

# Fähigkeiten adaptiver Produktionsanlagen

Verbindung von Produkt, Prozess und Ressource

In einer idealen Umsetzung von Plug-and-produce würde eine noch unbekannte Anlagenkomponente während des laufenden Betriebs angekoppelt, automatisch erkannt und funktional in die Produktionsabläufe integriert; und das ohne manuellen Integrationsaufwand. Ein Ansatz dafür ist die Modellierung von Fähigkeiten in der Produktion. Sind die Fähigkeiten einer Anlagenkomponente bekannt, können diese mit den Anforderungen an einen Produktionsschritt abgeglichen und gezielt eingesetzt werden. In diesem Beitrag werden die Fähigkeiten technischer Anlagen über ihre Relationen zu Produkt, Prozess und Ressource definiert. Das Spannungsfeld des resultierenden Frameworks entsteht durch die Notwendigkeit, Fähigkeiten sehr allgemein zu definieren, um alle potentiellen Einsatzmöglichkeiten zu berücksichtigen. Im Gegensatz dazu müssen für die Automatisierung eines konkreten Produktionsschritts spezifische Details operationalisiert werden.

**SCHLAGWÖRTER** Plug-and-produce / Adaptive Produktionsanlagen / Fähigkeiten in der Produktion

## **Skills in adaptive production systems – Relations of product, process, and resource**

Imagining an ideal implementation of the plug-and-produce concept, it would allow for plugging in new components at runtime, and integrate them automatically on a functional level. This contribution pursues this goal by modelling the skills provided by production resources. Production skills are defined based on their relations to the well-known concepts of product, process and resource (PPR). The resulting framework overcomes the dichotomy of modelling very generic skills to capture all possible use cases and providing the detailed information necessary to automate actual production hardware.

**KEYWORDS** plug and produce / adaptive production systems / production skills

**JULIUS PFROMMER**, Karlsruher Institut für Technologie  
**MIRIAM SCHLEIPEN**, Fraunhofer IOSB  
**JÜRGEN BEYERER**, Fraunhofer IOSB, Karlsruher Institut für Technologie

**D**urch volatile, globale Märkte sowie kürzere Produktlebenszyklen bei steigender Variantenzahl, werden die Anforderungen an die Wandlungsfähigkeit von Produktionsanlagen (gerade in der diskreten Fertigung) immer bedeutender [1]. In der Produktionstechnik wird nun seit einiger Zeit an der Unterstützung häufiger Produktwechsel und der Herstellung vieler Produktvarianten auf derselben Anlage geforscht. Auch neue Themen der letzten Jahre, etwa der Einsatz von mobilen Robotern oder 3D-Druckern zielen in diese Richtung. Diese sind (hardwareseitig) nicht mehr an die Herstellung eines bestimmten Produkts gebunden, sondern lassen sich ad hoc für unterschiedliche Aufgaben einsetzen.

Parallel dazu entwickelt sich die softwareseitige Steuerung von Produktionsanlagen weiter, um die Adaptivität der Systeme zu erhöhen [2]. Unter Plug-and-produce [3] wird dabei die vereinfachte Integration von Anlagenkomponenten durch standardisierte Kommunikation und Mechanismen der Selbstbeschreibung verstanden. Das SkillPro-Projekt (<http://www.skillpro-project.eu>), gefördert durch das 7. Rahmenprogramm der Europäischen Kommission, forscht an diesem Thema ausgehend von der Modellierung der Fähigkeiten von Anlagenkomponenten. Als Grundlage für die einheitliche Beschreibung von Produktionsanlagen wird AutomationML [4] verwendet. Dieser Beitrag stellt einen Ansatz zur Modellierung der Fähigkeiten technischer Anlagen vor. Diese sind über ihren Bezug zu den bekannten Konzepten Produkt, Prozess und Ressource (PPR) definiert.

## 1. PRODUKT, PROZESS UND RESSOURCE

Die Aufteilung in Produkt, Prozess und Ressource ist in der Produktionstechnik weit verbreitet [5, 6]. Auch in AutomationML wird PPR für die Modellierung verwendet [7] (siehe Bild 1).

- **Produkt:** Der Begriff Produkt bezeichnet die Werkstücke (End- und Zwischenprodukte), die auf einer

Anlage verarbeitet werden. Es ist zu beachten, dass hier oftmals ein Produkttyp beschrieben wird und kein physisch existentes Objekt. In diesem Beitrag wird davon ausgegangen, dass für alle vorkommenden (Zwischen-) Produkte ein entsprechender Typ existiert, auf den referenziert werden kann.

- **Prozess:** Ein Prozess ruft gezielt Änderungen von Produkteigenschaften hervor. Prozesse definieren zum Beispiel die Bearbeitung eines Werkstücks, einen logistischen Transport innerhalb der Anlage, oder das Fügen mehrerer Werkstücke. Ein Prozess besteht dabei aus einem oder mehreren (geordneten) Ausführungsschritten (siehe [6] für eine Übersicht der Definitionen von Prozess in verschiedenen Normen und Standards).
- **Ressource:** Als Ressource werden alle materiellen und immateriellen Anlagenkomponenten bezeichnet, die bei der Ausführung eines Prozesses und der Fertigung eines Produkts zum Einsatz kommen. Beispiele für Ressourcen sind Roboter, Werkzeugmaschinen, Software oder Logistikinfrastruktur.

