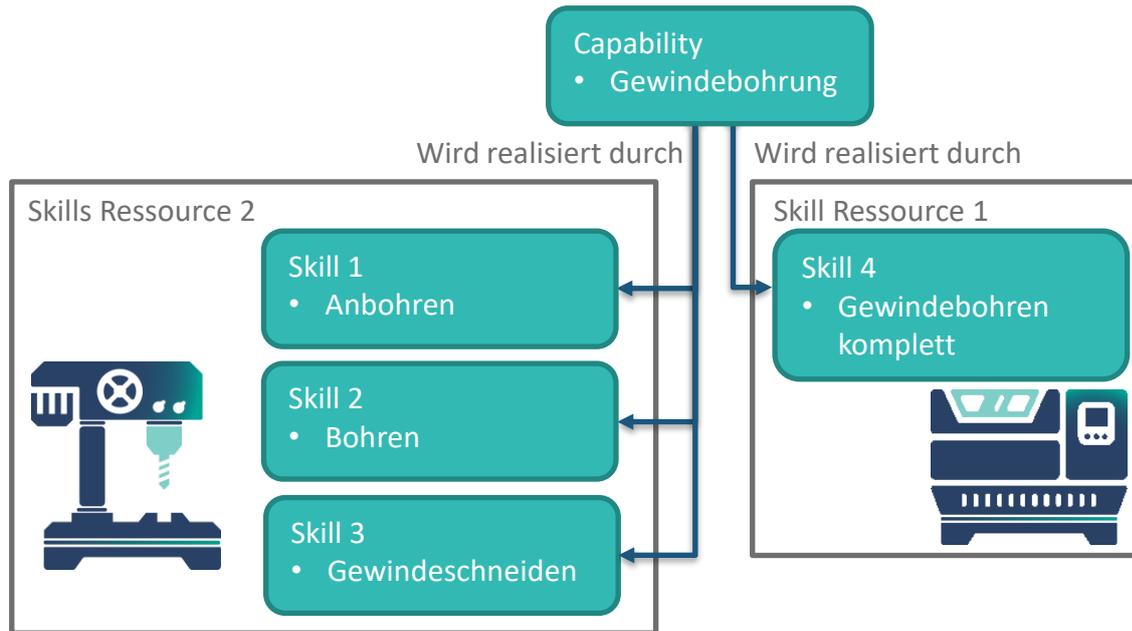


Abbildung 3.19: Die Komposition einer Capability kann durch einen oder mehrere nacheinander ausgeführte Skills realisiert werden

Im Folgenden wird die prozessspezifische oder die produktspezifische Beschreibung einer Capability genauer betrachtet. Zur produktspezifischen Beschreibung von Capabilities können Features eingesetzt werden, wie z. B. STEP- oder NX-Feature (Kapitel 3.2.1 / 3.2.2). Köcher et al. schlagen zur Beschreibung von Capabilities Normen wie die DIN 8580 für Fertigungsverfahren, VDI 2860 für Montage- und Handhabungstechnik (seit 2016 zurückgezogen) oder die VDI/VDE 3682 für eine formalisierte Prozessbeschreibung vor [Kö20]. Sarkar verwendet unter anderem produktspezifische, als auch prozessspezifische Beschreibungen für Capabilities von Fertigungsmaschinen (*SurfaceFinishCapability*) [Sa20]. Aufgrund des Fokus auf die spanende Bearbeitung wird zunächst die DIN 8580, als prozessspezifische Beschreibung von Capabilities, genauer betrachtet. DIN 8580 verweist für Fräsprozesse auf DIN 8589-3 und für Bohren, Senken und Reiben auf die DIN 8589-2. In Abbildung 3.20 sind die Normen bereits für die mögliche Beschreibung von Services analysiert worden. DIN 8589-3 beschreibt die Fertigungsprozesse des FräSENS spezifischer und wird genauer betrachtet. Ein Auszug von Fräsprozessen mit Ordnungsnummer (ON) aus DIN 8589-3:2003 ist in Abbildung 3.20 aufgelistet. Die Fräsverfahren werden in die übergeordneten Verfahren wie PlanfräSEN, RundfräSEN, ProfilfräSEN oder FormfräSEN untergliedert. Die Gruppen werden teilweise noch weiter untergliedert, so enthält FormfräSEN die weiteren Kategorien FreiformfräSEN, NachformfräSEN, Kinematisch-FormfräSEN und NC-FormfräSEN.

Das Beschreiben von Capabilities durch Prozesse nach DIN 8589-3:2003 bringt den großen Nachteil mit sich, dass ein eindeutiges Zuordnen eines Features zu einer Capability nicht möglich ist. Der Lösungsraum würde sich bei einer prozessspezifischen Beschreibung stark vergrößern. Grund dafür ist die allgemeingültigere Beschreibung von Prozessen im Gegensatz zu Fertigungs-Features. Somit kann ein Fertigungsprozess nicht eindeutig einem Feature zugeordnet werden,

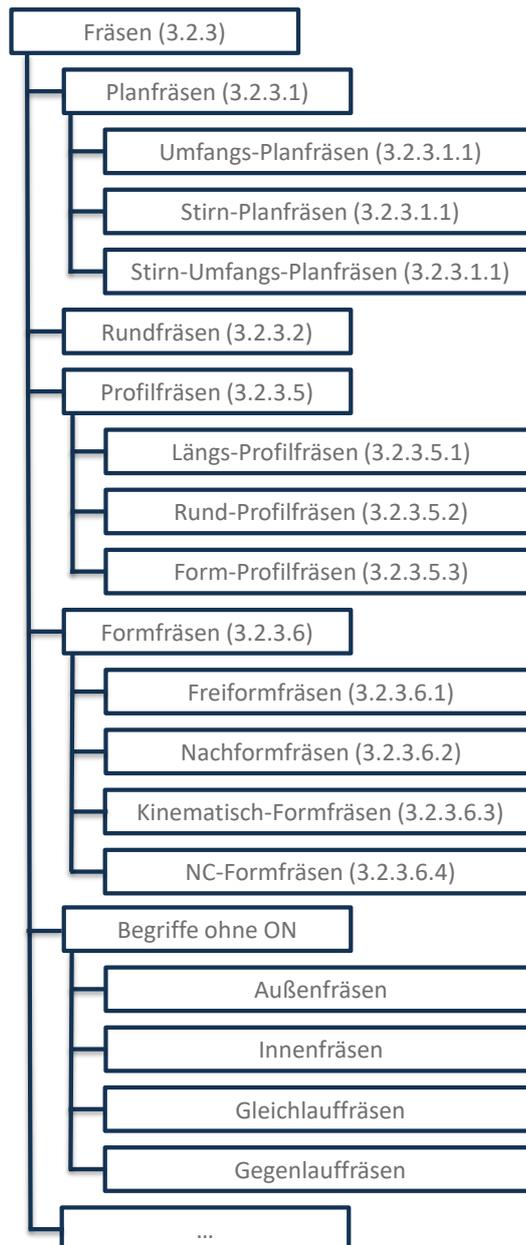


Abbildung 3.20: Auszug von Fräsprozessen in Anlehnung an [DIN 8589-3]

sondern in der Regel immer mehreren Features. Eine Rechtecktasche kann demnach bspw. durch Formfräsen oder Innenfräsen gefertigt werden:

- Formfräsen nach [DIN 8589-3]: „Fräsen mit gesteuerter Vorschubbewegung.“
- Innenfräsen nach [DIN 8589-3]: „Fräsen von am Werkstück innen liegenden Flächen.“

Der Lösungsraum, um hieraus eine entsprechende Capability zu identifizieren, wird somit kaum eingeschränkt. Auch die Unterkategorien des Formfräsens spezifizieren nur verschiedene Technologiearten und verbessern das Problem nicht. Nach DIN 8589-3:2003 wird z. B. definiert, dass beim Freiformfräsen die „Vorschubbewegung von Hand frei“ gesteuert wird, beim kinematischen Formfräsen „die Vorschubbewegung durch ein mechanisches Getriebe kinematisch gesteuert wird“ und beim NC-Formfräsen „die Vorschubbewegung durch eingegebene Daten

gesteuert wird“ [DIN 8589-3]. Prinzipiell kann eine Rechtecktasche sowie viele andere Form-Features mit jedem dieser Fertigungsprozesse gefertigt werden. Gleiches gilt für das Innenfräsen, hier kann jedes Form-Feature betrachtet werden, das im Inneren des Werkstücks liegt. STEP-NC beschreibt in [ISO 14649-11] durch *machining\_operations* das Fräsen, ebenfalls sehr allgemeingültig. Ein Überblick der Fräsoperationen aus STEP-NC sind in Abbildung 3.21 dargestellt. Auch hier ist ein mehrdeutiges Matching von Features auf die Beschreibungen der Fräsoperationen möglich, daher werden diese für eine Definition von Capabilities ebenfalls nicht verwendet. Allerdings referenziert eine *machining\_operation* in STEP über den *workingstep* auf *manufacturing\_feature*. Diese wiederum referenzieren auf die in Kapitel 3.2.1 definierten *machining\_feature*.

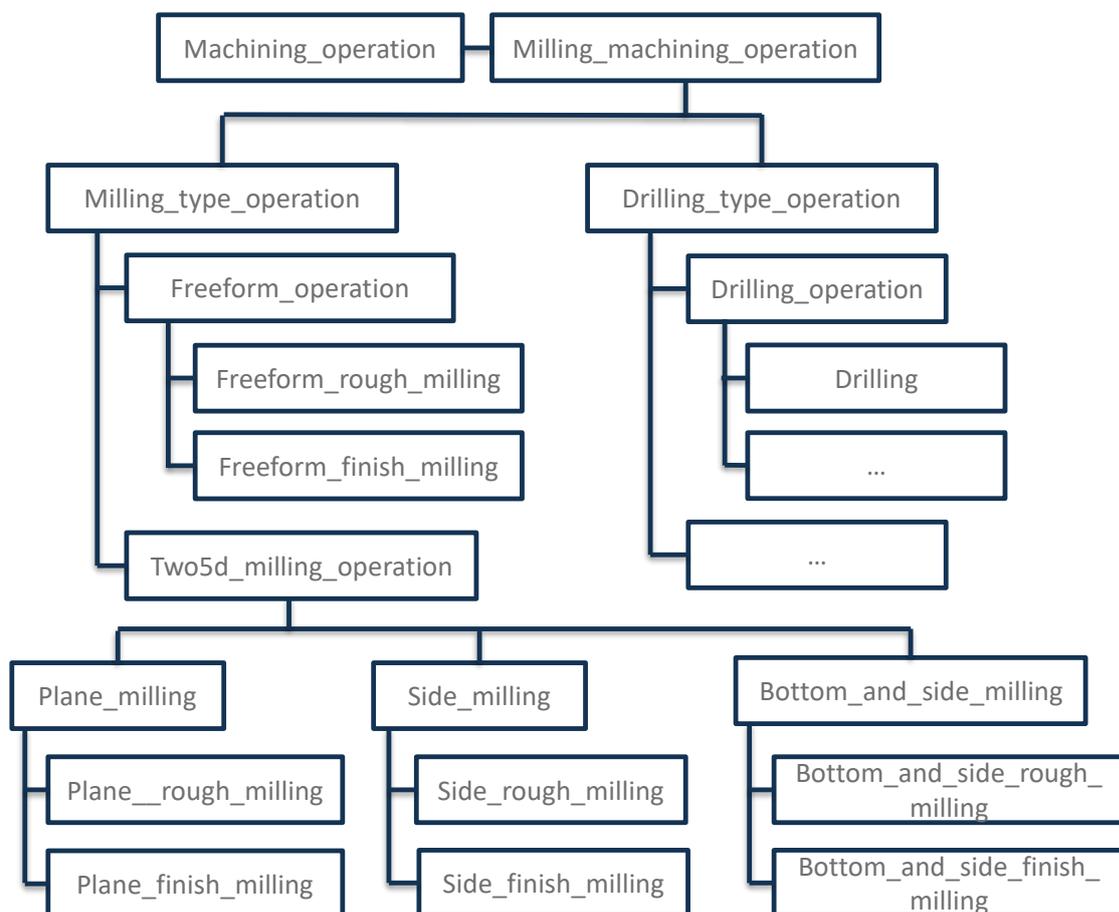


Abbildung 3.21: Übersicht der *Milling machining operations* mit Fokus auf *Milling type operations* nach STEP-NC [ISO 10303-238; ISO 14649-11; ISO 14649-12]

Um den Aufwand in einem Zuordnungsprozess von einem Form-Feature zu einem Fertigungsschritt zu reduzieren, ist die Variante, Capabilities auf Basis von Form-Features zu beschreiben und hieraus Fertigungs-Feature abzuleiten, für die Fräsbearbeitung effektiver. Ist eine Capability als Fertigungs-Feature beschrieben, wird eine Zuordnung der Form-Features eines Produktes auf die vorhandenen Capabilities möglich. Somit würde sich der Lösungsraum gegenüber einer prozessspezifischen Beschreibung im Matching stark einschränken (Abbildung 3.22).

Capability produktspezifische Beschreibung vs. Capability prozessspezifische Beschreibung

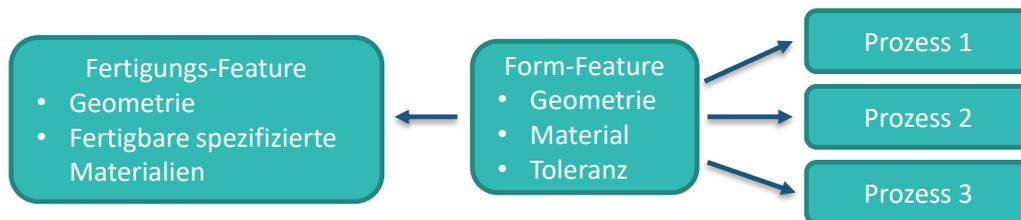


Abbildung 3.22: Komplexität des Matchings bei produkt- bzw. prozessspezifischer Beschreibung von Capabilities

Das Capability-TM kann in der VWS eines CPPM hinterlegt und zusätzlich in einer zentralen Datenbank gesichert werden, wie in Abbildung 3.5 dargestellt. Eine Sammlung aller Capabilities einer Produktionsumgebung ist sinnvoll, da Capabilities generell auch kapselungsfähig sind. Das modellierte Capability-TM ist in Abbildung 3.23 dargestellt.

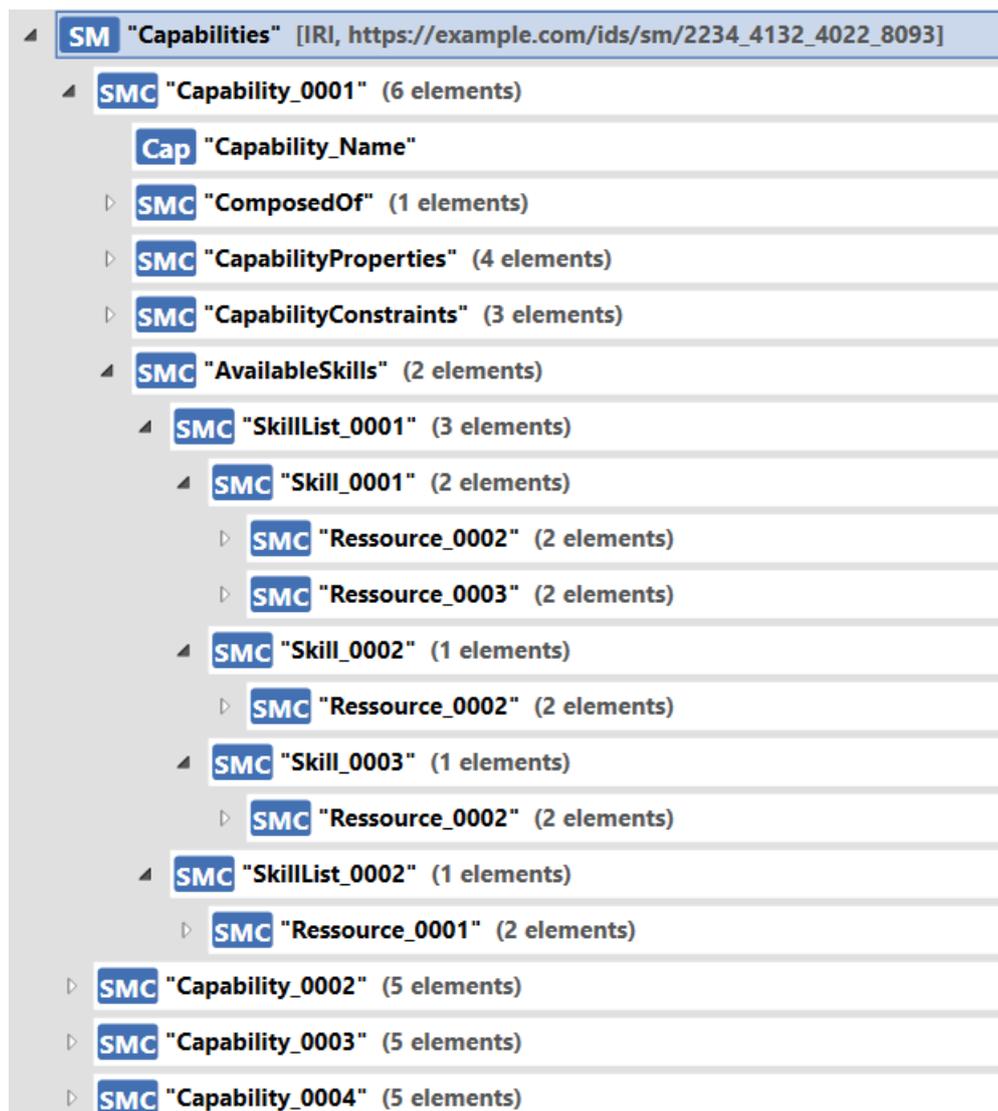


Abbildung 3.23: VWS-TM zur Beschreibung von Capabilities

Die hierarchische Strukturierung von Capabilities wird durch *RelationElements* mit dem Namen *ComposedOf* dargestellt. Somit würden bspw. zwei Capabilities von unterschiedlichen CPPM eine gemeinsame Capability ergeben oder granular definierte Capabilities zu einer gemeinsamen zusammengefasst werden. Capabilities benötigen des Weiteren *Properties*, die eine Capability genauer spezifizieren. Z. B. können in den *Properties* fertigbare Materialien beschrieben werden. *CapabilityConstraints* beschreiben notwendige *Pre-*, *Postconditions* und *Invariants*, dies sind Zustände, die vor, nach oder während der Ausführung der Capability erfüllt sein müssen. Die *SubmodelElementCollection* mit dem Namen *RealizedBy* stellt eine Relation zu Skills her, die für die Ausführung der beschriebenen Capability genutzt werden müssen. Eventuell müssen mehrere Skills nacheinander ausgeführt werden, um eine Capability umzusetzen. Eine genauere Modellierung des TM folgt in Kapitel 4.2 und zeigt die Verbindung zwischen dem Capability- und Skill-TM im Detail.

### 3.3.3 Matching von Capabilities und Features

Es ist sehr aufwendig, einen einheitlichen Bezeichnungsstandard für Features, Capabilities und Skills zu schaffen, den alle Unternehmen befolgen. So ist es z. B. beim G-Code geschehen, dass dieser zwar eine standardisierte Beschreibung von Befehlen für die Maschinensteuerung abdeckt, die Hersteller diesen allerdings mit proprietären Speziallösungen erweitert haben. So unterscheidet sich dieser von Steuerungshersteller zu Steuerungshersteller im Detail. Aufgrund etlicher unterschiedlicher CAD-Formate und damit einhergehender unterschiedlicher Definitionen von Features gibt es generell zwei Problematiken im Matching von Features auf Capabilities und Skills:

#### 1. Vielzahl an unterschiedlichen Featuredefinitionen

Es existiert eine Vielzahl von unterschiedlich definierten Features in der Featureidentifikation: Es gibt zwar mit dem STEP-AP 242 einen Standard, der einheitliche *machining\_Feature* definiert, allerdings wurde in der Recherche kein System zur Featureidentifikation von STEP-AP 242-Bauteilen identifiziert. Die in CAD-Systeme integrierten Featureerkennungen verwenden in der Regel eigens definierte Features. Dies wurde am Beispiel Siemens NX in Kapitel 3.2.2 gezeigt. Beim Export einer CAD-Konstruktion in einen bestehenden Standard wie STEP-AP 242 oder beim Import einer STEP-AP 242 Datei ist mit Informationsverlusten des digitalen Modells zu rechnen. Eine Featureextraktion sollte daher idealerweise mit den nativen Formaten der CAD-Hersteller umgesetzt werden. So können mögliche Informationsverluste im Export der Dateien vermieden werden. Die am Markt existierende Vielzahl unterschiedlicher Feature-Spezifikationen führt zu der Problematik, dass ähnliche Feature auf unterschiedliche Arten beschrieben und definiert werden. Plattformen wie Spanflug oder up2parts (Kapitel 2.5) unterstützen bereits die Analyse und somit vermutlich auch die Featureextraktion aus unterschiedlichen Dateiformaten. Der Kunde kann aus Formaten, wie z. B. Technische Zeichnungen

als PDF, 3D-Modellen in STEP oder verschiedenen herstellerspezifischen Formaten wählen. Durch welche technischen Mittel Feature dabei erkannt und in welche Kategorien Feature von den Herstellern eingeordnet werden, ist nicht ersichtlich und geschütztes Firmenwissen. Es ist davon auszugehen, dass zwischen den Plattformen kein bekannter einheitlicher Standard zur Abspeicherung von Featureinformationen existiert. Somit werden die Produktinformationen in Abhängigkeit des Exports durch unterschiedliche Spezifikationen beschrieben. Es ist auch nicht ersichtlich, ob die Plattformen fertigungsrelevante Parameter extrahieren, wie z. B. die genauen Positionen eines Features.

## 2. Fehlende Standards für Capabilities und Skills

In der Ausarbeitung wurden verschiedene Möglichkeiten diskutiert, Capabilities bzw. Skills zu beschreiben. Bisher sind allerdings keine Bemühungen erkennbar, dass ein möglicher Standard für Skills oder Capabilities in der spannenden Bearbeitung entsteht. In Kapitel 3.3.2 wurde diskutiert, dass es sich anbietet, im Bereich der Fräsbearbeitung bereits existierende *machining\_feature* aus STEP-AP 242 oder G-Code-Zyklen zu nutzen.

Durch die fehlende Integration existierender herstellerübergreifender Standards im Bereich der Featureidentifikation sowie fehlende Standardisierung im Bereich der Capabilities und Skills entsteht in Zukunft eine Vielzahl von Matching-Möglichkeiten. Um eine grobe Prozessplanung zu ermöglichen, werden die Features möglichen Capabilities zugewiesen. Eine Capability verweist wiederum auf mögliche Skills, die das Feature realisieren kann. Es entstehen mehrere mögliche Produktionspfade, die auf Durchführbarkeit geprüft werden müssen. Hinzu kommt die Problematik möglicher unterschiedlicher Benennungen. Abbildung 3.22 zeigt ein einfaches Beispiel für die m:n Beziehung an Matching-Möglichkeiten, die durch die Verwendung von Standards in der Benennung von Capabilities, Skills und Features entstehen. In Abbildung 3.24 zeigen die türkisfarbenen Kästchen die Verwendung eines einheitlichen Standards. Die Dienstleistungsbeschreibung wird an dieser Stelle vernachlässigt, da diese zunächst benötigt wird, damit ein Unternehmen mit den voraussichtlich benötigten Capabilities identifiziert werden kann. Bei einem komplett herstellerübergreifenden geltenden Standard, sowohl für Features, Capabilities als auch Skills, ist das Zuordnen des zugehörigen Objektes wenig komplex. Jedem Feature werden in diesem Fall eine Capability und definierte Skills zugeordnet (siehe Abbildung 3.24). Die Freiheiten von Unternehmen in der Software- und Systementwicklung werden allerdings stark eingeschränkt. Wird sich auf einen herstellerübergreifenden Standard für Features oder Skills geeinigt, entstehen eine Vielzahl von Kombinationsmöglichkeiten, relevante Produktionsschritte zu identifizieren. Hersteller können sich bei dieser Lösung an einem existierenden herstellerübergreifenden Standard orientieren. Allerdings wird weder ein einzelner Standard zur Produktbeschreibung, noch ein einzelner Standard für Skills als realistisch angesehen. Die bereits vielen existierenden unterschiedlichen Featuredefinitionen und Herstellerformate im CAD-Bereich

sprechen gegen eine Einigung auf einen Standard. Die Skills bilden das Know-how der Hersteller ab, da diese schlussendlich die Qualität eines Bauteiles beeinflussen. Auch aufgrund der vielen verschiedenen Steuerungs- und Maschinenhersteller am Markt, ist es als schwierig zu erachten, einen übergreifenden Standard durchzusetzen. Wird die Capability mit häufig benötigten Fertigungs-Features standardisiert, würde eine Verbindung sowohl von Skill als auch von Feature Seite möglich werden. Dies wäre eine sehr effektive Lösung, da sich CAD-Feature und Skills angliedern können. Bisher fehlt es in Produktionsumgebungen an der Beschreibung von Capabilities. Somit wäre dies ein geeigneter Punkt, eine Standardisierung zu beginnen. Nutzt jeder Teilnehmer in einem Produktionsnetzwerk eigene Begriffe für mögliche Features, Skills und Capabilities nimmt die Komplexität im Matching stark zu und sollte vermieden werden.

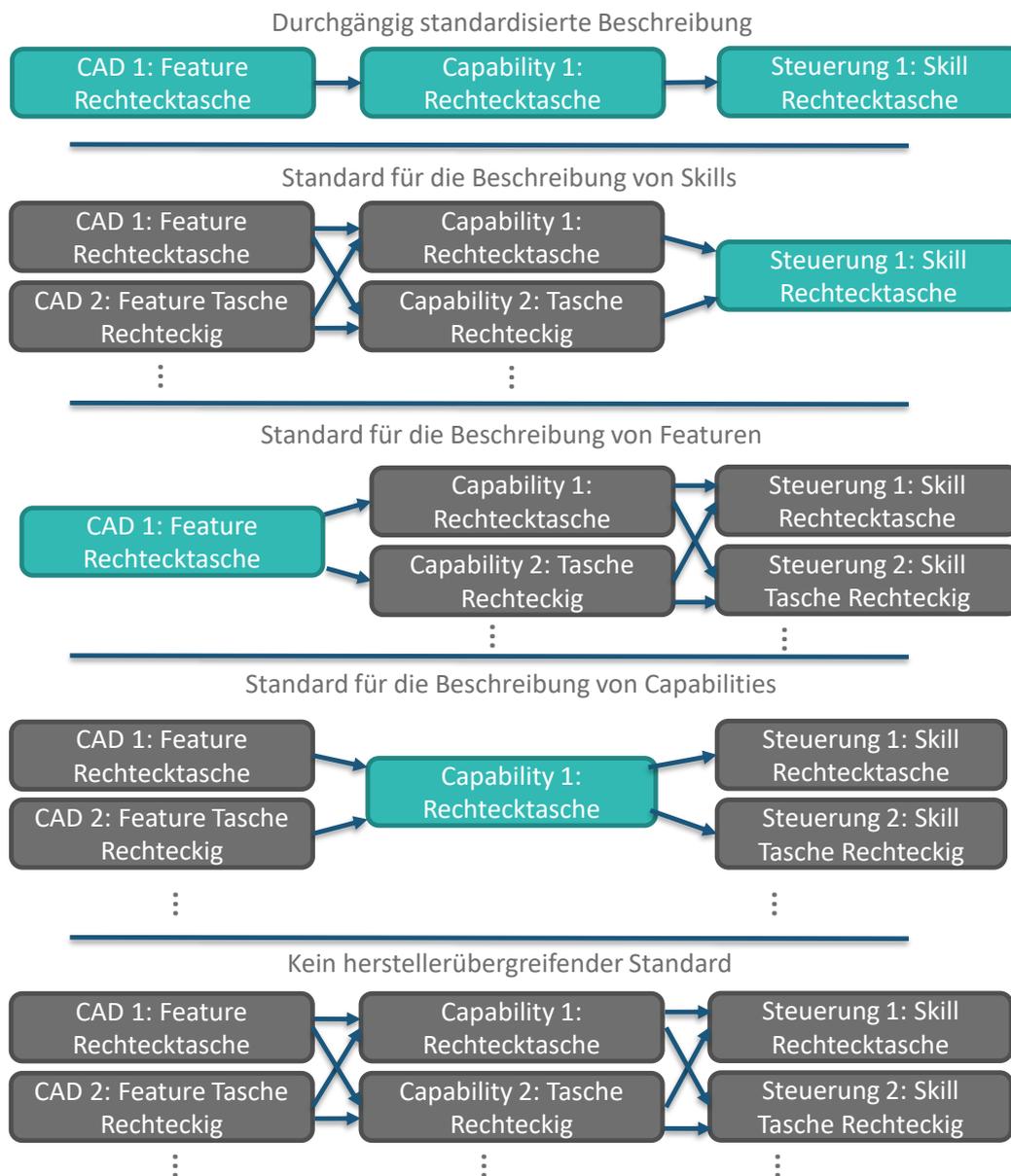


Abbildung 3.24: Darstellung der verschiedenen Matchingoptionen zwischen Features, Capabilities und Skills (Türkise Kästchen stellen einen einheitlichen Standard für alle Unternehmen dar)

Die sinnvollste Lösung ist, die Capabilities zu standardisieren. Dies kann im Rahmen von z. B. ECLASS als Fähigkeit definiert werden. So kann sich zu Beginn an einem Standard wie den *machining\_features* aus ISO 10303 orientiert werden. Die bereits existierenden Featurebeschreibungen können verwendet werden. Die Skills, die Fertigungs-Know-how von Unternehmen abbilden, bleiben somit frei anpassbar. Je nach Bedarf können in Produktionsumgebungen Skills entwickelt und eingesetzt werden. Gleiches gilt für die Featuredefinitionen. Es könnten parallel mehrere Definitionen existieren, die auf eine Capability gematcht werden. Mehrere Forschungsprojekte bieten Ansätze und zeigen Lösungen für die Definition von Features oder das Matching von Features auf Capabilities mit Ontologien.

Die *ManuService* Ontologie definiert eine Produktontologie auf Basis von Fräs-, Dreh- und Bohr-features, um benötigte Dienstleistung im CM identifizieren zu können [ManuService; LWX19]. Die Features sind in Anlehnung an ISO 14649, proprietäre CAM-Feature und fertigungsrelevante Features definiert worden. Alle Features werden mit einem IRI (*Internationalized Resource Identifier*) beschrieben. Dieser kann in der Produkt-VWS hinterlegt werden, um einen eindeutigen Verweis auf ein spezifiziertes Feature zu ermöglichen. Die *ManuService* Ontologie kann für die Umsetzung von Features mit Skills nicht eingesetzt werden. Hierzu müsste diese noch um weitere geometrische Parameter in der Featurebeschreibung, z. B. die eindeutige Position und Ausrichtung eines Features, erweitert werden.

Die *Semantically Integrated Manufacturing Planning Model* (SIMPM) Ontologie identifiziert auf Basis von Produktfeatures, Capabilities und Funktionen mögliche Prozessabläufe [Sa20]. Die Capabilities (*ManufacturingProcessCapabilities*) werden aus Maschinen- und Werkzeugfähigkeiten zusammengesetzt. Mit den *Manufacturing Processes* werden Funktionen der Fertigungsressourcen eingegrenzt. Algorithmen ermöglichen ein Matching eines Prozessablaufes mit Hilfe der Ontologie. Die *ManufacturingProcessCapabilities* sind prozessspezifisch und die Funktionen produktspezifisch beschrieben.

