



Software für Industrie 4.0 (Vorlesung & Übung)

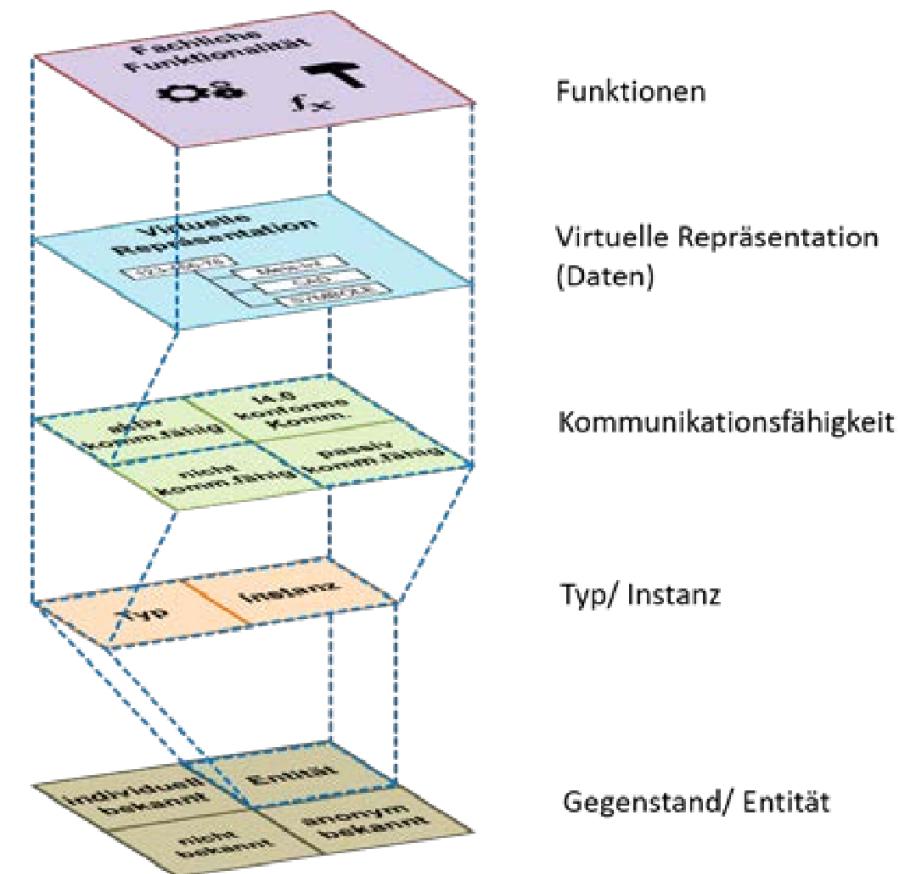
Verwaltungsschalen (AAS) und Interaktionsmuster



- Eine I4.0-Komponente zeichnet sich laut [GMA 2014] durch folgende **Eigenschaften** aus:
 - Sie besitzt einen **kommunikationsfähigen Softwareteil**, der sie zu einem vollwertigen Dienstsystemteilnehmer im I4.0-Netzwerk macht.
 - Sie ist bezüglich der CP-Klassifikation entweder eine **CP43- oder eine CP44-Komponente**.
- Als Dienstsystemteilnehmer im I4.0-Netzwerk werden gemäß [GMA 2014] noch folgende weitere Eigenschaften verlangt:
 - Sie ist im Netzwerk eindeutig identifizierbar.
 - Sie unterstützt die für ein I4.0-System allgemein standardisierten Dienstfunktionen und Zustände.
 - Sie bietet für Ihre Funktionalität und Daten einen der Aufgabe angemessenen Schutz.
 - Sie ist ihrer Aufgabe angemessen robust und verfügbar.
 - Sie besitzt die für ihre Aufgabe erforderlichen Echtzeiteigenschaften.
 - Sie unterstützt die für ein I4.0-System standardisierte Semantik.

I4.0-Komponente (Wiederholung)

- Neben Daten kann eine I4.0-Komponente auch eine **fachliche Funktionalität** besitzen
- Die **virtuelle Repräsentation** hält Daten zu dem Gegenstand. Diese Daten können entweder „auf/in“ der I4.0-Komponente selbst gehalten und durch eine I4.0-konforme Kommunikation der Außenwelt zur Verfügung gestellt werden.
- Um Eigenschaften einer I4.0-Komponente bereitstellen zu können, muss min. ein Informationssystem eine Verbindung zum Gegenstand halten (min. passive **Kommunikationsfähigkeit**)
- Gegenstände können als **Typ** oder als **Instanz** bekannt sein.
- Um Daten und Funktionen an einen Gegenstand binden zu können, muss dieser als **Entität** vorliegen.

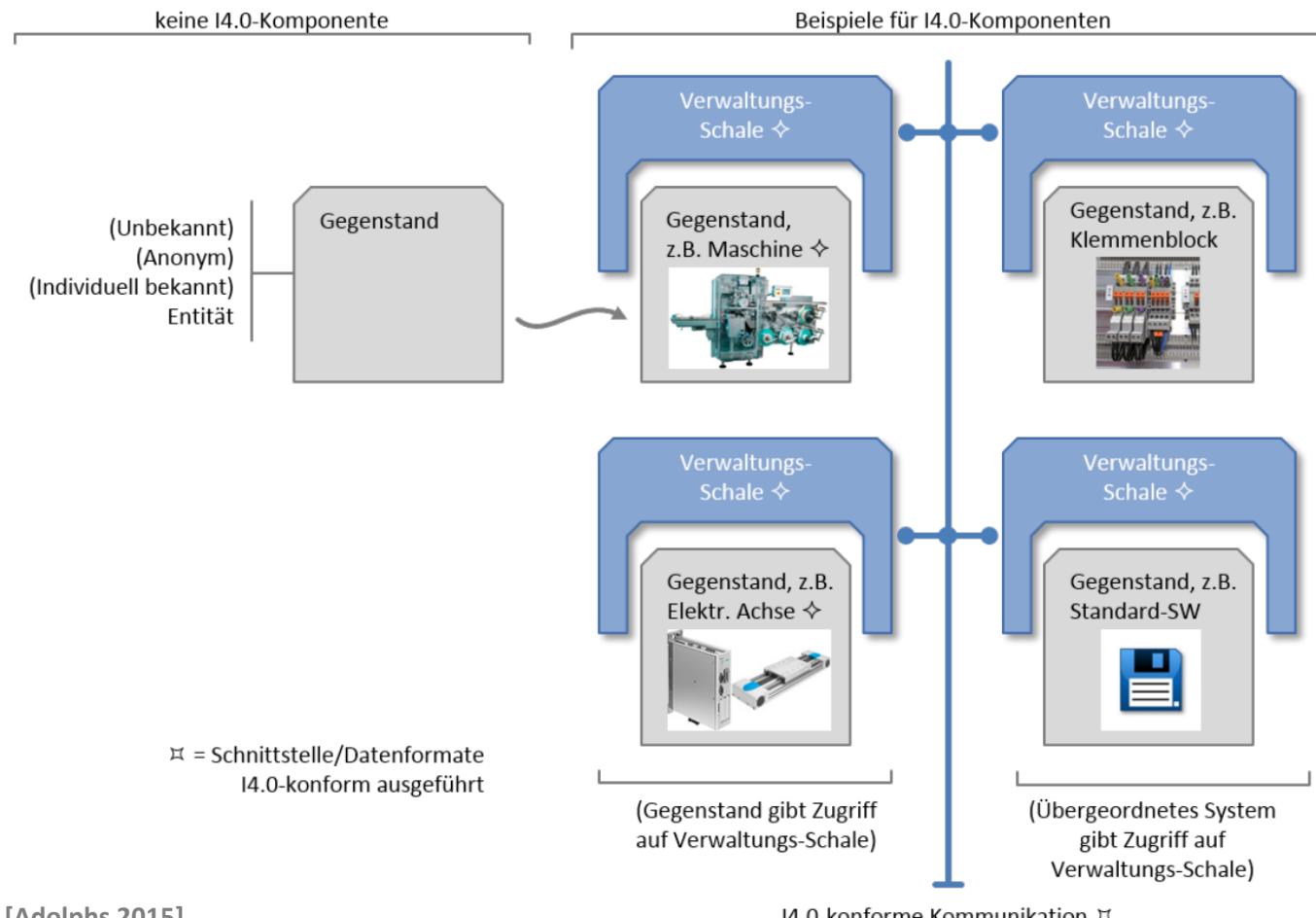
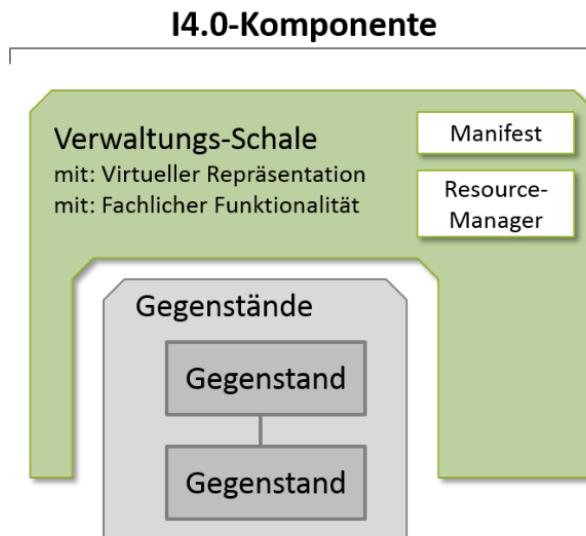


Quelle: [Adolphs 2015]

Verwaltungsschale: Idee

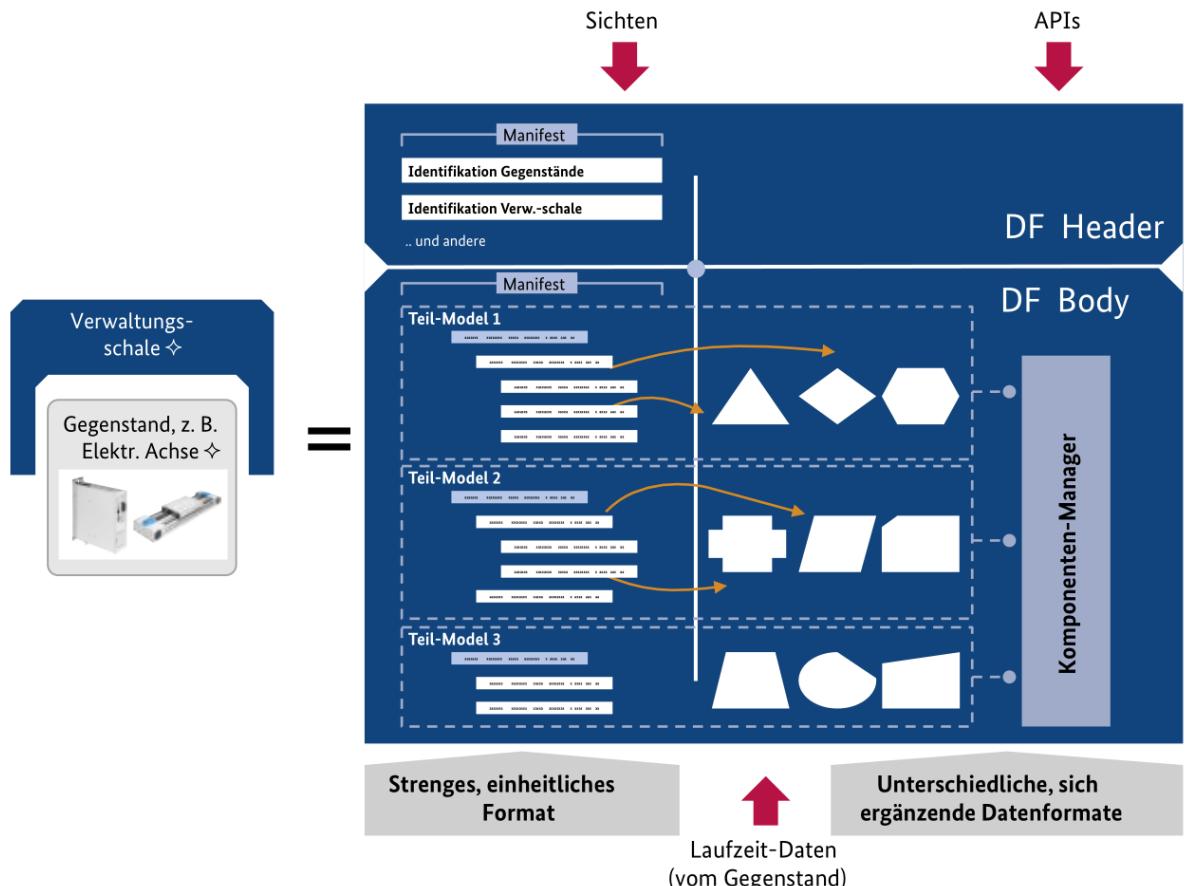
- Idee: Eine „Verwaltungs-Schale“ macht einen Gegenstand zu einer I4.0-Komponente.

- Die Verwaltungs-Schale umfasst
 - die virtuelle Repräsentation und
 - die fachliche Funktionalität des Gegenstands



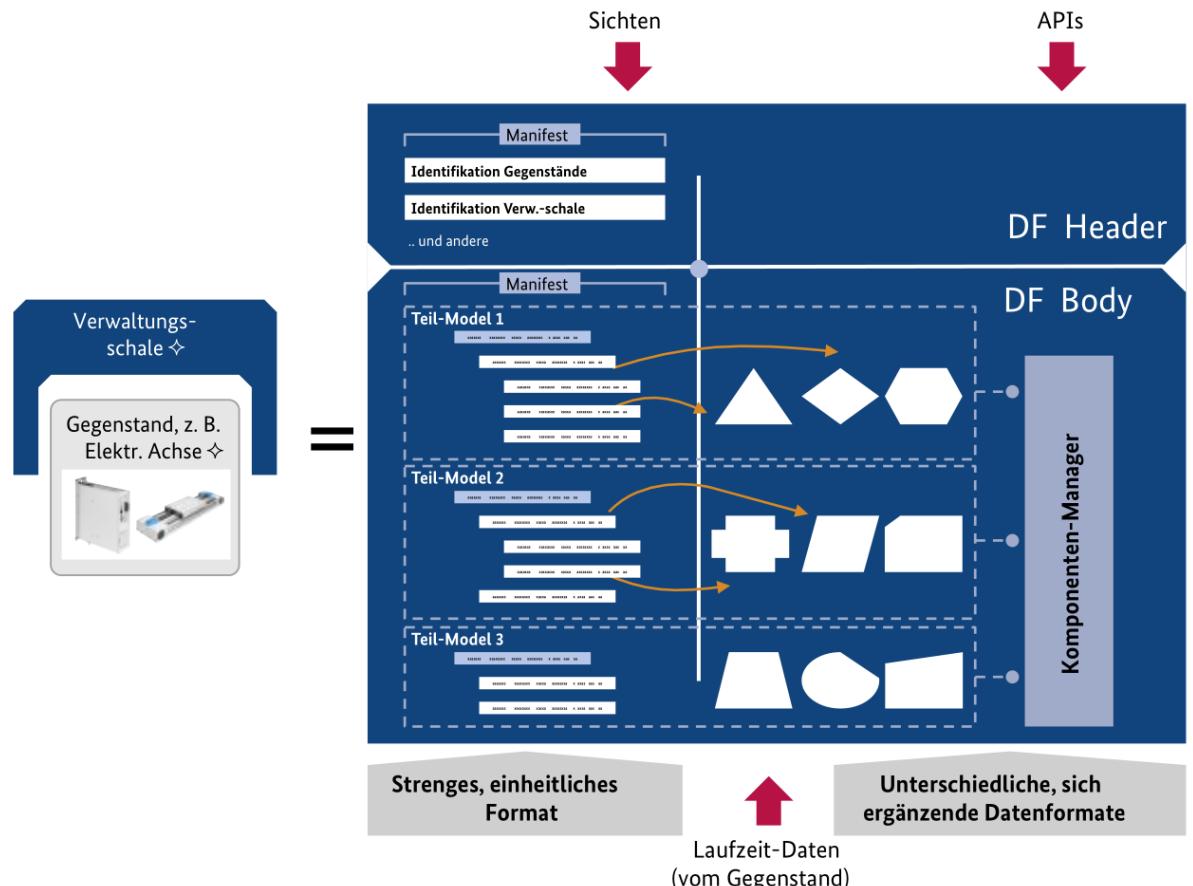
Quelle: [Adolphs 2015]

Verwaltungsschale: Grobstruktur [Adolphs 2016]



- Im **Header** sorgt eine Liste von Merkmalen für eine Identifikation und Bezeichnung der konkreten *Gegenstände* und der *Verwaltungsschale* und verweist gegebenenfalls auf ausgewählte Fähigkeiten der Gegenstände und Sichten.
- Im **Body** findet sich der *Komponenten-Manager*, welcher einzelne Teilmodelle innerhalb der Verwaltungsschale verwaltet. Jedes Teilmodell verfügt über hierarchisch organisierte Merkmale, welche auf Daten und Funktionen referenzieren.
- Die Gesamtheit der Merkmale aller Teilmodelle bildet somit das **Manifest der Verwaltungsschale**, welches damit als eindeutig aufzufindendes Inhaltsverzeichnis aller Daten und Funktionen dienen kann.