Die *Manufacturing Resource Capability Ontologie* (MaRCO) beinhaltet ein Produkt- sowie ein Capability-Modell [Jä18; Jä19]. Das Capability-Modell greift wiederum auf Ressourceninformationen zurück. Ein Prozess-Taxonomie-Modell importiert die Informationen aus dem Produkt wie auch aus dem Capability-Modell und ermöglicht somit in Zukunft die automatisierte Erstellung eines Fertigungsprozesses. Die Verteilung in verschiedene Modelle soll die Komplexität reduzieren, sodass Ontologien über die Zeit erweitert werden können. Fräsprozesse sind im Prozess-Modell unter maschineller Bearbeitung und Materialentfernung gelistet. Capabilities sind somit prozessorientiert beschrieben. Die Beschreibung des Fräsprozesses orientiert sich an DIN 8580, wird im Detail allerdings nicht weiter aufgesplittet.

Die Literatur zeigt, dass sich Ontologien für den Matching-Prozess eignen sowie die eindeutige Referenzierung von Features und Capabilities mit einem IRI ermöglichen. Die IRI's können wiederum in einer Produkt-VWS hinterlegt werden und einen eindeutigen Verweis auf ein Feature herstellen. Um möglichst variabel Ontologien einbinden zu können, ist der Ansatz aus [Jä18;

Jä19] sinnvoll. So kann ein durch verschiedene Standards modular erweiterbares Matching aufgesetzt werden, siehe (Abbildung 3.25). Der Aufbau einer Ontologie für den Matching-Prozess wird in der Ausarbeitung nicht detaillierter betrachtet.

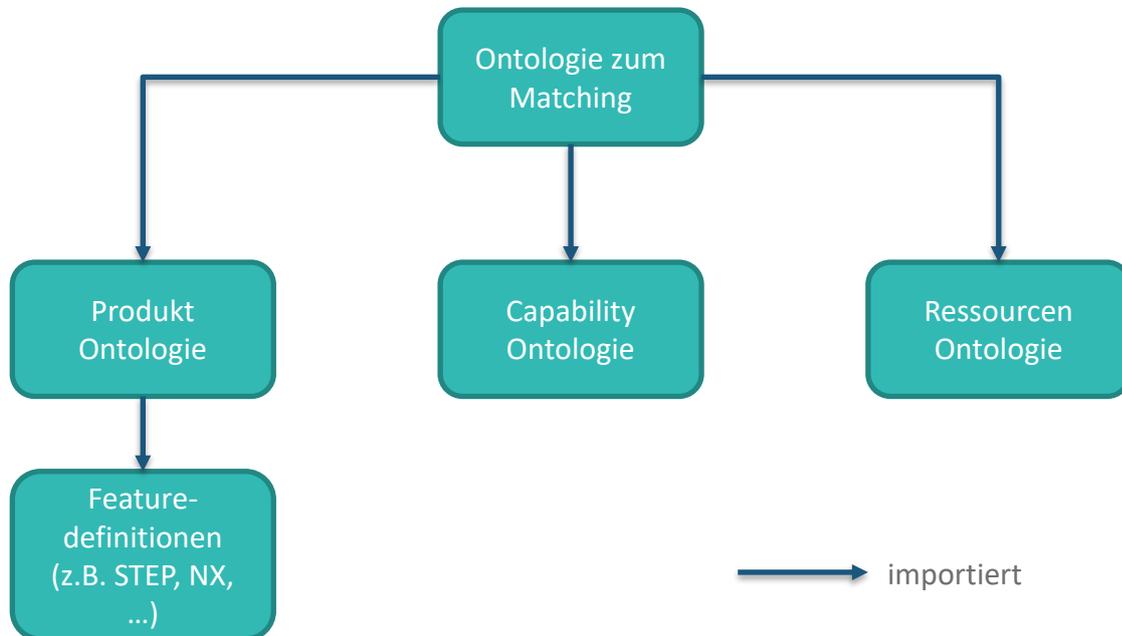


Abbildung 3.25: Möglicher modularer Aufbau einer Ontologie für das Matching in Anlehnung an [Jä18; Jä19]

Als Hindernis in der Einbindung verschiedener Standards in das gesamte Matching ist der Abgleich fertigungsrelevanter geometrischer Parameter zu sehen. Wird eine Capability durch einen Skill umgesetzt, müssen geometrische Parameter wie Position eines Features, Längen, Breiten, Radien usw. eindeutig definiert werden. Die Position einer Tasche könnte z. B. im Zentrum oder in einer der Ecken definiert sein. Die Problematik wird an einem Beispiel gezeigt und ähnliche Feature aus der Norm STEP und Siemens NX verglichen. Hierbei wurde die Zuordnung einzelner Parameter geprüft. Unter der Annahme, dass ein STEP-Feature auf ein NX-Feature gemappt werden muss, wurden die Parameter genauer betrachtet. In der Zuordnung einiger Parameter verbleibt ein gewisser Interpretationsspielraum. Durch die hohe Komplexität der STEP-Norm, teilweise mangelnde Genauigkeit in der Beschreibung (z. B. durch fehlende Konstruktionszeichnungen) oder die Existenz allgemein gehaltener Feature-Typen, ist es als Anwender sehr aufwendig, sich in die Normung einzuarbeiten. Teilweise entstehen im Vergleich der existierenden Feature Mehrdeutigkeiten. Das Problem wird am Beispiel einer Rechtecktasche diskutiert. In Abbildung 3.26 ist die ARM-Entität des *Machining\_Feature* einer *Rectangular\_Closed\_Pocket* aus [ISO 10303-1814] dargestellt. Es wird farblich markiert, ob mögliche NX-Attributsvariablen zu diesen Parametern zugeordnet werden können. Einige der im Feature enthaltenen Parameter sind nicht aufeinander abbildbar. So wird z. B. in STEP ein optionales *base\_radius* Objekt definiert, das den Übergangsradius zwischen der Oberfläche eines Körpers und den Wänden

einer Tasche definiert, während im NX-Feature nur eine Übergangsfase (*TOP\_CHAMFER*) als Parameter vorgesehen ist.

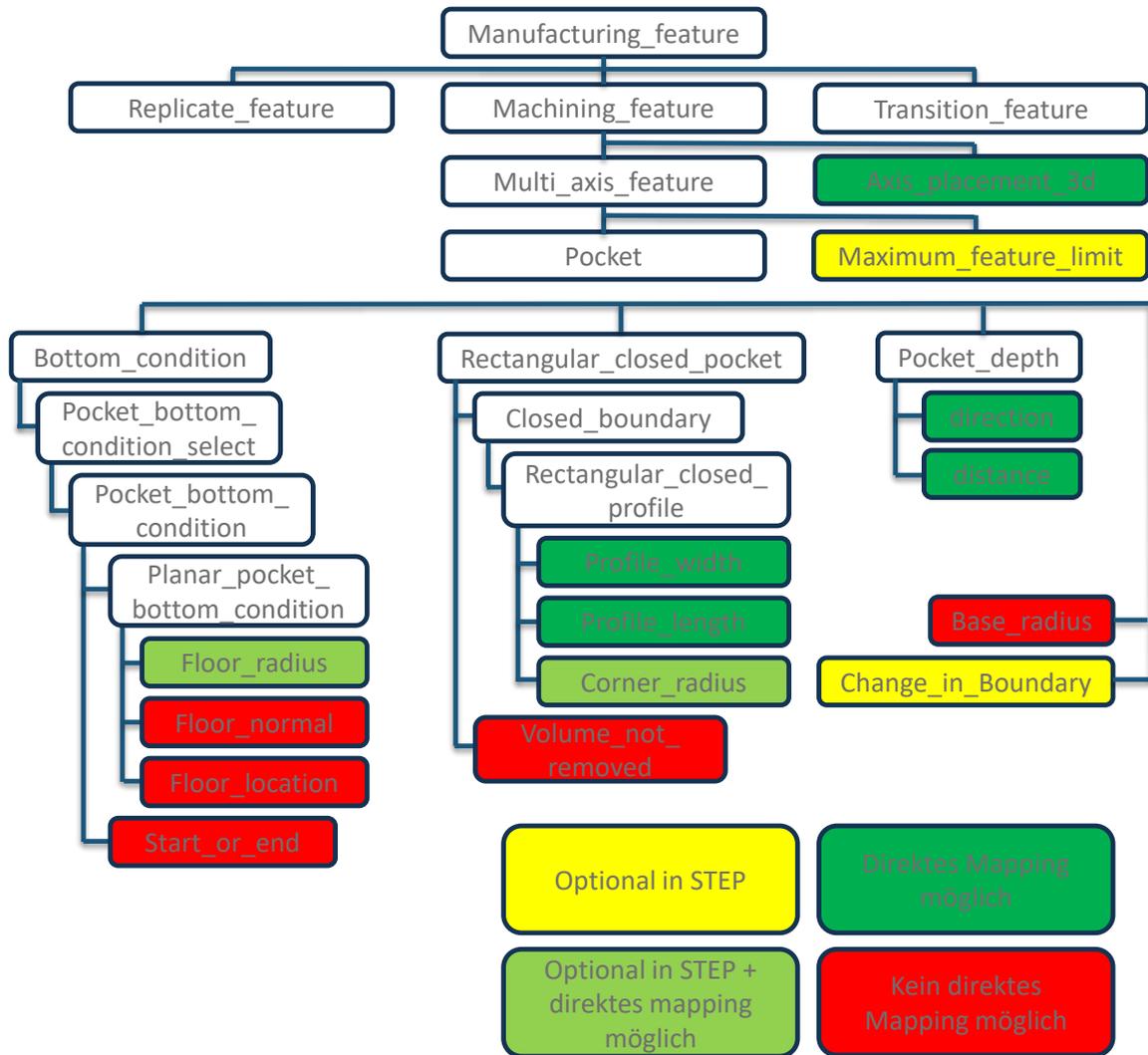


Abbildung 3.26: Mapping der NX-Attributvariablen auf STEP-ARM-Entitäten am Beispiel einer Rechtecktasche

Im Weiteren werden die Attribute des NX-Feature *POCKET\_RECTANGULAR\_STRAIGHT* mit den jeweiligen Attributen einer STEP *Rectangular\_Pocket* gegenübergestellt. Attribute, die gemeinsam in einer Tabellenzeile abgebildet sind, können direkt verglichen werden. Alleinstehende Attribute müssen über Umwege extrahiert oder berechnet werden. Die entsprechende Tabelle 1 ist im Anhang A abgebildet. Ein großer Unterschied in der Feature-Definition zwischen NX und STEP liegt z. B. in der Darstellung von Bohrungen. Während Gewindebohrungen in NX als eigenständiges Feature existieren (*STEP1HOLE\_THREAD*), werden diese in STEP in zwei Features gesplittet (*Round\_hole*, *Thread*) und mit einem *Compound\_Feature* Objekt verknüpft. Als generelle Anforderungen an ein Feature-Mapping verschiedener Definitionen bleibt festzuhalten, dass vorab nicht davon ausgegangen werden kann, dass ein vollständiges Mapping zweier Featuretypen möglich ist. Zunächst müssen zwei ähnelnde Geometrien identifiziert werden und die beiden Feature-Koordinatensysteme aufeinander abgebildet werden. Dies kann durch Transla-

tionen und Rotationen des Ursprungskoordinatensystems und aller weiteren davon abhängigen Parameter geschehen. Anschließend muss für alle weiteren Attribute eines Features ein entsprechendes Attribut im Zielsystem gefunden werden. Teilweise wurde identifiziert, dass dies nur durch weitere Berechnungen (z. B. Durchmesser in Radien umrechnen) möglich ist. Auch kann es sein, dass weitere verknüpfte Feature untersucht werden müssen, um ein vollständiges Mapping zu ermöglichen. Ist diese Option nicht möglich, ist das Mapping unvollständig und die Anpassung einer der Featuredefinitionen notwendig.

Als Fazit bleibt festzuhalten, dass es zum jetzigen Zeitpunkt die sinnvollste Lösung ist, einen gemeinsamen Standard an der Capability Schnittstelle zu erarbeiten. Auf diesen können Hersteller ihre Features mappen und Anlagenproduzenten ihre Skills anbinden. So bleibt die Freiheit auf Seiten der Features und Skills erhalten, ein bestehendes Konzept zu erarbeiten oder zu wählen. Der Matching-Aufwand wird stark reduziert, wenn eine herstellerübergreifende Ontologie zwischen Features und Capabilities entsteht. Mit Ontologien können auch komplett standard-unabhängige Lösungen entwickelt werden. Soll eine Ontologie allerdings firmenübergreifend genutzt werden, entsteht bei standardunabhängigen Lösungen ein mehrmaliger Aufwand, das Matching der Features und deren Parameter zu definieren.

### 3.4 Skill-Schnittstelle des CPPM

Für die Fertigung von Features soll die CAM-Programmierung durch die Definition von Capabilities und Skills sowie autonom agierende Werkzeugmaschinen ersetzt werden (siehe Abbildung 3.27). Damit alle CPPM einheitlich angesteuert werden können, ist eine klar definierte Skill-Schnittstelle notwendig.

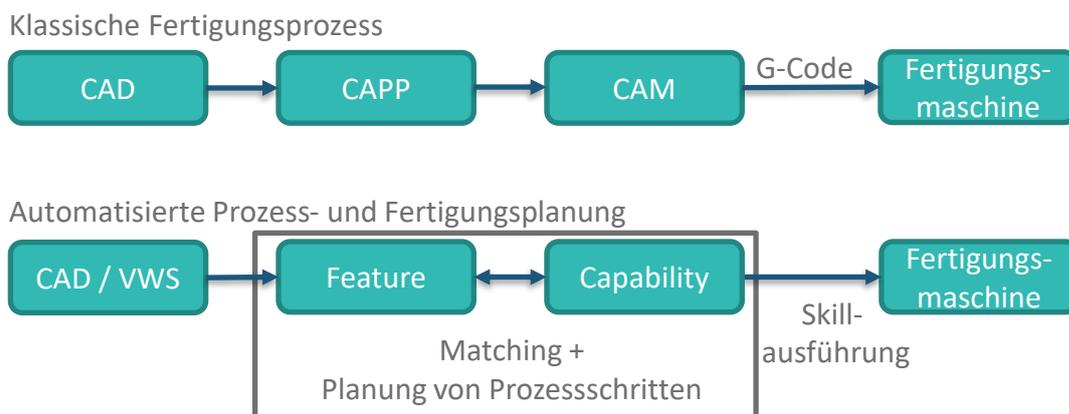


Abbildung 3.27: Vergleich der klassischen Fertigungskette und einer automatisierten Prozess- und Fertigungsplanung

In der definierten Produktionsumgebung kommt den CPPM somit eine tragende Rolle als autonom handelnde Fertigungsmaschine zu. Nach außen müssen die Fähigkeiten der Fertigungsmodule als Capabilities semantisch beschrieben werden, sodass ein Matching der Produktan-

forderungen auf die Produktionsressourcen möglich wird und über den Skill der gewünschte Fertigungsschritt ausgeführt werden kann. Die Skills sollen möglichst einfach beschrieben sein, sodass Werker z. B. intuitiv Skills nutzen und überprüfen können. [Ma18a] beschreibt die Problematik der Detailtiefe in der Skillbeschreibung (ebenso anwendbar auf die Capability) und gibt dort die verschiedenen Möglichkeiten für die Benennung und Definition von Skills vor: Atomare oder kombinierte Skills, prozessunspezifische oder prozessspezifische Beschreibung sowie eine produktspezifische bzw. produktunspezifische Beschreibung. In der Handhabungstechnik existieren häufig prozessspezifische Anforderungen, da diese z. B. den Transport durch eine Fabrik bewältigen und somit eingesetzt werden, um eine Prozesskette zwischen Fertigungsschritten zu beschreiben [VD23]. Capabilities können in diesem Fall häufig ausreichend genau beschrieben werden, um zu entscheiden, ob etwa ein Roboter in einem bestimmten Bauraum ein Produkt mit einer ausreichenden Genauigkeit greifen kann. Hierzu werden die Fähigkeiten der Einzelkomponenten eines CPPM beschrieben, die einander gewisse Anforderungen an einen Prozessschritt zusichern.

Im Bereich der spanenden Bearbeitung werden eine Vielzahl von Parametern geliefert, die für einen einzelnen Fertigungsschritt beachtet werden müssen, z. B. Toleranzmaße und Anforderungen an die Oberflächenrauheit. Die Prozessparameter (wie Vorschub, Werkzeugwahl, Featuretyp, Material, Einspannung und Abmaße des Werkstückes, Kühlung, ...) beeinflussen die Qualität (z. B. Oberflächenrauheiten oder Toleranzen) eines Werkstückes. Aufgrund der vielen Abhängigkeiten kann die Beschreibung einer Capability im Bereich der spanenden Bearbeitung selten mit ihren erreichbaren Qualitätsanforderungen in einer statischen Beschreibung geschehen. Es reicht in diesem Fall nicht aus zu prüfen, ob das CPPM das richtige Feature in der richtigen Form fertigen kann. Vor der Fertigung muss entschieden werden, ob die Maschine die gewünschten Qualitätsansprüche einhalten kann, da die Abhängigkeiten zwischen den Qualitätsansprüchen sowie den Prozessparametern nicht in dem VWS-TM Capability abbildbar sind. Um den Matching-Prozess komplett zu automatisieren, wird festgelegt, dass das CPPM „die Entscheidung treffen muss“, ob die geforderten Qualitätsansprüche eingehalten werden können. Hierzu könnten Simulationsmodelle, Knowledge Graphen oder neuronale Netze angewendet werden. Evtl. muss für komplexere Entscheidungen, wie die Einspannung eines Bauteils, ein Mensch Vorgaben treffen. Generell kann die Machbarkeitsprüfung immer durch einen erfahrenen Mitarbeiter durchgeführt werden. Für die Machbarkeitsprüfung wird der Begriff Feasibility-Check eingeführt:

*Der **Feasibility-Check** ermöglicht es, mit einem langfristigen Zeithorizont im Voraus zu prüfen, ob ein CPPM für einen spezifischen Produktionsschritt geeignet ist. Das Ergebnis des Feasibility-Checks gibt Auskunft, ob der gewünschte Skill mit dem gewünschten Parametern erfolgreich gefertigt werden kann. Des Weiteren können im Feasibility-Check wichtige Key-Performance-Indikatoren bereits berechnet werden, z. B. der kalkulierte Preis für den Fertigungsschritt, der geschätzte Energieverbrauch oder die geschätzte Fertigungszeit.*

Mit Hilfe der Informationen des Feasibility-Checks können Produktionsprozesse in Zukunft nach unterschiedlichen Gesichtspunkten optimiert werden. Da die Machbarkeitsanalyse stark mit dem benötigten Skill sowie dessen vorgegebenen Parametern zusammenhängt, soll der Feasibility-Check in die Skill-Schnittstelle integriert werden. Es wird somit die Anforderung definiert, dass mehrere Agenten die Planung und Durchführung des Produktionsprozesses eines CPPM vornehmen (siehe Abbildung 3.28). Die Agenten müssen nicht zwingend auf der Steuerung des Moduls implementiert sein, sondern können auch Services auf einem Edge Cluster, Server oder in der Cloud aufrufen. Im Folgenden wird erläutert, welche Prüfungen durchgeführt werden müssen, bevor ein Skill umgesetzt werden kann. Gemeinsam mit den Agenten bildet die Werkzeugmaschine somit ein Holon, in dem Agenten die Planung von Fertigungsschritten vornehmen (vgl. Kapitel 2.2.1 Abbildung 2.10).

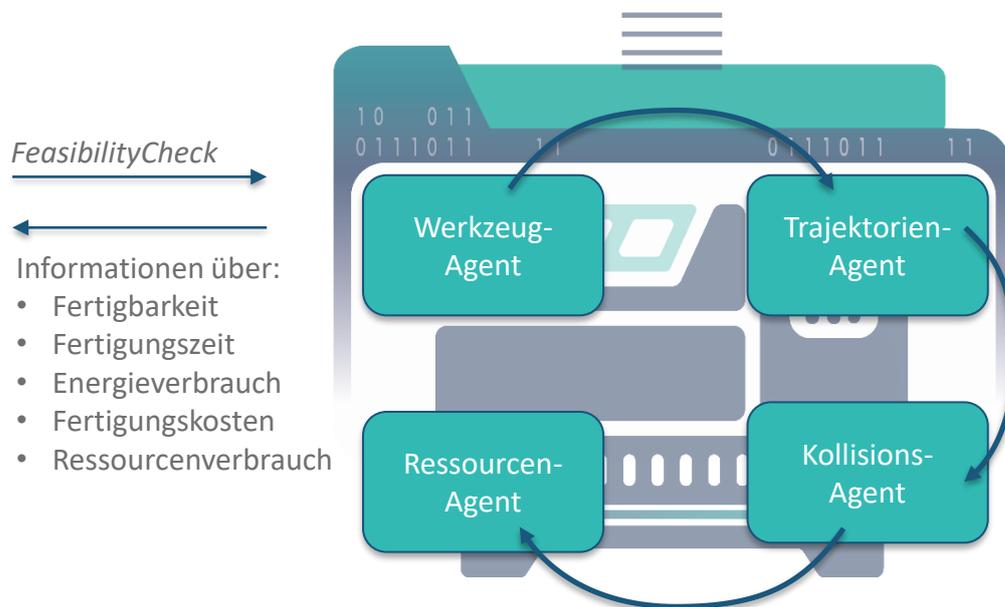


Abbildung 3.28: Agenten, die einen Feasibility-Check für eine Werkzeugmaschine durchführen

Um einen Fertigungsschritt in der spanenden Bearbeitung automatisiert zu prüfen, werden mindestens die folgenden Agenten benötigt:

- **Werkzeugagent:** Der Werkzeugagent entscheidet anhand der Maschinenbestückung, welches der Werkzeuge am geeignetsten für den angefragten Skill ist. Anhand des Materials, des zu fertigenden Feature, der vorgegebenen Oberflächenrauheit und Toleranzen, trifft der Agent die Entscheidung, welches der vorhandenen Werkzeuge einsetzbar ist und berechnet die Fertigungsparameter. Des Weiteren muss geprüft werden, ob mehrere Bearbeitungsschritte innerhalb eines Skills notwendig sind, z. B. neben dem Schrappvorgang auch ein Schlichtvorgang.
- **Trajektorienagent:** Der Trajektorienagent plant anhand des ausgewählten Werkzeuges und der vorgegebenen Bearbeitungsschritte die Bearbeitungsstrategie im Feature. Es wird ein kollisionsfreies An- und Abfahren zum benötigten Feature berechnet. Hierfür

kommen verschiedene Ansätze in Frage. Die Bewegungen können vorab berechnet werden oder die Trajektorie wird online anhand der vorhandenen Produkt- und Umgebungsinformationen geplant.

- **Kollisionsagent:** Der Agent prüft ständig, ob der Fertigungsprozess ohne Kollisionen durchgeführt werden kann. Auch hier existiert die Möglichkeit einer online und offline Kollisionsprüfung. Generell sollte schon vor Starten des Fertigungsauftrages an einem CPPM geprüft werden, ob die erforderlichen Schritte ohne Kollisionen durchgeführt werden können.
- **Ressourcenagent:** Mit Hilfe von Kalkulationsalgorithmen und maschinellem Lernen, soll das CPPM Voraussagen treffen, wie hoch die geschätzten Kosten, der Energieverbrauch und der Zeitaufwand ist. Somit wird in Zukunft die Voraussage des Ressourcenverbrauchs möglich und die Produktion kann nach ökologischen Gesichtspunkten optimiert werden. Die Idee, dass Vorhersagen von Energieverbräuchen in der Produktion eingesetzt werden können, ist z. B. in [Mo20] beschrieben.

Nach dem Feasibility-Check werden die Parameter über die Skill-Schnittstelle zurückgegeben, sodass eine Gesamtprozessplanung stattfinden kann. In der Planung können geeignete Maschinen ausgewählt und Prozesse nach verschiedenen Faktoren (Kosten, Zeit, Energieverbrauch) optimiert werden.

Zwischen dem Feasibility-Check und der Ausführung eines Skills, kann ein längerer Zeitraum liegen. Um kurz vor Ausführung eines Skills dessen Verfügbarkeit zu prüfen, wird der Begriff Kontext-Check eingeführt:

*Der **Kontext-Check** ermöglicht kurz vor Ausführung eines Skills an einem CPPM zu überprüfen, ob das CPPM zurzeit in der Lage ist, den gewünschten Skill zu realisieren. Mit Hilfe des Kontext-Checks können mögliche Fehler z. B. vor der Bestückung eines CPPM identifiziert werden.*

Der Kontext-Check ermöglicht vor allem das Identifizieren von Fehlern, die zwischen der Planung eines Prozesses und der Ausführung aufgetreten sind. Ist z. B. ein benötigter Fräser einer Werkzeugmaschine nicht mehr verfügbar, kann noch vor der Beladung der Maschine das fehlende Werkzeug registriert werden und eine Umplanung des Prozesses stattfinden.

### 3.4.1 OPC UA-Skills

Jedes CPPM in der Produktionsumgebung wird in Zukunft durch eine VWS beschrieben. In Abbildung 2.6 wurden die Möglichkeiten zum automatisierten Ansteuern mit unterschiedlichen VWS-Typen beschrieben. Im Folgenden liegt der Fokus besonders auf der Struktur eines Skills und einer Technologie als Schnittstelle zur sicheren Ansteuerung. [Ba16] zeigt in einer Recherche zu möglichen Integrationstechnologien und [LXW20] in einer Untersuchung über existieren-

de Automatisierungsstandards, dass OPC UA einen ausgezeichneten herstellerübergreifenden Standard bildet. Mit OPC UA ist die semantische Beschreibung von Maschinen möglich (siehe Kapitel 2.1.4). Wie aus [Bu19; Ho14; Sc16] hervorgeht, wird OPC UA das Potential zugesagt zum De-facto-Standard der I4.0-Kommunikation zu werden. [Lo13] nutzt *Device Profile for Web Services* (DPWS) zur Ansteuerung komponentenabhängiger Basisservices in der Feldebene (hier als Skills bezeichnet) und verweist auf OPC UA als vielversprechende Alternative. [LXW20] ermittelt in seiner Auswertung gegenüber existierenden Standards wie PackML eine fehlende Möglichkeit der Ausführung von Prozessen für OPC UA. Allerdings ist die PackML Zustandsmaschine inzwischen als OPC UA *Companion Specification* OPC 30050 verfügbar und kann als Template für die Schnittstelle des Skills in Betracht gezogen werden. Zur Modellierung der VWS in OPC UA existiert die *Companion Specification* OPC 30270. Die Spezifikation beschreibt im Detail, wie sich Informationen aus der VWS auf einem OPC UA-Server darstellen lassen. In [PI20c] wird der Skills als *Operation* einer Fähigkeit (Capability) in der VWS modelliert. Je nach Anwendung wird davon ausgegangen, dass es verschiedene Typen eines Skills geben kann. So können laut [PI20c] Skills in verschiedenen Ausprägungen auftreten, z. B. als *State* oder *Trigger* Variablen, Operationen, Funktionsblöcke oder semantischen Protokolle. Zur Modellierung des Skills wird das herstellerunabhängige Kommunikationsframework OPC UA verwendet.

Das Ziel eines OPC UA-Skills ist, mit Hilfe einer möglichst einfachen Zustandsmaschine einen definierten Prozess auf einer Maschine auszuführen. Zunächst wurden die folgenden Anforderungen zur Modellierung eines Fertigungsskills im Bereich der spanenden Bearbeitung identifiziert:

- Ein Skill soll eine Art Funktionsblock sein, mit dem Produktionsschritte gestartet, angehalten, gestoppt, gesperrt und resettet werden können und ähnelt somit dem Konzept aus IEC 61499 (siehe Kapitel 2.1.3).
- Der aktuelle Status der Skills muss überwachbar sein. Somit muss der aktuelle Status sowie die letzte Transition des Skillzustandes als Variable einsehbar sein und wichtige Transitionen (z. B. Maschinenausfälle) als OPC UA-Event ausgegeben werden. Teilnehmer können so die Informationen im Netzwerk abrufen bzw. den relevanten Informationen folgen.
- Die Möglichkeit, individuelle Parameter vor dem Starten eines Skills zu übertragen, muss gegeben sein.
- Ein Feasibility-Check muss in das Skill-Modell integriert werden. Dies ermöglicht im Voraus zu prüfen, ob eine Ausführung des Skills mit den erforderlichen Parametern realisierbar ist. Die Berechnung bzw. Planung des Skills kann unter Umständen eine längere Zeit in Anspruch nehmen. Daher soll die Machbarkeitsanalyse als Zustandsmaschine ausgeführt werden. Das Ergebnis soll in einem Node nach abgeschlossener Prüfung hinterlegt werden.

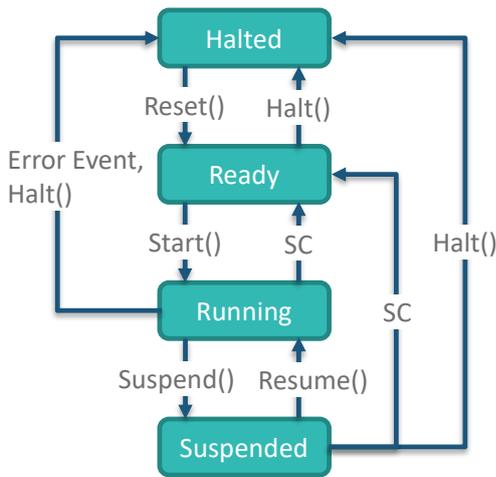
- Ein Kontext-Check ist zu integrieren, um kurz vor Ausführung eines Skills zu prüfen, ob die Maschine in der Lage ist, den gewünschten Skill auszuführen. Der Kontext-Check soll somit aufdecken, ob zwischen Feasibility-Check und Ausführung des Skills mögliche Fehler oder Defekte an der Maschine aufgetreten sind, die eine erfolgreiche Ausführung des Skills verhindern.
- Es muss die Möglichkeit bestehen, Informationen nach der Ausführung eines Skills zurückzuführen. Unter einem Node sollen somit Parameter nach der Ausführung übergeben werden können. Dies könnten z. B. Qualitätsinformationen oder bei einer Objekterkennung, Informationen über identifizierte Gegenstände sein.
- Es soll die Möglichkeit bestehen, dass mehrere Clienten gleichzeitig auf einen Server zugreifen. So soll z. B. ein Client einen Skill starten können und ein zweiter Client parallel eine Machbarkeitsprüfung ausführen. Dabei muss über einen Lock-Mechanismus abgesichert werden, dass nicht 2 Clienten zeitgleich dasselbe Parameterset beschreiben.

In zwei existierenden Veröffentlichungen wurden hierzu bereits Zustandsautomaten in OPC UA definiert und angewendet. Zum einen verwendete [DZ18] die OPC UA Spezifikation 10 [OPC 40502] für OPC UA Programme, um einen einfachen Zustandsautomaten für Skills zu definieren. Dieser basiert auf vier Zuständen, die über Methoden getriggert werden können. In [DZ18] wird gezeigt, dass komplexere Zustandsautomaten wie bspw. PackML hinter dem einfach verständlichen OPC UA-Programm gekapselt werden können. Neben dem aktuellen Zustand kann über eine Variable auch immer die letzte Transition eines Skills eingesehen werden. So hat bspw. ein Client, der auf die Maschine zugreift, die Möglichkeit, die Maschine zu überwachen. Die in BaSys entwickelte Führungskomponente nutzt PackML, um verschiedene komplexe Typen von Skills anzusteuern [GWE18]. In der Fachabteilung *Integrated Assembly Solutions* (IAS) des Verbandes Deutscher Maschinen- und Anlagenbau e. V. (VDMA) wurde ein *Skill Execution Metamodel* standardisiert und in einem Demonstrator auf der Messe Automatica 2018 präsentiert [Zi19].