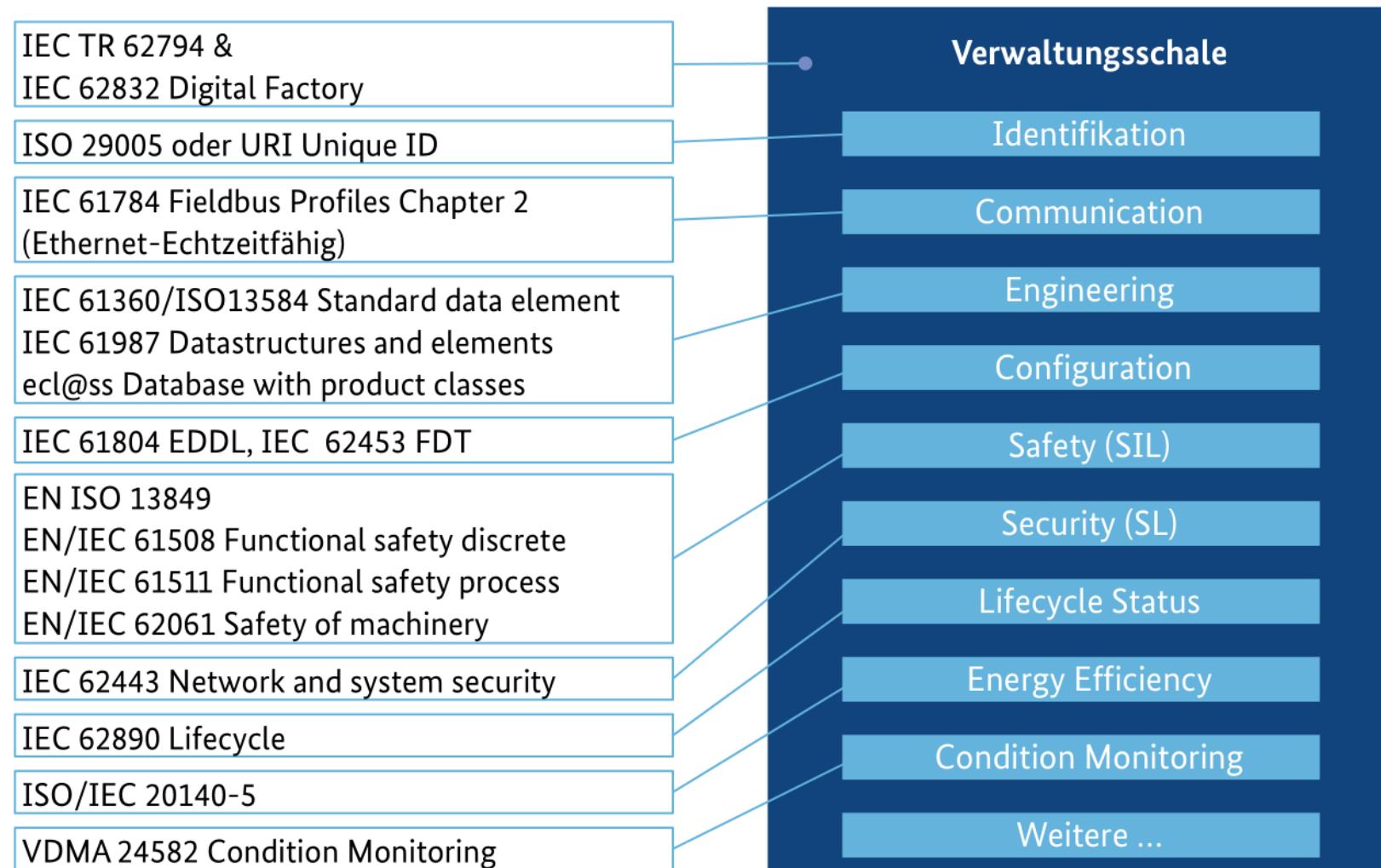
Verwaltungsschale: Grobstruktur [Adolphs 2016]



- Die Merkmalsstrukturen liegen in einem **strengen, einheitlichen Format** vor, während für die unterschiedlichen Daten und Funktionen unterschiedliche, **sich ergänzende Datenformate und Zugriffsmethoden** möglich sind.
- Nach außen hin kann die Verwaltungsschale ggf. **Laufzeit-Daten** (vom Gegenstand) aufnehmen und abbilden, z. B. die Ist-Position eines Motors
- Eine **serviceorientierte API** soll Dienste des Komponenten-Managers nach außen zur Verfügung stellen, z.B.
 - für die Pflege der Merkmale, Daten und Funktionen innerhalb der Verwaltungsschale,
 - für die Adressierung und Identifikation von Verwaltungsschalen und Gegenständen, oder
 - eine Suche nach Merkmalen und referenzierten Daten und Funktionen.

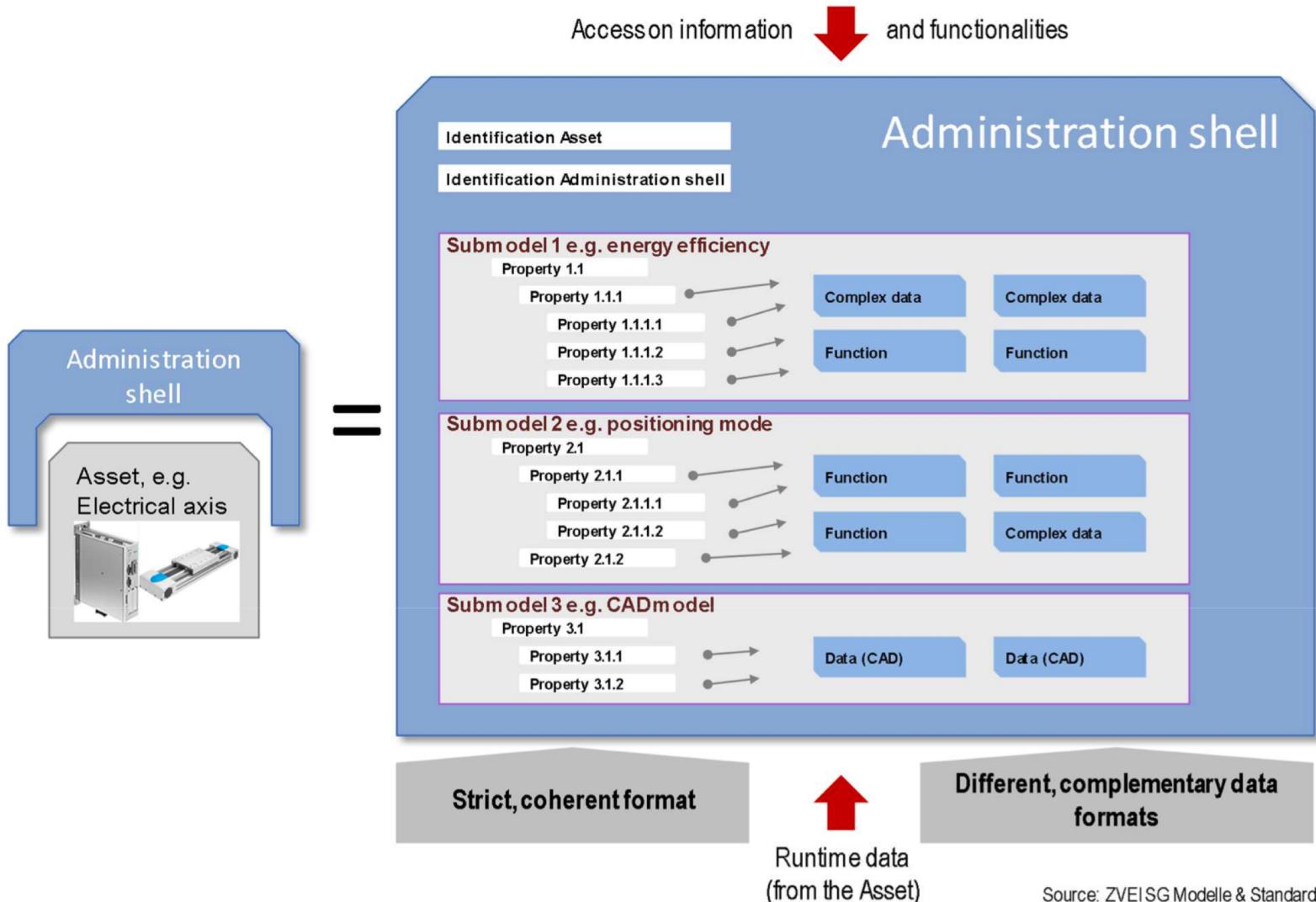
Verwaltungsschale Teil-Module (Sub-Modules)

Beispiele für Inhalte der Verwaltungsschale



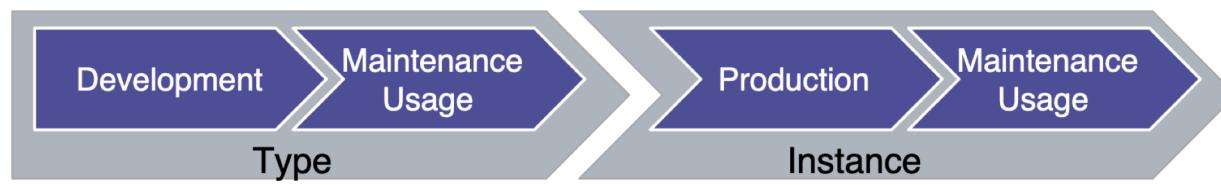
Quelle: [Adolphs 2016]

Verwaltungsschale: Beispiel



Source: ZVEI SG Modelle & Standards

Lebenszyklus (1)



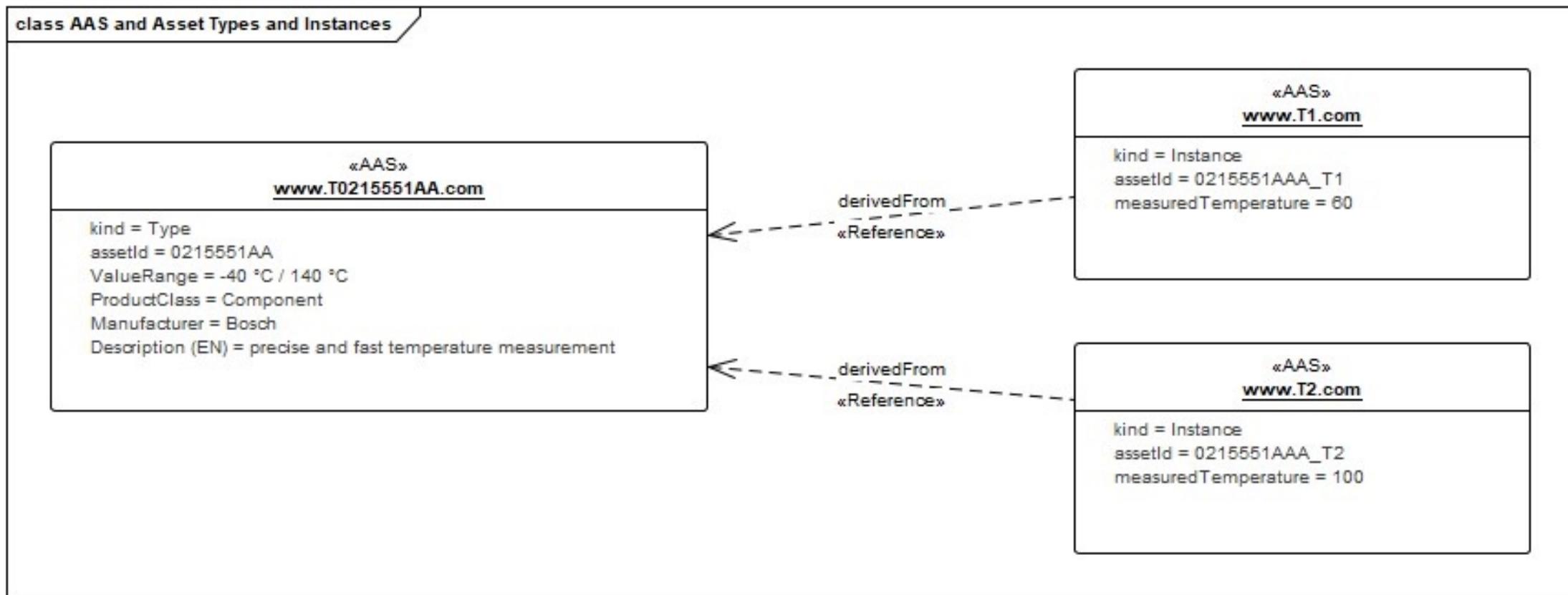
Phase		Inhalte (Beispiele)
Typ	Development	<ul style="list-style-type: none">• Unterscheidungs-Merkmale und Funktionalitäten• Design-Artefakten<ul style="list-style-type: none">• CAD Zeichnungen• Schaltpläne• Eingebettete Software
	Usage/ Maintenance	<ul style="list-style-type: none">• „Externe“ Informationen<ul style="list-style-type: none">• Technische Datenblätter• Schnittstellenbeschreibungen• Präventive Wartungsintervalle
Instanz	Production	<ul style="list-style-type: none">• Basiert auf Typ-Beschreibungen• Informationen über Produktion, Logistik und Tests<ul style="list-style-type: none">• Grundkalibrierungsdaten• Charge
	Usage/ Maintenance	<ul style="list-style-type: none">• Sensorwerte• Nutzungsdaten• ...

Lebenszyklus (2)



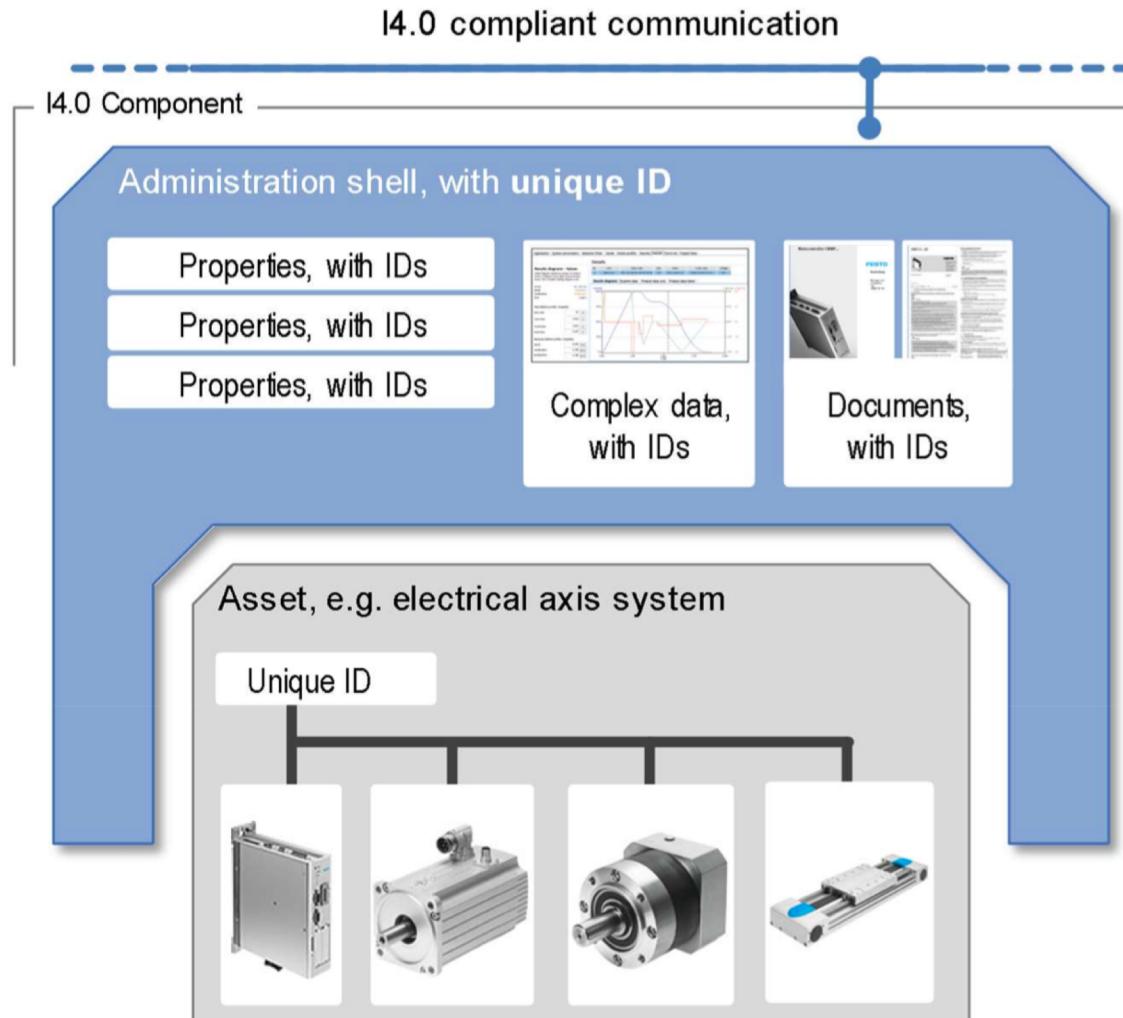
Institute for
Software & Systems
Engineering

Beispiel: Temperatursensoren



- Identifikation wird benötigt für die genaue Zuordnung von verschiedenen Entitäten der Verwaltungsschale zu gewährleisten und um Relationen zwischen diesen herzustellen
- Mögliche Entitäten
 - Verwaltungsschale
 - Gegenstand
 - Submodell Instanzen und Typen
 - Eigenschaften von externen Quellen (z. B. eCI@ss Instanz)
- Es gibt zwei normgerechte Identifikationstypen
 - IRDI (International Registration Data Identifier): z. B. 27-02-26-04 (Servo-Motor)
 - URI (Uniform Resource Identifier): z. B. <http://eo.dbpedia.org/page/Servomotor>

Identifikation (2)



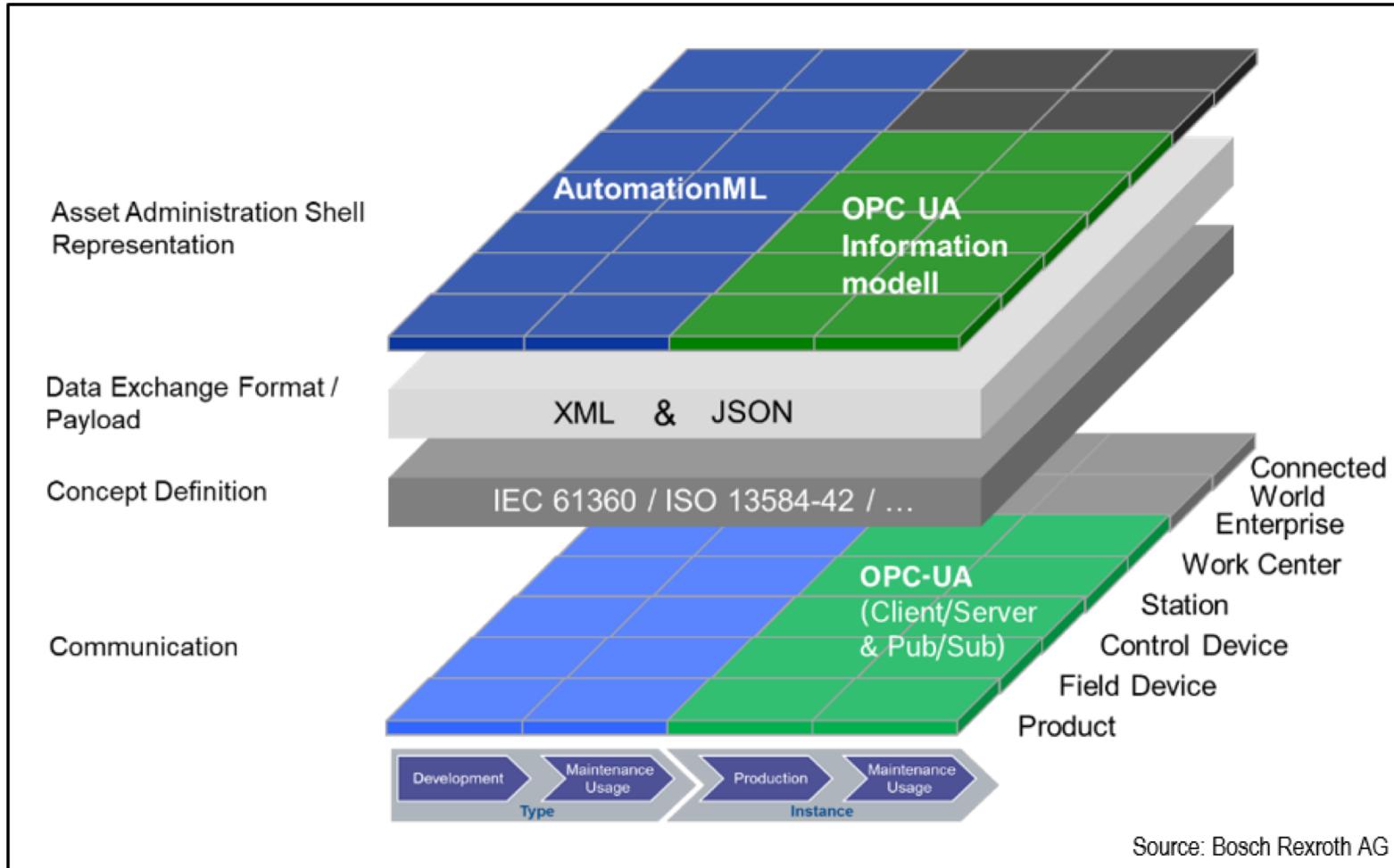
Datenaustausch zwischen unterschiedlichen Systemen (1)

Es ist möglich I40-konforme Informationen zwischen verschiedenen System im gesamten RAMI Modell auszutauschen. OPC UA dient als Hauptaustauschformat. Jedoch sind für verschiedene Anwendungen andere Austauschformate nötig.

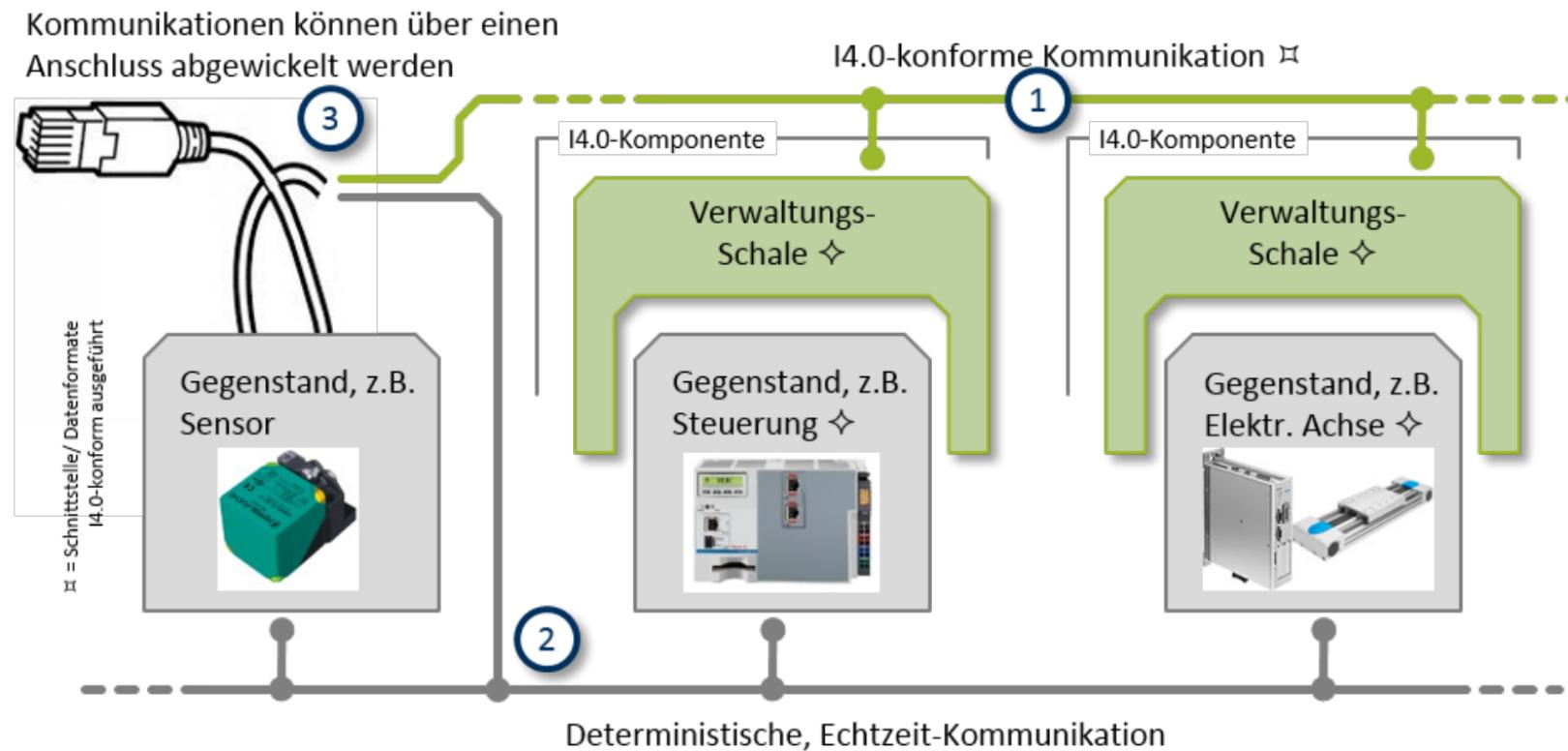
Datenformat	Verwendung
OPC UA Informations-Modell	Zugriff auf alle Verwaltung- und Livedaten
AutomationML	Austausch von Typ- und Instanz-Informationen aus dem Engineering-Prozess
XML, JSON	Serialisierung von Informationen für technische Kommunikation
RDF	Vernetzung von Informationen (Semantische Technologien)

Datenaustausch zwischen unterschiedlichen Systemen (2)

Datenaustauschformate für AAS im RAMI4.0



Kapselfähigkeit und Vernetzung einer I4.0-Komponente



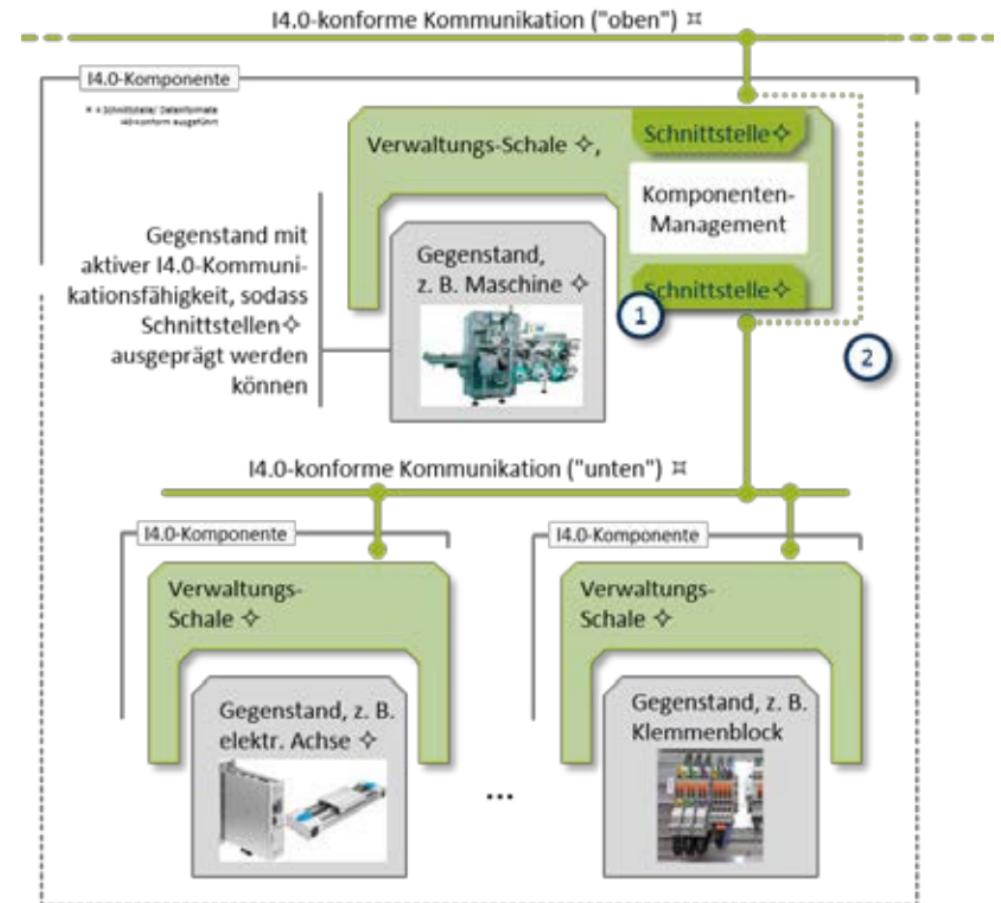
1. I4.0-Komponenten sollen alle möglichen Querverbindungen innerhalb einer I4.0-Fabrik eingehen bzw. aufbauen können.
2. Diese Vernetzung darf nicht zur Einschränkung der Kernfunktionalität führen. Die Fähigkeit, diesen Kernbereich störungsfrei zu erhalten, selbst wenn die „äußere“ Vernetzung Störungen unterliegt, wird als „kapselfähig“ bezeichnet.
3. Echtzeit-Ethernet-Protokolle ermöglichen es, beide Kommunikationen über die gleiche Kommunikationsinfrastruktur (Anschlüsse, Stecker, Zwischenstationen) abzuwickeln. Beide Kommunikationsarten sind aber logisch weiterhin getrennt.

Aufbau von I4.0-Komponenten

I4.0-Komponente aus mehreren Gegenständen

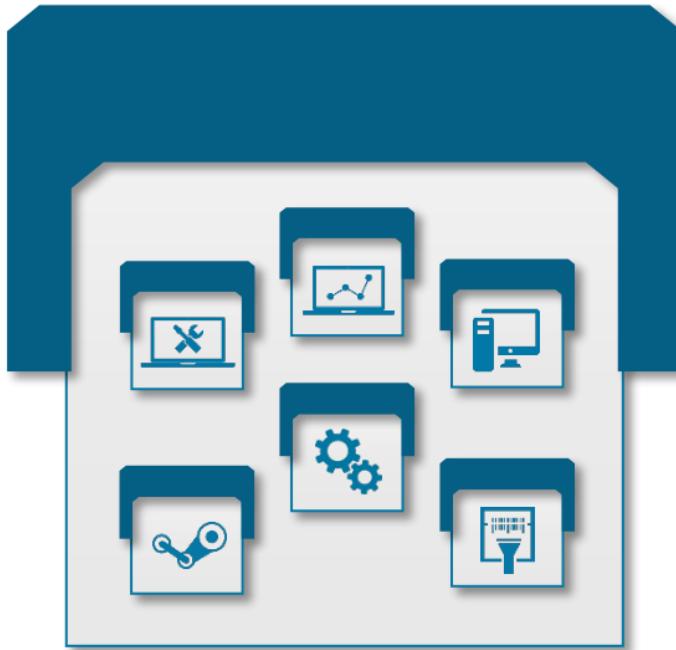


Schachtelbarkeit von I4.0-Komponenten

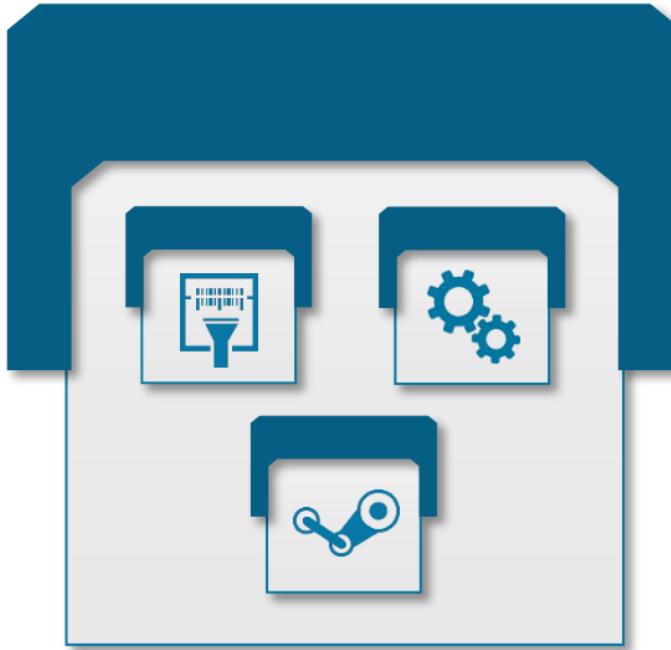


I4.0-Komponente & Hierarchieebenen

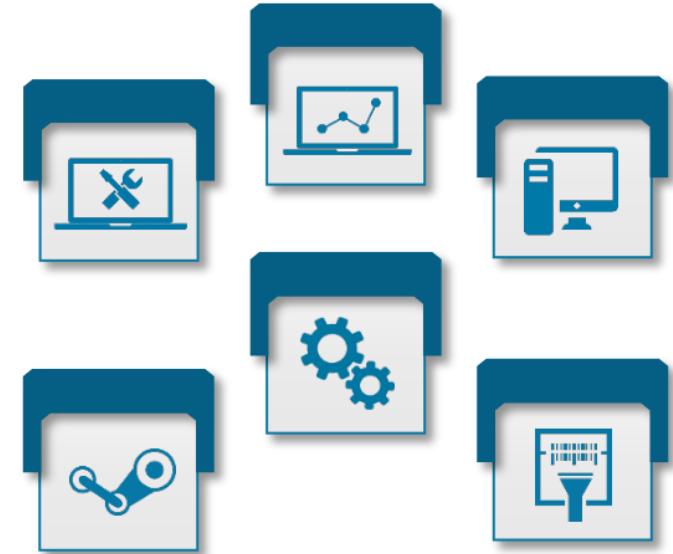
Station



Control Device

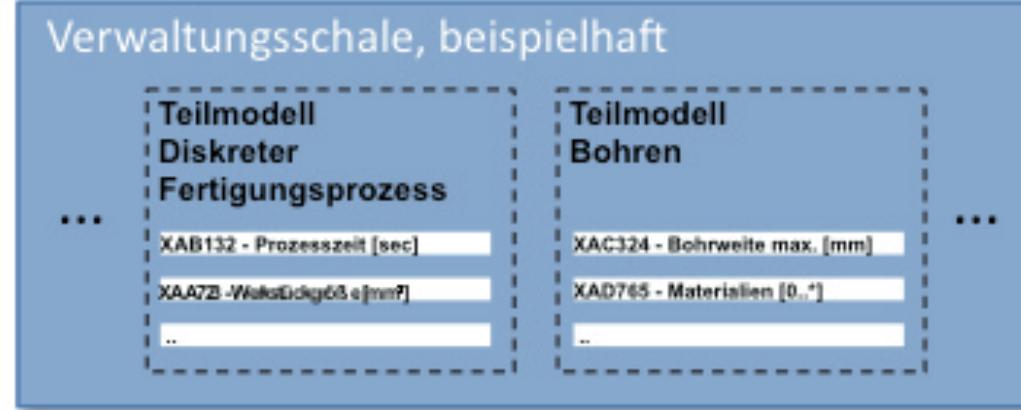


Field Device



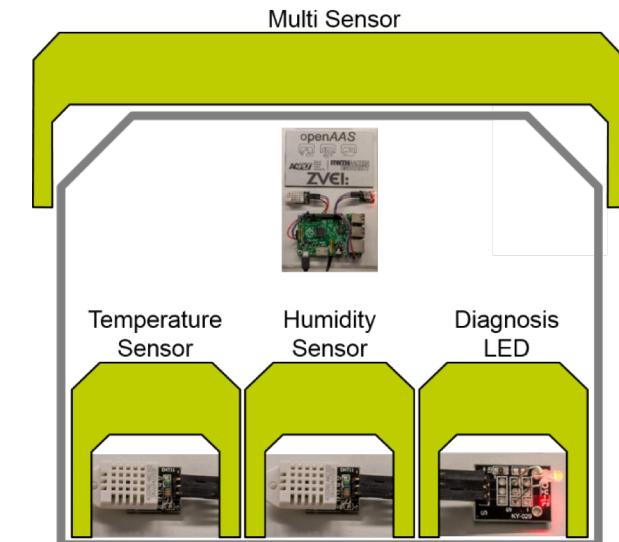
Interaktionsmuster mit der AAS

»Ist der Fertigungsprozess **Bohren** möglich, mit der
Werkstückgröße $\leq 500 \times 300 \text{mm}$, einem
Bohrungsdurchmesser von $\leq 12 \text{mm}$, dem
Material V2A-Stahl, in einer **Prozesszeit $\leq 3 \text{sec}$** ?«