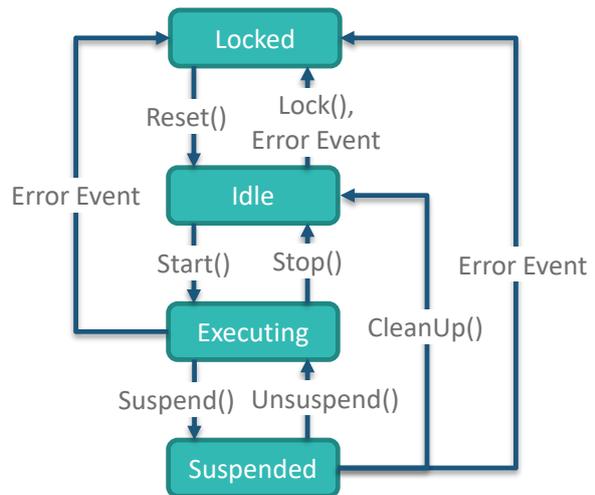
Die von [DZ18] und [Zi19] definierten Modelle sind ähnlich und unterscheiden sich nur in den Bezeichnungen sowie einige zusätzliche Methoden. In Abbildung 3.29 sind die beiden Modelle dargestellt. Die wichtigste Änderung ist eine Methode, die ermöglicht einen Skill von *Executing* in *Idle* zurückzusetzen und ist für die Fertigung als relevant zu betrachten. Hinzugekommen ist eine Methode, die es ermöglicht, Startparameter eines Skills zu setzen. Die Startparameter ermöglichen eine Vorgabe bestimmter Variablen, um die Ausführung des Skills zu beeinflussen. So kann z. B. für einen Bewegens-Skill die gewünschte Position und Geschwindigkeit einer Achse definiert werden.

Im Vergleich hierzu wird die PackML *State Machine* auf Basis der OPC UA *Companion Specification* für PackML [OPC 30050] dargestellt. Diese ist deutlich umfangreicher und aufwendiger in der Implementierung.

OPC UA Program Finite State Machine



VDMA Skill Execution Metamodel



SC: internal State Change

PackML State Machine

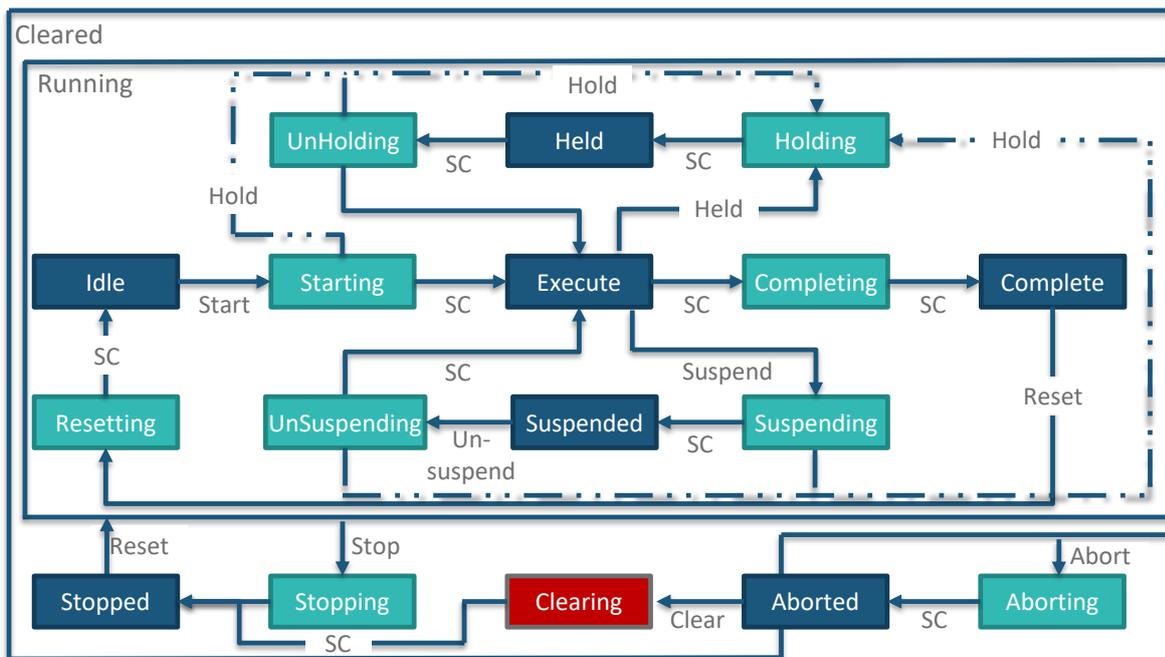


Abbildung 3.29: OPC UA Program Finite State Machine (oben links) in Anlehnung an [DZ18; OPC 40502], VDMA Skill Execution Metamodel (oben rechts) in Anlehnung an [Zi19] und PackML State Machine in Anlehnung an [OPC 30050] (unten)

Die in der Literatur analysierten Skill-Modelle zeigen, dass im Vergleich zum PackML-Modell, einige Zustände vereinfacht werden können. Das Modell zur Ansteuerung einer Maschine wird somit weniger komplex. Auf Basis der in diesem Kapitel gestellten Anforderungen wurden die in der Literatur existierenden Modelle mit zusätzlichen Informationen erweitert. Somit wird sichergestellt, dass alle relevanten Daten mit einer Maschine über die Schnittstelle ausgetauscht werden können. Das konzipierte Modell ist in Abbildung 3.30 dargestellt.

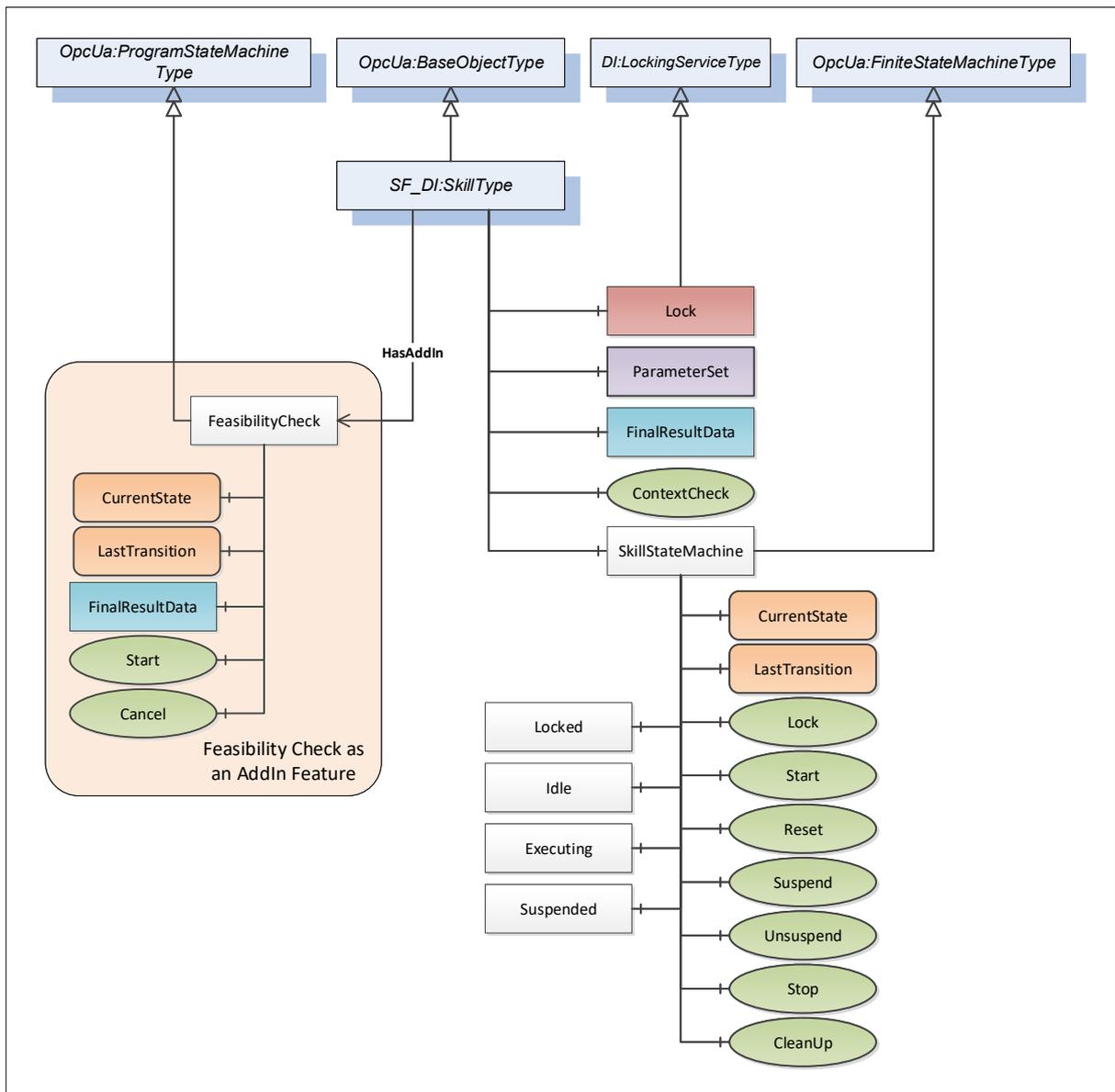


Abbildung 3.30: Darstellung eines OPC UA-Nodesets für einen Skill inklusive eines Feasibility-Checks [Vo21a]

Die einzelnen OPC UA *Nodes* unter dem Skill haben die folgende Erläuterung:

- *Lock*: Der *Locking Service* bewerkstelligt, dass mehrere Clients zeitgleich einen Skill nutzen können. Evtl. kann es mehrere Clients zur Planung und zur Ausführung von Skills geben. Somit wird verhindert, dass die Clients zeitgleich auf ein Parameterset eines Skills zugreifen und diese gegenseitig überschreiben. Für den Lock-Mechanismus kann die OPC UA-Spezifikation 100 *Devices* [OPC 10000-100] verwendet werden, in der ein *LockingServiceType* beschrieben ist.
- *ParameterSet*: Enthält mehrere Variablen, durch die ein Skill parametrisiert werden kann. Die Parameter ermöglichen eine individuelle Ausführung für einen spezifischen Auftrag. Es kann Variablen geben, die verpflichtend für die Ausführung eines Skills benötigt werden und Variablen, die nicht zwingend benötigt werden.

- *FinalResultData*: Enthält Variablen, die Informationen nach Ausführung eines Skills zurückgeben. So kann z. B. eine Fehlermeldung bei nicht erfolgreicher Ausführung oder Toleranzinformationen nach einer Qualitätsprüfung zurückgegeben werden.
- *ContextCheck*: Der *ContextCheck* ist ein Methodenaufruf, der kurz vor Ausführung eines Skills prüft, ob die Maschine mit dem geforderten *ParameterSet* in der Lage ist, den Skill auszuführen. Der *ContextCheck* ermöglicht kurzfristig auftretende Fehler / fehlende Werkzeuge oder ähnliche Probleme rechtzeitig zu identifizieren.
- *SkillStateMachine*: Enthält alle benötigten Methoden, um einen Skill zu steuern (Starten, Stoppen, Blockieren und Resetten) und Variablen, um den aktuellen Status eines Skills zu prüfen. Die *SkillStateMachine* wurde aus [Zi19] entnommen, da die Zustandsmaschine in einem VDMA Arbeitskreis entwickelt wurde und universal einsetzbar ist.
- *FeasibilityCheck*: Im *FeasibilityCheck* ist eine Zustandsmaschine integriert, die eine Abfrage ermöglicht, ob ein Skill in der gewünschten Konfiguration ausführbar ist. Zum Starten werden die vorausgesetzten Parameter in das *ParameterSet* geschrieben und die Start-Methode des *FeasibilityCheck* aufgerufen. Über die Variablen in *FinalResultData* des *FeasibilityChecks* kann nach erfolgreicher Prüfung die benötigte Zeit, der voraussichtliche Energieverbrauch sowie die anfallenden Kosten abgefragt werden. Da eine Simulation, Planung und Prüfung der Skills mehrere Sekunden oder Minuten dauern kann, wurde der Skill als Zustandsmaschine und nicht als einfacher Methodenaufruf ausgeführt. Im Vergleich zur *SkillStateMachine* besitzt der *FeasibilityCheck* nur die Methode *Start* und *Cancel*. Mit den Methoden kann der *FeasibilityCheck* gestartet oder abgebrochen werden. Ein *SuspendedState* ist nicht vorgesehen, da Simulationen, Berechnungen oder Kollisionsprüfungen in der Regel nicht beliebig pausiert und fortgesetzt werden kann. Da nur komplexere Skills bzw. Skills, die individuelle Parameter zulassen, einen *FeasibilityCheck* benötigen, wurden diese Nodes als OPC UA *Add-In* nach [OPC 10001-7] eingebunden. Das *Add-In* kann somit bei Bedarf eingefügt werden.

Der Ablauf zur Planung eines Prozesses und der Ausführung eines benötigten Skills zwischen einem OPC UA-Client (z. B. Agent) und einem OPC UA-Server (z. B. CPPM) ist in Abbildung 3.31 dargestellt. Zunächst verbindet sich der Client mit dem OPC UA-Server des benötigten CPPM und sucht nach dem benötigten Skill-Node. Die Parameter zur Ausführung des Skills werden in das *ParameterSet* geschrieben und die *Start*-Methode des *FeasibilityCheck* ausgeführt. Die Agenten kalkulieren nacheinander, mit welchem Werkzeug der benötigte Skill umgesetzt werden kann, planen eine Trajektorie, prüfen diese auf mögliche Kollisionen und berechnen den voraussichtlichen Energie-, Ressourcenverbrauch, Kosten und Dauer. Aus der *FinalResultData* des *FeasibilityCheck* kann sich ein Planungsalgorithmus diese Informationen abrufen und kalkuliert, ob die Maschine im Prozess eingesetzt wird oder nicht. Ist das CPPM im Prozess eingeplant, wird zum entsprechenden Zeitpunkt das benötigte Produkt zum CPPM transportiert. Kurz vor Ausführung des Skills wird ein Kontext-Check ausgeführt. Hierzu werden die benötigten Para-

meter in das *ParameterSet* des Skills geschrieben und die *Start* Methode *ContextCheck* aufgerufen. Wird die Methode erfolgreich aufgerufen, kann der Skill ausgeführt werden. Kann der Skill in diesem Moment nicht ausgeführt werden, können über den Methodenaufruf Fehlercodes zurückgegeben werden. Zum Ausführen des Skills wird die Start-Methode in der *SkillStateMachine* getriggert und der *CurrentState* wechselt in *Executing*. Nach erfolgreicher Ausführung des Skills wechselt der Status des CPPM zurück in *Idle*.

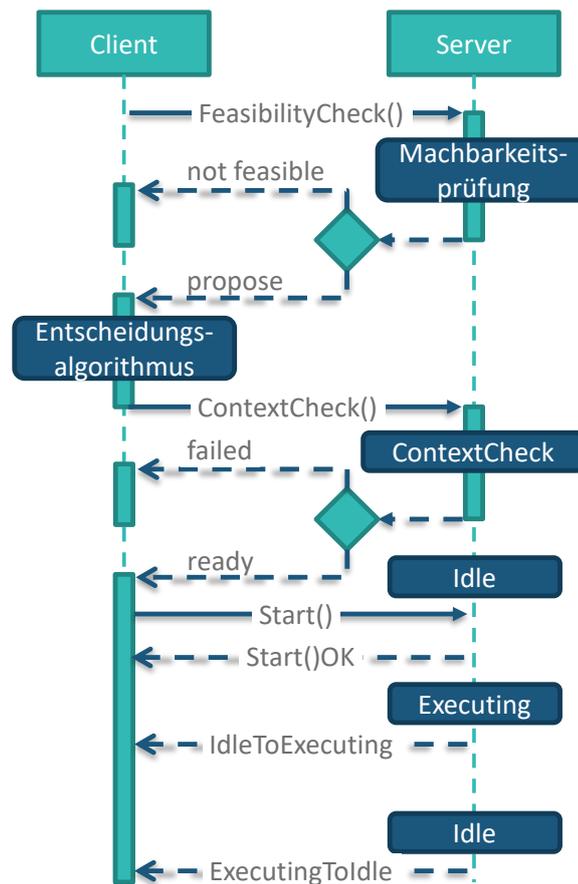


Abbildung 3.31: Kommunikationsablauf eines OPC UA-Clients und Servers in der Fertigungsplanung und -ausführung in Anlehnung an [Vo21a]

Neben der individuellen Ansteuerung von Maschinenfähigkeiten bietet das definierte Skill-Nodeset mit dem Feasibility-Check einen Zugangspunkt zum digitalen Zwilling für die Produktionsplanung eines CPPM. Dieser ist notwendig, um in Zukunft eine individuelle Prozessplanung auf Ebene der CPPM automatisiert durchführen zu können. Wird eine Maschine in das Fabriknetzwerk eingebunden, ist somit nicht nur die Steuerung der Skills möglich, sondern auch der Zugangspunkt zu einem Teil des digitalen Zwillings eines CPPM gegeben.

### 3.4.2 VWS-Teilmodell für Skills

Der Skill kann durch eine Schnittstellenbeschreibung in einem VWS-TM eines CPPM beschrieben werden, sodass ein Agent die Schnittstelle interpretieren kann. Die Beschreibung referenziert

auf den OPC UA-Server, der die technologische Schnittstelle des CPPM bildet (siehe Abbildung 3.30 in Kapitel 3.4.1). Ein erarbeitetes Konzept für das VWS-TM des Skills ist in Abbildung 3.32 und 3.33 dargestellt. Ein Skill besteht aus den SMC *StateMachine* und *SkillInterface*. Die *StateMachine* definiert die Zustände und Transitionen des Skills, während im *SkillInterface* alle notwendigen Informationen zur Ansteuerung eines Skills gesammelt werden. Im *SkillInterface* wird die Art der Kommunikationstechnologie beschrieben, die zum Einsatz kommt. Unter *Trigger* sind alle Methoden zusammengefasst, mit denen die Zustandsmaschine eines Skills beeinflusst werden kann. Ist ein Skill über eine VWS erreichbar, könnten die *Trigger* als *Operation* umgesetzt werden [PI20c]. In dieser Arbeit wird auf die Methoden des OPC UA-Servers eines CPPM referenziert. Unter *InputParameters* wird das *ParameterSet* des Skills abgebildet. Die SMC *ResultParameter* enthält alle Parameter des OPC UA-Nodes *FinalResultData*.

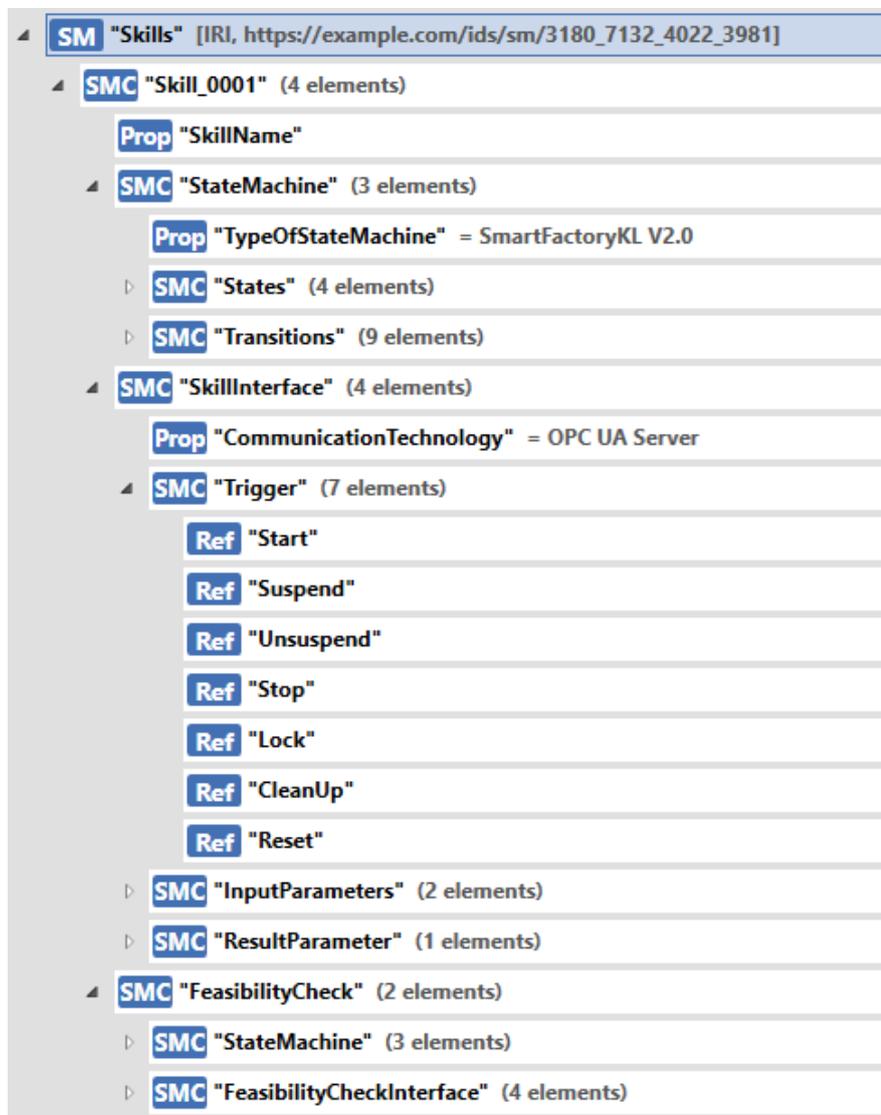


Abbildung 3.32: Modellierung eines VWS-TM für Skills

Die Transitionen sind in Abbildung 3.33 separat dargestellt. Sie wurden mit einem *AnnotatedRelationshipElement* in der VWS umgesetzt. Hierdurch wird ein Ausgangs- und ein Endzustand miteinander verbunden und durch eine Referenz erweitert. Die Referenz *Trigger* verweist auf die

Methode, die eine entsprechende Transition auslöst. Anstatt einer Methode kann eine Transition auch durch ein internes *Error-Event* ausgelöst werden. So ist es möglich, alle Zustandsübergänge eines Skills abzubilden.

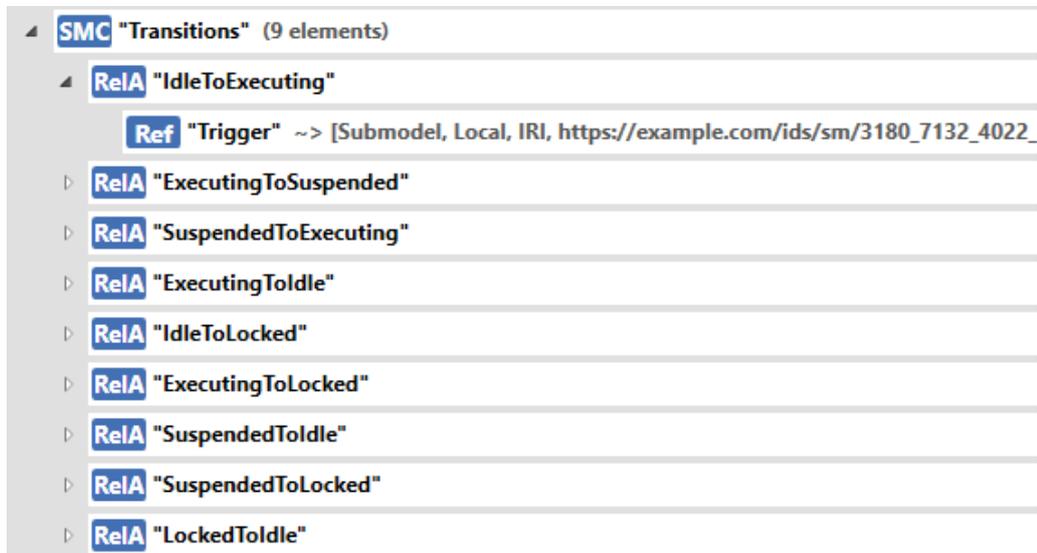


Abbildung 3.33: Modellierung von Transitionen im VWS-TM für Skills

# 4 Implementierung und Validierung

Im Folgenden werden zu den in Kapitel 1 gestellten Forschungsfragen implementierte Lösungsansätze gezeigt. Abbildung 4.1 gibt einen Überblick über den Kapitelinhalt.

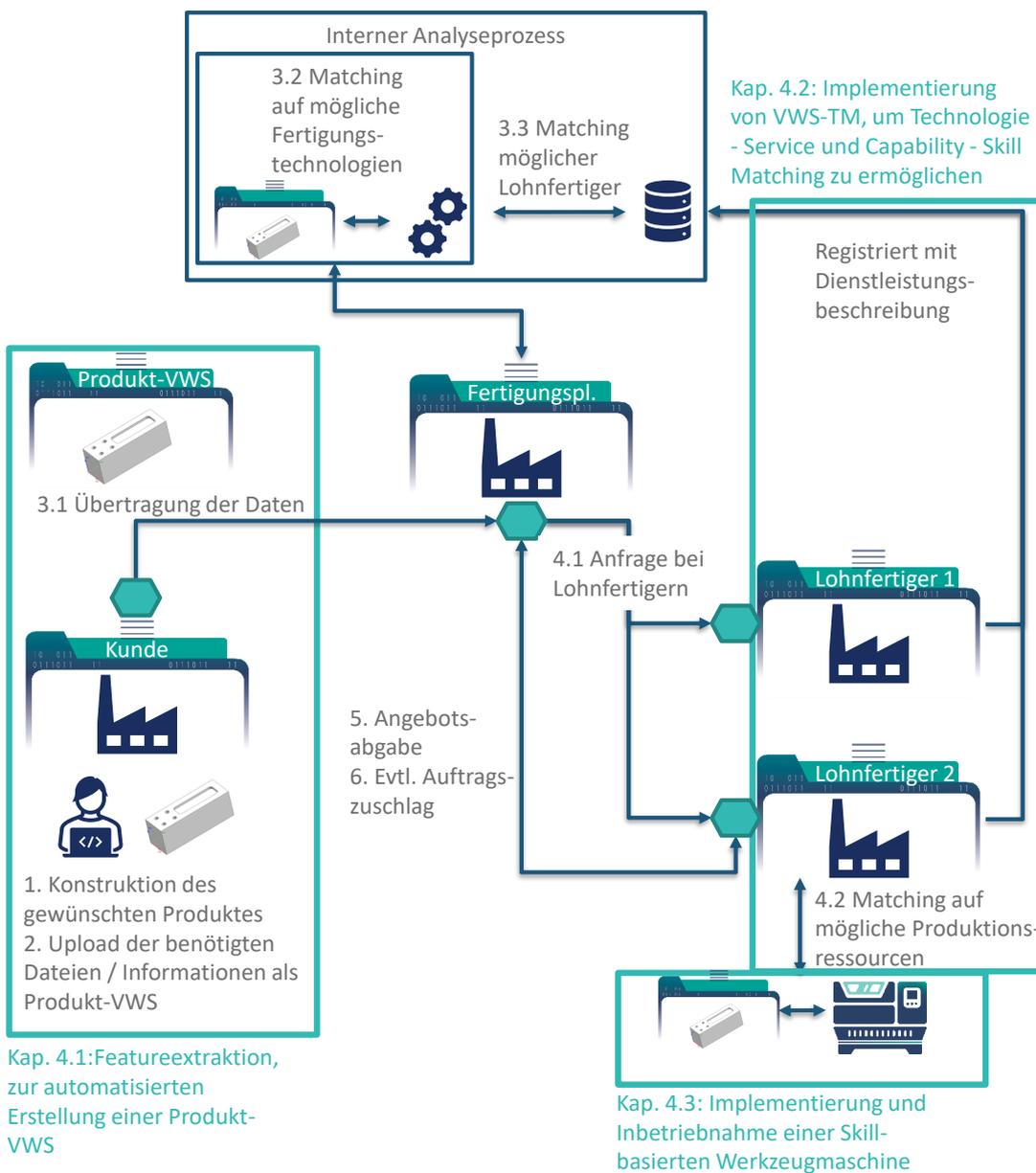


Abbildung 4.1: Gliederung Kapitel 4

Zu Forschungsfrage 1 wird eine automatisierte Extraktion von Features aus einer CAD-/ CAM-Umgebung implementiert. Somit werden die benötigten Informationen zur Erzeugung einer Produkt-VWS automatisiert bereitgestellt. Ein Capability- und Service-TM wird in einer VWS modelliert, um individuell beschriebene Skills einer Produktionsumgebung zu referenzieren und so ein dynamisches Matching von Produktanforderungen auf Capabilities und Skills zu ermöglichen (Forschungsfrage 2). Des Weiteren wird eine Fräsmaschine mit einem OPC UA-Interface in Betrieb genommen. Es wird die Fertigung von den in Forschungsfrage 1 extrahierten Features anhand von Skills gezeigt (Forschungsfrage 3).

## 4.1 Extraktion von Features aus CAD/CAM-Software

Zur Erstellung der Produkt-VWS ist das automatisierte Auslesen und Abspeichern der relevanten Produktdaten wichtig. Hierzu wurde eine automatische Extraktion von Features umgesetzt. Zur Extraktion mehrerer Features wurde die Software Siemens NX 1872 verwendet. Mit der API von NXOpen besteht die Möglichkeit, eine Featureerkennung automatisiert zu starten [@SiemensPLM1]. Die identifizierten Features können im Fertigungsmodul NX CAM über drei unterschiedliche Modi identifiziert werden [@SiemensPLM2]. Die Formelement-Identifikation analysiert Bauteile, die nach dem Konzept *Design by Feature* konstruiert wurden, die *Manuelle Erkennung* erfordert die Interaktion des Anwenders und dessen Auswahl von Formelementen sowie Flächen [@SiemensPLM2]. Zur Umsetzung der automatisierten Extraktion wurde die *Parametrische Erkennung* verwendet. Hierbei werden Basisfeature des NX Portfolios erkannt [@SiemensPLM2]. Abbildung 3.11 (in Kapitel 3.2.2) gibt einen Überblick der extrahierbaren NX-Features. Das konstruierte Bauteil zeigt, dass unterschiedlichste Features zuverlässig erkannt wurden, mit Ausnahme des Features vom Typ *CORNER\_NOTCH\_ROUND\_CONVEX*. Das analysierte Bauteil wurde in NX konstruiert und im *.prt* Format analysiert, um Informationsverluste zu vermeiden. Bei der Analyse von STEP-Dateien kann der Export und Import des STEP-Formates zu Informationsverlust führen. Dies betrifft primär PMI, z. B. Toleranzangaben. Daher wurde das etablierte Austauschformat zur automatisierten Extraktion der Features nicht verwendet.

Um den Anwendern die Extraktion der Features möglichst einfach zu gestalten, wurde ein Software-Plug-in mit einer Bedienoberfläche implementiert. Die Software soll dabei die folgenden Funktionalitäten erfüllen:

- Ausgabe einer Liste mit allen geometrischen Features des CAD-Modells
- Bereitstellen der Feature-Parameter
- Die Erstellung einer Fertigungsreihenfolge ermöglichen
- Suche nach aktiven Maschinen im Netzwerk und Unterstützung der Abfrage eines Feasibility-Checks

- Ansteuern von Skills in der vorhandenen Fabrik

Der erste Teil der grafischen Oberfläche ist in Abbildung 4.2 dargestellt. Mit dem *FindFeatures*-Button, werden alle Features des aktuell geöffneten Bauteils automatisch ausgegeben.