open Asset Administration Shell (openAAS)

- Offene Entwicklung der RWTH Aachen und des ZVEI
- Erste Implementierung der Verwaltungsschale (asset administration shell)
 - Asset Administration Shell (structure)
 - Message based interaction model
 - Property model
 - Lifecycle model
- openAAS verwendet den OPC UA stack open62541
- URL: <https://github.com/acplt/openAAS>



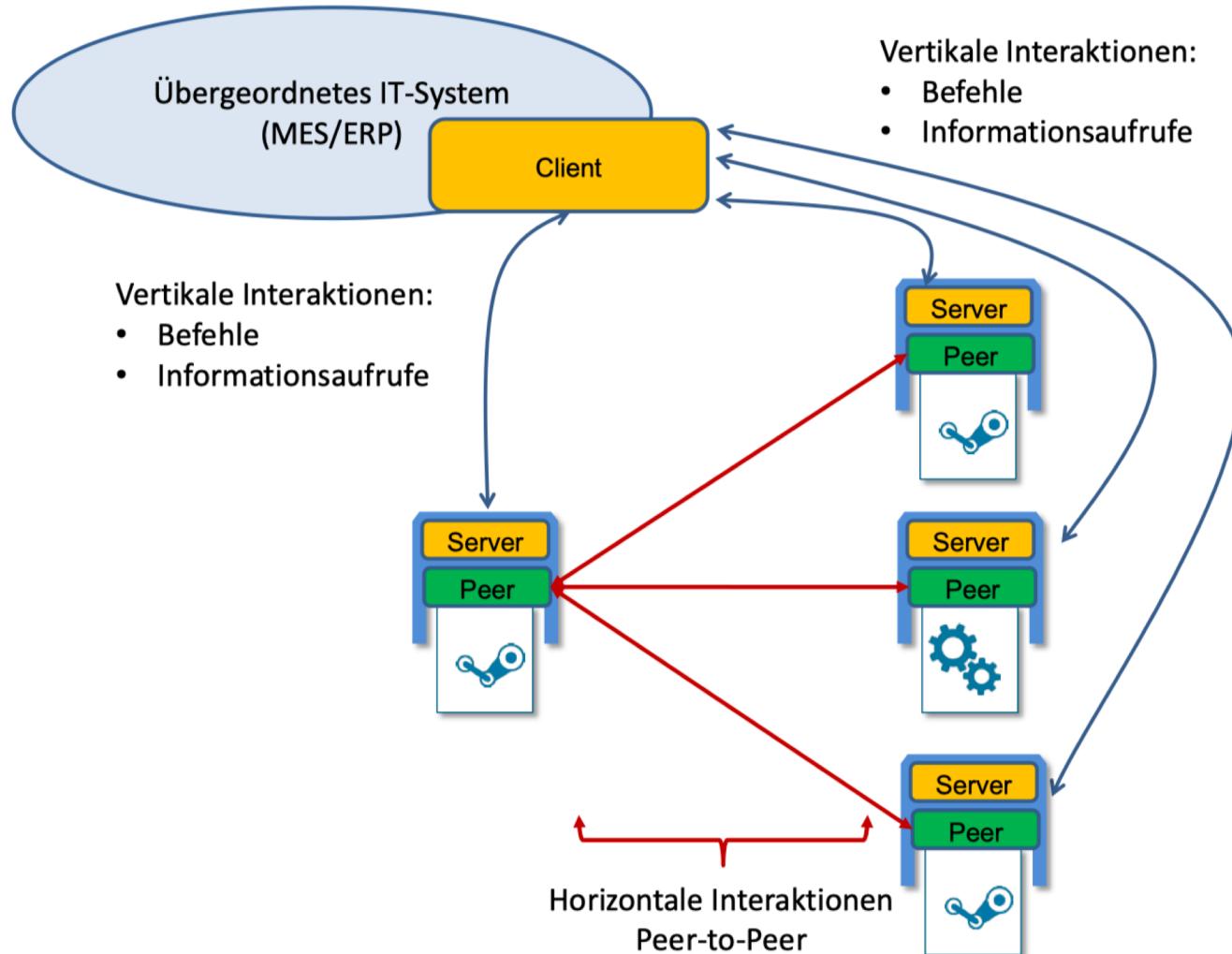
Vertikale Interaktion (Objekt-/dienstorientiert)

- Funktionalität durch Operationen zugreifbar
- Dienstsystem, z. B. SOA-basiert
 - Synchron
 - Hierarchisch
 - Client/Server
 - Enge Kopplung
- geeignet für Kommunikations-, Informations- und Plattformdienste

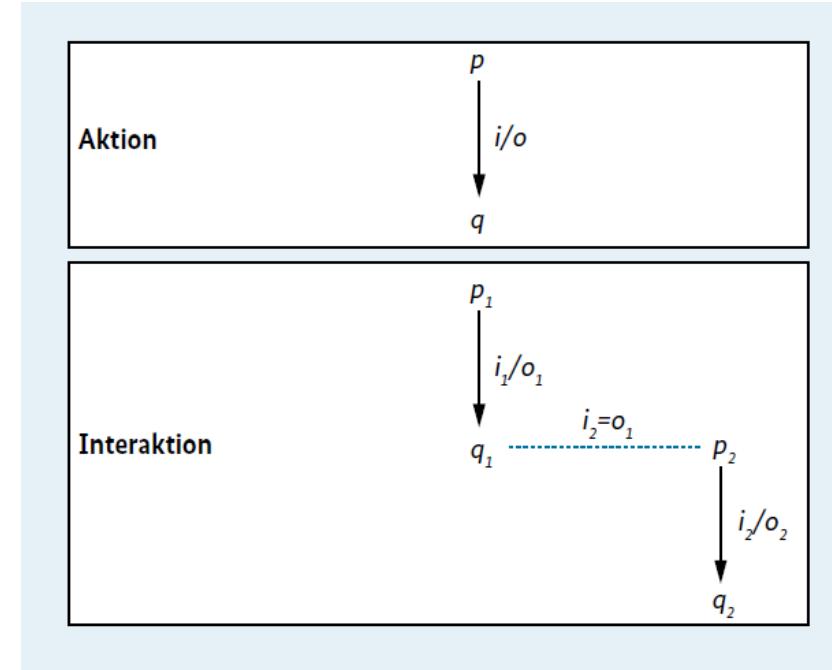
Horizontale Interaktion (Protokollorientiert)

- Abstraktion der Funktionalität durch Automaten
- Protokollautomaten
 - Asynchron
 - Horizontal
 - Peer-to-Peer
 - Lose Kopplung
- geeignet für Anwendungsaufgaben, z. B. Verhandlung, komplexe Maschinensteuerungen

Interaktion zwischen I4.0-Komponenten



- Als **Aktion** wird ein Zustandsübergang eines Systems bezeichnet, der sich aus den Eingabewerten, den inneren Zustandswerten vor und nach der Transition sowie den Ausgabewerten zusammensetzt.
- Als **Interaktion** wird die Kombination von Zustandsübergängen wenigstens zweier Systeme bezeichnet, bei der die Ausgabewerte eines Systems („Sender“) die Eingabewerte des weiteren Systems („Empfänger“) darstellen.

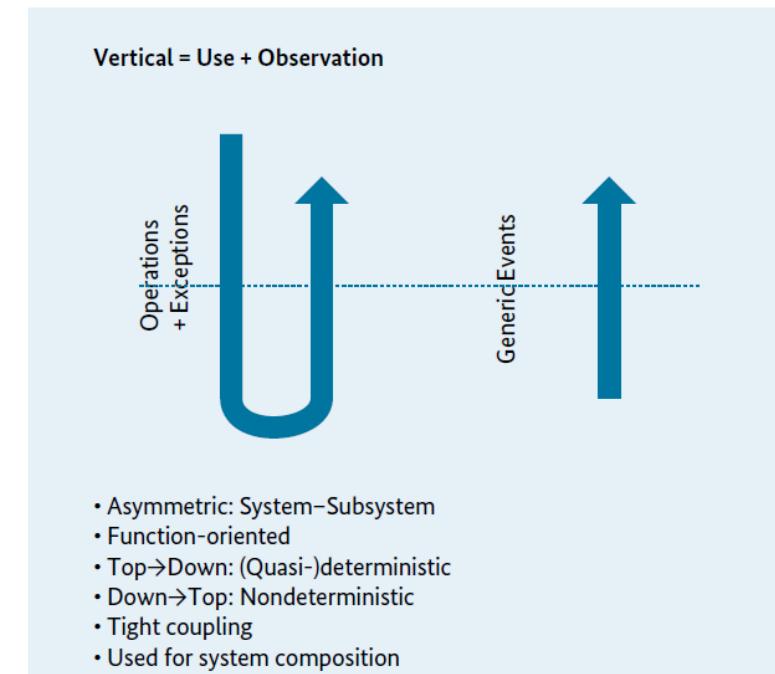


- Interaktionen lassen sich auf Basis der Art der Informationsverarbeitung von Sender und Empfänger klassifizieren:
 - **Synchronizität:** Der Sender wartet auf die Berechnung des Ergebnisses des Empfängers (synchroenes Verhalten des Senders) oder nicht (asynchrones Verhalten).
 - **Determinismus:** Der Zustandsübergang des Empfängers ist aus Sicht des Senders durch die von ihm gesendeten Informationen vollständig festgelegt (deterministisches Verhalten des Empfängers) oder nicht (nicht-deterministisches Verhalten).
 - **Zustandsbehaftung:** Der Zustandsübergang des Empfängers basiert aus Sicht des Senders ausschließlich auf den zuletzt gesendeten Informationen (zustandsloses Verhalten des Empfängers) oder auf zusätzlichen inneren Zustandswerten (zustandsbehaftetes Verhalten).

- Werden innerhalb von Interaktionen die Rollen von Sender und Empfänger vertauscht, lassen sich zwei grundsätzlich verschiedene Klassen von Interaktionen unterscheiden:
 - Bei **symmetrischen Interaktionen** verhalten sich alle Parteien bezogen auf die drei angeführten Dimensionen auf dieselbe Art und Weise
 - Bei **asymmetrischen Interaktionen** verhalten sich die Parteien unterschiedlich.

Formale Grundlagen des Interaktionsmodells

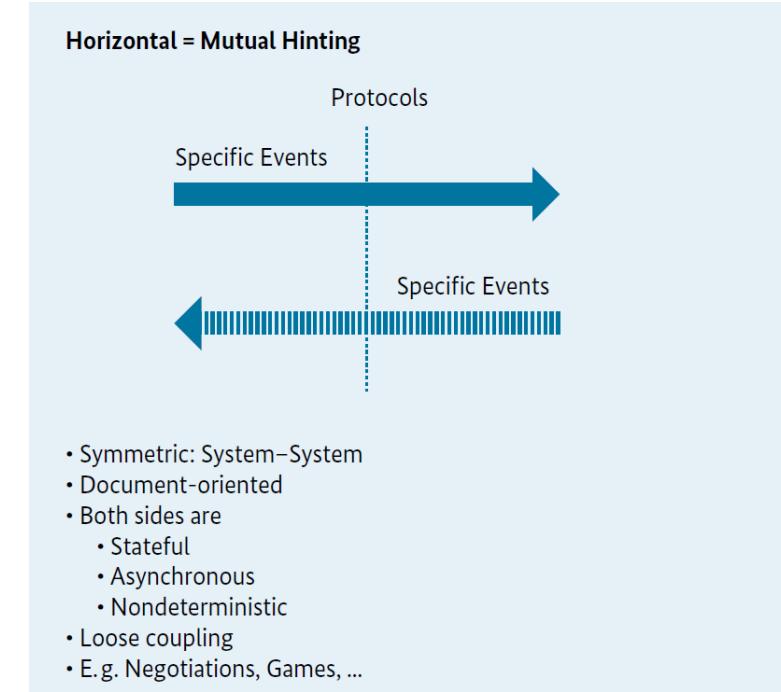
- Die wichtigste asymmetrische Interaktionsform ist die „**Verwendung**“:
 - Die Richtung wird durch den Determinismus vorgegeben. Der Verwender bestimmt über die verwendete Komponente. Umgekehrt ist die Beziehung nichtdeterministisch.
 - Durch die Verwendungsbeziehung entsteht aus logischer Sicht eine enge Kopplung im Sinne einer Teil-Ganzes-Beziehung.
- Syntaktisch lässt sich der **Top-Down-Anteil** der Verwendung durch eine objektorientierte Operation beschreiben, d.h. als Funktion, die auf einem definierten Datenkontext operiert. Gegebenenfalls wird ein nicht-deterministischer Teil mittels Ausnahmebehandlung separiert, der dann den Kontrollfluss beeinflusst.
- Der **Down-Top-Anteil** der Verwendung ist ein generisches Event. Dienste sind Vertreter dieser Interaktionsform.



Quelle: [Bock 2016]

Formale Grundlagen des Interaktionsmodells

- Die wichtigste symmetrische Interaktionsform ist das „**Protokoll**“. Hier agiert jede Partei im Allgemeinen aus Sicht der anderen Parteien asynchron, nicht-deterministisch und zustandsbehaftet.
 - Auf Grund des Nicht-Determinismus lässt sich den transportierten Informationen in der Regel keine imperative Semantik zuordnen.
 - „Dokument“-Charakter: Der eine Interaktionspartner dokumentiert seinen Zustandsübergang auf eine Art und Weise, dass die davon betroffenen Interaktionspartner ihrerseits ihren Zustandsübergang durchführen können usw.
- Diese Interaktionsform kommt zum Einsatz, wenn die Unwägbarkeiten zwischen den Interaktionspartnern so groß sind, dass sie nicht mehr Ausnahmen sondern einen wesentlichen Anteil der Beziehung zwischen den Akteuren darstellen.
- Dies ist der Fall, wenn die Akteure in weitere Interaktionen auf derselben semantischen Ebene eingebunden werden und gewisse Freiheiten benötigen, um alle ihre Interaktionen zu koordinieren.
- Diese Interaktionsform kommt zum Einsatz, wenn Applikationen „lose“ gekoppelt miteinander interagieren sollen.

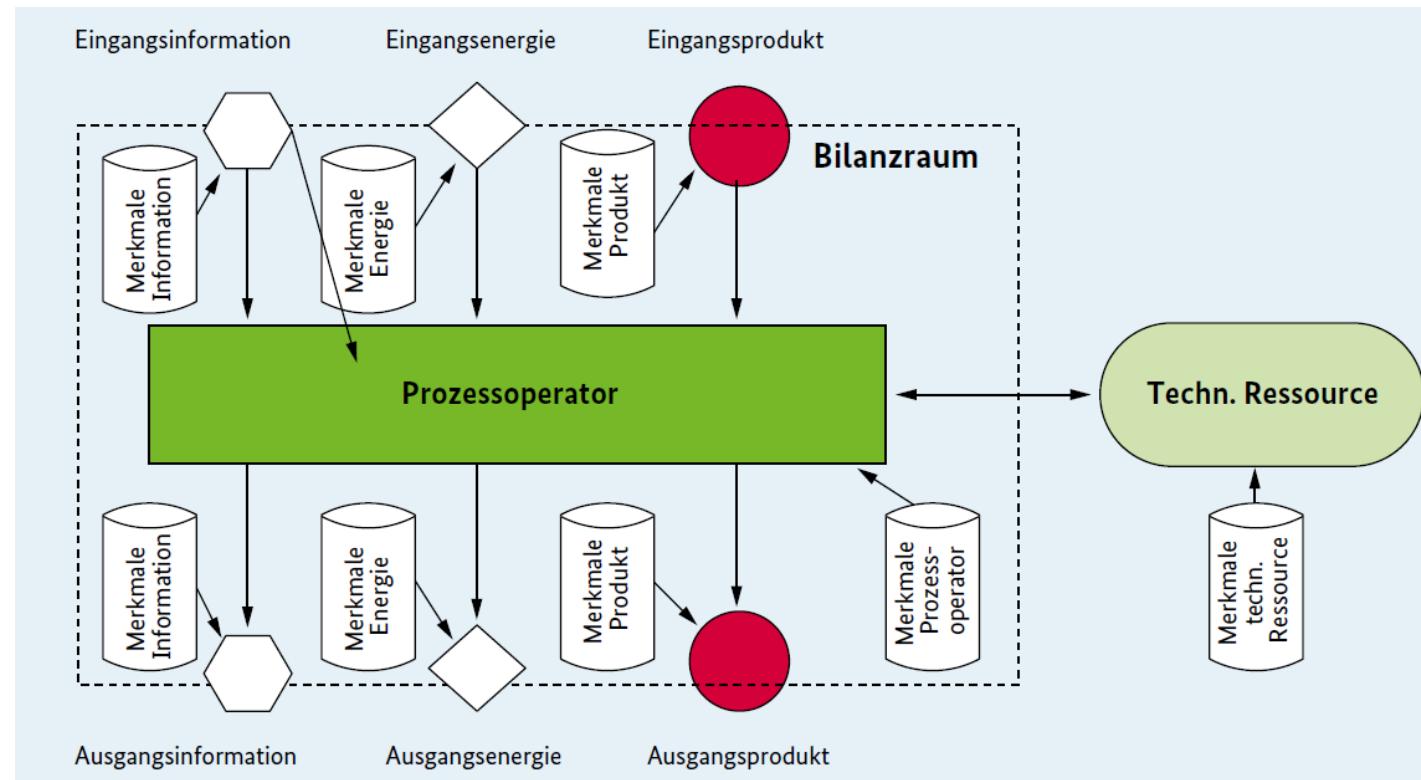


Quelle: [Bock 2016]

- Im Rahmen von Industrie 4.0 sind nicht nur Produktionsplanungs- und Steuerungsaufgaben, sondern auch Arbeitsplanungsaufgaben zu automatisieren, um automatisch und in Echtzeit angemessen auf die momentane Situation reagieren zu können.
- Nach [Bock 2016] ist die **Formalisierte Prozessbeschreibung (VDI/VDE 3682)** ein Vorgehensmodell, um
 - eine *durchgängige informationstechnische Prozessbeschreibung* und
 - eine über den gesamten Lebenszyklus eindeutige, strukturierte, vollständige, technisch wie kognitiv wiedergewinnbare Abbildung von Informationen
 - für *bestimmungsgemäßen Betrieb und Planung einer Anlage* zu entwickeln.

Grafische Repräsentation der formalisierten Prozessbeschreibung

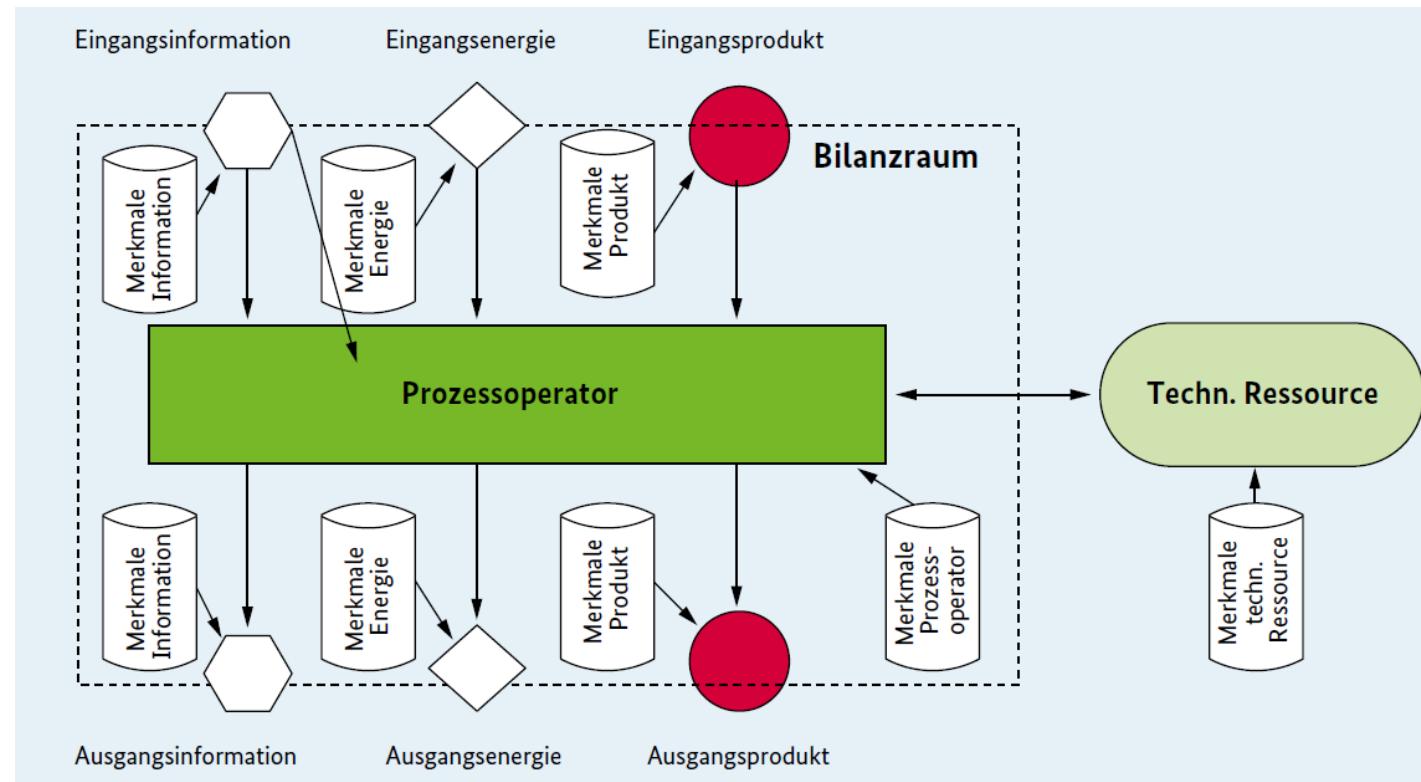
- Ein Produktionsprozess wird durch ein Netz beschrieben, bei dem physikalisch, chemisch, biologisch oder verfahrens- oder fertigungstechnisch definierte Zustände durch Prozessoperatoren verbunden sind.
- Durch diese Verbindung wird ein Anfangszustand (*ante*) in einen Endzustand (*post*) überführt.
- Ein Zustand wird hierbei stets
 - durch ein oder mehrere Produkte (Materialien, Stoffe etc.),
 - eine oder mehrere Energien und Informationenbeschrieben.



Quelle: [Bock 2016]

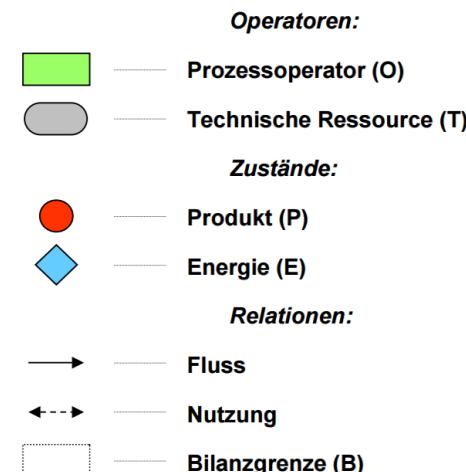
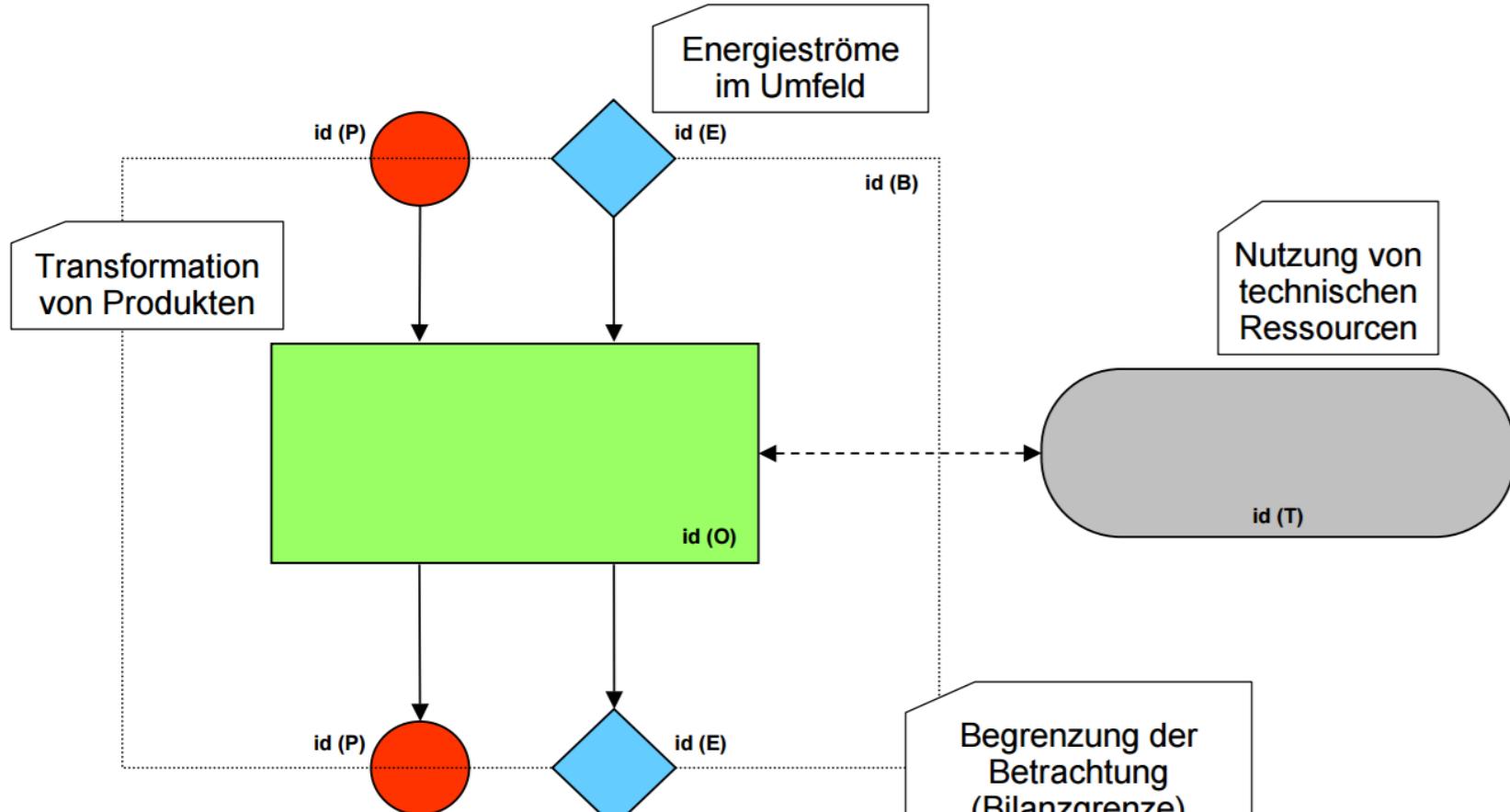
Grafische Repräsentation der formalisierten Prozessbeschreibung

- Der Prozessoperator wandelt somit Produkte, Energien und Informationen in neue Produkte, Energien und Informationen um und nutzt dabei technische Ressourcen wie zum Beispiel Anlagen, um die Umwandlung zu realisieren.
- Eine Abgrenzung des Systems zur Umwelt wird mit Hilfe des Bilanzraums sichergestellt, welcher das Netz an definierten Punkten umgibt.
- Die Strukturierung des Prozesses innerhalb der Bilanzgrenze erfolgt mit Hilfe von Komposition und Dekomposition ausschließlich an Prozessoperatoren.



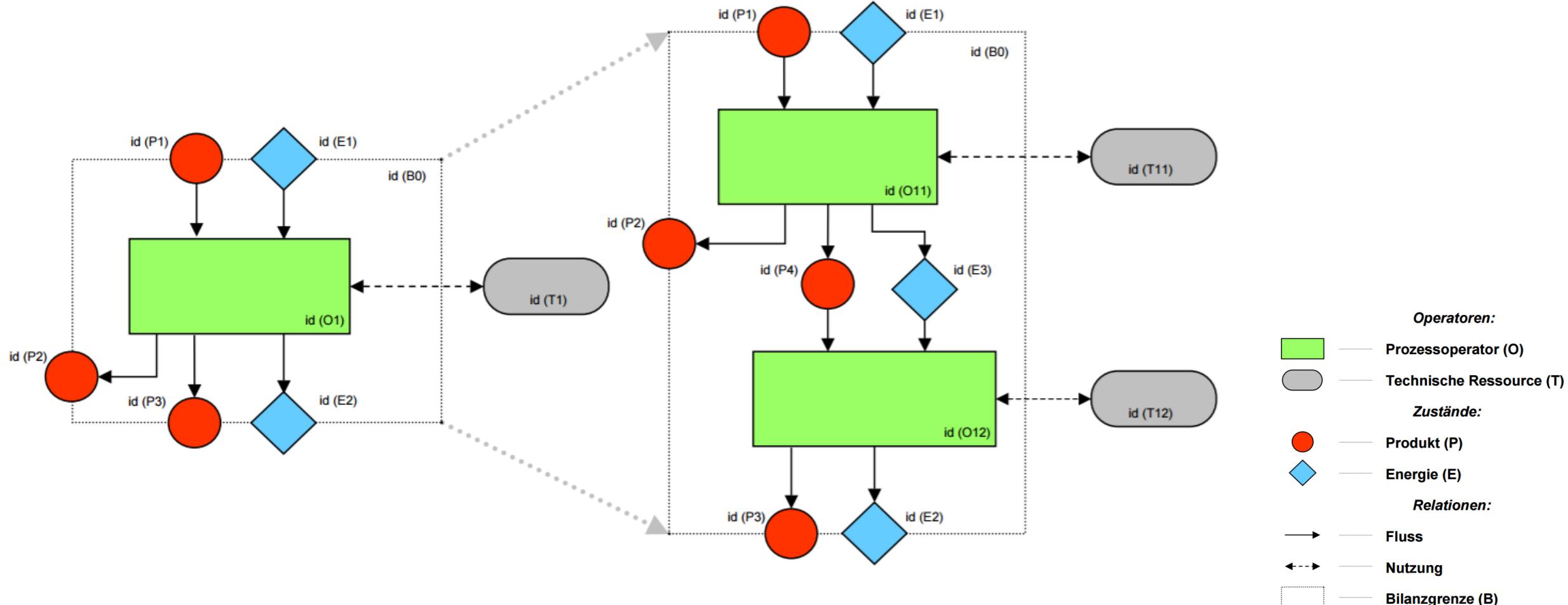
Quelle: [Bock 2016]

Grafische Repräsentation der formalisierten Prozessbeschreibung



Quelle: [Polke 2004]