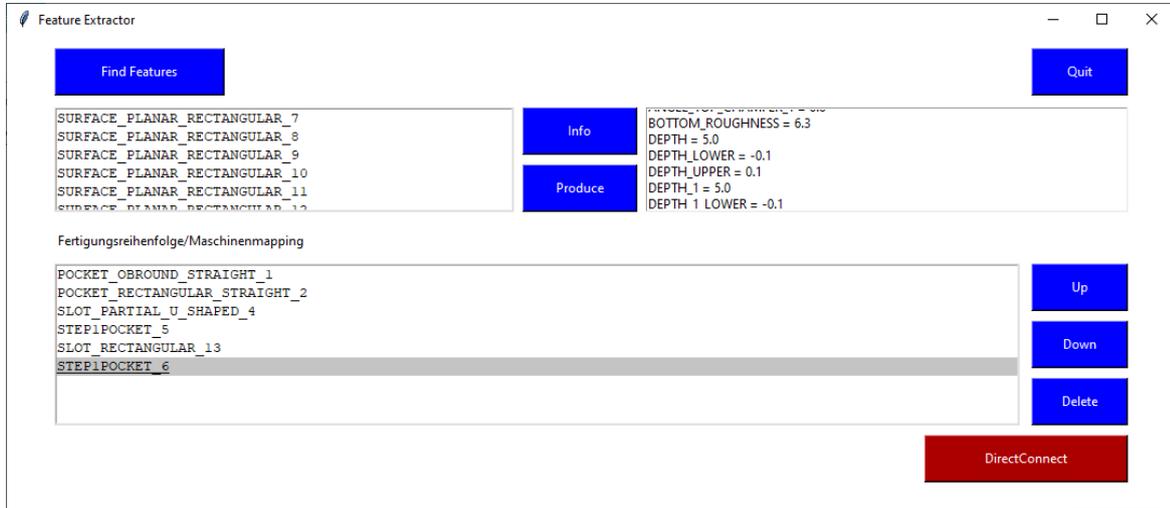


Abbildung 4.2: Fenster 1 der Bedieneroberfläche zum Auslesen der Feature und zur Erstellung einer Fertigungsreihenfolge

Bei Verwendung des *FindFeatures*-Button wird mit dem Befehl *NXOpen.Session.GetSession().Parts.Work.CAMFeatures* auf die vorhandenen Features zugegriffen und eine iterierbare Liste erzeugt. Die Speicherstruktur der NX-CAM-Features und der zugehörigen Parameter ist in Abbildung 4.3 dargestellt.

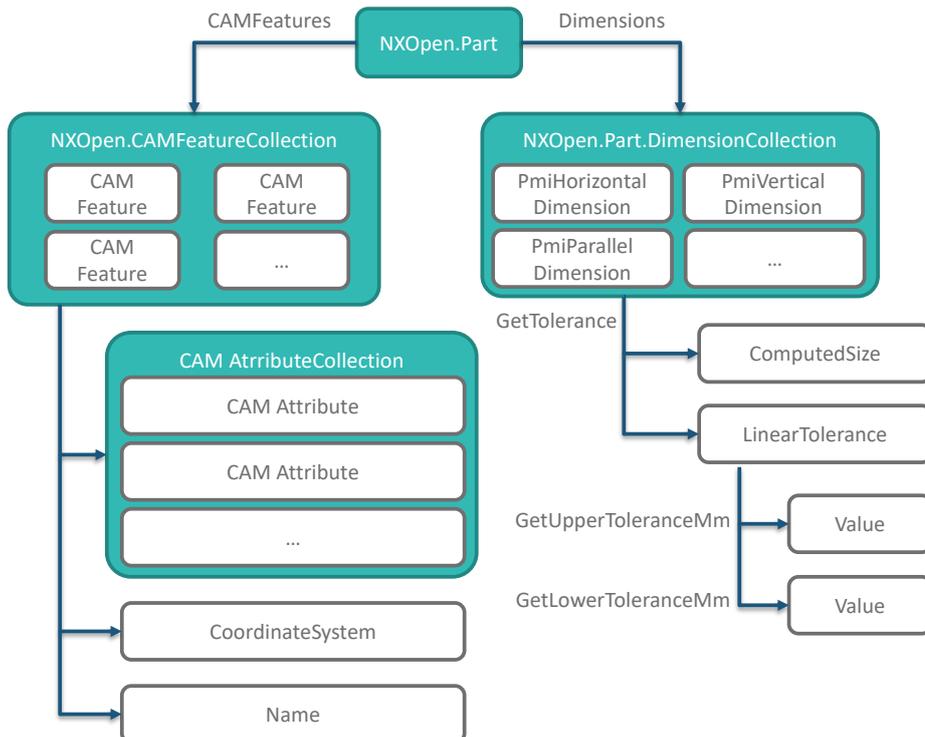


Abbildung 4.3: Speicherstruktur der CAM Feature- und Dimensions-Objekte in NX

Der Typ und die Parameter jedes Features werden ausgelesen, aufeinander gemappt und in einer Python *Dictionary*-Struktur gespeichert. Die Attributvariablen eines CAM-Feature-Objektes enthalten die Maßangaben eines Features, allerdings keine Toleranzdaten, die an einem Punkt, einer Kante oder einer Fläche eines Features platziert wurden. Somit werden Toleranzangaben, die der Konstrukteur an einer Kante des Features vorgenommen hat, nicht in die Parameter übernommen. Die Toleranzangaben können durch den Befehl `workPart.PMIManager.Pmis` ausgelesen werden. Hierzu werden zusätzlich die PMI zur Definition von Längen- und Breiten-Toleranzen durchsucht und anhand der Basislänge dem Feature-Parameter zugeordnet. So können die Featureinformationen mit Toleranzwerten erweitert werden. Toleranzen zwischen einzelnen Features wurden in der implementierten Software nicht abgebildet werden.

Der *Info*-Button der Bedieneroberfläche ermöglicht, die extrahierten Parameter eines ausgewählten Features in einem separaten Fenster zu betrachten. So können diese durch einen Anwender noch einmal geprüft werden. Mit Hilfe von *Produce* lässt sich ein Feature in die Fertigungsliste übertragen (siehe Abbildung 4.2). Der Anwender kann an dieser Stelle die gewünschte Fertigungsreihenfolge mit den Buttons *Up*, *Down* und *Delete* verändern. In der Fertigung muss z. B. beachtet werden, dass überlagernde Features von oben nach unten abgearbeitet werden. Nach Definition der Fertigungsreihenfolge kann sich der Anwender über den Button *DirectConnect* mit dem Maschinenpark über die definierte OPC UA-Schnittstelle verbinden. An den Maschinen kann anschließend ein Feasibility-Check ausgeführt und die Skills gestartet werden. Die erforderlichen Schritte bis zur Ausführung eines Skills sind in Kapitel 4.3 erläutert.

Die in der Bedieneroberfläche dargestellten Featureinformationen können in eine Produkt-VWS exportiert werden. Somit kann in einem verteilten Fertigungsnetzwerk mit Hilfe der Produkt-VWS nach möglichen Lohnfertigern gesucht werden. Je nach Definition der Features und Zuordnung in eine bestimmte Prozesskategorie, kann das Feature von einer Fertigungsplattform interpretiert und einem benötigten Service zugeordnet werden. Beispielhaft wurden die Informationen der Featureextraktion in die Produkt-VWS überführt. Hierzu wurden zum einen ein Feature spezifisch abgebildet und zum anderen die allgemeinen Produktspezifikationen wie Material und Form des Bauteils.

Das NX-Feature *Pocket\_Rectangular* würde z. B. wie folgt mit allen relevanten Parametern in die definierte Produkt-VWS als *MillingFeature* exportiert werden (siehe Abbildung 4.4). Die Parameter des Featureextraktor wurden hierzu in ein Submodel übertragen, das für NX-Features angelegt wurde. In dem Feature wurde eine schematische Skizze des Features als *File* hinterlegt. Hauptsächlich enthält die SMC eines Features die geometrischen Parameter. Parameter, die typischerweise ohne Wert hinterlegt sind, wurden mit dem Zusatz *Empty* versehen. In die restlichen Parameter wurde der jeweilige Wert der Featureextraktion übertragen. Parameter ohne Wert können als optionale Parameter einer Capability und eines Skills angesehen werden. Mit *Qualifiern* zu einer *Property* kann definiert werden, ob ein Parameter verpflichtend oder optional ist.

SM	"PartSpecification" [IRI, <a href="https://example.com/ids/sm/9111_3131_8012_7308">https://example.com/ids/sm/9111_3131_8012_7308</a> ]
SMC	"NXMillingFeatures" (5 elements)
SMC	"NX_PocketRectangularStraight" (58 elements)
File	"NX_PocketRectangularStraight_Schematic" -> /aasx/files/Pocket_Recta
Prop	"AngleTopChamfer" = 0 @ {Multiplicity=One}
Prop	"BottomRadius" = 0 @ {Multiplicity=One}
Prop	"BottomRadiusLower" = -0.1 @ {Multiplicity=One}
Prop	"BottomRadiusUpper" = 0.1 @ {Multiplicity=One}
Prop	"BottomRoughness" = 6.3 @ {QualityAssurance=true} @ {Cerification=I/
Prop	"BottomRoughnessAllowance_Empty" @ {Multiplicity=One}
Prop	"BottomRoughnessDirectionOfLay_Empty" @ {Multiplicity=One}
Prop	"BottomRoughnessProcess_Empty" @ {Multiplicity=One}
Prop	"BottomRoughnessRoughnessCutoff_Empty" @ {Multiplicity=One}
Prop	"BottomRoughnessRoughnessMin_Empty" @ {Multiplicity=One}
Prop	"BottomRoughnessRoughnessSpacing_Empty" @ {Multiplicity=One}
Prop	"BottomRoughnessRoughnessValue_Empty" @ {Multiplicity=One}
Prop	"BottomRoughnessWaviness_Empty" @ {Multiplicity=One}
Prop	"Chamfer_Empty" @ {Multiplicity=One}
Prop	"Color" @ {Multiplicity=One}
Prop	"Comment_Empty" @ {Multiplicity=One}
Prop	"Depth" = 3 @ {Multiplicity=One}
Prop	"DepthLower" = -0.1 @ {Multiplicity=One}
Prop	"DepthTopChamfer" = 0 @ {Multiplicity=One}
Prop	"DepthUpper" = 0.1 @ {Multiplicity=One}
Prop	"Length" = 55 @ {Multiplicity=One}
Prop	"LengthLower" = -0.3 @ {Multiplicity=One}
Prop	"LengthUpper" = 0.3 @ {Multiplicity=One}
Prop	"MachiningRule_Empty" @ {Multiplicity=One}
Prop	"Name_Empty" @ {Multiplicity=One}
Prop	"Radius" = 3.5 @ {Multiplicity=One}
Prop	"RadiusLower" = -0.1 @ {Multiplicity=One}

Abbildung 4.4: Auszug eines NX\_PocketRectangularStraight Features in der Produkt-VWS

Parameter, wie das Material sowie die geometrische Form und Größe, können ebenfalls aus NX extrahiert und unter *PhysicalProperties* ergänzt werden (siehe Abbildung 4.5). So kann aus NX die *BoundingBox* einer CAD-Konstruktion genutzt werden, um die Außengeometrie zu identifizieren. Mit Hilfe der *BoundingBox* kann ein Lohnfertiger benötigtes Rohmaterial identifizieren.

Des Weiteren können die geometrischen Maße genutzt werden, um mögliche Fertigungsmaschinen zu identifizieren.

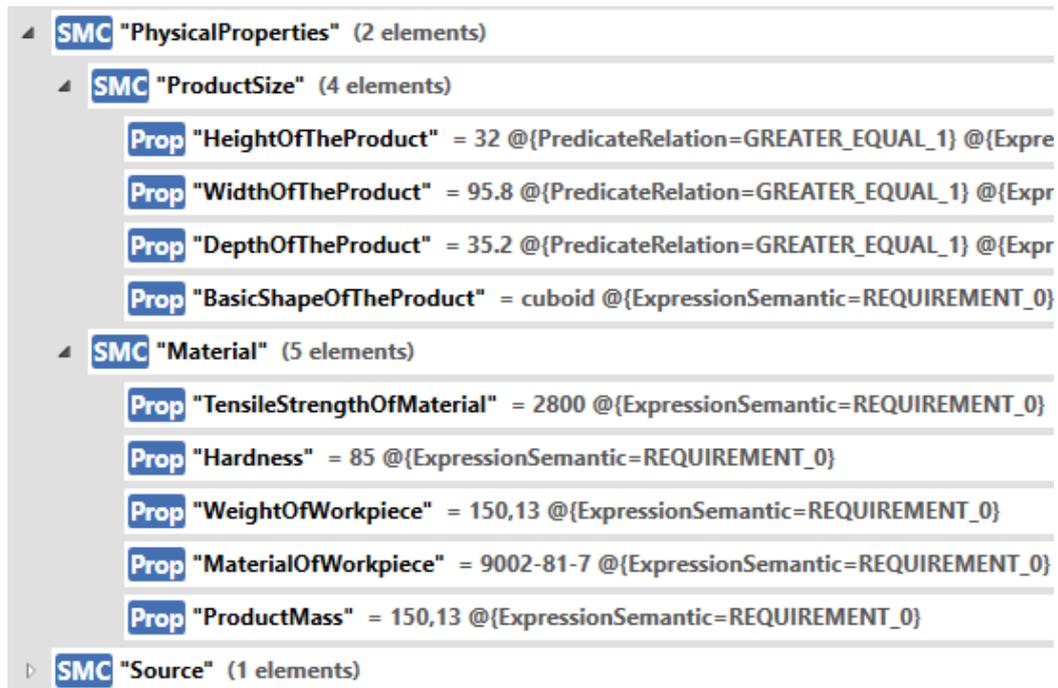


Abbildung 4.5: Material- sowie Geometrieinformationen des Produktes

## 4.2 Teilmodelle für ein Mapping von Services, Capabilities und Skills

In Kapitel 3.3 und 3.4.2 wurde ein VWS-TM für Services, Capabilities und Skills definiert. In Kapitel 3.3.3 wurde auf Literatur verwiesen, die ein Mapping zwischen Features und Capabilities auf Basis von Ontologien ermöglicht. Ebenso wurde in Kapitel 3.3.2 gezeigt, dass eine Capability sich gegebenenfalls aus mehreren Capabilities zusammensetzt oder auf mehrere Skills zur Realisierung referenziert. Dieses Kapitel beschreibt, wie Services für die spanende Bearbeitung modelliert werden können. Des Weiteren wird ein VWS-TM modelliert, das mehrere Capabilities und Skills mit unterschiedlichen Parametern in einem VWS-TM für Capabilities einer Produktionsumgebung verknüpft.

Anhand der in der Produkt-VWS hinterlegten Informationen und des beigefügten CAD-Modells, kann auf benötigte Services zur Fertigung eines Bauteils geschlossen werden. Die benötigten Services können, wie in Abbildung 3.16 gezeigt, in verschiedenen Abstraktionsstufen vorliegen. Die benötigten Services bilden in diesem Fall die Technologiekette ab (siehe Abbildung 3.7). Als Beschreibung von Technologien bietet sich in diesem Fall z. B. die DIN 8580, DIN 8589-2 oder DIN 8589-3 an. Ein VWS-TM zur Anfrage von benötigten Services ist in Abbildung 4.6 dargestellt und wird als *RequestForServices* bezeichnet.

SM	"RequestForServices"	[IRI, https://example.com/ids/sm/5194_3182_1022_3917]
SMC	"RawPartProcurement"	(3 elements)
Ref	"RefRawPart"	~> [Submodel, Local, IRI, https://example.com/ids/sm/9111_3131_8012_7308],[Submodel
Prop	"Norm"	= DIN8580 - 1 PrimaryForming
Prop	"CallForProposalStatus"	= done
SMC	"Milling"	(3 elements)
Ref	"NX_PocketRectangularStraight"	~> [Submodel, Local, IRI, https://example.com/ids/sm/9111_3131_801
Prop	"Norm"	= DIN 8589-3 Milling
Prop	"CallForProposalStatus"	= done
SMC	"Drilling"	(6 elements)
Ref	"NX_Step1Pocket"	~> [Submodel, Local, IRI, https://example.com/ids/sm/9111_3131_8012_7308],[Subr
Ref	"NX_Step1Pocket"	~> [Submodel, Local, IRI, https://example.com/ids/sm/9111_3131_8012_7308],[Subr
Ref	"NX_Step1Pocket"	~> [Submodel, Local, IRI, https://example.com/ids/sm/9111_3131_8012_7308],[Subr
Ref	"NX_Step1Pocket"	~> [Submodel, Local, IRI, https://example.com/ids/sm/9111_3131_8012_7308],[Subr
Prop	"Norm"	= DIN 8589-2 Drilling
Prop	"CallForProposalStatus"	= running

Abbildung 4.6: VWS-TM zur Beschreibung von Serviceanfragen

Die beschriebenen Technologien müssen auf die Services eines Service Providers gematcht werden. Hier wird in Abbildung 4.7 ein VWS-TM für Dienstleistungen in der spanenden Bearbeitung dargestellt.

Zur Definition der Fräs-Dienstleistung wurden die Vorarbeiten aus Kapitel 3.3.2 verwendet und ein Service anhand der ECLASS *MillingWork\_(subcontracting)* spezifiziert. Die *GeneralConditions* beschreiben die Fertigung möglicher Materialien und die minimalen bzw. maximalen Abmaße von Bauteilen. Unter *InputFactors* sind die benötigten Vorgaben gesammelt, um ein Angebot zu erhalten (CAD, Modell, Technische Zeichnung). *Qualifier* geben an, ob die Vorgabe von Daten und Informationen zwingend notwendig oder optional sind. Die Angebotserstellung kann nach gewissen Optimierungen geschehen (energiesparsam, kostengünstig oder schnelle Lieferung). Die *QualitativeData* wird in Fräsprozessen hauptsächlich durch die Toleranzen im CAD oder der Technischen Zeichnung vorgegeben. Weitere Normen, die im Service angeboten werden, wie die ISO 9001 oder IATF 16949 und sich auf Qualitätsansprüche beziehen, sind hier hinterlegt. Die *SMC Quantity* spezifiziert die mögliche Wiederholung von Aufträgen und gibt minimale sowie maximale Losgrößen vor. *TermsOfDeliveryAndPayment* beschreiben Zahlungsbedingungen oder auch *IncoTerms*, die mit einer Auslieferung verbunden sind. Hier kann auch ein Preis der Dienstleistung hinterlegt werden. Da sich die Arbeit auf die Fertigung individueller Produkte fokussiert, wird der Preis erst mit einem entsprechenden Angebot übermittelt. Die *Location* referenziert auf die Adresse der Fertigungsstätte. Die *Capabilities* sind nur intern für ein Un-

ternehmen ersichtlich und referenzieren auf die verfügbaren Capabilities, um einen Service zu realisieren.

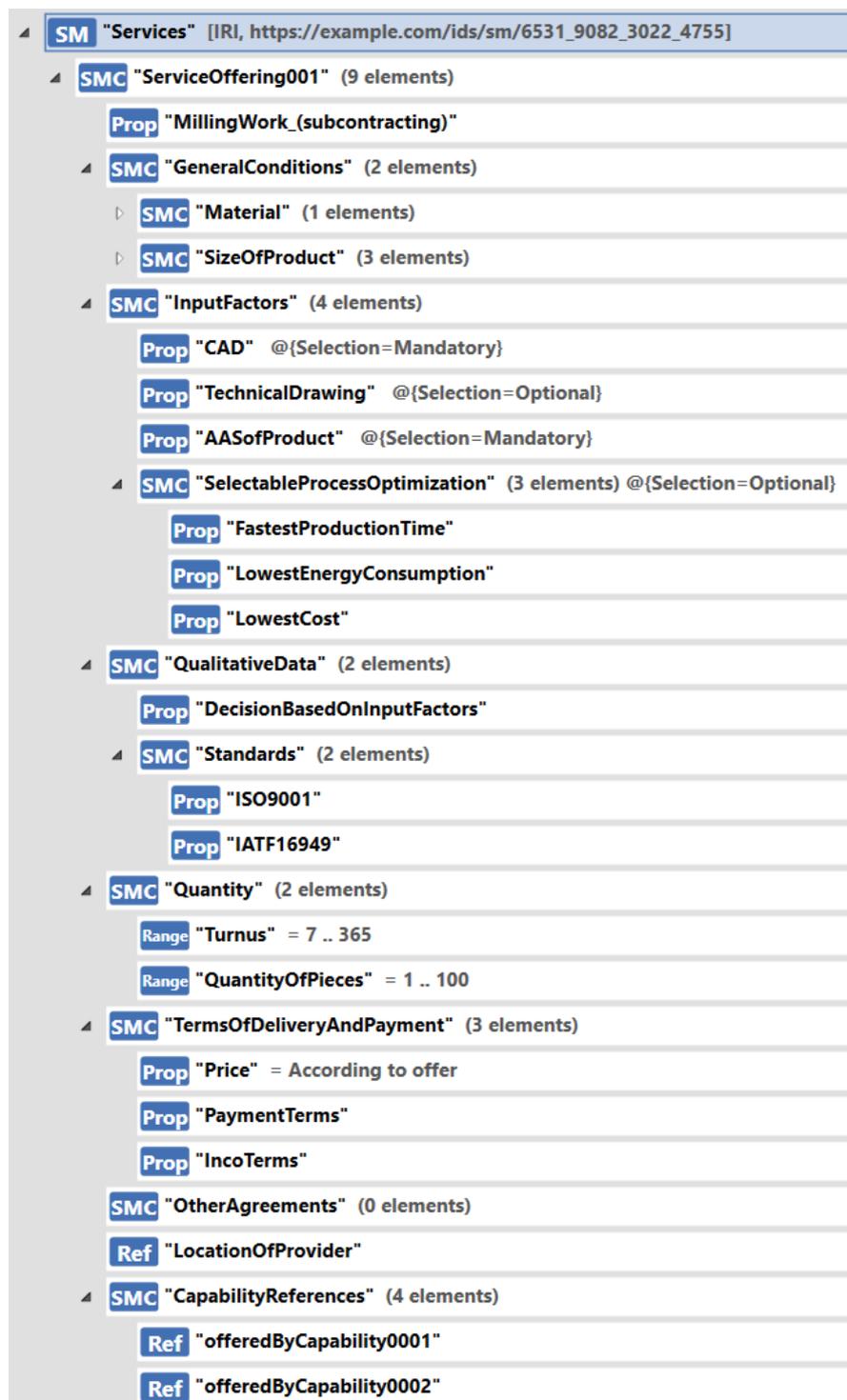


Abbildung 4.7: VWS-TM für die Beschreibung von Services

Abbildung 4.8 stellt das VWS-TM einer Capability dar, um Fräs-Capabilities zu spezifizieren. Der Fokus liegt auf der Komposition von Capabilities und der Referenz auf verfügbare Skills. Zur Verknüpfung mehrerer Capabilities zu einer übergeordneten Capability, wird in der SMC *ComposedOf* eine Relation angelegt, diese ist in Kapitel 3.3.2 dargestellt. Des Weiteren kann eine

Capability in einer Produktionsumgebung evtl. durch mehrere Skills umgesetzt werden (siehe Abbildung 3.19).



Abbildung 4.8: VWS-TM zum Sammeln von Capabilities einer Produktionsumgebung, das auf mögliche Skill-Sequenzen verweist

Jeder Skill, der durch eine Capability realisierbar ist, sollte durch diese referenziert werden. Hierzu wurde in Abbildung 4.8 eine SMC mit der Bezeichnung *AvailableSkills* angelegt. Verschiedene Kombinationen von Skills können genutzt werden, um eine Capability zu realisieren. Jede Skill Kombination, die eine Capability realisiert, wird als *SkillList* bezeichnet. Die *SkillList* kann sich wiederum aus mehreren Skills zusammensetzen. So kann, wie in Abbildung 3.19 gezeigt, die Capability "Gewindebohrung" durch die drei Skills Anbohren, Bohren, Gewinde schneiden (*SkillList\_001*) oder durch den Skill Gewindebohrung (*SkillList\_002*) gefertigt werden. Ein Skill kann evtl. durch mehrere Ressourcen in einer Produktionsumgebung realisiert werden. Der Skill einer Ressource wird durch die *RealizedBy Relation* in der SMC *Ressource* mit der Capability in eine Beziehung gesetzt. Sind in der *SkillList* mehrere Skills gelistet, müssen diese nacheinander ausgeführt werden. Sind in einem Skill-TM mehrere Ressourcen gelistet, gibt es verschiedene CPPM, die den Skill bereitstellen. Um das Modell kompakt darzustellen, werden nicht alle *CapabilityProperties* aufgeführt. Die *CapabilityProperties* beinhalten alle notwendigen Parameter

eines Features, um dieses fertigen zu können (vgl. mit Parametern eines Features in Abbildung 4.4).

Die *Properties* einer Capability beschreiben bestimmte Werte, die eine Capability spezifizieren. Die *Properties* werden evtl. als Parameter für den Skill benötigt. Die Beziehung zwischen einer *Property* und den *InputParameters* eines Skills, werden mit einer *AnnotatedRelationship* verknüpft, die als *PropertyToParameterMapping* bezeichnet wird (siehe Abbildung 4.9). Mit Hilfe der VWS-Elemente, die unterhalb einer *AnnotatedRelationship* hinzugefügt werden können, werden die Beziehungen zwischen *Property* und *Parameter* genauer spezifiziert. So wird durch die *Relation* der Zusammenhang zwischen dem Skill-Parameter und einer *Property* hergestellt. Es kann z. B. spezifiziert werden, ob ein Wert zur Ausführung eines Skills zwingend erforderlich ist oder es sich um einen optionalen Parameter handelt. Es können Rechenoperationen zur Umrechnung von Parametern vorgesehen werden, die durch *Operations* hinterlegt werden. Durch die Einbindung von *Operations* kann somit beispielsweise das Problem unterschiedlicher Feature-Spezifikationen gelöst werden. Unterschiedliche Definitionen von Features, die eine Umrechnung, z. B. von Radius in Durchmesser oder die Transformation von Koordinatensystemen benötigen, kann mit Hilfe der *AnnotatedRelationship* umgesetzt werden. Aus dem Capability-TM, das eine gesamte Produktionsumgebung zusammenfasst, werden somit alle Maschinen ersichtlich, die für die Ausführung eines Fertigungsschrittes in Frage kommen. Die *Properties* einer Capability können des Weiteren auf einen Skill gemappt werden.

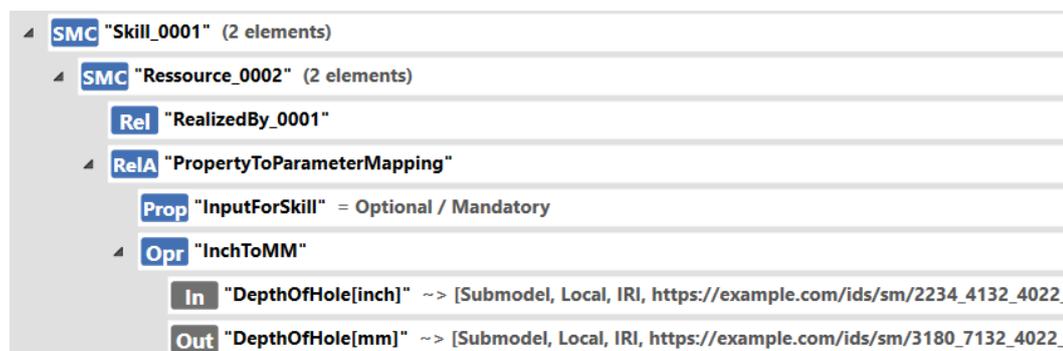


Abbildung 4.9: Verwendung von *AnnotatedRelations* zur Verbindung von *Capability-Properties* und Skill-Parametern

Ist das Matching von Feature-Parametern auf eine Capability durch z. B. Ontologien möglich, können alle weiteren Relationen zwischen Capabilities und Skills durch eine in der Produktionsumgebung existierenden VWS repräsentiert werden. Eine ausführende Software hat somit die Möglichkeit, auf entsprechende Ressourcen zuzugreifen und einen Feasibility-Check oder die Ausführung eines Skills durchzuführen. In diesem Kapitel wurden somit alle VWS-TM spezifiziert, die für ein Matching von einer Produkthanfrage über Services, Capabilities und Skills notwendig sind.

### 4.3 Skillbasierte Ansteuerung eines Fräsroboters

Zur Ansteuerung einer Fräsmaschine mit Skills steht der in Abbildung 4.10 dargestellte KUKA KR 300 Roboter mit einer Frässpindel zur Verfügung. In der Roboterzelle befinden sich in einem Werkzeugmagazin verschiedene Werkzeuge. Zur skillbasierten Ansteuerung wird die Robotersteuerung mit einem Industrie-PC erweitert, um das OPC UA-Nodeset aus Kapitel 3.4.1 zu implementieren.

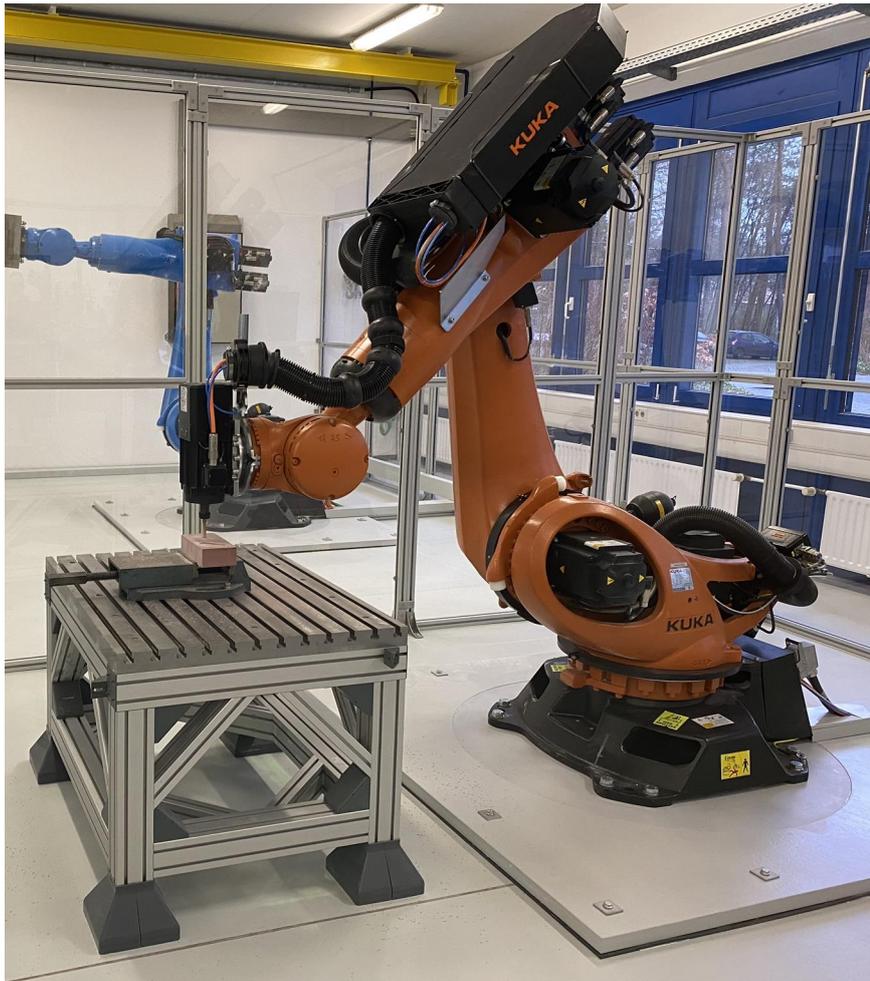


Abbildung 4.10: Roboterzelle mit Frässpindel, die zur Validierung eingesetzt wird

Die vorhandene Hardwareausstattung ist in Abbildung 4.11 dargestellt. Der Roboter wurde zusätzlich mit einem Beckhoff Control Panel ausgestattet. Der zwischengeschaltete Industrie-PC mit Bedienoberfläche dient als übergeordnete Steuerung (*Edge*). Dieser hostet den OPC UA-Server der Roboterzelle im Netzwerk und stellt das Skill-Modell dar. Des Weiteren ist der Industrie-PC eine Art Koordinator, der den entsprechenden Code auf der Robotersteuerung startet oder Entscheidungen zur Werkzeugwahl trifft. Die Kommunikation zur KR C4 Robotersteuerung eines KUKA KR 300 findet über eine EtherCAT Bridge statt. Ein Funktionsblock auf dem Industrie-PC koordiniert die Übermittlung von Informationen zwischen OPC UA-Server und der KR C4.