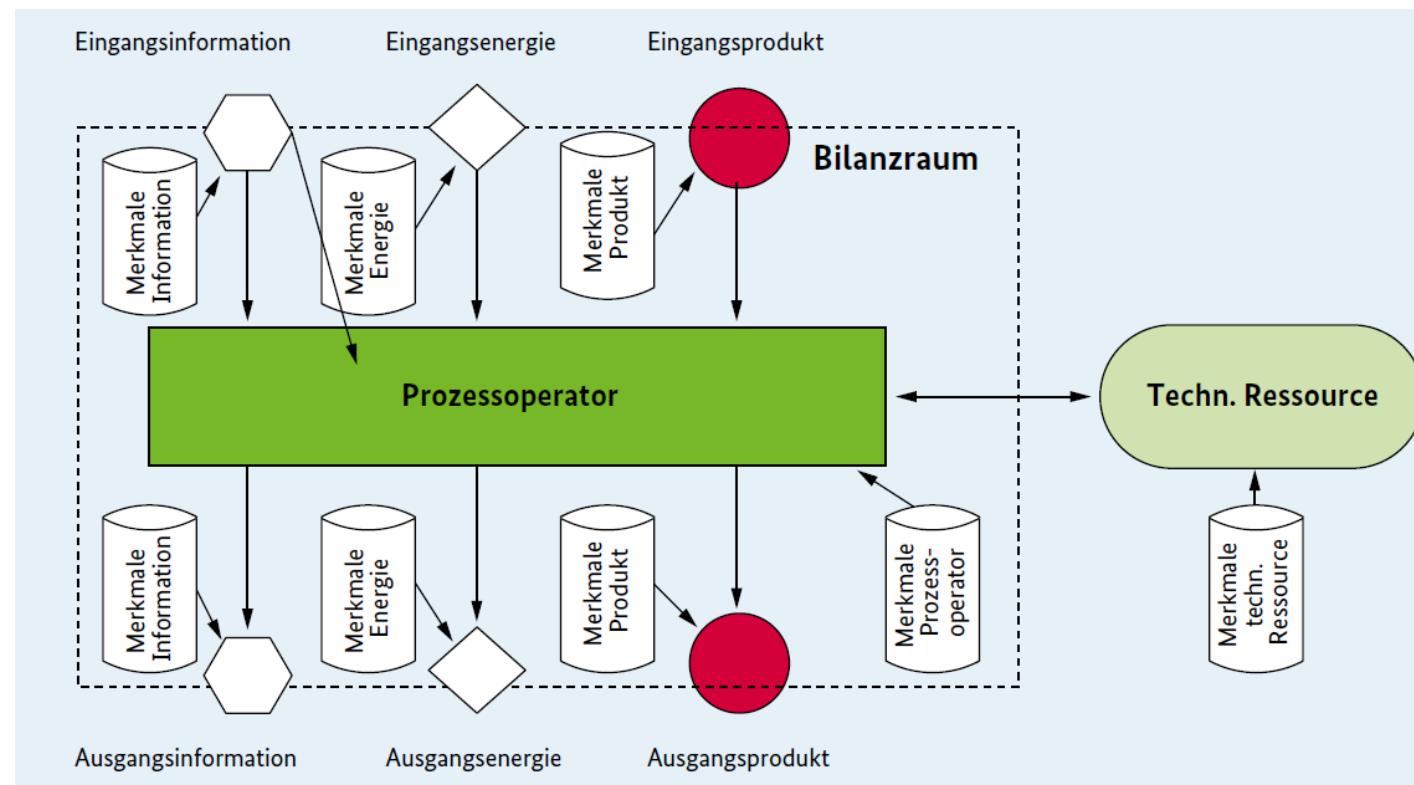
Grafische Repräsentation der formalisierten Prozessbeschreibung



Quelle: [Polke 2004]

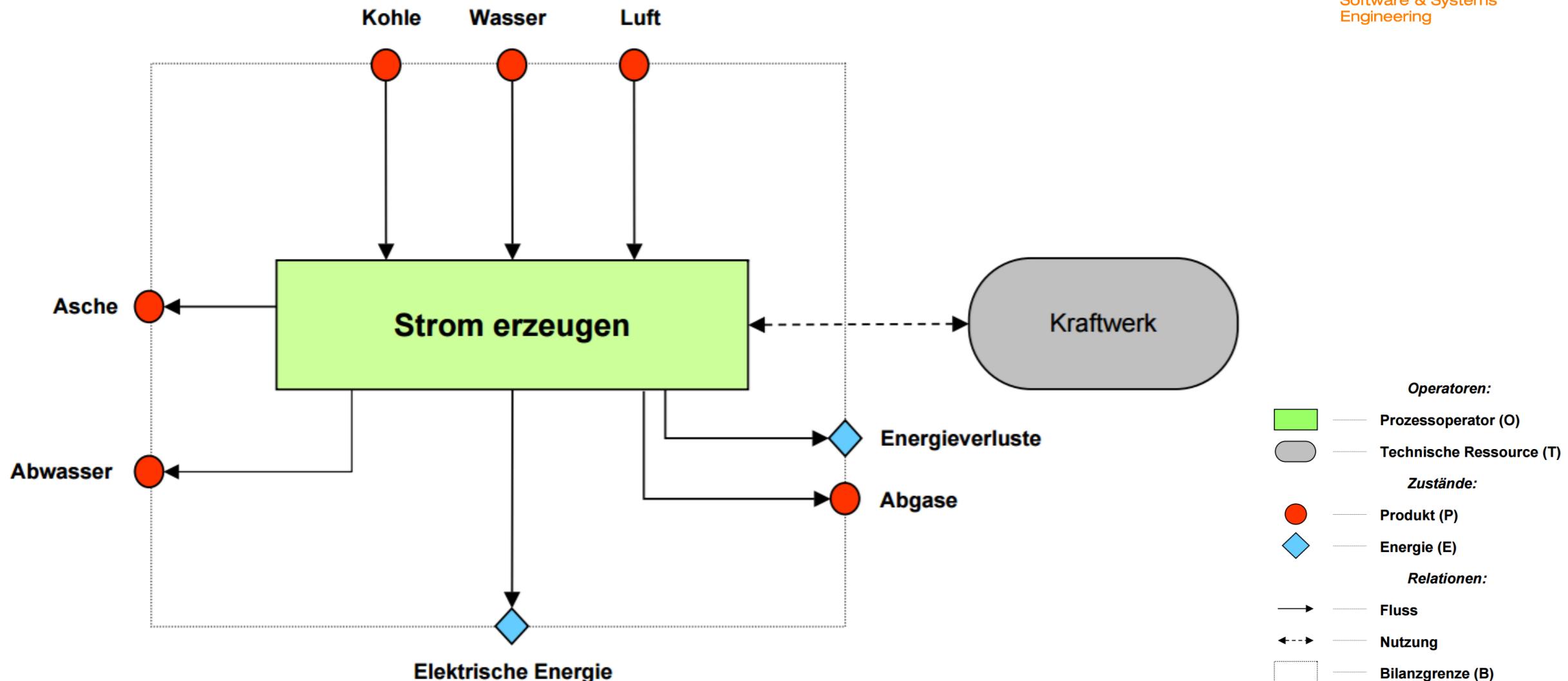
Grafische Repräsentation der formalisierten Prozessbeschreibung

- Jedes Element des Graphen (Produkt, Energie, Information, Prozess und technische Ressource) wird mit einem Satz an Merkmalen beschrieben, die in einem Informationsmodell gehalten werden.
- Die Verbindung aus Eingangsprodukt, transformierendem Prozess, Ausgangsprodukt und technischer Ressource wird als PPR-Bindung (Produkt-, Prozess-, Ressource-Bindung) bezeichnet.



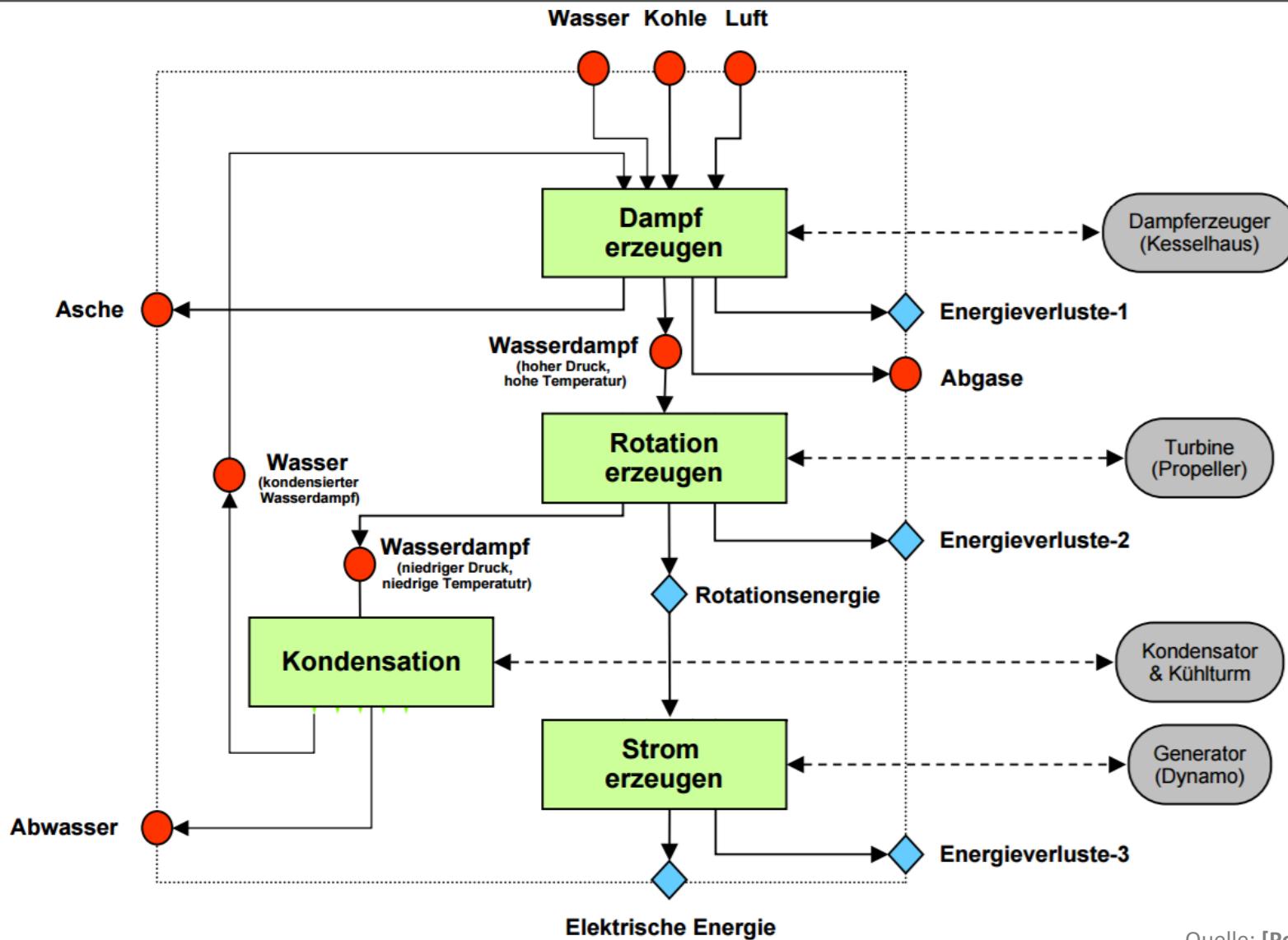
Quelle: [Bock 2016]

Beispiel für die formalisierte Prozessbeschreibung (I)

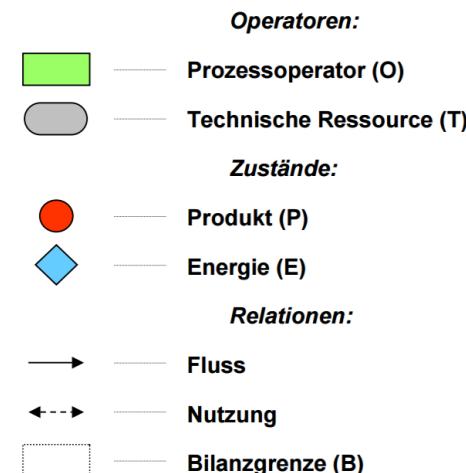


Quelle: [Polke 2004]

Beispiel für die formalisierte Prozessbeschreibung (I)



Quelle: [Polke 2004]

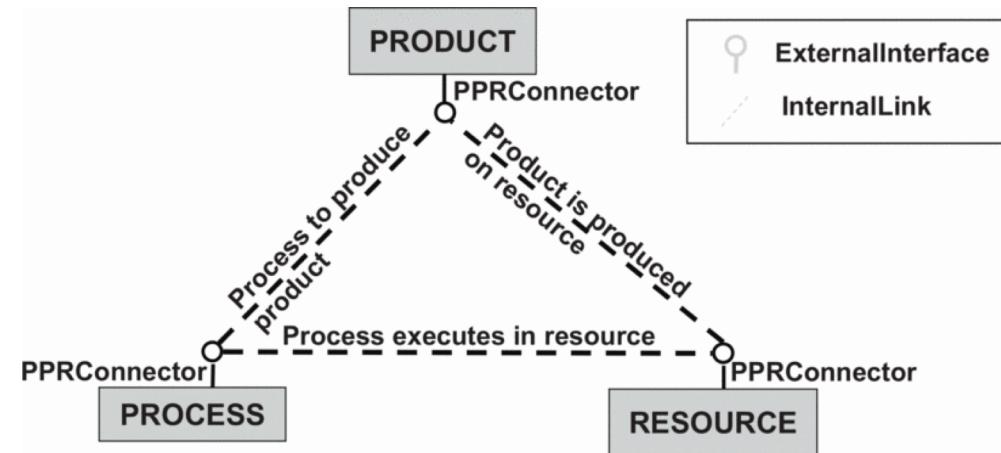


Beispiel für die formalisierte Prozessbeschreibung (II)

- Einbringen einer Bohrung in ein Werkstück:
 - Veränderung geometriebezogenen Merkmale des Eingangsprodukts
 - Es können auch weitere Merkmale des Produkts von Bedeutung sein, wie z. B. die Materialzähigkeit.
 - Die Produktumwandlung wird durch den Prozessoperator, d.h. das Bohren, beschrieben.
- Eine Bohrmaschine, eine Laserschneidmaschine oder eine Fräsmaschine kann die technische Ressource sein, die die Produktumwandlung ausführt.
- Zu den Merkmalen des Prozesses gehören z.B. Toleranzen der Bohrung, Oberflächenrauigkeit und die Dauer der Umwandlung.
- Die technische Ressource hat Merkmale, die Fähigkeiten der Maschine beschreiben, wie z.B. mögliche Bohrdurchmesser, Rotationsgeschwindigkeiten, den Bereich der Eingriffswinkel und Qualitätsmerkmale (z.B. Toleranzen).

Dynamisches Erstellen der PPR-Bindung

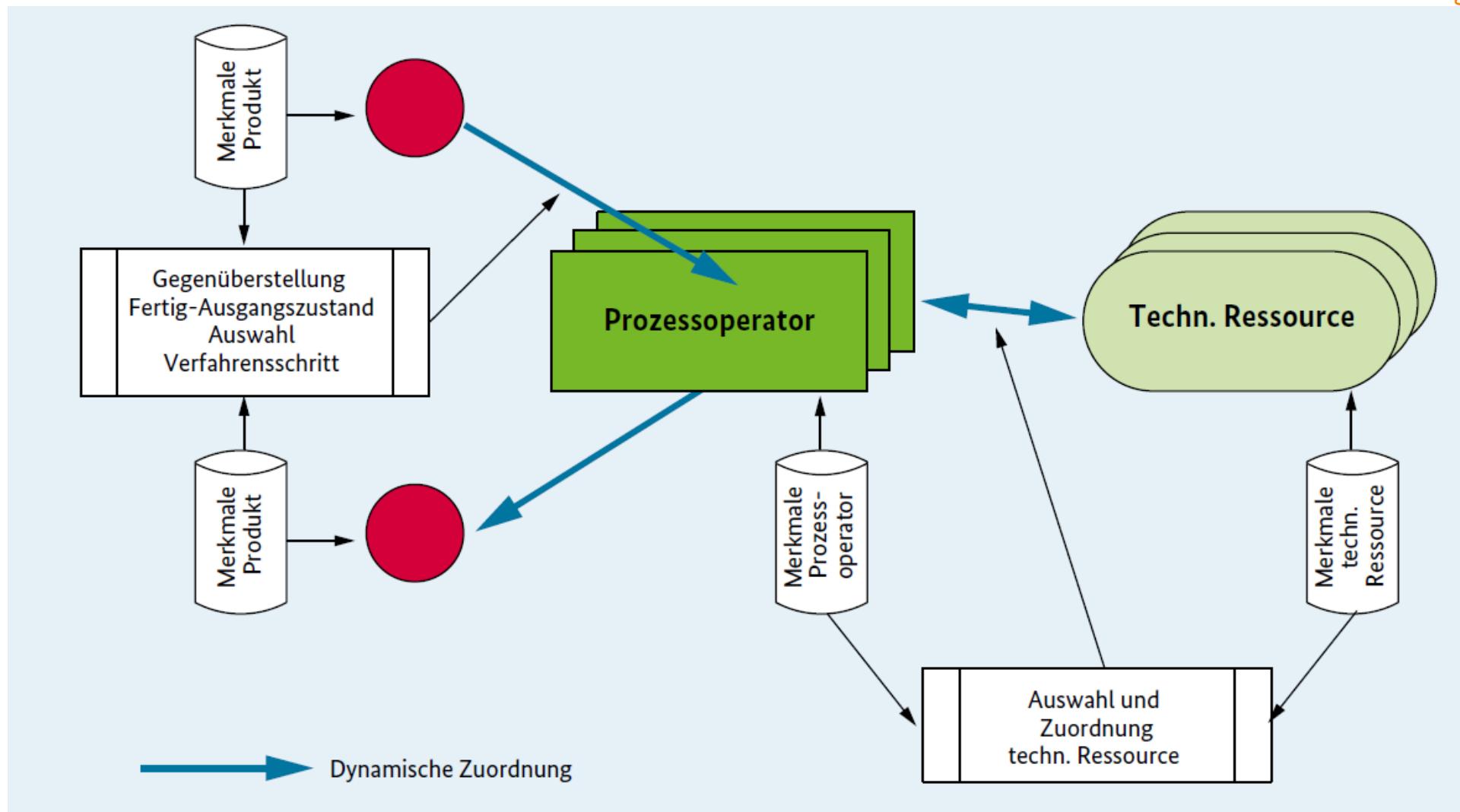
- Um einen Produktionsschritt durchzuführen, muss die Umwandlung vom Eingangs- zum Ausgangsprodukt festgelegt und das dafür benötigte Produktionssystem (technische Ressource) ausgewählt werden.



Quelle: [Schleipen 2009]

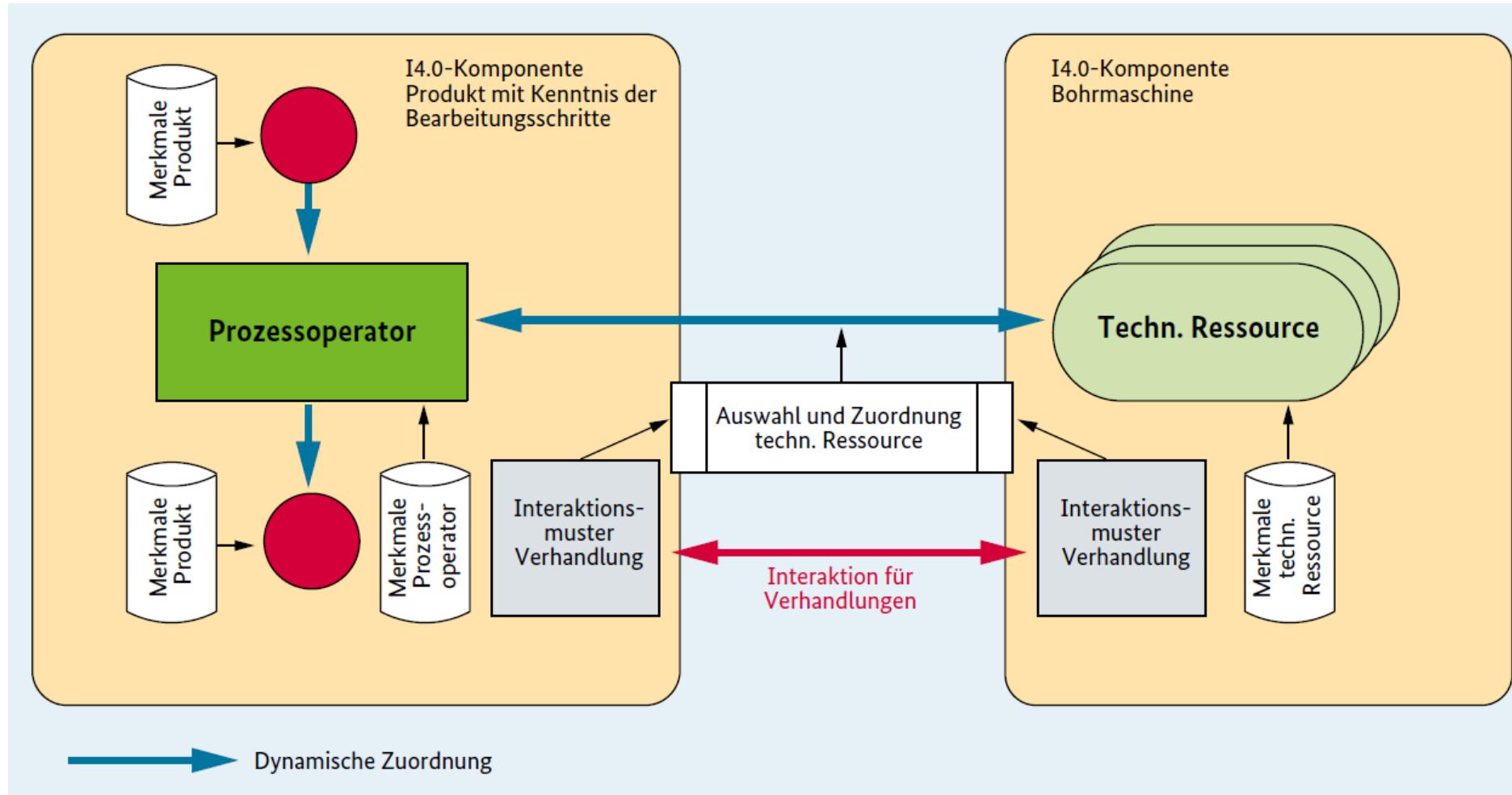
- In I4.0-Systemen soll diese Zuordnung des Prozessschrittes zu den technischen Ressourcen und gegebenenfalls auch der Prozessschritt für die Produktumwandlung nicht manuell, sondern durch Zuordnung während des operativen Betriebs erfolgen.

Dynamisches Erstellen der PPR-Bindung



Quelle: [Bock 2016]

Zuordnung von Prozessoperator und technischer Ressourcen



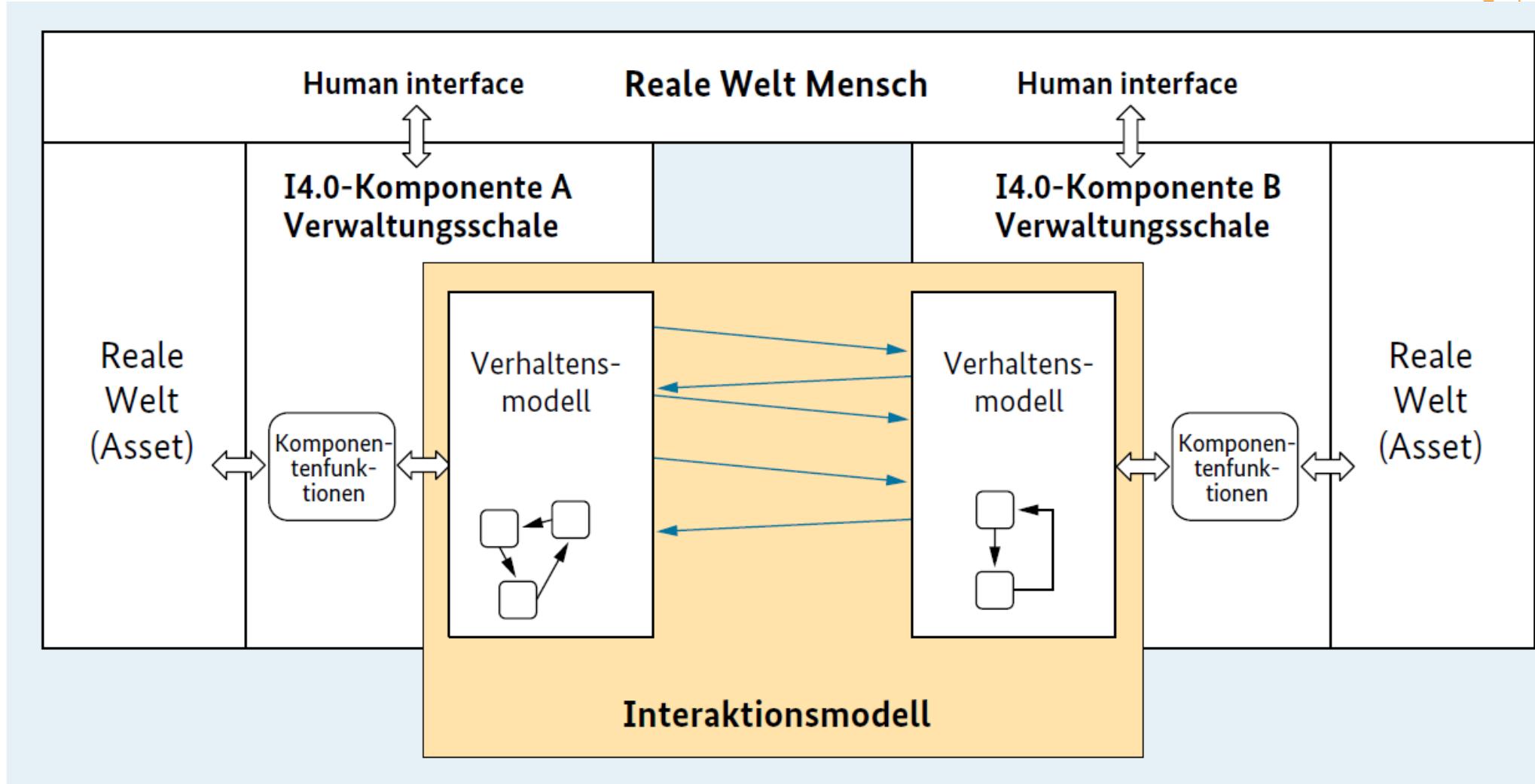
Quelle: [Bock 2016]

- Die Zuordnung von Prozessoperator und technischer Ressourcen soll durch **Verhandlung zwischen den I4.0-Komponenten** erfolgen. Diese **Verhandlung benötigt Interaktionen**.
- Weitere Interaktionen werden während des operativen Betriebs zur Steuerung und Beobachtung der Produktionsprozesse sowie für Instandhaltungs-, Wartungs- und Diagnoseprozesse an den technischen Ressourcen benötigt.
- **Für einzelne Interaktionen werden Muster vereinbart**, damit zwischen den Interaktionsteilnehmern ein gegenseitiges Verständnis möglich ist.
- Verhandlungsinteraktionen und Interaktionen im operativen Prozess und in Instandhaltungsprozessen sind typische Muster, die beschrieben werden.

Interaktionen zwischen I4.0-Komponenten

- Die Funktionen der Assets einer I4.0-Komponente werden über das Interaktionsmodell zugreifbar gemacht. Die Assets werden durch ihre Komponentenfunktionen in der Verwaltungsschale steuerbar.
- Das Interaktionsmodell verwendet Nachrichten zwischen mindestens zwei I4.0-Komponenten, die Zustandsübergänge in der I4.0-Komponente hervorrufen können und somit das Verhalten beeinflussen.
- Das Interaktionsmodell ist das Bindeglied zwischen den I4.0-Komponenten, das letztlich zu einem I4.0-System führt.
- Das Verhaltensmodell beruht auf gekoppelten Automaten.

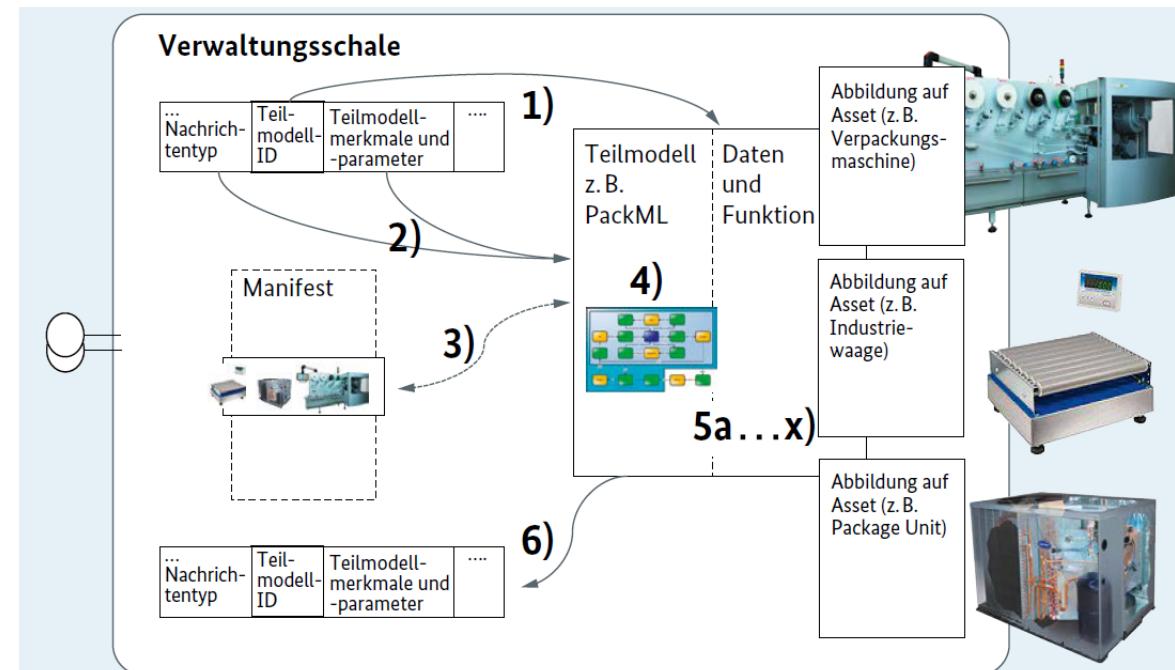
Interaktionen zwischen I4.0-Komponenten



Quelle: [Bock 2016]

Prinzipieller Ablauf von Interaktionsmustern

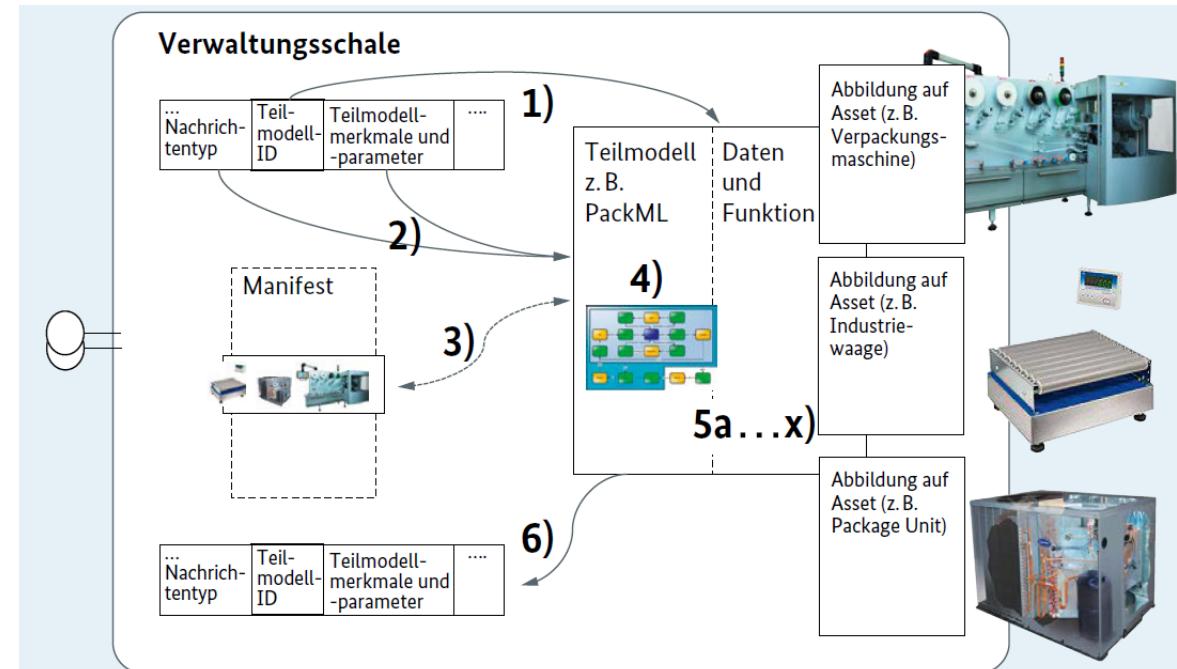
- Mit einer Folge von Nachrichten, die zwischen den I4.0-Komponenten ausgetauscht werden, entsteht eine Interaktion zwischen diesen. Die Nachrichten, die Abfolge der Nachrichten für die Erfüllung der Aufgabe sowie das Verhalten der beteiligten Teilmodelle bilden zusammen ein Interaktionsmuster.
- Folgende Schritte werden durchlaufen:
 1. Zuordnung der eingehenden Nachricht zu einem Teilmodell
 2. Übernahme der entsprechenden Parameter
 3. Überprüfung mittels Eintragung im Manifest und den lokalen Daten und Funktionen, ob Parameter zum Teilmodell passen
 4. Interpretation der Aufgabe
 5. Abwicklung der Aufgabe mit den lokalen Daten und Funktionen
 6. Ggf. Bereitstellung einer Antwort



Quelle: [Bock 2016]

Prinzipieller Ablauf von Interaktionsmustern

- Die Inhalte der Nachrichten, d.h. die Parameter und Merkmale, werden anhand der im Manifest enthaltenen Teilmodelle der I4.0-Komponente überprüft.
- Dies ist möglich, weil die Parameter und Merkmale durch IDs in den Nachrichten benannt werden, die damit auf ihre Typbeschreibungen im Manifest verweisen.
- Dadurch können Teilmodelle individuelle Parameter- und Merkmalsignaturen aufweisen, die durch die Eintragung im Manifest für die Partner erkundbar ist.