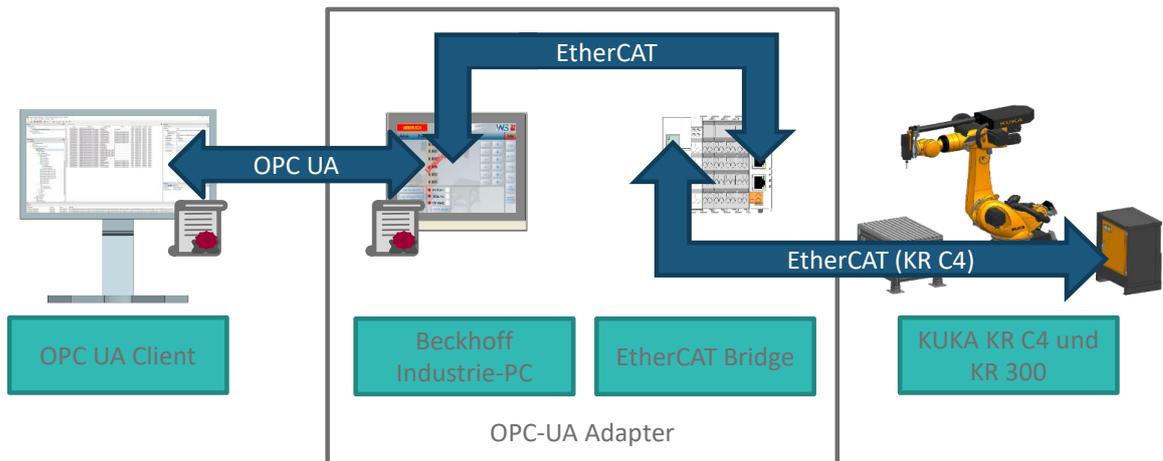


Abbildung 4.11: Hardwareaufbau einer Fräsumgebung zur Ansteuerung mit Hilfe eines OPC UA-Servers in Anlehnung an [Vo20]

Auf der Robotersteuerung wurden Fertigungsabläufe für Rechtecktaschen, Nuten, Bohrungen und Kreistaschen implementiert. Die Werkzeugpfade sind generisch programmiert, sodass unterschiedliche Größen und Orientierungen abgebildet werden können. Je nach Auswahl eines Werkzeuges und der Parameter wird der Werkzeugpfad generiert (siehe Abbildung 4.12).

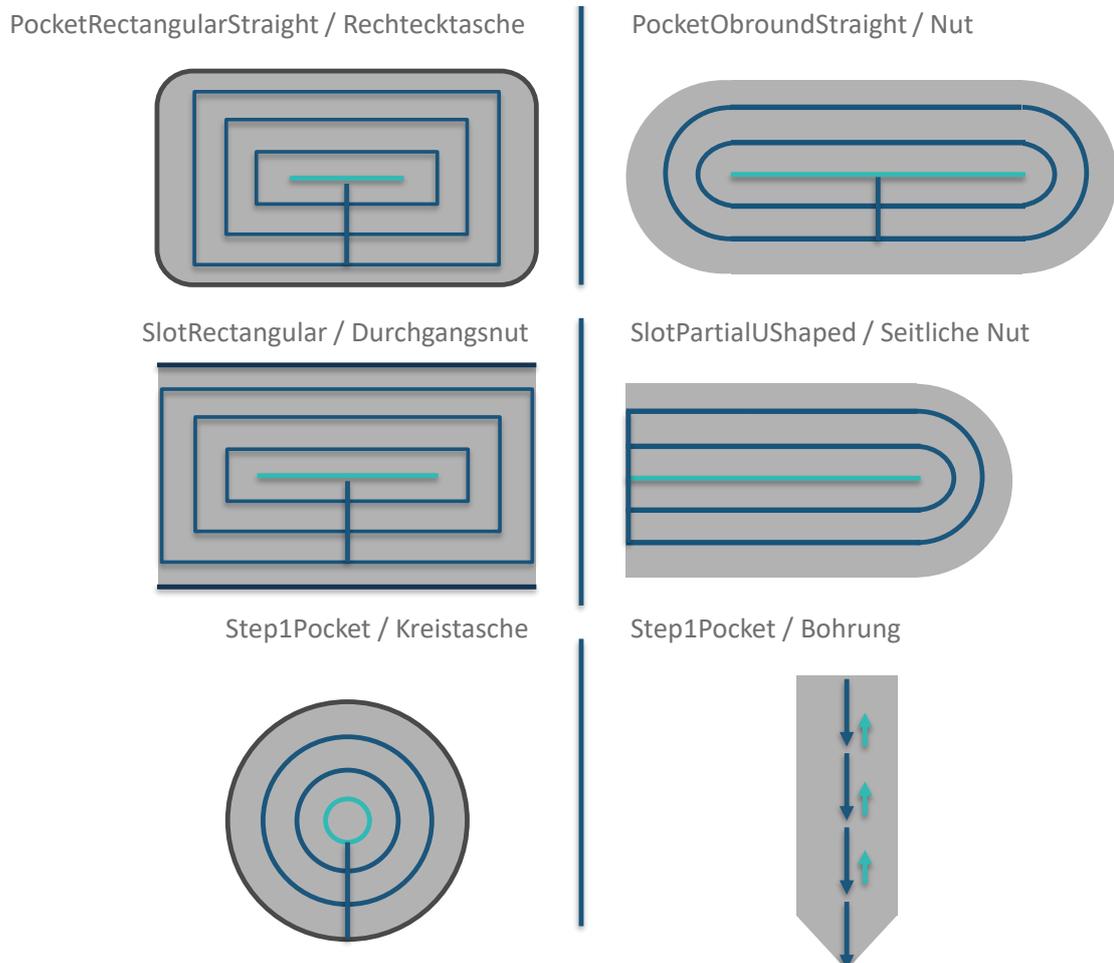


Abbildung 4.12: Übersicht der implementierten Fräskills

In [Vo20] wird die implementierte Anfahrstrategie erläutert, die es ermöglicht von allen Seiten ein Fertigungs-Feature anzufahren. Hierzu wird ein senkrecht zum Feature positionierter Vektor benötigt (in der Regel die Z-Achse). Die Z-Achse des Werkzeugkoordinatensystems, muss entsprechend positioniert werden, sodass die Vektoren übereinander liegen. Auf einer Kuppel, die anhand der Bauteilgröße berechnet wird, positioniert sich der Roboter zum Anfahrpunkt und -vektor (siehe Abbildung 4.13). Kollisionen können somit für Feature, die auf der nach oben ausgerichteten Fläche des Werkstückes liegen, vermieden werden. Ist der Anfahrvektor geneigt, kann das Anfahren nur durch Zustimmung eines Werkers auf dem Control Panel geschehen. Des Weiteren ist je nach Anfahrt einer Außenfläche die Rotationsrichtung der Frässpindel zu definieren, um eine Kollision mit dem Roboterarm und Werkstück zu vermeiden.

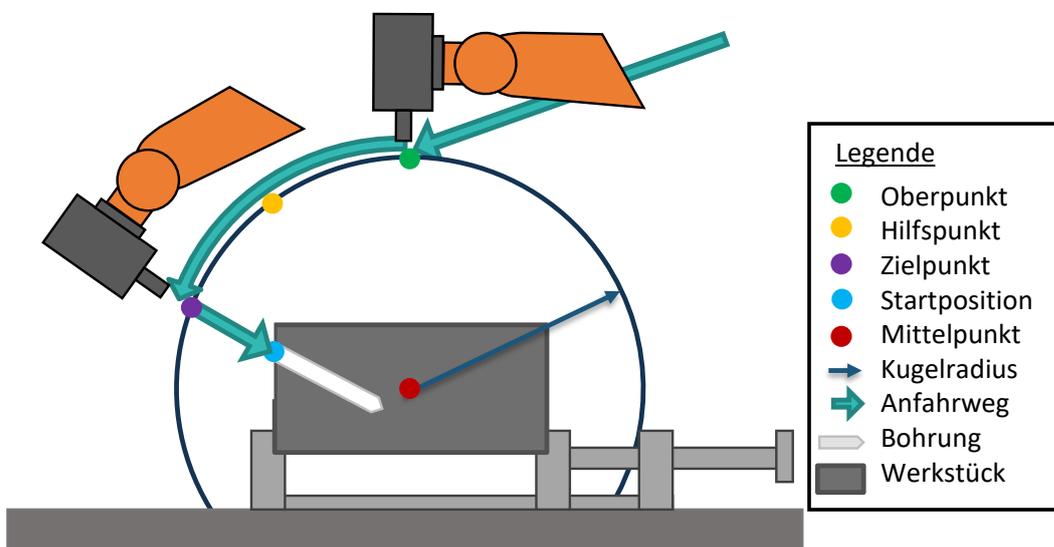
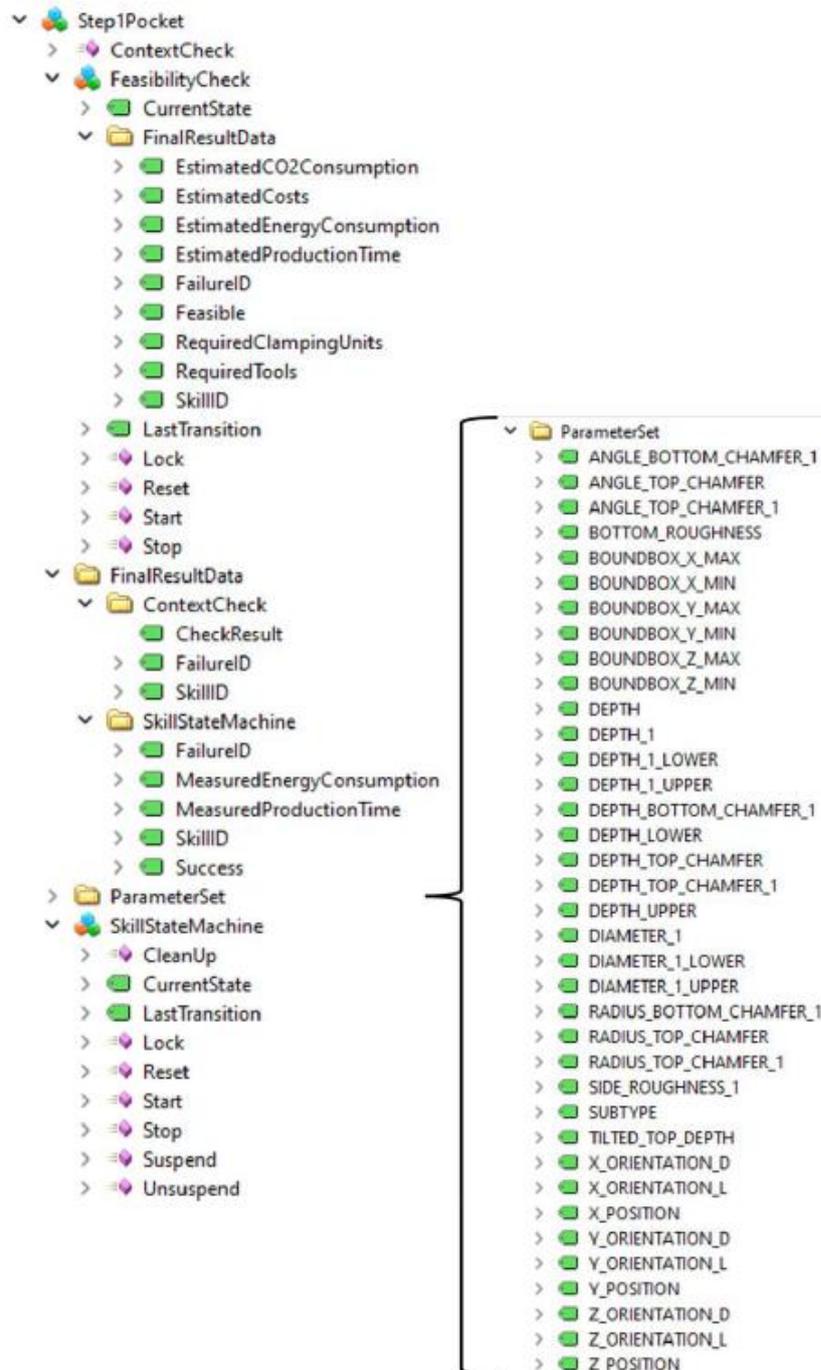


Abbildung 4.13: Anfahrstrategie des Roboters zu einem Feature zur Kollisionsvermeidung in Anlehnung an [Vo20]

Die Skills werden über den OPC UA-Server im Netzwerk über den Industrie-PC dargestellt. Der OPC UA-Server enthält die in Kapitel 3.4.1 vorgestellte *StateMachine* und ein *ParameterSet*. Über das *ParameterSet* können alle benötigten Parameter zur Durchführung eines Skills übertragen werden. Das Objekt *FinalResultData* wurde zusätzlich für Ergebnisse des Kontext-Checks sowie Informationen der Skillausführung untergliedert. Der in Kapitel 3.4.1 vorgestellte *Lock* wurde vernachlässigt, da in der Implementierung nur ein Client auf den Server zugreift. Der *Lock* Mechanismus kann auch vernachlässigt werden, wenn davon ausgegangen wird, dass nur eine Softwarekomponente einen Skill auf einem CPPM ausführen darf und ein zusätzliches *ParameterSet* im *FeasibilityCheck* angelegt wird. Die Einschränkung von Nutzergruppen auf bestimmte Nodes sowie die maximale Anzahl von Nutzern einer Gruppe kann auf dem Server definiert werden. So könnten Clients zur Planung bspw. nur auf den *FeasibilityCheck* zugreifen und Clients zur Ausführung nur auf die ausführenden Methoden des Skills. Der OPC UA-Skill einer *Step1Pocket* ist in Abbildung 4.14 dargestellt.

Abbildung 4.14: OPC UA-Struktur des *STEP1Pocket* Skills

Die Benennung der Skills sowie die Bezeichnung der Skill-Parameter wurde identisch zu den NX-Features vorgenommen. So wird für das Matching der einfachste Fall der skillbasierten Fertigung betrachtet (vergleiche Kapitel 3.3.3). Durch die identische Definition von Features, Capabilities und Skills sowie der Parameter und Properties können die Produkthanforderungen direkt auf benötigte Capabilities und Skills gematcht werden. Durch die einheitliche Bezeichnung konnte das Matching vereinfacht werden. Die Features werden somit direkt auf einen vorhandenen Skill gematcht.

In Kapitel 4.1 wird das Anlegen einer Fertigungsreihenfolge der Features erläutert. Ist dieser Schritt im CAD Plug-in abgeschlossen, kann über den Button *DirectConnect* ein neues Fenster geöffnet werden (siehe Kapitel 4.1, Abbildung 4.2). Das sich öffnende Fenster ist in Abbildung 4.15 dargestellt. Der Anwender kann in diesem einen Feasibility-Check, mit allen im Plug-in hinterlegten Maschinen, durchführen.

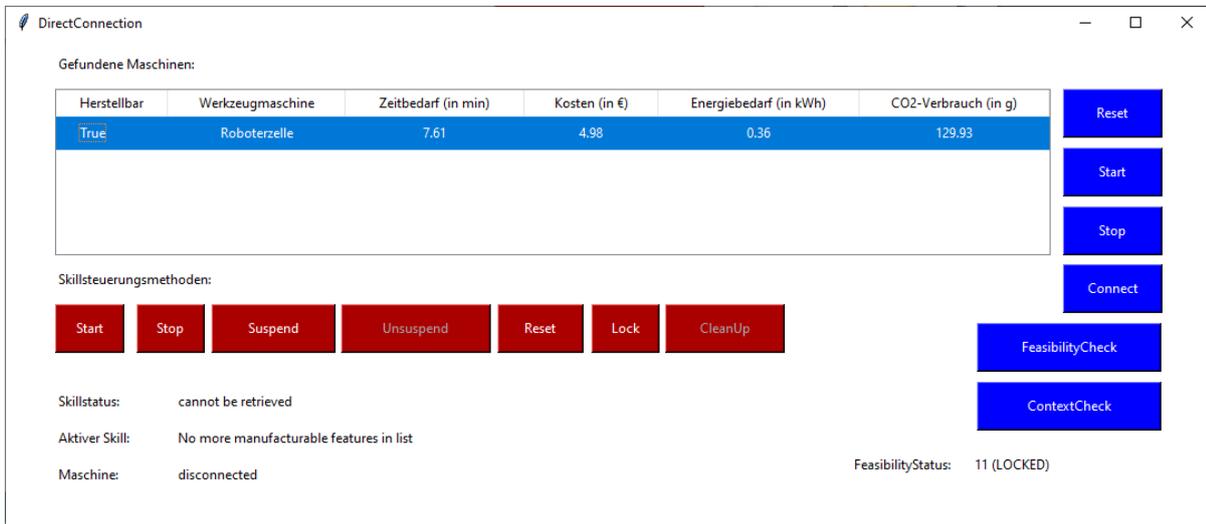


Abbildung 4.15: Zweite Seite der GUI zum Starten des Feasibility-Checks. Nach durchgeführten Feasibility-Check werden die möglichen Fertigungsmaschinen angezeigt

Zur Durchführung eines Feasibility-Checks wird mit allen verfügbaren Maschinen eine Verbindung hergestellt und für die in der Fertigungsreihenfolge vorgegebenen Skills jeweils ein Feasibility-Check durchgeführt. Hierzu verbindet sich ein OPC UA-Client mit den vorgegebenen CPPM. Der Client sucht anhand des Namens eines Features bzw. der Capability, die in einer Liste gespeichert werden, nach einem identischen benannten Skill auf dem OPC UA-Server des CPPM. Ist der Skill vorhanden, wird ein Feasibility-Check getriggert. Dieser Vorgang wird wiederholt, bis für jeden notwendigen Skill ein Feasibility-Check durchgeführt wurde. Durch den Feasibility-Check wird für jeden Skill der jeweilige Datensatz mit den folgenden Informationen generiert: Fertigbarkeit des Skills, Zeitbedarf, Kosten, Energiebedarf und CO<sub>2</sub>-Verbrauch. Über den *Node FinalResultData* des Feasibility-Checks werden die Daten an den Clienten zurückgegeben. Im Feasibility-Check werden einige rudimentäre Prüfungen durchgeführt. Zunächst wird geprüft, ob ein Schaftfräser oder ein Bohrer für die Durchführung des angefragten Fertigungsschrittes vorhanden ist. Im Falle einer Bohrung muss z. B. der Bohrer mit dem exakten Durchmesser und einer ausreichenden Werkzeuglänge vorliegen. Mit Hilfe der Drehzahl ( $n$ ), Anzahl der Werkzeugschneiden ( $z$ ), dem Zahnvorschub ( $f_z$ ) sowie des maximal zulässigen Vorschubes eines Werkzeugs, kann der Vorschub berechnet werden. Die Parameter für die Schnittgeschwindigkeit ( $v_c$ ) und die Vorschubgeschwindigkeit ( $v_f$ ) sind auf dem Industrie-PC, in Abhängigkeit des Werkstückmaterials fix hinterlegt und orientieren sich an Bearbeitungstabellen aus [Go20]. Anhand des vorhandenen Materials, den hinterlegten Werkzeugparametern und dem Werkzeugdurchmesser ( $d$ ) wird die Drehzahl ( $n$ ) des Werkzeugs bestimmt [Go20]:

$$n = \frac{v_c}{\pi \cdot d} \quad (4.1)$$

Mit der berechneten Drehzahl, Anzahl der Werkzeugschneiden und dem Zahnvorschub kann der zu verwendende Vorschub berechnet werden [Go20]:

$$v_f = n \cdot z \cdot f_z \quad (4.2)$$

Werden Werkzeug- und Parametertabellen in Zukunft ausführlicher mit VWS beschrieben, könnte das System ebenfalls allgemeingültig für Werkzeuge umgesetzt werden. In diesem Fall wurden die benötigten Parameter auf dem Industrie-PC hinterlegt.

Je nach zu fertigendem Feature wird anhand des Werkzeugs, die zurückzulegende Strecke ( $s_{\text{Werkzeug}}$ ), Anzahl Zustellungen ( $Z$ ) und mit Hilfe des Vorschubes ( $v_f$ ) die benötigte Zeit für einen Fertigungsschritt kalkuliert ( $t_{\text{Werkzeugweg}}$ ). Hinzu werden die Zeiten für das Anfahren bzw. Abfahren ( $t_{\text{fahrt}}$ ), die Werkzeugwechselfdauer ( $t_{\text{WZW}}$ ) und die Verweilzeit ( $t_{\text{verw}}$ ) nach einem Skill kalkuliert. Die benötigte Zeit zur Ausführung eines Skills ( $t_{\text{skill}}$ ) wird folgendermaßen berechnet:

$$t_{\text{skill}} = \frac{s_{\text{Werkzeug}} \cdot Z}{v_f \cdot 60} + \frac{t_{\text{fahrt}} + t_{\text{WZW}} + t_{\text{verw}}}{60} \quad (4.3)$$

Mit Hilfe der Dauer und einem geschätzten Energiebedarfs kann der Energieverbrauch ( $W$ ) und die CO<sub>2</sub>-Emission ( $E_{\text{CO}_2}$ ) kalkuliert werden. Der Leistungsbedarf des Roboters wurde mit 2,8 kW angenommen:

$$W = 2,8 \text{ kW} \cdot t \quad (4.4)$$

Laut [UmweltBundesamt] wird pro Kilowattstunde im deutschen Strommix 366 g CO<sub>2</sub> emittiert. Somit ergibt sich für die voraussichtliche CO<sub>2</sub>-Emission:

$$E_{\text{CO}_2} = W \cdot 366 \frac{\text{g}}{\text{kWh}} \quad (4.5)$$

Die Kosten ( $K$ ) basieren vereinfacht auf der Summe von Energieverbrauch ( $W$ ) und Stromkosten ( $SK$ ) sowie einem konstanten Wert für die Fertigungskosten ( $FK$ ) aller Skills. In diesen sollen alle Kosten wie Verbrauch von Betriebsstoffen, Verschleiß usw. mit einfließen. Ein Gewinnzusatz ( $GM$ ) von 15 % wird auf die Kosten erhoben:

$$K = \sum (W \cdot SK + FK) \cdot (1 + GM) \quad (4.6)$$

Die Ergebnisse für den finalen Energiebedarf, Kosten, Zeit und die CO<sub>2</sub>-Emission werden dem User in der Bedienoberfläche des CAD Plug-in angezeigt (siehe Abbildung 4.15). Ist eines der Features auf der Maschine nicht vorhanden, wird dieses rot hinterlegt. Ist der Feasibility-Check *false*, wird das Feature orange hinterlegt. Wird ein *Timeout* im Feasibility-Check erreicht, wird das Feature gelb markiert.

Wird die gewünschte Maschine zur Fertigung ausgewählt, kann der Bediener diese über den *Reset* und *Start*-Button die Fertigung aller gewählten Feature beginnen. Der OPC UA-Client verbindet sich mit der ausgewählten Maschine. Anhand des Featurenamens wird der zugehörige Skill im OPC UA-Nodeset des CPPM gesucht. Die Variablen werden in das Parameterset geschrieben und der Skill getriggert. Während des Fertigungsprozesses kann der Bediener über die Nutzeroberfläche mit dem CPPM interagieren und bspw. den Prozess anhalten und wieder fortsetzen. Zwischen jedem Skill fährt der Roboter wieder auf den in Abbildung 4.13 dargestellten Oberpunkt zurück. Der Roboter verweilt dort für eine definierte Zeit, bis der nächste Skill getriggert wird. Folgt kein weiterer Skillaufruf, fährt der Roboter in die Ausgangslage zurück. Die aktuellen Skillzustände werden ständig auf der Bedieneroberfläche angezeigt.

Für die Konstruktion und die Fertigung von rechteckigen Bauteilen wurde die Regel definiert, dass diese in positiver X-, Y-, Z-Richtung vom Ursprung des globalen Koordinatensystems im CAD konstruiert werden (siehe Abbildung 4.16). So ist in der Fertigung immer der Bezug zum Ursprung des Bauteils möglich. Dies ist wichtig, um den Nullpunkt eines eingespannten Bauteils vor der Fertigung zu ermitteln und gegebenenfalls Koordinatentransformationen während der Fertigung durchführen zu können. Zum Einmessen des Bauteils wurde ein Skill implementiert, der mit Hilfe eines Messtasters das Bauteil aus X-, Y- und Z-Richtung einmisst. Die Fertigung kann anschließend gestartet werden. Ein Bauteil mit einer Tasche und 4 Bohrungen (Abbildung 4.16) sowie ein Bauteil, das alle implementierten Fertigungsskills enthält (Abbildung 4.17) wurde skillbasiert gefertigt.

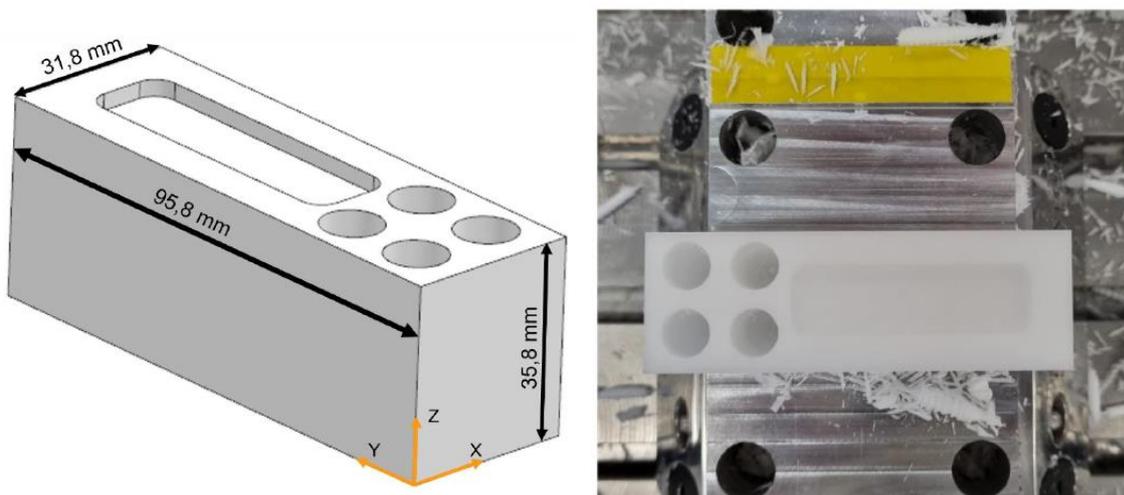


Abbildung 4.16: Gefertigtes Bauteil 1

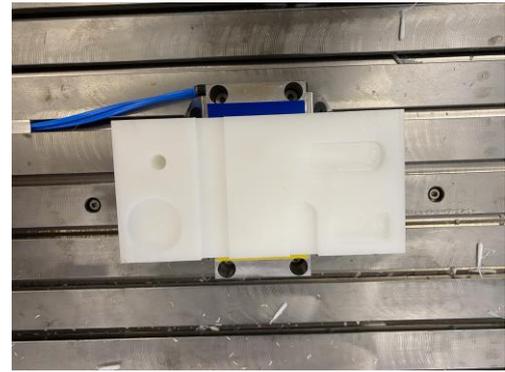
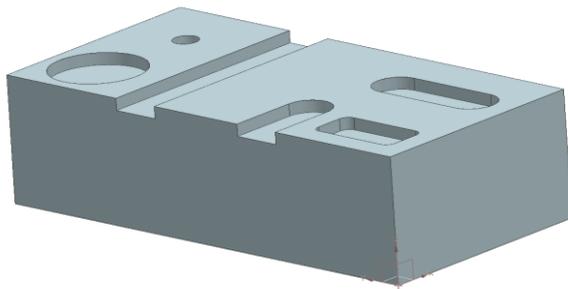


Abbildung 4.17: Gefertigtes Bauteil 2

Der Fertigungsablauf wurde im Video [[@SF-Live](#); [@YouTube](#)] aufgezeichnet. Die Implementierung und Fertigung haben gezeigt, dass die Produktion auf Basis einer Produktbeschreibung mit Features und die Fertigung durch Skills möglich ist. Vorteile in dem Verfahren ergeben sich vorwiegend in der Fertigung von kleinen Losgrößen. Für neue Bauteile entfällt die erneute Programmierung an der Maschine. Einige Einschränkungen sind in der skillbasierten Fertigung zum aktuellen Implementierungsstand vorhanden, die in Zukunft noch verbessert werden können. Die Außenkontur des Rohteils wurde auf Maß vorbereitet. Bei Fräsprozessen ist zunächst eine Rohteilbearbeitung von allen Seiten des Rohteils notwendig, um die gewünschten Außenmaße und Oberflächengenauigkeiten des Bauteils zu erzeugen. Hierzu können Flächenfeatures aus dem CAD einbezogen werden. Die Bearbeitungsdauer in einer Maschine wird mit dem hier implementierten skillbasierten Ansatz zunehmen, da jeder Skill einzeln angesteuert wird. Prozesse, die etwa die gleichen Werkzeuge benötigen, werden bisher nicht zusammengeführt. Ein Algorithmus, der die Fertigungsreihenfolge optimiert, oder die Anpassung der Skill-Architektur können eine Lösung für die längere Fertigungszeit sein. Einige der Features aus der Featureerkennung sind nicht fertigbar. Wird z. B. eine Tasche mit einer freien Kontur konstruiert, zwei Rechtecktaschen übereinandergelegt oder die Außenwände einer Tasche angeschrägt, werden diese als *GeneralPocket* identifiziert. Eine *GeneralPocket* gibt keine Informationen über die exakte Kontur einer Tasche zurück. Zur Fertigbarkeit müsste das programmierte CAD Plug-in erweitert werden und relevante geometrische Informationen ausgeben. Komplexere Formen wie Freiformflächen sind in NX nicht als Feature identifizierbar. Im Stand der Technik wurde gezeigt, dass hierfür bereits Lösungen entwickelt wurden.

Das vorgestellte Konzept könnte mit einer *Skill-ID* erweitert werden. Dabei würden im Feasibility-Check alle benötigten Parameter in das *ParameterSet* geschrieben und der Feasibility-Check entsprechend ausgeführt werden. Im *FinalResultData* würde zusätzlich eine *Skill-ID* zurückgegeben werden. Unter dieser *Skill-ID* speichert das CPPM das individuell auszuführende Programm oder individuelle Parameter, die ein Programm beeinflussen, ab. Wird das CPPM für den entsprechenden Prozess benötigt, muss für den Kontext-Check sowie das Starten des Skills, die *Skill-ID* anstatt eines gesamten Parametersets vorgegeben werden. Hierzu wäre

allerdings zusätzlich noch eine Erweiterung notwendig, um nicht benötigte Programme bzw. eine *Skill-ID* löschen zu können. Das Konzept der *Skill-ID* wurde in Abbildung 4.18 visualisiert.

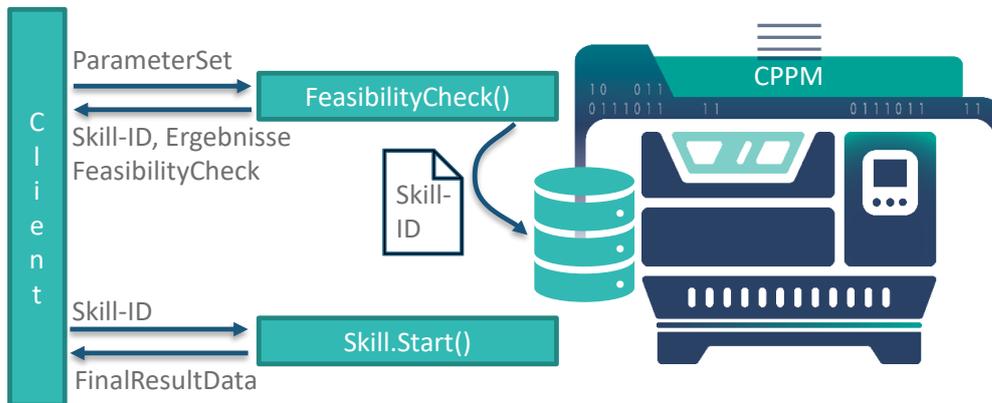


Abbildung 4.18: Einbindung einer *Skill-ID* in den Feasibility-Check und Skillaufruf in Anlehnung an [Vo21a]

## 5 Fazit und Ausblick

### 5.1 Zusammenfassung

Die Arbeit zeigt einen Lösungsansatz für die individuelle spanende Bearbeitung durch die Umsetzung von herstellerunabhängigen Technologien und einer definierten Semantik. Ein Lösungskonzept für einen Marktplatz inklusive der Anbindung von Unternehmen wurde auf Basis von RAMI 4.0 und vorhandenen Technologien aufgezeigt. Zur Beschreibung der Unternehmensfunktionen und der Ansteuerung der Feldebene wurden Services, Capabilities und Skills definiert. Hierzu ist zunächst die Extraktion von Informationen aus CAD-Systemen und die Beschreibung von Produktinformationen auf Basis der VWS wichtig. Entsprechend wurden vorhandene Standards und CAD-Systeme auf die Extraktion von geometrischen Funktionen untersucht. Eine VWS für das Produkt mit den relevanten Informationen wurde definiert. Um die Produktinformationen für das Matching eines geeigneten Produktionsprozesses bereitzustellen, wurden Dienstleistungs- und Capability-Beschreibungen als VWS-TM angefertigt. Mögliche Ansätze zum Matching mit Ontologien wurden diskutiert, bilden allerdings nicht den Fokus dieser Arbeit. Im Anschluss wurde eine Modulschnittstelle auf Basis von OPC UA konzeptioniert und somit die flexible Ansteuerung der Ressourcen über Skills ermöglicht. Die Konzepte wurden im Anschluss validiert und zwei verschiedene Musterteile im CAD konstruiert und gefertigt. Es konnte gezeigt werden, dass auf Basis des vorliegenden CAD-Objektes, und der Zuordnung von Features, Services, Capabilities und Skills die individuelle Fertigung mit reduziertem Programmieraufwand möglich wird.

### 5.2 Fazit

Das folgende Fazit ergibt sich zu den gestellten Forschungsfragen. Zu Forschungsfrage 1:

*Wie kann eine herstellerunabhängige RAMI 4.0 Produktionsbeschreibung umgesetzt werden, um marktplatzbasierte Verfahren in der spanenden Bearbeitung zu ermöglichen?*

Das Ziel war die Erstellung einer Produkt-Verwaltungsschale zur Beschreibung aller relevanten Produktinformationen. Mit dieser soll eine automatisierte Prozessplanung für individuelle Aufträge durchführbar werden.

Ein Template einer Produkt-VWS wurde in Kapitel 3.2 erarbeitet. Ein Produkt, das durch Fräsbearbeitung weiterverarbeitet wird, kann durch Features beschrieben werden. Die erarbeitete Produkt-VWS ermöglicht, benötigte Technologien zur Fertigung des Produktes zu identifizieren. Eine Herausforderung bleiben die vielen unterschiedlichen Featuredefinitionen. Ein einheitlicher Standard wird sich für die Beschreibung der Produktfeatures kaum durchsetzen, vor allem da einheitliche Standards wie STEP in der Praxis in der Datenübertragung zu Informationsverlusten führen. Die erarbeitete Produkt-VWS ermöglicht die Beschreibung von Features durch unterschiedliche Featurestandards. Ob eine existierende Featurebeschreibung alle benötigten Parameter zur Fertigung enthält, muss separat geprüft werden. Es existieren bspw. die in Kapitel 3.3.3 betrachteten Ontologien, die zur Produktbeschreibung von Features verwendet werden können. Allerdings müssen diese durch Positionsparameter und Bearbeitungsvektoren erweitert werden, sodass die Fertigung der Features möglich ist. In Kapitel 4.1 wurde gezeigt, dass mit Hilfe einer automatisierten Featureextraktion relevante Parameter der Produkt-VWS aus CAD-Objekten automatisiert ausgelesen werden können. Hierzu wurde die API der Software NX verwendet. Die implementierte Umsetzung ist somit auf die Standardfeatures von NX eingeschränkt. Komplexe Freiformflächen können beispielsweise nicht erkannt werden. Auch existieren vereinzelt nicht fertigbare Features, wie die *GeneralPocket*. Die automatisierte Generierung einer Produkt-VWS aus dem CAD heraus, wurde innerhalb dieser Arbeit nicht implementiert. Es wurde allerdings gezeigt, dass relevante Fertigungsparameter auslesbar sind, und ein entsprechendes VWS-TM für eine Produkt-VWS entwickelt. Somit können alle fertigungsrelevanten Informationen in dem TM hinterlegt werden. Die Feature-Parameter wurden innerhalb des implementierten CAD Plug-in weiterverarbeitet, um im Kapitel 4.3 die Fertigung mit Skills zu zeigen.

Die existierenden Marktplätze (siehe Kapitel 2.5) basieren ebenso auf einer Analyse von CAD-Dateien. Es ist davon auszugehen, dass im Hintergrund ähnliche Features analysiert werden, um eine Kostenkalkulation zu erstellen. Die dort verwendeten Tools zur Extraktion von Features bieten unter Umständen mehr Funktionen als die API der betrachteten CAD-/CAM-Software.

Zu Forschungsfrage 2:

*Wie können Dienstleistungen und Fähigkeiten von Unternehmen in der spanenden Bearbeitung herstellerübergreifend beschrieben werden, sodass die Identifikation eines Produktionsablaufes möglich wird?*

Ziel war die Definition von Verwaltungsschale-Teilmodellen, die eine Dienstleistungs- und Fähigkeitsbeschreibung standardisieren. Somit können geeignete Unternehmen zur Fertigung kontaktiert und eine Prozessplanung anhand der Produkt-Verwaltungsschale stattfinden.

Ein VWS-TM für Dienstleistungen wurde in Kapitel 3.3 erarbeitet und gezeigt, dass mögliche Dienstleistungen mit ECLASS oder bestehenden Normen auf einem bestimmten Abstraktions-

level beschrieben werden können. In welcher Abstraktionsstufe Dienstleistungen beschrieben werden, ist vom Typ der Anfrage und dem Anwendungsfall abhängig. Zunächst ist davon auszugehen, dass Lohnfertiger eine allgemeine Prozessbeschreibung ihrer Dienstleistung angeben, z. B. Zerspanung oder Fräsen. Spezifischere Beschreibungen von Fertigungsschritten sind denkbar, erfordern allerdings eine Standardisierung, um den Lösungsraum für die Dienstleistungssuche zu begrenzen. Parameter, wie die verarbeitbaren Materialien, geometrische Einschränkungen, Fertigungszeiten, Zahlungsinformationen oder Zertifikate und Normen, spezifizieren eine Dienstleistungsbeschreibung (siehe Kapitel 4.2). Anhand der geometrischen Features, des Materials und der geometrischen Form eines Bauteils sowie mögliche zusätzliche Eingaben eines Kunden, ist es möglich einen benötigten Prozess wie Drehen, Fräsen oder Bohren zu identifizieren. Hier wird auch auf die in Kapitel 2.4 analysierten Marktplätze verwiesen, die diese Analyse von Bauteilen bereits automatisiert haben. Um einen möglichen Fertigungsweg für ein individuell angefragtes Produkt zu spezifizieren, können im Unternehmen Capabilities definiert werden. Zur Definition von Capabilities kommen verschiedene Standards in Frage. Es wurden prozess- und produktspezifische Beschreibungsarten für die Capability betrachtet. Am sinnvollsten wurde eine produktspezifische Beschreibung von Capabilities für die spanende Bearbeitung erachtet. Somit sollten sich Capabilities an den Features der Produktbeschreibung orientieren. Das Mapping eines Features auf eine Capability kann durch eine Ontologie geschehen. Existierende Lösungsansätze wurden in Kapitel 3.3.3 beschrieben. Ein VWS-TM für die Beschreibung einer Capability wurde erstellt. Die Abbildung von Kompositionen mehrerer Capabilities wurde gezeigt (Kapitel 3.3). In Kapitel 4.2 wurde die Referenz einer VWS auf Skills zur Fertigung definiert. Es wurde beachtet, dass sowohl unterschiedliche Fertigungsreihenfolgen als auch Hierarchiestufen von Skills abgebildet werden können. Das Konzept wurde möglichst generisch umgesetzt, sodass Unternehmen die Möglichkeiten besitzen, ihre eigenen Skills zu definieren und zu nutzen. So sollen die Vorteile und das Know-how von Unternehmen in der Implementierung und Automatisierung erhalten und geschützt bleiben. Das Modell ermöglicht keinen parallelisierten Ablauf von Skills. Im Falle dieser Ausarbeitung und Implementierung war ein paralleler Ablauf von Skills nicht erforderlich. Aktuell wird ein standardisiertes Capability-Modell aktiv in der IDTA diskutiert [@IDTA] und die erarbeiteten Lösungen aus Kapitel 3.3.2 und 4.2 in die Standardisierung eingebracht.

Zu Forschungsfrage 3:

*Wie sieht eine herstellerunabhängige Maschinenschnittstelle und deren Beschreibung aus, um individuell geplante Produktionsabläufe für Fräsaufträge zu fertigen?*

Ziel war die Definition einer standardisierten Schnittstelle zur individuellen und herstellerunabhängigen Einbindung von Produktionsressourcen in die Produktionsplanung und deren flexible Ansteuerung.

Als herstellerunabhängiges Kommunikationsframework für CPPM wurde OPC UA gewählt. Die Anforderungen an ein OPC UA-Nodeset wurden in Kapitel 3.4.1 herausgearbeitet und dies in Kapitel 3.4.2 zusätzlich mit einem VWS-TM beschrieben. Das OPC UA-Nodeset wurde erarbeitet und auf einer Industriesteuerung implementiert (siehe Kapitel 4.3). Von der Extraktion der Features bis zur Fertigung wurde gezeigt, dass die skillbasierte Bearbeitung für Standardfeature möglich ist. Das Matching zwischen Features, Capabilities und Skills wurde in der Implementierung vereinfacht durchgeführt (siehe Kapitel 4.3). Die Bezeichnung des Features und der Feature-Parameter sowie der Capabilities, Skills und der Skill-Parameter wurden identisch gewählt. Somit ist ein direktes Matching zwischen benötigten Features und vorhandenen Skills möglich. Um Entscheidungen für die Fertigbarkeit der angefragten Skills zu ermöglichen, wurde ein Feasibility-Check implementiert. Die Feasibility-Checks wurden bisher umgesetzt, können jedoch noch optimiert werden, um genauere Kalkulationen zu ermöglichen. Für den Energieverbrauch wurden grobe Schätzwerte angenommen. Die Werkzeugauswahl kann generischer programmiert und noch optimiert werden. Es können noch keine Aussagen über die Fertigbarkeit einer gewünschten Qualität getroffen werden und eine Kollisionsprüfung fehlt. Des Weiteren fehlt eine Prüfung, ob das benötigte Rohmaterial eingespannt werden kann. Zur Kostenkalkulation dienen Schätzwerte als Grundlage. Für die automatisierte Fertigung von kleinen Losgrößen wird der Feasibility-Check ein wichtiger Faktor, um genaue Aussagen über die Fertigbarkeit eines Bauteils vorab automatisiert zu ermöglichen. Es besteht in diesem Themengebiet weiterhin Forschungsbedarf. Die Grenzen der Fertigbarkeit durch Standardfeatures wurden erläutert. Die Definitionen von Features müssen erweitert werden, um eine größere Vielfalt abdecken zu können bzw. die Systeme zur Extraktion von Features optimiert werden. Die Prozesse einer skillbasierten Ansteuerung dauern länger, da Feature sequenziell gefertigt werden und mögliche ähnliche Skills evtl. nicht kombiniert werden. Würde z. B. ein Gewindebohrungs-Skill mit identischen Abmaßen mehrfach gefertigt werden, wird innerhalb jedes Skills ein oder mehrere Werkzeugwechsel durchgeführt. Werkzeugwechsel könnten minimiert werden, wenn die Maschine z. B. vorab eine Planung über alle benötigten Skills vornehmen würde oder es könnten Skills eingeführt werden, die z. B. Muster fertigen. Die Reihenfolge der Fertigungsschritte wird bisher durch einen Bediener vorgenommen. Auch dieser Prozess könnte anhand der Position der Features automatisiert werden. In der *Industrial Digital Twin Association e. V.* (IDTA) wurde an einem TM *Control Component* gearbeitet, das genutzt werden kann, um verschiedenen Ausführungsschnittstellen, z. B. Skills zu beschreiben [@IDTA]. Dieses bezieht sich vorerst auf die Beschreibung des Skills, seine Modi, Parameter und mögliche Fehlerfälle. Ergänzende Nodesets wie die Beschreibung der Schnittstelle stehen noch aus.

### 5.3 Ausblick

Zusammenfassend wurde gezeigt, dass Bauteile, die durch Features vollständig beschreibbar sind, innerhalb einer marktplatzbasierten Produktionsarchitektur gefertigt werden können. Un-

ternehmen müssen hierfür ihre Dienstleistung anhand von Unternehmens-Verwaltungsschalen in einem vertrauten Datenraum oder einem Dienstleister zur Verfügung stellen. Zur Kommunikation zwischen Unternehmen eignen sich Gaia-X-Konnektoren. Die Konnektoren und das Gaia-X-Netzwerk befindet sich zurzeit in der Entwicklung. Aufbauend auf den Ergebnissen der Dissertation wurden hierzu bereits weitere Ausarbeitungen des Instituts veröffentlicht [Ju23; Si23a; Si23b; Vo24]. Die Erzeugung dynamischer Lieferketten und der digitale Vertragsabschluss muss entwickelt werden, um das beschriebene Szenario umzusetzen. Handelt es sich um die Bestellung fix definierter Produkte, können die Verhandlungen direkt zwischen Unternehmen stattfinden. Bei individuellen Produktionsanfragen ist davon auszugehen, dass eine Fertigungsplattform die Vermittlung von Fertigungsaufträgen übernimmt. Anhand der erstellten Produkt-VWS können benötigte Dienstleistungen identifiziert und an die entsprechenden Unternehmen vermittelt werden. Durch das Matching der Features auf definierte Capabilities findet in den Unternehmen eine Prüfung statt, ob ein Fertigungsprozess umgesetzt werden kann. Hierzu könnten in Zukunft Ontologien zum Einsatz kommen. Mit diesen können herstellereigene Features auf unabhängig definierte Capabilities gemappt werden. Vor allem ist im Bereich der Definition von Capabilities zu prüfen, ob sich in Zukunft Standards etablieren, um das Matching zu vereinfachen. Für die spanende Bearbeitung wurde erarbeitet, dass die Capabilities produktspezifisch beschrieben werden sollten. So müssten Features und die benötigten *Properties* als Capabilities definiert werden. Ein VWS-TM Capabilities befindet sich zurzeit in der IDTA in einem Standardisierungsprozess, in das die hier ausgearbeiteten Konzepte einfließen [@IDTA]. Mit Hilfe eines Standards könnte effektiver am Matching und der Planung individueller Prozessketten gearbeitet werden. Als Planungssystem würde sich ein Agent anbieten, der mögliche Prozessketten miteinander vergleicht und nach verschiedenen Faktoren wie Kosten, Zeit und Energieverbrauch optimiert. Wurde eine benötigte Capability erfasst, könnte der Agent bei allen referenzierten Skills zunächst einen Feasibility-Check durchführen. So werden erste Prognosen über den Fertigungsauftrag kalkuliert. Ein vereinfachter Feasibility-Check wurde im Rahmen der Arbeit umgesetzt, allerdings werden Funktionen wie Kollisionsprüfungen und Vorhersagen zur erreichbaren Qualität als offene Forschungsfragen angesehen. Ist ein Prozess geplant, kann ein Ausführungsagent die benötigten CPPM ansteuern. Weiterführende Untersuchungen des Feasibility-Checks und der Skillausführung wurden in [Wa23; Wa24] veröffentlicht. Anhand der Beschreibung des Skills in der VWS, ist es für Softwarekomponenten möglich diese zu interpretieren. Zu empfehlen ist es auch weniger komplexe Skill-Modelle zu standardisieren. Durch die Relation der *Properties* einer Capability und der Parameter eines Skills ist es möglich, diese miteinander zu verknüpfen und notwendige Umrechnung zu hinterlegen. Des Weiteren können einzelne oder mehrere Skills eine Capability abbilden. So können Unternehmen ihre eigenen Skills definieren, um ihr Produktions-Know-how abzubilden oder bereits bestehende Software anzubinden. Eine vollständige Implementierung des Konzeptes auf Basis von VWS wäre im nächsten Schritt sinnvoll. Zur Generierung oder den Zugriff auf VWS existiert Open-Source-Software, wie Basyx [@BaSyx] oder FAAst [@FA<sup>3</sup>ST], die hierfür eingesetzt werden kann. Hierzu sollten neben den Capability

und den Skill (*Control Component*, MTP) TM [IDTA] auch die TM der Produkt-VWS in einen Standard überführt werden. Die hier definierten TM repräsentieren nur einen Auszug der TM, die in Zukunft eine Produkt-VWS enthalten wird. Die Produkt-VWS eignet sich, wie in [PI22] vorgestellt, als Produktpass und würde somit die Grundlage einer Kreislaufwirtschaft bilden. Um das erarbeitete Konzept von Services, Capabilities und Skills für Forschung und Industrie zu konsolidieren, wurde aktiv in der Plattform Industrie 4.0 an einem CSS (Capability, Skill, Service)-Modell mitgearbeitet. Die hier diskutierten Inhalte sind in das CSS-Modell eingeflossen. Das Modell wurde in [Di22], [Kö23] veröffentlicht. Ebenso wurde in einem VDMA Whitepaper ein Vorschlag für eine vereinfachte einheitliche OPC UA-Skill-Schnittstelle veröffentlicht sowie die Anwendung des CSS-Modells in der spannenden Bearbeitung genauer betrachtet [VD23].

# Anhang

## A Appendix A

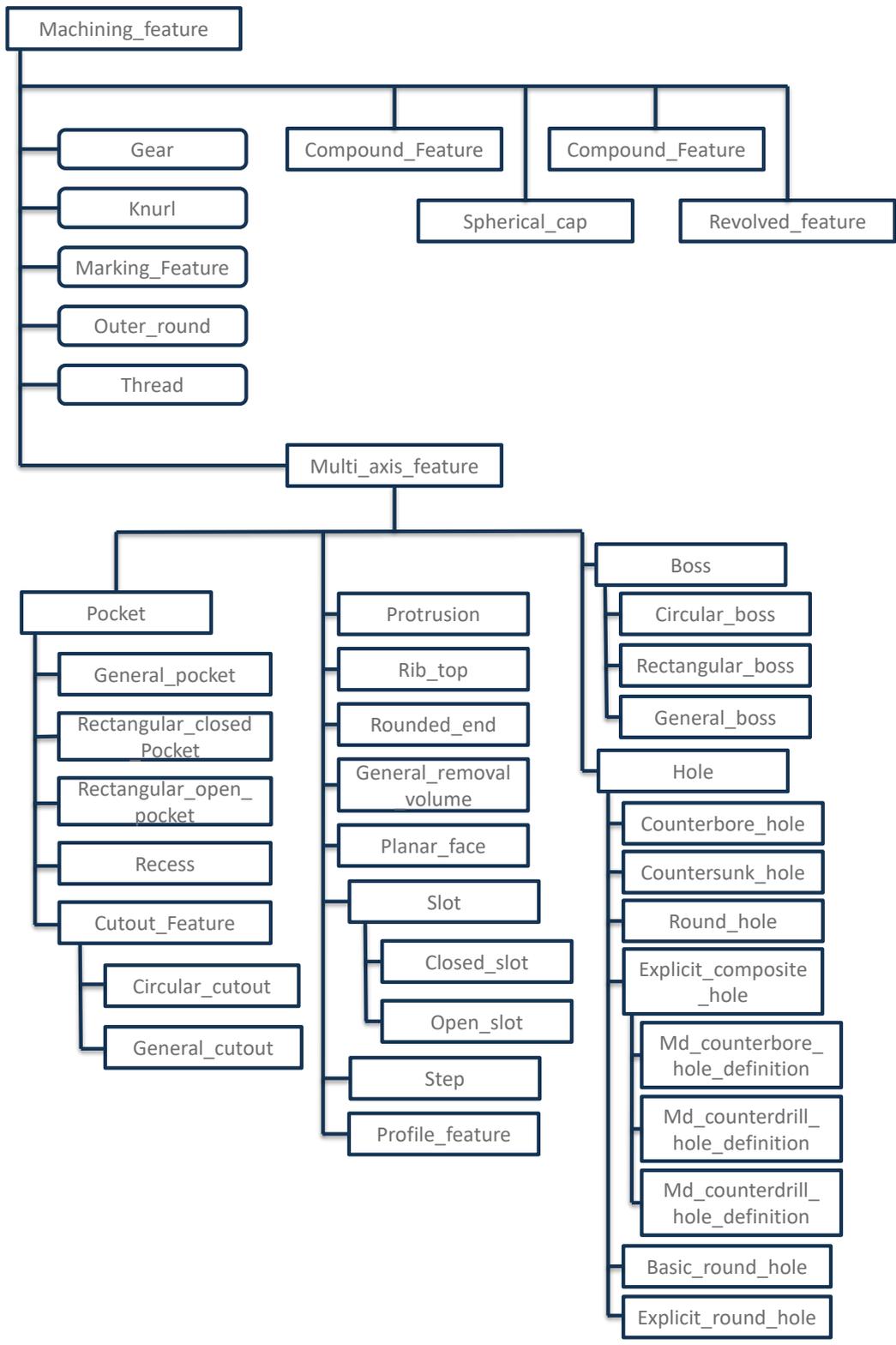


Abbildung 1: Übersicht STEP Machining Feature nach [ISO 10303]

<b>STEP-Variablenbezeichnung</b>	<b>NX-Variablenbezeichnung</b>
profile_width	WIDTH
	WIDTH
	WIDTH_LOWER
	WIDTH_UPPER
profile_length	LENGTH
	LENGTH_LOWER
	LENGTH_UPPER
corner_radius	RADIUS
	RADIUS_LOWER
	RADIUS_UPPER
floor_radius	BOTTOM_RADIUS
	BOTTOM_RADIUS_LOWER
	BOTTOM_RADIUS_UPPER
pocket_depth	DEPTH
	DEPTH_LOWER
	DEPTH_UPPER
placement	X_ORIENTATION_D
	X_ORIENTATION_D
	X_ORIENTATION_L
	X_POSITION
	Y_ORIENTATION_D
	Y_ORIENTATION_L
	Y_POSITION
	Z_ORIENTATION_D
	Z_ORIENTATION_L
	Z_POSITION
base_radius	
change_in_boundary	
	TILTED_TOP_DEPTH
	DEPTH_TOP_CHAMFER
	BOTTOM_ROUGHNESS
	BOTTOM_ROUGHNESS_MATERIAL_REMOVAL
	BOTTOM_ROUGHNESS_MODIFIER
	BOTTOM_ROUGHNESS_STANDARD
	SIDE_ROUGHNESS
	SUBTYPE

Tabelle 1: Vergleich der Parameter eines NX- und STEP-Feature *Pocket Rectangular*

# Literaturverzeichnis

## Monografien und Artikel

- [Ad15] Adamson, G.; Wang, L.; Holm, M.; Moore, P.: Cloud manufacturing – a critical review of recent development and future trends. *International Journal of Computer Integrated Manufacturing* /, S. 1–34, 2015, ISSN: 0951-192X.
- [AN17] Awad, R.; Naumann, M.: Lean Intelligent Assembly Automation (LIAA): Final Report, 2017, URL: <https://cordis.europa.eu/docs/results/608/608604/final1-liaa-r04-final-report-v-1-00-submitted.pdf>, Stand: 20. 07. 2020.
- [Au17] Autodesk: Bitkom Research. Welche der folgenden Arten von Software für technische Anwendungen sind in Ihrem Unternehmen im Einsatz bzw. plant Ihr Unternehmen zukünftig einzusetzen?, 2017, URL: <https://de.statista.com/statistik/daten/studie/785353/umfrage/einsatz-von-software-fuer-technische-anwendungen-in-unternehmen/>, Stand: 28. 08. 2022.
- [AWM17] Adamson, G.; Wang, L.; Moore, P.: Feature-based control and information framework for adaptive and distributed manufacturing in cyber physical systems. *Journal of Manufacturing Systems* 43/, S. 305–315, 2017, ISSN: 02786125.
- [Ba14] Bauer, W.; Schlund, S.; Marrenbach, D.; Ganschar, O.: Industrie 4.0 - Volkswirtschaftliches Potenzial für Deutschland, 2014, URL: <https://www.bitkom.org/sites/main/files/file/import/Studie-Industrie-40.pdf>, Stand: 12. 03. 2022.
- [Ba15] Backhaus, J. C. S.: Adaptierbares aufgabenorientiertes Programmiersystem für Montagesysteme, Dissertation, Herbert Utz Verlag GmbH, 2015.
- [Ba16] Bangemann, T.; Riedl, M.; Thron, M.; Diedrich, C.: Integration of Classical Components Into Industrial Cyber–Physical Systems. *Proceedings of the IEEE* 104/5, S. 947–959, 2016, ISSN: 1558-2256.

- [Ba17] Balta, E. C.; Jain, K.; Lin, Y.; Tilbury, D.; Barton, K.; Mao, Z. M.: Production as a service: A centralized framework for small batch manufacturing. In: 2017 13th IEEE International Conference on Automation Science and Engineering (CASE). IEEE, Piscataway, NJ, 2017, ISBN: 9781509067817.
- [Ba18] Balta, E. C.; Lin, Y.; Barton, K.; Tilbury, D. M.; Mao, Z. M.: Production as a Service: A Digital Manufacturing Framework for Optimizing Utilization. IEEE Transactions on Automation Science and Engineering 15/4, S. 1483–1493, 2018, ISSN: 1545-5955.
- [BD18] Brandenbourger, B.; Durand, F.: Design Pattern for Decomposition or Aggregation of Automation Systems into Hierarchy Levels. In: Proceedings 2018 IEEE 23rd International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, Piscataway, NJ, S. 895–901, 2018, ISBN: 978-1-5386-7108-5.
- [BD19] Belyaev, A.; Diedrich, C.: Aktive Verwaltungsschale von I4.0-Komponenten: Erscheinungsformen von Verwaltungsschalen. In: AUTOMATION, Autonomous Systems and 5G in Connected Industries. 2019.
- [Be18] Bernstein, W. Z.; Hedberg, T. D.; Helu, M.; Feeney, A. B.: Contextualising manufacturing data for lifecycle decision-making. International journal of product lifecycle management 10/4, S. 326–347, 2018.
- [Be20a] Becker, K.; Heymann, S.; Okon, M.; Schnebel, B.; Schöppenthau, F.; Stojanovic, L.; Watson, K.: Usage of Standards in the Smart Factory Web Testbed: An Industrial Internet Consortium White Paper - Version 1.0, 2020, URL: [https://www.iiconsortium.org/pdf/Usage\\_of\\_Standards\\_in\\_Smart\\_Factory\\_Web\\_TB\\_White\\_Paper\\_2020-06-29.pdf](https://www.iiconsortium.org/pdf/Usage_of_Standards_in_Smart_Factory_Web_TB_White_Paper_2020-06-29.pdf), Stand: 09. 12. 2020.
- [Be20b] Bertram, P.: Entwicklung eines kontextsensitiven, modularen Assistenzsystems für manuelle Tätigkeiten. VDI Verlag, Düsseldorf, 2020, ISBN: 9783183040223.
- [Bö15] Böge, A., Hrsg.: Handbuch Maschinenbau: Grundlagen und Anwendungen der Maschinenbau-Technik. Springer Vieweg, Wiesbaden, 2015, ISBN: 9783658065973.
- [Br10] Broy, M.: Cyber-Physical Systems: Innovation durch softwareintensive eingebettete Systeme ; [acatech Symposium, hrsg. von Broy, M., Berlin, 2010.

- [Bu19] Burger, A.; Koziolk, H.; Rückert, J.; Platenius-Mohr, M.; Stomberg, G.: Bottleneck Identification and Performance Modeling of OPC UA Communication Models. In (Apte, V., Hrsg.): Proceedings of the 2019 ACM/SPEC International Conference on Performance Engineering. ACM Conferences, ACM, New York, NY, S. 231–242, 2019, ISBN: 9781450362399.
- [Bu20] Bundesamt für Sicherheit in der Informationstechnik: Sicherheitsanalyse Open Platform Communications Unified Architecture (OPC UA), 2020, URL: [https://www.bsi.bund.de/DE/Publikationen/Studien/OPCUA/OPCUA\\_node.html](https://www.bsi.bund.de/DE/Publikationen/Studien/OPCUA/OPCUA_node.html), Stand: 19.05.2020.
- [BW21] Brecher, C.; Weck, M.: Werkzeugmaschinen Fertigungssysteme 3. Springer Berlin Heidelberg, Berlin, Heidelberg, 2021, ISBN: 978-3-662-46568-4.
- [Ch94] Christensen, J. H.: Holonic Manufacturing Systems: Initial Architecture and Standards Directions. In: Proceedings of the 1st Euro Workshop on Holonic Manufacturing Systems. 1994.
- [Ci17] Cisneros-Cabrera, S.; Ramzan, A.; Sampaio, P.; Mehandjiev, N.: Digital Marketplaces for Industry 4.0: A Survey and Gap Analysis. In (Camarinha-Matos, L. M.; Afsarmanesh, H.; Fornasiero, R., Hrsg.): Collaboration in a data-rich world. Bd. 506, IFIP Advances in Information and Communication Technology, Springer, Cham, S. 18–27, 2017, ISBN: 978-3-319-65150-7.
- [Co17] Colombo, A. W.; Karnouskos, S.; Kaynak, O.; Shi, Y.; Yin, S.: Industrial Cyberphysical Systems: A Backbone of the Fourth Industrial Revolution. IEEE Industrial Electronics Magazine 11/1, S. 6–16, 2017, ISSN: 1932-4529.
- [Cr18] Cruz Salazar, L. A.; Mayer, F.; Schütz, D.; Vogel-Heuser, B.: Platform Independent Multi-Agent System for Robust Networks of Production Systems. IFAC-PapersOnLine 51/11, S. 1261–1268, 2018, ISSN: 24058963.
- [Da17] Danny, P.; Ferreira, P.; Lohse, N.; Guedes, M.: An AutomationML model for plug-and-produce assembly systems. In: 2017 IEEE 15th International Conference on Industrial Informatics (INDIN). IEEE, Piscataway, NJ, S. 849–854, 2017, ISBN: 978-1-5386-0837-1.
- [Da18] Danny, P.; Ferreira, P.; Lohse, N.; Dorofeev, K.: An Event-Based AutomationML Model for the Process Execution of Plug-and-

- Produce' Assembly Systems. In: Proceedings IEEE 16th International Conference on Industrial Informatics (INDIN). IEEE, Piscataway, NJ, S. 49–54, 2018, ISBN: 978-1-5386-4829-2.
- [Di12] Dietz, T.; Schneider, U.; Barho, M.; Oberer-Treitz, S.; Drust, M.; Hollmann, R.; Haegele, M.: Programming System for Efficient Use of Industrial Robots for Deburring in SME Environments. In: ROBOTIK 2012; 7th German Conference on Robotics. S. 1–6, 2012.
- [Di22] Diedrich, C.; Belyaev, A.; Blumenfeld, R.; Bock, J.; Grimm, S.; Hermann, J.; Klausmann, T.; Köcher, A.; Maurmaier, M.; Meixner, K.; Peschke, J.; Schleipen, M.; Schmitt, S.; Schnebel, B.; Stephan, G.; Volkmann, M.; Wannagat, A.; Watson, K.; Winter, M.; Zimmermann, P.: Information model for capabilities, skills & services, Techn. Ber., 2022, URL: <https://www.plattform-i40.de/IP/Redaktion/EN/Downloads/Publikation/CapabilitiesSkillsServices.html>, Stand: 14. 03. 2023.
- [Do17] Dorofeev, K.; Cheng, C.-H.; Guedes, M.; Ferreira, P.; Profanter, S.; Zoitl, A.: Device adapter concept towards enabling plug&produce production environments. In: 2017 22nd IEEE International Conference on Emerging Technologies and Factory Automation. IEEE, Piscataway, NJ, S. 1–8, 2017, ISBN: 978-1-5090-6505-9.
- [Do19] Dorofeev, K.; Profanter, S.; Cabral, J.; Ferreira, P.; Zoitl, A.: Agile Operational Behavior for the Control-Level Devices in Plug&Produce Production Environments. In: Proceedings, 2019 24th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, Piscataway, NJ, S. 49–56, 2019, ISBN: 978-1-7281-0303-7.
- [DW19] Dorofeev, K.; Wenger, M.: Evaluating Skill-Based Control Architecture for Flexible Automation Systems. In: Proceedings of the IEEE International Conference on Emerging Technologies And Factory Automation (ETFA). 2019.
- [DZ18] Dorofeev, K.; Zoitl, A.: Skill-based Engineering Approach using OPC UA Programs. In: Proceedings IEEE 16th International Conference on Industrial Informatics (INDIN). IEEE, Piscataway, NJ, S. 1098–1103, 2018, ISBN: 978-1-5386-4829-2.
- [EI19] Elfaham, H.; Grothoff, J.; Deppe, T.; Azarmipour, M.; Epple, U.: Recipe Based Skill Matching. In: Proceedings, 2019 IEEE 17th Inter-

national Conference on Industrial Informatics (INDIN). IEEE, Piscataway, NJ, S. 500–507, 2019, ISBN: 978-1-7281-2927-3.