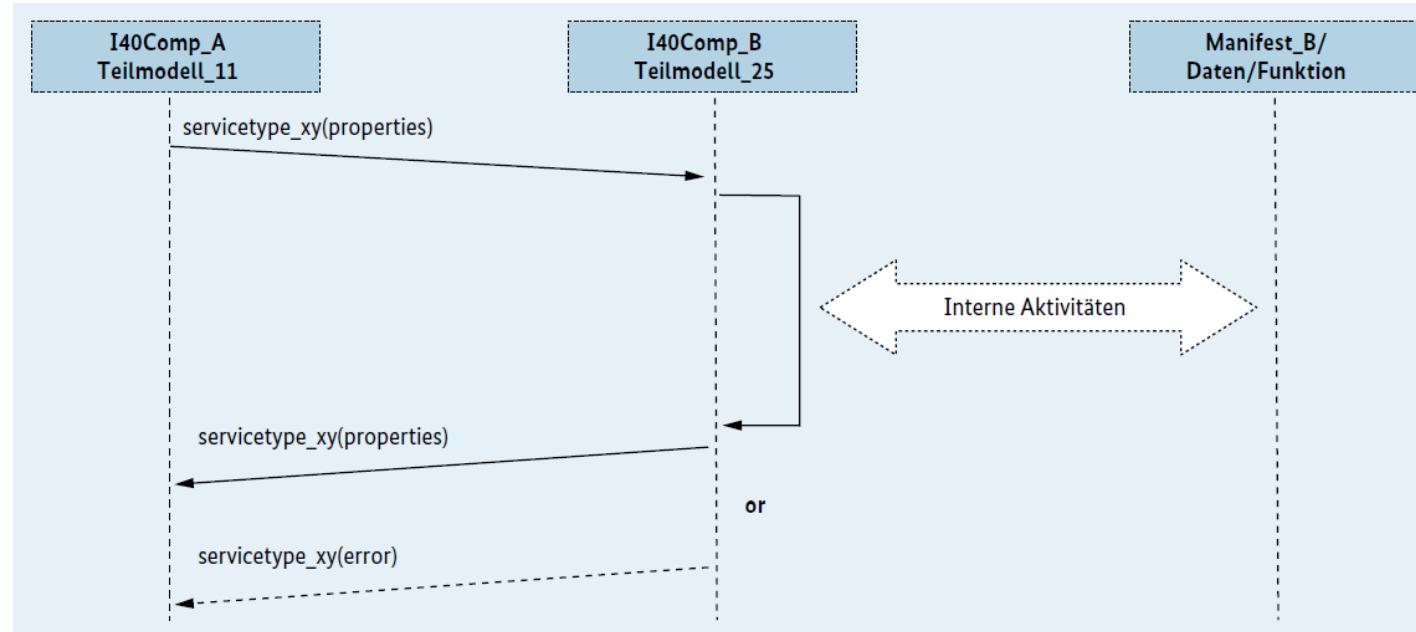
Quelle: [Bock 2016]

Beispiele Interaktionsmuster

Interaction type	Kurzbeschreibung
Feststellung der Identität	Die I4.0-Komponenten, die miteinander zusammenarbeiten wollen, stellen gegenseitig ihre Identität fest → Input von AG3
Vereinbarung der Security-Maßnahmen	Die I4.0-Komponenten, die miteinander zusammenarbeiten wollen, legen die zu verwendenden Security-Maßnahmen fest und stellen diese ein → Input von AG3
Anfrage für eine Aufgabe an eine I4.0-Komponente	Anfrage an eine durch ein Teilmodell bereitgestellte Funktionalität der Verwaltungsschale, ob die geforderten Ausprägungen der Merkmale zur Nutzung des Teilmodells durch die I4.0-Komponente, repräsentiert durch die Eintragungen der Merkmale im Manifest, erfüllt werden können. Angefragte I4.0-Komponente hat mit einer positiven oder negativen Entscheidung zu antworten.
Aushandlung einer Aufgabe mit einer I4.0-Komponente	Abfrage der Merkmalausprägungen der durch ein Teilmodell bereitgestellten Funktionalität der Verwaltungsschale. Gegebenenfalls notwendige Anpassung der angeforderten Merkmalausprägung wird von der anfragenden I4.0-Komponente vorgenommen.
Beauftragung einer Aufgabe an eine I4.0-Komponente	Auftragserteilung mit Angabe der zu verwendenden Merkmalwerte.
Durchführung einer Aufgabe in einer I4.0-Komponente	Direkte Steuerung der durch das Teilmodell bereitgestellten Funktionalitäten. Konfiguration deterministischer Echtzeitkommunikation durch in Teilmodellen referenzierte Signale.
Beenden einer Aufgabe initiiert durch die beauftragte I4.0-Komponente	Abbruch einer Aufgabe, wenn dies durch den Zustand oder andere begleitende Bedingungen der beauftragten I4.0-Komponente erforderlich ist.
Beenden einer Aufgabe initiiert durch die beauftragende I4.0-Komponente	Abbruch einer Aufgabe, wenn dies durch den Zustand oder andere begleitende Bedingungen der beauftragenden I4.0-Komponente erforderlich ist.
Melden von Störungen von einer I4.0-Komponente bei der Verhandlung	Melden von Störungen ohne Abbruch der Verhandlung.
Melden von Störungen von einer I4.0-Komponente bei der Auftragsabarbeitung	Melden von Störungen ohne Abbruch des Auftrags durch die beauftragte I4.0-Komponente. Es kann ein Fehlerzustand eingenommen werden.

Prinzipieller Ablauf von Interaktionsmustern

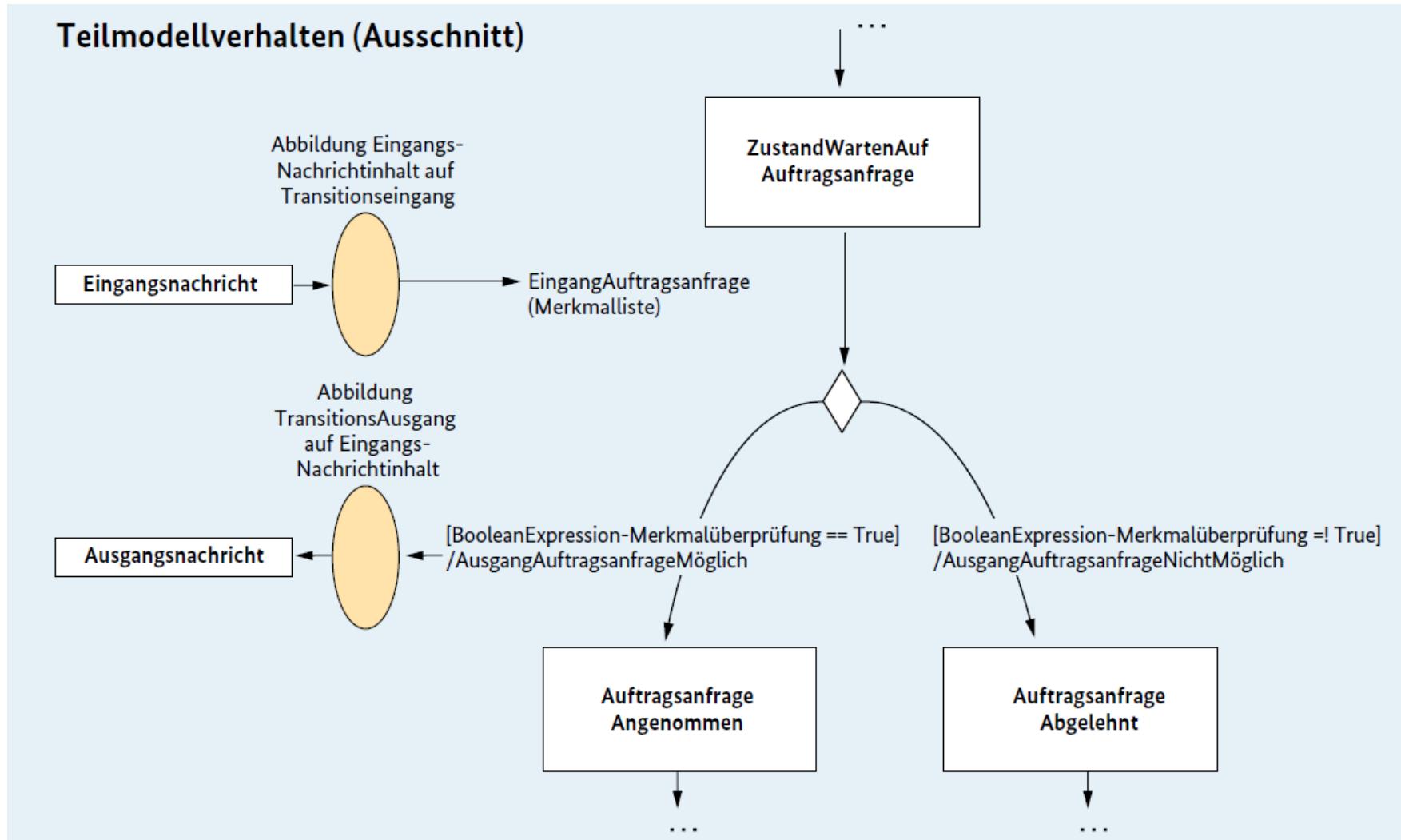
- Ein Interaktionsmodell besteht aus
 - der Definition von interagierenden Partnern (I4.0-Komponenten, Manifest, Teilmodell),
 - den Nachrichtenhalten (z.B. IDs und Merkmale von Vereinbarungen, Aufträge),
 - den Abläufen entsprechend dem Verhalten der Teilmodelle (z.B. durch Sequenzdiagramme) und
 - zusätzlichen Regeln zur Behandlung interner Daten und Funktionen (z.B. zur Überprüfung von Anforderungen und Zusicherungen).



Quelle: [Bock 2016]

- Die **Interaktion** löst bei dem empfangenden Interaktionspartner **unterschiedliche Zustandsübergänge** aus bzw. kann in Abhängigkeit der aktuellen Zustände unterschiedlich behandelt werden.
- Deshalb werden als Beschreibungsmodelle **Automaten** verwendet.
- Beispiel:
 - I4.0-Komponente, die einen Roboter ansteuert.
 - Der Roboter kann Transportaufträge ausführen und ist auch eine I4.0-Komponente.
 - Die Anfrage lautet, ein Werkstück mit einem bestimmten Gewicht und einem passenden Greifwerkzeug zu verschiedenen Punkten im dreidimensionalen Raum zu bewegen. In dem Beispiel wird ein KUKA Roboter verwendet.
 - Alle Interaktionen zur Abarbeitung von Teilaufgaben, wie die Abwicklung der Auftragsanfrage oder die Bearbeitung des Auftrages selbst, werden in eigenständigen Sessions eingebettet.

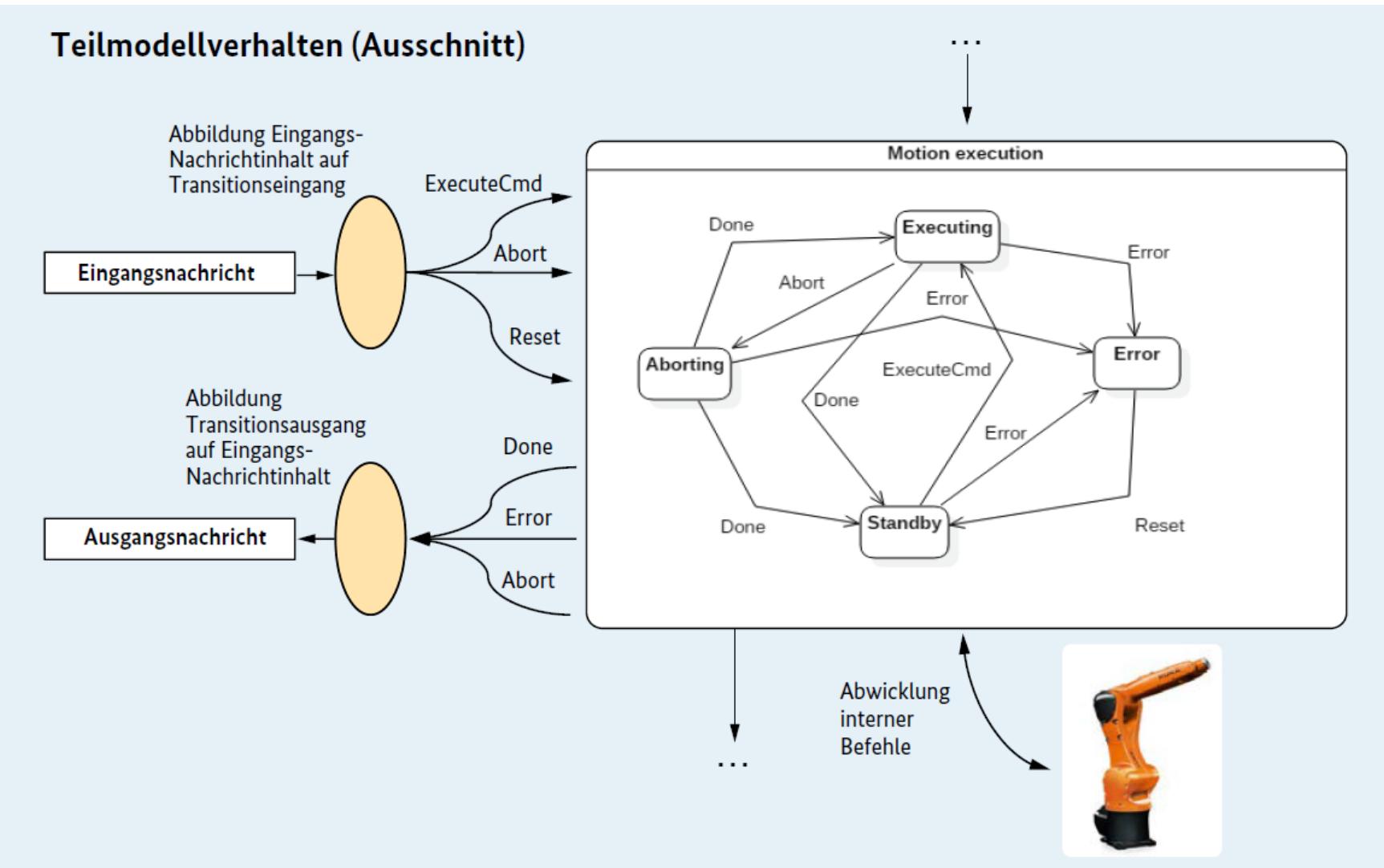
Verhaltensmodell von Interaktionsmustern



Quelle: [Bock 2016]

- Ein Dienst kann in Erweiterung des oben dargestellten Beispiels auch aus einer Folge von mehreren Eingangs- und Ausgangsnachrichten und damit auch Zustandsübergängen bestehen.
- Beispiel (Fortsetzung):
 - Sind Vorbereitungen für die eigentliche Aufgabe abgeschlossen, können direkte Befehle im Rahmen eines Teilmodells erteilt werden.
 - Dafür müssen alle Voraussetzungen für die aktiven Bewegungsaufgaben erfüllt sein (z.B. Auftrag muss angenommen sein, Roboter muss im richtigen Betriebsmodus sein, das richtige Greifwerkzeug muss am Roboter vorhanden sein, die unmittelbare Umgebung muss auf die Bewegung des Roboters gefasst sein).
 - Dann ist die I4.0-Komponente des Roboters in dem dafür geeigneten Zustand (MotionExecution), in welchem alle die Befehle an den Roboter gegeben werden, welche die gewünschten Aktionen des Roboters auslösen.
 - Der sichtbare Automat ist eine Projektion des Roboterverhaltens. Die Robotersteuerung aktualisiert die Zustände in der projizierten Zustandsmaschine.

Verhaltensmodell von Interaktionsmustern



Quelle: [Bock 2016]

- Verwaltungsschalen zur Vervollständigung einer Komponente zu einer I4.0-Komponente
 - Virtuelle Repräsentation
 - Einheitliche Schnittstellen
- Interaktion zwischen I4.0-Komponenten
 - Horizontale Interaktion (Synchron)
 - Client-Server
 - Vertikale Interaktion (Asynchron)
 - Peer-to-Peer
 - Dynamische PPR-Bindung
 - Interaktionsmuster
 - Verhaltensmodelle

Empfohlene und weiterführende Literatur

- [Adolphs 2015] Peter Adolphs et al.: „Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)“; Statusreport; April 2015; VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik und ZVEI e.V.
- [Adolphs 2016] Peter Adolphs et al.: „Struktur der Verwaltungsschale“; Ergebnispapier; April 2016; BMWi
- [ISA 88] Batch Control Part 1: Models and Terminology; ANSI/ISA-88.01-1995
- [Gilchrist 2016] Alasdair Gilchrist: “Industry 4.0”; apress; 2016
- [GMA 2014] „Gegenstände, Entitäten, Komponenten“; Statusreport; April 2014; VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik

Empfohlene und weiterführende Literatur



Institute for
Software & Systems
Engineering

- [Adolphs 2015] Peter Adolphs et al.: „Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)“; Statusreport; April 2015; VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik und ZVEI e.V., https://www.vdi.de/fileadmin/user_upload/VDI-GMA_Statusreport_Refenzarchitekturmodell-Industrie40.pdf
- [Bock 2016] Jürgen Bock et al.: „Weiterentwicklung des Interaktionsmodells für Industrie 4.0-Komponenten“; Diskussionspapier; Nov. 2016; Plattform Industrie 4.0; <http://www.plattform-i40.de/I40/Redaktion/DE/Downloads/Publikation/interaktionsmodell-i40-komponenten-it-gipfel.pdf?blob=publicationFile&v=10>
- [Schleipen 2009] Schleipen, Miriam, and Rainer Drath. "Three-view-concept for modeling process or manufacturing plants with AutomationML." Emerging Technologies & Factory Automation, 2009. ETFA 2009. IEEE Conference on. IEEE, 2009.