- [ENV19] Ellwein, C.; Neff, S.; Verl, A.: Cloud Manufacturing: An Automated Literature Review. *Procedia CIRP* 86/, S. 251–256, 2019, ISSN: 22128271.
- [Eu19] European Commission: The European Green Deal, Brussels, 2019, URL: [https://eur-lex.europa.eu/resource.html?uri=cellar:b828d165-1c22-11ea-8c1f-01aa75ed71a1.0002.02/DOC\\_1&format=PDF](https://eur-lex.europa.eu/resource.html?uri=cellar:b828d165-1c22-11ea-8c1f-01aa75ed71a1.0002.02/DOC_1&format=PDF), Stand: 17. 03. 2022.
- [Fe89] Fehr, E.: *Semantik von Programmiersprachen*. Springer, Berlin und Heidelberg, 1989, ISBN: 9783642702716.
- [Fr14] Franzkowiak, M.: *Methodik zur Strukturierung von Vorrichtungssystemen in der Lohnfertigung*, Dissertation, München: Technische Universität München, 2014.
- [Ga19] Gao, W.; Zhang, C.; Hu, T.; Ye, Y.: An intelligent CNC controller using cloud knowledge base. *The International Journal of Advanced Manufacturing Technology* 102/1-4, S. 213–223, 2019, ISSN: 0268-3768.
- [Ga21] Gaia-X European Association for Data and Cloud AISBL: Gaia-X Architecture Document: 21.12 Release, 2021, URL: [https://gaia-x.eu/sites/default/files/2022-01/Gaia-X\\_Architecture\\_Document\\_2112.pdf](https://gaia-x.eu/sites/default/files/2022-01/Gaia-X_Architecture_Document_2112.pdf), Stand: 17. 03. 2022.
- [GB04] Giret, A.; Botti, V.: Holons and agents. *Journal of Intelligent Manufacturing* 15/5, S. 645–659, 2004, ISSN: 0956-5515.
- [Ge19] Gerrikagoitia, J. K.; Unamuno, G.; Urkia, E.; Serna, A.: Digital Manufacturing Platforms in the Industry 4.0 from Private and Public Perspectives. *Applied Sciences* 9/14, S. 2934, 2019.
- [Ge21] Gesamtmetall - Gesamtverband der Arbeiterverbände der Metall- und Elektro-Industrie e.V.: *Die Metall- und Elektro-Industrie in der Bundesrepublik Deutschland - Zahlen 2021*, 2021, URL: [https://www.gesamtmetall.de/sites/default/files/epaper/epaper-Zahlenheft\\_2021/epaper/ausgabe.pdf](https://www.gesamtmetall.de/sites/default/files/epaper/epaper-Zahlenheft_2021/epaper/ausgabe.pdf), Stand: 09. 06. 2022.
- [Go20] Gomeringer, R.; Kilgus, R.; Menges, V.; Oesterle, S.: *Tabellenbuch Metall: XXL, mit Formelsammlung und CD*. Europa-Lehrmittel, Hanan, 2020, ISBN: 9783808516850.

- [GWE18] Grothoff, J. A.; Wagner, C. A.; Epple, U.: BaSys 4.0: Metamodell der Komponenten und Ihres Aufbaus; 1st ed, 2018.
- [He17a] Herrero, H.; Moughlbay, A. A.; Outón, J. L.; Sallé, D.; de Ipiña, K. L.: Skill based robot programming: Assembly, vision and Workspace Monitoring skill interaction. *Neurocomputing* 255/, S. 61–70, 2017, ISSN: 09252312.
- [He17b] Herrero, H.; Outón, J. L.; Puerto, M.; Sallé, D.; López de Ipiña, K.: Enhanced Flexibility and Reusability through State Machine-Based Architectures for Multisensor Intelligent Robotics. *Sensors (Basel, Switzerland)* 17/6, 2017.
- [He19] Hermann, J.; Rübel, P.; Birtel, M.; Mohr, F.; Wagner, A.; Ruskowski, M.: Self-description of Cyber-Physical Production Modules for a product-driven manufacturing system. *Procedia Manufacturing* 38/, S. 291–298, 2019, ISSN: 23519789.
- [He20] Hermann, J.; David, A.; Wagner, A.; Ruskowski, M.: Considering interdependencies for a dynamic generation of process chains for Production as a Service. *Procedia Manufacturing* 51/, S. 1454–1461, 2020, ISSN: 23519789.
- [He21] Hermann, J.: Dynamische Generierung alternativer Fertigungsfolgen im Kontext von Production as a Service: Dissertation. VDI Verlag, 2021, ISBN: 9783186705020.
- [Ho14] Hoppe, S.: Standardisierte horizontale und vertikale Kommunikation: Status und Ausblick. In (Bauernhansl, T.; ten Hompel, M.; Vogel-Heuser, B., Hrsg.): *Industrie 4.0 in Produktion, Automatisierung und Logistik*. Springer Vieweg, Wiesbaden, S. 325–341, 2014, ISBN: 978-3-658-04681-1.
- [HRR20] Hermann, J.; Rubel, P.; Ruskowski, M.: Development of a system architecture enabling a dynamic generation of process chains for Production as a Service. In: *2020 IEEE Conference on Industrial Cyberphysical Systems (ICPS)*. IEEE, S. 15–20, 2020, ISBN: 978-1-7281-6389-5.
- [Hu13] Hu, S. J.: Evolving Paradigms of Manufacturing: From Mass Production to Mass Customization and Personalization. *Procedia CIRP* 7/, S. 3–8, 2013, ISSN: 22128271.
- [In19] International Data Spaces Association: Reference Architecture Model: Version 3.0, 2019, URL: <https://internationaldataspaces>.

org/wp-content/uploads/IDS-Reference-Architecture-Model-3.0-2019.pdf, Stand: 19. 03. 2022.

- [Jä18] Järvenpää, E.; Hylli, O.; Siltala, N.; Lanz, M.: Utilizing SPIN Rules to Infer the Parameters for Combined Capabilities of Aggregated Manufacturing Resources. IFAC-PapersOnLine 51/11, S. 84–89, 2018, ISSN: 24058963.
- [Jä19] Järvenpää, E.; Siltala, N.; Hylli, O.; Lanz, M.: The development of an ontology for describing the capabilities of manufacturing resources. Journal of Intelligent Manufacturing 30/2, S. 959–978, 2019, ISSN: 0956-5515.
- [Ju22] Jungbluth, S.; Hermann, J.; Motsch, W.; Pourjafarian, M.; Sidorenko, A.; Volkmann, M.; Zoltner, K.; Plociennik, C.; Ruskowski, M.: Dynamic Replanning using Multi-Agent Systems and Asset Administration Shells. In: 2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, S. 1–8, 2022, ISBN: 978-1-6654-9996-5.
- [Ju23] Jungbluth, S.; Witton, A.; Hermann, J.; Ruskowski, M.: Architecture for shared production leveraging asset administration shell and Gaia-X. In: 2023 IEEE 21st International Conference on Industrial Informatics (INDIN). IEEE, Lemgo, Germany, Juli 2023.
- [JW98] Jennings, N. R.; Wooldridge, M. J.: Agent Technology: Foundations, Applications, and Markets, Berlin und Heidelberg, 1998.
- [Ka20a] Katti, B.: Ontology-Based Approach to Decentralized Production Control in the Context of Cloud Manufacturing Execution Systems, Dissertation, Kaiserslautern: TU Kaiserslautern, 2020.
- [KHM18] Kolberg, D.; Hermann, J.; Mohr, F.: SmartFactoryKL Systemarchitektur für Industrie 4.0-Produktionsanlagen: Whitepaper SF-1.2: 04/2018, 2018, URL: [https://smartfactory.de/wp-content/uploads/2018/04/SF\\_WhitePaper\\_SystemArchitecture\\_1-2\\_DE\\_XS.pdf](https://smartfactory.de/wp-content/uploads/2018/04/SF_WhitePaper_SystemArchitecture_1-2_DE_XS.pdf), Stand: 14. 03. 2022.
- [Kl14] Kleinemeier, M.: Von der Automatisierungspyramide zu Unternehmenssteuerungsnetzwerken. In (Bauernhansl, T.; ten Hompel, M.; Vogel-Heuser, B., Hrsg.): Industrie 4.0 in Produktion, Automatisierung und Logistik. Springer Vieweg, Wiesbaden, S. 571–579, 2014, ISBN: 978-3-658-04681-1.

- [Kl17] Klöber-Koch, J.; Pielmeier, J.; Grimm, S.; Brandt, M. M.; Schneider, M.; Reinhart, G.: Knowledge-Based Decision Making in a Cyber-Physical Production Scenario. *Procedia Manufacturing* 9/, S. 167–174, 2017, ISSN: 23519789, URL: <http://www.sciencedirect.com/science/article/pii/S2351978917301324>, Stand: 27. 08. 2022.
- [Ko18] Kolberg, D.: Entwicklung einer Referenzarchitektur zur Realisierung von Methoden der Lean Production mittels digitaler Technologien. 2018.
- [Kö20] Köcher, A.; Hildebrandt, C.; Caesar, B.; Bakakeu, J.; Peschke, J.; Scholz, A.; Fay, A.: Automating the Development of Machine Skills and their Semantic Description. In: 2020 25th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, S. 1013–1018, 2020, ISBN: 978-1-7281-8956-7.
- [Kö23] Köcher, A.; Belyaev, A.; Hermann, J.; Bock, J.; Meixner, K.; Volkmann, M.; Winter, M.; Zimmermann, P.; Grimm, S.; Diedrich, C.: A reference model for common understanding of capabilities and skills in manufacturing, 2023, URL: <https://doi.org/10.1515/auto-2022-0117>, Stand: 12. 02. 2023.
- [KP09] Kramer, T.; Proctor, F.: Feature-based Process Planning Based on STEP. In (Xu, X.; Nee, A. Y., Hrsg.): *Advanced Design and Manufacturing Based on STEP*. Springer Series in Advanced Manufacturing, Springer, London, S. 23–48, 2009, ISBN: 978-1-84882-738-7.
- [Kr01] Kramer, T.; Huang, H.; Messina, E.; Proctor, F.; Scott, H.: A feature-based inspection and machining system. *Computer-Aided Design* 33/9, S. 653–669, 2001, ISSN: 00104485.
- [Kr11] Kretz, D.; Teich, T.; Militzer, J.; Neumann, T.: Implementing ISO standard 10303 application protocol 224 for automated process planning. *Robotics and Computer-Integrated Manufacturing* 27/4, S. 729–734, 2011, ISSN: 07365845.
- [Ku17] Kuss, A.; Dietz, T.; Ksensow, K.; Verl, A.: Manufacturing Task Description for Robotic Welding and Automatic Feature Recognition on Product CAD Models. *Procedia CIRP* 60/, S. 122–127, 2017, ISSN: 22128271.
- [Ku18] Kubota, T.; Liu, C.; Mubarok, K.; Xu, X.: A Cyber-Physical Machine Tool Framework based on STEP-NC, 2018.

- [KX09] Kramer, T.; Xu, X.: STEP in a Nutshell. In (Xu, X.; Nee, A. Y., Hrsg.): Advanced Design and Manufacturing Based on STEP. Springer Series in Advanced Manufacturing, Springer, London, S. 1–22, 2009, ISBN: 978-1-84882-738-7.
- [LBT10] Liu, Z.; Bu, W.; Tan, J.: Motion navigation for arc welding robots based on feature mapping in a simulation environment. Robotics and Computer-Integrated Manufacturing 26/2, S. 137–144, 2010, ISSN: 07365845.
- [Le09] Leitão, P.: Agent-based distributed manufacturing control: A state-of-the-art survey. Engineering Applications of Artificial Intelligence 22/7, S. 979–991, 2009, ISSN: 09521976.
- [Le16] Leitao, P.; Karnouskos, S.; Ribeiro, L.; Lee, J.; Strasser, T.; Colombo, A. W.: Smart Agents in Industrial Cyber–Physical Systems. Proceedings of the IEEE 104/5, S. 1086–1101, 2016, ISSN: 1558-2256.
- [Le17] Leo Kumar, S. P.: State of The Art-Intense Review on Artificial Intelligence Systems Application in Process Planning and Manufacturing. Engineering Applications of Artificial Intelligence 65/, S. 294–329, 2017, ISSN: 09521976.
- [Le19] Lepuschitz, W.; Trautner, T.; Mayerhofer, M.; Fallah, S. M.; Ayatollahi, I.; Merdan, M.: Applying Ontologies in a Cloud Manufacturing System. In: Proceedings, IECON 2019 - 45th Annual Conference of the IEEE Industrial Electronics Society. IEEE, Piscataway, NJ, S. 2928–2933, 2019, ISBN: 978-1-7281-4878-6.
- [Li18] Liu, Y.; Zhao, G.; Zavalnyi, O.; Cao, X.; Cheng, K.; Xiao, W.: STEP-Compliant CAD/CNC System for Feature-Oriented Machining. Computer-Aided Design and Applications 16/2, S. 358–368, 2018.
- [Li19b] Liu, Y.; Wang, L.; Wang, X. V.; Xu, X.; Jiang, P.: Cloud manufacturing: key issues and future perspectives. International Journal of Computer Integrated Manufacturing 32/9, S. 858–874, 2019, ISSN: 0951-192X.
- [LL15] Lipman, R.; Lubell, J.: Conformance checking of PMI representation in CAD model STEP data exchange files. Computer-Aided Design 66/, S. 14–23, 2015, ISSN: 00104485.

- [LL19] Liu, X.; Li, Y.: Feature-based adaptive machining for complex free-form surfaces under cloud environment. *Robotics and Computer-Integrated Manufacturing* 56/, S. 254–263, 2019, ISSN: 07365845.
- [Lo13] Loskyll, M.: Entwicklung einer Methodik zur dynamischen kontextbasierten Orchestrierung semantischer Feldgerätefunktionalitäten: Zugl.: Kaiserslautern, Techn. Univ., Diss., 2013. Techn. Univ, Kaiserslautern, 2013, ISBN: 978-3-943995-29-9.
- [LRH08] Laguionie, R.; Rauch, M.; Hascoët, J.-Y.: Toolpaths Programming in an Intelligent Step-NC Manufacturing Context. 2008.
- [LRZ06] Lindemann, U.; Reichwald, R.; Zäh, M. F.: Individualisierte Produkte: Komplexität beherrschen, in *Entwicklung und Produktion*. Springer, Berlin und Heidelberg, 2006, ISBN: 3-540-25506-0.
- [LS15] Lee, E. A.; Seshia, S. A.: Introduction to embedded systems: A cyber-physical systems approach. LeeSeshia.org, Lulu, 2015, ISBN: 978-1-312-42740-2.
- [LSB16] Lehmann, C.; Städter, J. P.; Berger, U.: Anwendungsbeispiele zur Integration heterogener Steuerungssysteme bei robotergestützten Industrieanlagen. In (Vogel-Heuser, B.; Bauernhansl, T.; ten Hompel, M., Hrsg.): *Handbuch Industrie 4.0*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 1–14, 2016, ISBN: 978-3-662-45537-1.
- [Lu20] Lu, Y.; Liu, C.; Wang, K. I.-K.; Huang, H.; Xu, X.: Digital Twin-driven smart manufacturing: Connotation, reference model, applications and research issues. *Robotics and Computer-Integrated Manufacturing* 61/, S. 101837, 2020, ISSN: 07365845.
- [LWX19] Lu, Y.; Wang, H.; Xu, X.: ManuService ontology: a product data model for service-oriented business interactions in a cloud manufacturing environment. *Journal of Intelligent Manufacturing* 30/1, S. 317–334, 2019, ISSN: 0956-5515.
- [LXW20] Lu, Y.; Xu, X.; Wang, L.: Smart manufacturing process and system automation – A critical review of the standards and envisioned scenarios. *Journal of Manufacturing Systems* 56/, S. 312–325, 2020, ISSN: 02786125.
- [Ma18a] Malakuti, S.; Bock, J.; Weser, M.; Venet, P.; Zimmermann, P.; Wiegand, M.; Grothoff, J.; Wagner, C.; Bayha, A.: Challenges in Skill-based Engineering of Industrial Automation Systems \*. In: *Proceedings 2018 IEEE 23rd International Conference on Emerging*

Technologies and Factory Automation (ETFA). IEEE, Piscataway, NJ, S. 67–74, 2018, ISBN: 978-1-5386-7108-5.

- [Ma18b] Marks, T.: Wertschöpfungsketten der produzierenden Unternehmen in Deutschland. In (Marks, T., Hrsg.): Widerstandsfähigkeit der Wertschöpfungsketten der produzierenden Unternehmen in Deutschland: Dissertation. ifaa-Research, Springer Berlin Heidelberg, Berlin, Heidelberg, S. 97–112, 2018, ISBN: 978-3-662-59802-3.
- [Ma21] Magableh, G. M.: Supply Chains and the COVID–19 Pandemic: A Comprehensive Framework. *European Management Review* 18/3, S. 363–382, 2021, ISSN: 1740-4754.
- [MDN16] Mahmoud, H.; Dhokia, V.; Nassehi, A.: STEP-based Conceptual Framework for Measurement Planning Integration. *Procedia CIRP* 43/, S. 315–320, 2016, ISSN: 22128271.
- [Mi18] Miranda, F.; Martins, R.; Dorofeev, K.; Gentile, V.; Ferreira, P.; Guedes, M.: Towards a Common Manufacturing Service Bus to Enable Flexible Plug-and-Produce Automation. In: *ISR 2018: 50th International Symposium on Robotics*. VDE, Frankfurt am Main, 2018, ISBN: 9783800746996.
- [MM05] Marik, V.; McFarlane, D.: Industrial Adoption of Agent-Based Technologies. *IEEE Intelligent Systems* 20/1, S. 27–35, 2005, ISSN: 1541-1672.
- [Mo14] Monostori, L.: Cyber-physical Production Systems: Roots, Expectations and R&D Challenges. *Procedia CIRP* 17/, S. 9–13, 2014, ISSN: 22128271.
- [Mo20] Motsch, W.; David, A.; Sivalingam, K.; Wagner, A.; Ruskowski, M.: Approach for Dynamic Price-Based Demand Side Management in Cyber-Physical Production Systems. *Procedia Manufacturing* 51/, S. 1748–1754, 2020, ISSN: 23519789.
- [Mo21] Motsch, W.; Dorofeev, K.; Gerber, K.; Knoch, S.; David, A.; Ruskowski, M.: Concept for Modeling and Usage of Functionally Described Capabilities and Skills. In: *2021 26th IEEE International Conference on Emerging Technologies and Factory Automation (ETFA)*. IEEE, S. 1–8, 2021, ISBN: 978-1-7281-2989-1.
- [MVK06] Monostori, L.; Váncza, J.; Kumara, S.: Agent-Based Systems for Manufacturing. *CIRP Annals* 55/2, S. 697–720, 2006, ISSN: 00078506.

- [NNA06] Nassehi, A.; Newman, S. T.; Allen, R. D.: STEP-NC compliant process planning as an enabler for adaptive global manufacturing. *Robotics and Computer-Integrated Manufacturing* 22/5-6, S. 456–467, 2006, ISSN: 07365845.
- [OA06] OASIS SOA-RM Technical Committee: Service Oriented Architecture Reference Model, 2006, URL: <http://docs.oasis-open.org/soa-rm/v1.0/soa-rm.pdf>, Stand: 16. 12. 2020.
- [OJM20] Othman, M. A.; Jamaludin, Z.; Minhat, M.: Intelligent Control of CNC System Based on IEC 61499 Function Block Technology. In (Jamaludin, Z.; Mokhtar, M. N. A., Hrsg.): *Intelligent manufacturing and mechatronics. Lecture Notes in Mechanical Engineering*, Springer, Singapore, S. 176–185, 2020, ISBN: 978-981-13-9538-3.
- [Pe19] Perzylo, A.; Grothoff, J.; Lucio, L.; Weser, M.; Malakuti, S.; Venet, P.; Aravantinos, V.; Deppe, T.: Capability-based semantic interoperability of manufacturing resources: A BaSys 4.0 perspective. *IFAC-PapersOnLine* 52/13, S. 1590–1596, 2019, ISSN: 24058963.
- [PI18] Plattform Industrie 4.0: Diskussionspapier I4.0-Sprache: Vokabular, Nachrichtenstruktur und semantische Interaktionsprotokolle der I4.0-Sprache, hrsg. von Bundesministerium für Wirtschaft und Energie, 2018.
- [PI20a] 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 3.0), hrsg. von Federal Ministry for Economic Affairs and Energy, Berlin, 2020, URL: [https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/Details-of-the-Asset-Administration-Shell-Part1.pdf?\\_\\_blob=publicationFile&v=5](https://www.plattform-i40.de/PI40/Redaktion/DE/Downloads/Publikation/Details-of-the-Asset-Administration-Shell-Part1.pdf?__blob=publicationFile&v=5).
- [PI20b] Plattform Industrie 4.0: Verwaltungsschale in der Praxis: Wie Definiere ich Teilmodelle, beispielhafte Teilmodelle und Interaktion zwischen Verwaltungsschalen (Version1.0), hrsg. von Bundesministerium für Wirtschaft und Energie, 2020.
- [PI20c] Plattform Industrie 4.0: White Paper: Describing Capabilities of Industrie 4.0 Components, hrsg. von Bundesministerium für Wirtschaft und Energie, 2020.
- [PI22] Plociennik, C.; Pourjafarian, M.; Nazeri, A.; Windholz, W.; Knetsch, S.; Rickert, J.; Ciroth, A.; Precci Lopes, A. d. C.; Hagedorn, T.; Vogelgesang, M.; Benner, W.; Gassmann, A.; Bergweil-

- ler, S.; Ruskowski, M.; Schebek, L.; Weidenkaff, A.: Towards a Digital Lifecycle Passport for the Circular Economy. *Procedia CIRP* 105/ , S. 122–127, 2022, ISSN: 22128271.
- [Po14] Pobożniak, J.: Automation of CNC Machine Tool Programming Using STEP-NC (ISO 14649). *Applied Mechanics and Materials* 656/ , S. 206–214, 2014.
- [Ra12] Rauch, M.; Laguionie, R.; Hascoet, J.-Y.; Suh, S.-H.: An advanced STEP-NC controller for intelligent machining processes. *Robotics and Computer-Integrated Manufacturing* 28/3, S. 375–384, 2012, ISSN: 07365845.
- [Ra14] Rauschecker, U.; Stock, D.; Stöhr, M.; Verl, A.: Connecting factories and related IT environments to manufacturing clouds. *International Journal of Manufacturing Research* 9/4, S. 389, 2014, ISSN: 1750-0591.
- [RB18] Ribeiro, L.; Bjorkman, M.: Transitioning From Standard Automation Solutions to Cyber-Physical Production Systems: An Assessment of Critical Conceptual and Technical Challenges. *IEEE Systems Journal* 12/4, S. 3816–3827, 2018, ISSN: 1937-9234.
- [Re92] Reinisch, H.: Planungs- und Steuerungswerkzeuge zur impliziten Geräteprogrammierung in Roboterzellen: Zugl.: Erlangen, Nürnberg, Univ., Diss., 1992. Hanser, München, 1992, ISBN: 3446173803.
- [RH18] Ribeiro, L.; Hochwallner, M.: On the Design Complexity of Cyber-physical Production Systems. *Complexity* 2018/ , S. 1–13, 2018, ISSN: 1099-0526.
- [RN21] Russell, S. J.; Norvig, P.: *Artificial intelligence: A modern approach*. Pearson, Hoboken, NJ, 2021, ISBN: 9780134610993.
- [RS12] Rauschecker, U.; Stohr, M.: Using manufacturing service descriptions for flexible integration of production facilities to manufacturing clouds. In (Thoben, K.-D.; Katzy, B., Hrsg.): 2012 18th International ICE Conference on Engineering, Technology and Innovation (ICE 2012). IEEE, Piscataway, NJ, S. 1–10, 2012, ISBN: 978-1-4673-2275-1.
- [RSS13] Rauschecker, U.; Stöhr, M.; Schel, D.: 8th ASME Manufacturing Science and Engineering Conference, MSEC 2013. CD-ROM. In: r MSEC2013–1021, 9, 2013.

- [Ru20] Ruskowski, M.; Herget, A.; Hermann, J.; Motsch, W.; Pahlevannejad, P.; Sidorenko, A.; Bergweiler, S.; David, A.; Plociennik, C.; Popper, J.; Sivalingam, K.; Wagner, A.: Production Bots für Production Level 4. *atp magazin* 62/9, S. 62–71, 2020, ISSN: 2190-4111.
- [Sa20] Sarkar, A.: Semantic Agent Based Process Planning for Distributed Cloud Manufacturing, Dissertation, Ohio: Russ College of Engineering and Technology of Ohio University, 2020.
- [SB16a] Sanfilippo, E. M.; Borgo, S.: What are features? An ontology-based review of the literature. *Computer-Aided Design* 80/, S. 9–18, 2016, ISSN: 00104485.
- [SB16b] Schatz, A.; Bauernhansl, T.: Geschäftsmodell-Innovationen. In (Vogel-Heuser, B.; Bauernhansl, T.; ten Hompel, M., Hrsg.): *Handbuch Industrie 4.0*. Springer Berlin Heidelberg, Berlin, Heidelberg, S. 1–15, 2016, ISBN: 978-3-662-45537-1.
- [SC02] Suh, S.-H.; Cheon, S.-U.: A Framework for an Intelligent CNC and Data Model. *The International Journal of Advanced Manufacturing Technology* 19/10, S. 727–735, 2002, ISSN: 0268-3768.
- [Sc14] Schulte, S.; Hoenisch, P.; Hochreiner, C.; Dustdar, S.; Klusch, M.; Schuller, D.: Towards Process Support for Cloud Manufacturing. In (Grossmann, G., Hrsg.): *2014 IEEE 18th International Enterprise Distributed Object Computing Conference workshops and demonstrations (EDOCW 2014)*. IEEE, Piscataway, NJ, S. 142–149, 2014, ISBN: 978-1-4799-5470-4.
- [Sc16] Schleipen, M.; Gilani, S.-S.; Bischoff, T.; Pfrommer, J.: OPC UA & Industrie 4.0 - Enabling Technology with High Diversity and Variability. *Procedia CIRP* 57/, S. 315–320, 2016, ISSN: 22128271.
- [SCH02] Suh, S.-H.; Cho, J.-H.; Hong, H.-D.: On the architecture of intelligent STEP-compliant CNC. *International Journal of Computer Integrated Manufacturing* 15/2, S. 168–177, 2002, ISSN: 0951-192X.
- [SF07] Scholz-Reiter, B.; Freitag, M.: Autonomous Processes in Assembly Systems. *CIRP Annals* 56/2, S. 712–729, 2007, ISSN: 00078506.
- [SG16] Siepman, D.; Graef, N.: Industrie 4.0 – Grundlagen und Gesamtzusammenhang. In (Roth, A., Hrsg.): *Einführung und Umsetzung von Industrie 4.0*. Springer Gabler, Berlin und Heidelberg, S. 17–82, 2016, ISBN: 978-3-662-48504-0.

- [Si11] Siemens PLM Software: Hohe Produktivität bei der Teilefertigung: NX CAM, 2011, URL: [https://www.solid-system-team.de/fileadmin/user\\_upload/download/nx/NX-CAM.pdf](https://www.solid-system-team.de/fileadmin/user_upload/download/nx/NX-CAM.pdf), Stand: 17.03.2022.
- [Si14] Siemens PLM Software: NX CMM Inspection programming product review, 2014, URL: [https://www.plm.automation.siemens.com/en\\_us/Images/20647\\_tcm1023-94288.pdf](https://www.plm.automation.siemens.com/en_us/Images/20647_tcm1023-94288.pdf), Stand: 17.03.2022.
- [Si17] Siemens PLM Software: Machining Knowledge Editor Training, 2017.
- [Si23a] Simon, M.; Hermann, J.; Jungbluth, S.; Witton, A.; Volkmann, M.; Belyaev, A.; Urban, C.; Diedrich, C.; Rübél, P.; Ruskowski, M.: Realisierung einer Shared Production. *Atp Ed. - Autom. Prax.* 65/6-7, S. 99–109, Juni 2023.
- [Si23b] Simon, M.; Jungbluth, S.; Farrukh, A.; Volkmann, M.; Hermann, J.; Ruskowski, M.: Implementation of Asset Administration Shells in a Shared Production Scenario with Gaia-X. In: 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA). S. 1–8, 2023.
- [Sp13] Spath, D., Hrsg.: *Produktionsarbeit der Zukunft - Industrie 4.0: Studie.* Fraunhofer-Verl., Stuttgart, 2013, ISBN: 978-3-8396-0570-7.
- [SSM91] Shah, J.; Sreevalsan, P.; Mathew, A.: Survey of CAD/feature-based process planning and NC programming techniques. *Computer-Aided Engineering Journal* 8/1, S. 25, 1991, ISSN: 02639327.
- [Su02] Suh, S.; Chung, D.; Lee, B.; Cho, J.; Cheon, S.; Hong, H.; Lee, H.: Developing an integrated STEP-compliant CNC prototype. *Journal of Manufacturing Systems* 21/5, S. 350–362, 2002, ISSN: 02786125.
- [Su03] Suh, S. H.; Lee, B. E.; Chung, D. H.; Cheon, S. U.: Architecture and implementation of a shop-floor programming system for STEP-compliant CNC. *Computer-Aided Design* 35/12, S. 1069–1083, 2003, ISSN: 00104485.
- [SZM19] Slavkovic, N.; Zivanovic, S.; Milutinovic, D.: An indirect method of industrial robot programming for machining tasks based on STEP-NC. *International Journal of Computer Integrated Manufacturing* 32/1, S. 43–57, 2019, ISSN: 0951-192X.

- [Ta15] Tao, F.; Zhang, L.; Liu, Y.; Cheng, Y.; Wang, L.; Xu, X.: Manufacturing Service Management in Cloud Manufacturing: Overview and Future Research Directions. *Journal of Manufacturing Science and Engineering* 137/4, 2015, ISSN: 1087-1357.
- [TDM21] Thomas, R. G.; Deepak Lawrence, K.; Manu, R.: STEP AP 242 Managed Model-based 3D Engineering: An Application Towards the Automation of Fixture Planning. *International Journal of Automation and Computing* /, 2021, ISSN: 1476-8186.
- [To19] Toquica, J. S.; Zivanovic, S.; Bonnard, R.; Rodriguez, E.; Alvarez, A. J.; Ferreira, J. C. E.: STEP-NC-based machining architecture applied to industrial robots. *Journal of the Brazilian Society of Mechanical Sciences and Engineering* 41/8, 2019, ISSN: 1678-5878.
- [Tr09] Trentesaux, D.: Distributed control of production systems. *Engineering Applications of Artificial Intelligence* 22/7, S. 971–978, 2009, ISSN: 09521976.
- [VD13] VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik: Cyber-Physical Systems: Chancen und Nutzen aus Sicht der Automation, 2013, URL: <https://www.vdi.de/ueber-uns/presse/publikationen/details/cyber-physical-systems-chancen-und-nutzen-aus-sicht-der-automation>, Stand: 13. 03. 2022.
- [VD23] VDMA Verband Deutscher Maschinen- und Anlagenbau: Fähigkeiten in der Produktionsautomatisierung. Konsolidierung des Konzepts aus Sicht des Maschinen- und Anlagenbaus mit dem Schwerpunkt OPC UA, 2023.
- [Vo18] Vogel-Heuser, B.; Ryashentseva, D.; Salazar Cruz, L.; Ocker, F.; Hoffmann, M.; Brehm, R.; Bruce-Boye, C.; Redder, M.; Lüder, A.: Agentenmuster für flexible und rekonfigurierbare Industrie 4.0/CPS-Automatisierungs- bzw. Energiesysteme. In: 19. Leitkongress der Mess- und Automatisierungstechnik. VDI-Berichte, VDI Verlag GmbH, Düsseldorf, 2018, ISBN: 9783180923307.
- [Vo20] Volkmann, M.; Legler, T.; Wagner, A.; Ruskowski, M.: A CAD feature-based manufacturing approach with OPC UA skills. *Procedia Manufacturing* 51/, S. 416–423, 2020, ISSN: 23519789, URL: <http://www.sciencedirect.com/science/article/pii/S2351978920319144>, Stand: 27. 08. 2022.
- [Vo21a] Volkmann, M.; Sidorenko, A.; Wagner, A.; Hermann, J.; Legler, T.; Ruskowski, M.: Integration of a feasibility and context check into

an OPC UA skill. IFAC-PapersOnLine 54/1, S. 276–281, 2021, ISSN: 24058963.

- [Vo21b] Volz, F.: Sharing of Asset Administration Shells with the International Data Spaces. In (Plattform Industrie 4.0, Hrsg.): Shaping a globally secure Industrie 4.0 Ecosystem. S. 92–103, 2021.
- [Vo24] Volkmann, M.; Wagner, A.; Hermann, J.; Ruskowski, M.: Asset Administration Shells and GAIA-X Enabled Shared Production Scenario. In (Silva, F. J. G.; Ferreira, L. P.; Sá, J. C.; Pereira, M. T.; Pinto, C. M. A., Hrsg.): Flexible Automation and Intelligent Manufacturing: Establishing Bridges for More Sustainable Manufacturing Systems. Springer Nature Switzerland, Cham, S. 187–199, 2024.
- [VVZ15] Verein Deutscher Ingenieure e.V.; Verband der Elektrotechnik, Elektronik und Informationstechnik e.V.; ZVEI: Status Report - Reference Architecture Model Industrie 4.0 (RAMI4.0), 2015, URL: [https://www.zvei.org/fileadmin/user\\_upload/Themen/Industrie\\_4.0/Das\\_Referenzarchitekturmodell\\_RAMI\\_4.0\\_und\\_die\\_Industrie\\_4.0-Komponente/pdf/5305\\_Publikation\\_GMA\\_Status\\_Report\\_ZVEI\\_Reference\\_Architecture\\_Model.pdf](https://www.zvei.org/fileadmin/user_upload/Themen/Industrie_4.0/Das_Referenzarchitekturmodell_RAMI_4.0_und_die_Industrie_4.0-Komponente/pdf/5305_Publikation_GMA_Status_Report_ZVEI_Reference_Architecture_Model.pdf), Stand: 13. 01. 2021.
- [Wa01] Wang, L.: Integrated design-to-control approach for holonic manufacturing systems. Robotics and Computer-Integrated Manufacturing 17/1-2, S. 159–167, 2001, ISSN: 07365845.
- [Wa03] Wallace, A.: Sequential resource allocation utilizing agents. International Journal of Production Research 41/11, S. 2481–2499, 2003, ISSN: 0020-7543.
- [Wa13] Wang, L.: Machine availability monitoring and machining process planning towards Cloud manufacturing. CIRP Journal of Manufacturing Science and Technology 6/4, S. 263–273, 2013, ISSN: 17555817.
- [Wa14] Wannenwetsch, H.: Integrierte Materialwirtschaft, Logistik und Beschaffung. Springer-Vieweg, Berlin und Heidelberg, 2014, ISBN: 978-3-642-45022-8.
- [Wa15] Wang, L.: An overview of function block enabled adaptive process planning for machining. Journal of Manufacturing Systems 35/, S. 10–25, 2015, ISSN: 02786125.

- [Wa23] Wagner, A.; Lamoth, S.; Volkmann, M.; Hermann, J.; Ruskowski, M.: Interaction between FeasibilityCheck, PreconditionCheck and SkillExecution in skill-based machining. In: 2023 IEEE 28th International Conference on Emerging Technologies and Factory Automation (ETFA). IEEE, Sinaia, Romania, Sep. 2023.
- [Wa24] Wagner, A.; Volkmann, M.; Hermann, J.; Ruskowski, M.: Machining of individualized milled parts in a skill-based production environment. In: Flexible Automation and Intelligent Manufacturing: Establishing Bridges for More Sustainable Manufacturing Systems. Lecture notes in mechanical engineering, Springer Nature Switzerland, Cham, S. 283–292, 2024.
- [We18] Weyer, S.: Methodik zur informationstechnischen Integration Cyber-Physischer Produktionssysteme innerhalb der Digitalen Fabrik, Dissertation, Technische Universität Kaiserslautern, 2018.
- [We96] Weber, C.: What is a Feature and What is its Use? – Results of FEM-EX Working Group I. In: 29th International Symposium on Automotive Technology and Automation 1996 (ISATA 96). 1996.
- [Wi08] Wildemann, H.: Einkaufspotentialanalyse: Programme zur partnerschaftlichen Erschließung von Rationalisierungspotentialen. TCW Transfer-Centrum-Verl., München, 2008, ISBN: 9783929918649.
- [WJF06] Wang, L.; Jin, W.; Feng, H.-Y.: Embedding machining features in function blocks for distributed process planning. International Journal of Computer Integrated Manufacturing 19/5, S. 443–452, 2006, ISSN: 0951-192X.
- [WLM13] Wang, W.; Li, Y.; Ma, Y.: Towards a Feature-based Agent-driven NC Tool Path Generation to Support Design and Process Changes. Computer-Aided Design and Applications 10/4, S. 603–618, 2013.
- [Wo06] Wosnik, M.; Kramer, C.; Selig, A.; Klemm, P.: Enabling feedback of process data by use of STEP-NC. International Journal of Computer Integrated Manufacturing 19/6, S. 559–569, 2006, ISSN: 0951-192X.
- [Wu13] Wu, D.; Greer, M. J.; Rosen, D. W.; Schaefer, D.: Cloud manufacturing: Strategic vision and state-of-the-art. Journal of Manufacturing Systems 32/4, S. 564–579, 2013, ISSN: 02786125.

- [WXD07] Wang, H.; Xu, X.; Des Tedford, J.: An adaptable CNC system based on STEP-NC and function blocks. *International Journal of Production Research* 45/17, S. 3809–3829, 2007, ISSN: 0020-7543.
- [XN06] Xu, X. W.; Newman, S. T.: Making CNC machine tools more open, interoperable and intelligent—a review of the technologies. *Computers in Industry* 57/2, S. 141–152, 2006, ISSN: 01663615.
- [Xu06] Xu, X. W.: Realization of STEP-NC enabled machining. *Robotics and Computer-Integrated Manufacturing* 22/2, S. 144–153, 2006, ISSN: 07365845.
- [XWN11] Xu, X.; Wang, L.; Newman, S. T.: Computer-aided process planning – A critical review of recent developments and future trends. *International Journal of Computer Integrated Manufacturing* 24/1, S. 1–31, 2011, ISSN: 0951-192X.
- [YL14] Yusof, Y.; Latif, K.: Survey on computer-aided process planning. *The International Journal of Advanced Manufacturing Technology* 75/1-4, S. 77–89, 2014, ISSN: 0268-3768.
- [YI19] Yli-Ojanperä, M.; Sierla, S.; Papakonstantinou, N.; Vyatkin, V.: Adapting an agile manufacturing concept to the reference architecture model industry 4.0: A survey and case study. *Journal of Industrial Information Integration* 15/, S. 147–160, 2019, ISSN: 2452414X.
- [ZF09] Zeppenfeld, K.; Finger, P.: *SOA und WebServices*. Springer-Verlag Berlin Heidelberg, Berlin, Heidelberg, 2009, ISBN: 9783540769910.
- [Zh17] Zhang, Y.; Zhang, G.; Liu, Y.; Di Hu: Research on services encapsulation and virtualization access model of machine for cloud manufacturing. *Journal of Intelligent Manufacturing* 28/5, S. 1109–1123, 2017, ISSN: 0956-5515.
- [Zi19] Zimmermann, P.; Axmann, E.; Brandenbourger, B.; Dorofeev, K.; Mankowski, A.; Zanini, P.: Skill-based Engineering and Control on Field-Device-Level with OPC UA. In: *IEEE ETFA 2019*. 2019.
- [ZSM18] Zivanovic, S.; Slavkovic, N.; Milutinovic, D.: An approach for applying STEP-NC in robot machining. *Robotics and Computer-Integrated Manufacturing* 49/, S. 361–373, 2018, ISSN: 07365845.

## Normen und Richtlinien

- [ANSI/ISA-TR88.00.02] ISA International Society of Automation: ANSI/ISA-TR88.00.02. Machine and Unit States. An implementation example of ANSI/ISA-88.00.01, 2015.
- [DIN 40912] DIN Deutsches Institut für Normung e.V.: DIN SPEC 40912. Kernmodelle - Beschreibung und Beispiele, 2014.
- [DIN 66025] DIN Deutsches Institut für Normung e.V.: DIN 66025. Programmaufbau für numerisch gesteuerte Arbeitsmaschinen.
- [DIN 8580] DIN Deutsches Institut für Normung e.V.: DIN 8580. Fertigungsverfahren - Begriffe, Einteilung, 2020.
- [DIN 8589-2] DIN Deutsches Institut für Normung e.V.: DIN 8589-2. Fertigungsverfahren Spanen: Teil 2: Bohren, Senken, Reiben; Einordnung, Unterteilung, Begriffe.
- [DIN 8589-3] DIN Deutsches Institut für Normung e.V.: DIN 8589-3. Fertigungsverfahren Spanen: Teil 3: Fräsen, 2003.
- [DIN 8593-0] DIN Deutsches Institut für Normung e.V.: DIN 8593-0. Fertigungsverfahren Fügen - Teil 0: Allgemeines; Einordnung, Unterteilung, Begriffe, 2003.
- [DIN EN 61499] DIN Deutsches Institut für Normung e.V.: DIN 61499. Funktionsbausteine für industrielle Leitsysteme. Berlin.
- [DIN EN 61499-1] DIN Deutsches Institut für Normung e.V.: DIN 61499-1. Funktionsbausteine für industrielle Leitsysteme. Teil1: Architektur, Berlin, 2014.
- [DIN EN 61512-1] DIN Deutsches Institut für Normung e.V.: DIN EN 61512-1: Chargenorientierte Fahrweise: Teil1: Modelle und Terminologie, Berlin.
- [DIN EN 62264-1] DIN Deutsches Institut für Normung e.V.: DIN EN 62264-1: Integration von Unternehmensführungs- und Leitsystemen, Berlin.
- [DIN EN ISO 9000] DIN Deutsches Institut für Normung e.V.: DIN EN ISO 9000. Qualitätsmanagementsysteme – Grundlagen und Begriffe, 2015.
- [DIN ISO 2768-1] DIN Deutsches Institut für Normung e.V.: DIN ISO 2768-1. Allgmeintoleranzen; Toleranzen für Längen- und Winkelmaße ohne einzelne Toleranzeintragung, 1991.

- [DIN ISO 2768-2] DIN Deutsches Institut für Normung e.V.: DIN ISO 2768-2. Allgemeintoleranzen; Toleranzen für Form und Lage ohne einzelne Toleranzeintragung, 1991.
- [DIN SPEC 91345] DIN Deutsches Institut für Normung e.V.: DIN SPEC 91345. Referenzarchitekturmodell Industrie 4.0 (RAMI4.0), 2016.
- [IATF 16949] IATF International Automotive Task Force: IATF 16949. Sanctioned Interpretations, 2016.
- [IEC 61499] IEC International Electrotechnical Commission: IEC 61499. Function blocks. 2012.
- [IEC 61499-1] IEC International Electrotechnical Commission: IEC 61499-1. Function blocks. Part 1: Architecture, 2012.
- [IEC 62714] IEC International Electrotechnical Commission: IEC 6274. Engineering data exchange format for use in industrial automation systems engineering - Automation Markup Language.
- [IEC 62714-1] IEC International Electrotechnical Commission: IEC 6274. Engineering data exchange format for use in industrial automation systems engineering - Automation Markup Language - Part 1: Architecture and general requirements, 2018.
- [IEC 62890] IEC International Electrotechnical Commission: IEC 62890. Industrial-process measurement, control and automation - Life-cycle-management for systems and components, 2020.
- [ISO 10303] ISO International Organization for Standardization: ISO 10303. Industrial automation systems and integration - Product data representation and exchange.
- [ISO 10303-1] ISO International Organization for Standardization: ISO 10303-1. Industrial automation systems and integration — Product data representation and exchange — Part 1: Overview and fundamental principles, 2021.
- [ISO 10303-1814] ISO International Organization for Standardization: ISO 10303-1814. Industrial automation systems and integration — Product data representation and exchange Part 1814: Application module: Machining features, 2019.
- [ISO 10303-203] ISO International Organization for Standardization: ISO 10303-203. Industrial automation systems and integration — Product data representation and exchange — Part 203: Application protocol:

- Configuration control-led 3D design of mechanical parts and assemblies, 2011.
- [ISO 10303-214] ISO International Organization for Standardization: ISO 10303-214: Industrial automation systems and integration — Product data representation and exchange — Part 214: Application protocol: Core data for automotive mechanical design processes, 2010.
- [ISO 10303-238] ISO International Organization for Standardization: ISO 10303-238. Industrial automation systems and integration — Product data representation and exchange — Part 238: Application protocol: Model based integrated manufacturing - Edition 3, 2022.
- [ISO 10303-242] ISO International Organization for Standardization: ISO 10303-242. Industrial automation systems and integration — Product data representation and exchange - Part 242: Application protocol: Managed model-based 3D engineering, 2021.
- [ISO 10303-442] ISO International Organization for Standardization: Industrial automation systems and integration — Product data representation and exchange — Part 442: Application module: AP242 managed model based 3D engineering, 2018.
- [ISO 10303-522] ISO International Organization for Standardization: ISO 10303-522. Industrial automation systems and integration — Product data representation and exchange Part 522: Application interpreted construct: Machining features, 2014.
- [ISO 14649] ISO International Organization for Standardization: ISO 14649. Industrial automation systems and integration - Physical device control; Data model for computerized numerical controllers.
- [ISO 14649-11] ISO International Organization for Standardization: ISO 14649-11. Industrial automation systems and integration — Physical device control — Data model for computerized numerical controllers — Part 11: Process data for milling, 2004.
- [ISO 14649-12] ISO International Organization for Standardization: ISO 14649-12. Industrial automation systems and integration — Physical device control — Data model for computerized numerical controllers — Part 12: Process data for turning, 2005.
- [ISO 6983] ISO International Organization for Standardization: ISO 6983. Automation systems and integration - Numerical control of machines - Program format and definitions of address words.

- [ISO 9001] ISO International Organization for Standardization: ISO 9001. Quality management systems — Requirements, 2015.
- [OPC 10000-10] OPC Foundation: OPC 10000-10. OPC UA Specification Part 10: Programs: Release 1.04, 2017, URL: <https://opcfoundation.org/developer-tools/specifications-unified-architecture/part-10-programs/>.
- [OPC 10000-100] OPC Foundation: OPC UA Specification Part 100: Devices: Release 1.02.02, 2020.
- [OPC 10001-7] OPC Foundation: OPC 10001-7. Amendment 7: Interfaces and AddIns: Release 1.04, 2019, URL: <https://opcfoundation.org/developer-tools/specifications-unified-architecture/errata-and-amendments>, Stand: 08. 10. 2020.
- [OPC 30050] OPC Foundation: OPC 30050 - UA for PackML (OMAC), 2020.
- [OPC 30270] OPC Foundation: OPC 30270. Release Candidate - UA Companion Specification for AAS, 2020.
- [OPC 40001-1] OPC Foundation; Verband Deutscher Maschinen- und Anlagenbau: OPC 40001-1: OPC UA for Machinery. Part1: Basic Building Blocks: Release 1.02.0, 2022.
- [OPC 40010-1] OPC Foundation; Verband Deutscher Maschinen- und Anlagenbau: OPC 40010-1: OPC UA Robotics Companion Specification. Part 1: Vertical Integration: Release 1.00, 2019.
- [OPC 40501-1] OPC Foundation; Verein Deutscher Werkzeugmaschinenfabriken e.V.: OPC 40501-1. OPC UA for Machine Tools. Part:1 Machine Monitoring and Job Overview, 2020.
- [OPC 40502] OPC Foundation; Verein Deutscher Werkzeugmaschinenfabriken e.V.: OPC 40502. OPC UA for Computerized Numerical Control (CNC) Systems: Release 1.0, 2017.
- [PAS 1018] DIN Deutsches Institut für Normung e.V.: PAS 1018. Grundstruktur für die Beschreibung von Dienstleistungen in der Ausschreibungsphase.
- [VDA 4961/2] VDA Verband der Automobilindustrie e.V.: VDA 4961/2. Kooperationsmodelle und SE-Checkliste zur Abstimmung der Datenlogistik in SE-Projekten, 2001.
- [VDI 2218] Verein Deutscher Ingenieure e.V.: VDI 2218. Informationsverarbeitung in der Produktentwicklung Feature Technology, 2003.

- [VDI 2860] Verein Deutscher Ingenieure e.V.: VDI 2860. Blatt 1: Montage- und Handhabungstechnik; Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbole (Zurückgezogen), 1982.
- [VDI 3682-1] Verein Deutscher Ingenieure; Verband der Elektrotechnik, Elektronik und Informationstechnik: VDI/VDE 3682-1. Formalisierte Prozessbeschreibungen: Konzept und grafische Darstellung, Berlin, 2015.
- [VDI/VDE 2193-1] Verein Deutscher Ingenieure e.V.; Verband der Elektrotechnik, Elektronik und Informationstechnik e.V.: VDI/VDE 2193-1. Sprache für I4.0-Komponenten. Part1: Struktur von Nachrichten, 2020.
- [VDI/VDE 2193-2] Verein Deutscher Ingenieure e.V.; Verband der Elektrotechnik, Elektronik und Informationstechnik e.V.: VDI/VDE 2193-2. Sprache für I4.0-Komponenten. Part2: Interaktionsprotokoll für Ausschreibungsverfahren, 2020.
- [VDI/VDE 3682-2] Verein Deutscher Ingenieure; Verband der Elektrotechnik, Elektronik und Informationstechnik: VDI/VDE 3682-2. Formalisierte Prozessbeschreibungen: Informationsmodell, Berlin.
- [VDI/VDE/NAMUR 2658-1] Verein Deutscher Ingenieure e.V.; Verband der Elektrotechnik, Elektronik und Informationstechnik e.V.; NAMUR: VDI/VDE/NAMUR 2658. Blatt 1: Automatisierungstechnisches Engineering modularer Anlagen in der Prozessindustrie - Allgemeines Konzept und Schnittstellen.

## Webseiten und Internetreferenzen

- [@4diac] Eclipse: 4diacTM: Open source infrastructure for distributed industrial process measurement and control systems based on the IEC 61499 standard, URL: <https://www.eclipse.org/4diac/>, Stand: 14. 03. 2022.
- [@AASX] AASX Package Explorer: GitHub Projekt, URL: <https://github.com/admin-shell-io/aasx-package-explorer#:~:text=AASX%20Package%20Explorer.%20AASX%20Package%20Explorer%20is%20a,screencasts%20%28both%20in%20English%20and%20German%29%20at%3A%20https%3A%2F%2Fadmin-shell-io.com%2Fscreencasts%2F.>, Stand: 02. 04. 2022.

- [@BaSys] BaSys 4.0: Basissystem Industrie 4.0: Eine offene Plattform für die vierte industrielle Revolution, URL: <https://www.basys40.de/>, Stand: 20.07.2020.
- [@BaSyx] BaSyx: Eclipse BaSyx open source platform, URL: <https://www.eclipse.org/basyx/>, Stand: 14.03.2022.
- [@Catia] Catia: Dassault Systèmes: Benutzerunterstützung R2019x: STEP Konzepte. Anforderungen, URL: <https://help.3ds.com/2019x/German/DSDoc/SstUserMap/sst-c-Requirements.htm?ContextScope=onpremise>, Stand: 06.01.2021.
- [@DSC] Fraunhofer Institut für Software und Systemtechnik: DataSpace-Connector, URL: <https://www.isst.fraunhofer.de/de/geschaeftsfelder/datenwirtschaft/technologien/Dataspace-Connector.html>, Stand: 17.03.2022.
- [@easy2parts] easy2parts: Website, URL: <https://www.easy2parts.com/>, Stand: 16.03.2022.
- [@EDC] EDC: Eclipse Dataspace Connector - Project, URL: <https://projects.eclipse.org/projects/technology.dataspaceconnector>, Stand: 17.03.2022.
- [@FA<sup>3</sup>ST] Fraunhofer IOSB: FAAAST - Fraunhofer Advanced AAS Tools (for Digital Twins), URL: <https://github.com/FraunhoferIOSB/FAAAST-Service>, Stand: 01.05.2022.
- [@FabOS] FabOS: Projektwebsite), URL: <https://www.fab-os.org/>, Stand: 21.01.2023.
- [@FabOS-Components] FabOS: Getting Started), URL: <https://docs.fab-os.org/docs/components/#eclipse-basyx>, Stand: 21.01.2023.
- [@Fraesteile] Frästeile Marktplatz: Website, URL: <https://fraesteile-marktplatz.de/>, Stand: 16.03.2022.
- [@Gaia-X] Bundesministerium für Wirtschaft und Klimaschutz: Gaia-X, URL: <https://www.bmwi.de/Redaktion/DE/Dossier/gaia-x.html>, Stand: 17.03.2022.
- [@IDTA] Industrial Digital Twin Association: AAS Submodel Templates, URL: <https://industrialdigitaltwin.org/en/content-hub/submodels>, Stand: 02.04.2022.

- [@manucloud] Manucloud: EU-Projekt Website, URL: <https://cordis.europa.eu/article/id/36001-manucloud-integrating-the-solar-cell-supply-chain-online/de>, Stand: 09. 12. 2020.
- [@ManuService] Lu, Y.; Xu, X.: ManuService Ontologie, URL: <https://mcloud.foe.auckland.ac.nz/Static/index-en.html>, Stand: 18. 04. 2022.
- [@NAMUR] NAMUR: NAMUR Open Architecture (NoA), URL: <https://www.namur.net/de/fokusthemen/namur-open-architecture.html>, Stand: 15. 04. 2022.
- [@OnShape] OnShape: Supported File Formats, URL: <https://cad.onshape.com/help/Content/translation.htm>, Stand: 06. 01. 2021.
- [@OPC-Foundation] OPC Foundation: Website: The Industrial Interoperability Standard, URL: <https://opcfoundation.org/>, Stand: 15. 01. 2021.
- [@OPC-References] OPC Foundation: OPC UA Online Reference: Online versions of OPC UA specifications and information models, URL: <https://reference.opcfoundation.org/>, Stand: 27. 08. 2022.
- [@OpenCascade] OpenCASCADE: Technology 7.6.0 dev, URL: [https://dev.opencascade.org/doc/overview/html/occt\\_user\\_guides\\_\\_step.html#occt\\_step\\_1\\_2](https://dev.opencascade.org/doc/overview/html/occt_user_guides__step.html#occt_step_1_2), Stand: 06. 01. 2021.
- [@openMOS] openMOS: Open Dynamic Manufacturing Operating System for Smart Plug-and-Produce Automation Components, URL: <https://www.openmos.eu/>, Stand: 20. 07. 2020.
- [@Orderfox] Orderfox: Website, URL: <https://www.orderfox.com/de/?>, Stand: 16. 03. 2022.
- [@Plattform4.0] Plattform Industrie 4.0: Asset Administration Shell Specifications, URL: <https://www.plattform-i40.de/IP/Redaktion/EN/Standardartikel/specification-administrationshell.html>, Stand: 14. 03. 2022.
- [@SF-Live] Volkmann, M.; Peschke, J.; Hermann, J.; Ruskowski, M.: SmartFactory-KL LIVE: Skill Based Production - Die Produktionsarchitektur der Zukunft?, 2022, URL: <https://www.youtube.com/watch?v=AawRuq1gIZQ>, Stand: 12. 03. 2022.
- [@SFKL] Transferinitiative SmartFactory Kaiserslautern e.V.: Shared Production Kaiserslautern, URL: <https://smartfactory.de/unsere-production-level-4-oekosystem-im-jahr-2025/>, Stand: 02. 04. 2022.

- [@SFweb] Smart Factory Web: Website, URL: [www.smartfactoryweb.de](http://www.smartfactoryweb.de), Stand: 05.01.2020.
- [@SiemensPLM1] Siemens PLM Software: NX Open, URL: [https://docs.plm.automation.siemens.com/tdoc/nx/1872/nx\\_api#uid:xid1162445](https://docs.plm.automation.siemens.com/tdoc/nx/1872/nx_api#uid:xid1162445), Stand: 01.04.2022.
- [@SiemensPLM2] Siemens PLM Software: NX - Formelemente suchen, URL: [https://docs.plm.automation.siemens.com/tdoc/nx/1872/nx\\_help#uid:xid1128418:index\\_mfgholemaking:fbm\\_select\\_features\\_ov](https://docs.plm.automation.siemens.com/tdoc/nx/1872/nx_help#uid:xid1128418:index_mfgholemaking:fbm_select_features_ov), Stand: 02.04.2022.
- [@SiemensPLM3] Siemens Product Lifecycle Management Software Inc.: NX-Hilfe: Datenkonvertierung. STEP Translator. URL: [https://docs.plm.automation.siemens.com/tdoc/nx/1872/nx\\_help#uid:xid1128422:index\\_xid458198](https://docs.plm.automation.siemens.com/tdoc/nx/1872/nx_help#uid:xid1128422:index_xid458198), Stand: 06.01.2021.
- [@SMACH] SMACH: SMACH Bibliothek in ROS, URL: <http://wiki.ros.org/smach>, Stand: 20.07.2020.
- [@Spanflug] Spanflug: Website, URL: <https://spanflug.de/>, Stand: 16.03.2022.
- [@STEP242] STEP AP 242: Project Website, URL: [www.ap242.org](http://www.ap242.org), Stand: 18.03.2021.
- [@UmweltBundesamt] Umwelt Bundesamt: Strom- und Wärmeversorgung in Zahlen, URL: <https://www.umweltbundesamt.de/themen/klima-energie/energieversorgung/strom-waermeversorgung-in-zahlen#Strommix>, Stand: 24.04.2022.
- [@up2parts] up2parts: Website, URL: <https://up2parts.com/>, Stand: 16.03.2022.
- [@VDI 2860 - Projekt] Verein Deutscher Ingenieure e.V.: VDI 2860 - Projekt: Handhabungsfunktionen, Handhabungseinrichtungen; Begriffe, Definitionen, Symbol, URL: <https://www.vdi.de/richtlinien/details/vdi-2860-handhabungsfunktionen-handhabungseinrichtungen-begriffe-definitionen-symbol>, Stand: 04.09.2022.
- [@VDMA] VDMA, Verband Deutscher Maschinen- und Anlagenbau: Website, URL: <https://www.vdma.org/>, Stand: 18.03.2022.
- [@Xometry] Xometry: Website, URL: <https://xometry.eu/de/>, Stand: 16.03.2022.

[@YouTube] Volkmann, M.: YouTube: Skill-based Production in der industriellen Anwendung. Ein autonomer Roboter bei der Arbeit. 2022, URL: <https://youtu.be/P3KsxH3eLNg>, Stand: 27. 08. 2022.

## Betreute studentische Arbeiten mit thematischer Relevanz

- [Br21] Brandauer, M.: Autonome Identifikation von Bearbeitungsparametern in Fertigungsmaschinen holonischer Produktionssysteme, Masterarbeit, 2021.
- [De20] Deller, M.: Extraktion geometrischer Features aus 3D-CAD-Modellen für die skill-basierte Produktion, Studienarbeit, 2020.
- [De21] Deller, M.: Programmierung eines CAD-User-Interfaces für die skill-basierte Produktion am Beispiel von Siemens NX 1872, Studienarbeit, 2021.
- [Ha21] Hamman, A.: Erstellung einer Ontologie zur Beschreibung von Produkten und Ressourcen in Produktionsumgebungen, Diplomarbeit, 2021.
- [Ka20b] Kauffmann, D.: Programmierung von autonomen Fertigungsoperationen auf einer Robotersteuerung, Masterarbeit, 2020.
- [Kr22] Kremser, P.: Optimierung einer skill-basierten Werkzeugmaschine sowie Erweiterung eines CAD-Plug-In zur Feature-basierten Prozessplanung, Studienarbeit, 2022.
- [Li19a] Lippert, C.: Konzeption und Inbetriebnahme einer Roboterzelle zur mechanischen Bearbeitung und deren CAM-Einbindung, Projektarbeit, 2019.
- [Wa20] Wagner, A.: Untersuchung und Umsetzung von Fertigungsszenarien in der autonomen Produktion an einer Roboterzelle mit Hilfe von OPC UA Skills, Masterarbeit, 2020.

# Lebenslauf

## Persönliche Daten

Name: Magnus Volkmann

Staatsangehörigkeit: Deutsch

## Ausbildung

2011 - 2017 Studium an der Technische Universität Kaiserslautern (Diplom)  
Studiengang: Maschinenbau & Verfahrenstechnik

2001 - 2010 Abitur am Max-Planck-Gymnasium, Ludwigshafen am Rhein

## Berufserfahrung

seit 07/2021 Rheinland-Pfälzische Technische Universität Kaiserslautern-Landau  
(ehemals Technische Universität Kaiserslautern)  
Lehrstuhl für Werkzeugmaschinen und Steuerungen (WSKL)  
Technologie-Initiative SmartFactory KL e.V., Kaiserslautern  
*Teamleiter*

seit 09/2017 Technische Universität Kaiserslautern  
Lehrstuhl für Werkzeugmaschinen und Steuerungen (WSKL)  
*Researcher*

12/2015 - 05/2017 Technische Universität Kaiserslautern  
Lehrstuhl für Strömungsmechanik und Strömungsmaschinen  
*Hilfswissenschaftler*