## 2. FÄHIGKEITEN VON PRODUKTIONSANLAGEN

In den vergangenen Jahren wurden vermehrt Fähigkeiten in der Produktion untersucht. Die Autoren des SIARAS-Projekts [8, 9, 10] verwenden eine Ontologie, um Fähigkeiten und ihre Relationen für die automatische Konfiguration eines Produktionssystems zu verwenden. Das ROSETTA-Projekt [11, 12, 13] zielt darauf ab, Roboter und ihre Fähigkeiten in einer Umgebung einzusetzen, in der sie mit menschlichen Arbeitskräften direkt zusammenarbeiten. Für das AutoPnP-Projekt [14, 15] wurde ebenfalls eine Beschreibung der Fähigkeiten von Plug-and-produce-Anlagenkomponenten entwickelt. Die Autoren bezeichnen zum einen generelle Bearbeitungsverfahren, wie Bohren oder Transportieren, als Fähigkeiten. Zum anderen können diesen Fähigkeiten Attribute, wie der Durchmesser des Bohrers, zugewiesen werden. In [16] wird eine Taxonomie von Fähigkeiten speziell für das

Fügen von Bauteilen definiert. Die Autoren von [17] stellen eine Ontologie für Fähigkeiten in der Produktion auf und ordnen, gemäß entsprechend definierter Anforderungen, Ressourcen einzelnen Produktionsschritten zu. Das OZONE-Projekt [6] definiert Aktivitäten als zentrales Element. Über die Verbindung zu Produkten und Ressourcen wird eingeschränkt, welche Aktivitäten im Rahmen eines Herstellungsvorgangs ausgeführt werden können. In [18] werden die PPR-Konzepte im Kontext von Fähigkeiten genutzt. Fähigkeiten werden als endliche Automaten modelliert, die mit allen drei PPR-Konzepten in Verbindung stehen. Die Urheber von [19] und [20] modellieren Fähigkeiten als agentenbasierte Systeme. Diese können ihre Fähigkeiten dynamisch kombinieren, um komplexere Aufgaben zu bewältigen. In der englischen Ausgabe der ISA-95/DIN EN 62264 [21] wird zwar von Capabilities gesprochen, dieser Begriff bezieht sich aber auf die Kapazität von Produktionsanlagen und nicht auf ihre Fähigkeiten. Das RoboEarth-Projekt [22, 23] beschäftigt sich mit dem Austausch von Informationen zwischen Robotern. Ziel ist das gemeinsame Erforschen und Beschreiben der Umwelt, ein Austausch erlernter Problemlösungsstrategien und die Kollaboration von Robotern in Situationen, die ein gemeinsames Handeln erfordern. Dafür werden in der RoboEarth-Ontologie auch Fähigkeiten von Robotern definiert. Deren Geltungsbereich ist dabei nicht auf das Produktionsumfeld beschränkt.

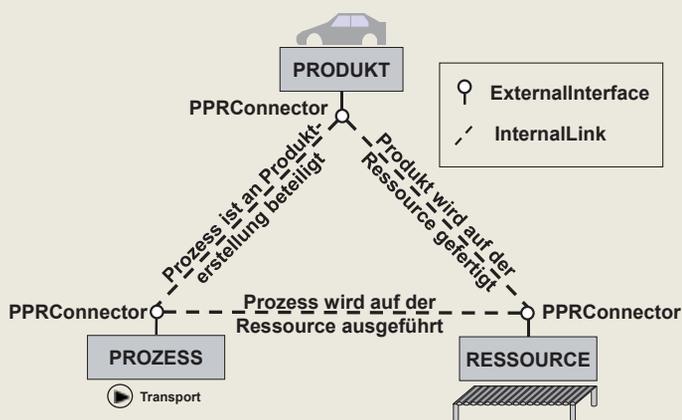
In den genannten Ansätzen werden identische Begriffe bezüglich Fähigkeiten teilweise semantisch unterschiedlich besetzt. Weitere Unterschiede gibt es in der Granularität (low-level, wie *Arm bewegen*, im Gegensatz zu high-level, wie *Motorhaube lackieren*). Auch sind die bisherigen Definitionen von Fähigkeiten meist stark in einer bestimmten informationstechnischen Verarbeitung verhaftet, zum Beispiel Ontologien oder objektorientierte Modellierung. Zusammengefasst sind drei Sichtweisen auf Fähigkeiten erkennbar:

- 1 | Eine Fähigkeit beschreibt das Potenzial, eine Tätigkeit aus einer bestimmten Domäne durchzuführen und kann einer technischen Anlage beschreibend zugewiesen werden. (Weder Subjekt noch Objekt sind bekannt.)
- 2 | Eine Fähigkeit beschreibt das Potenzial einer konkreten technischen Anlage, eine Tätigkeit aus einer bestimmten Domäne durchzuführen. Durch den Bezug zu einer technischen Anlage sind Details (besonders Limitierungen) der Fähigkeit bekannt und können für logisches Schließen verwendet werden. Die Fähigkeit bleibt dabei generisch genug, um in unterschiedlichen Situationen Anwendung zu finden. (Das Subjekt ist bekannt, das Objekt noch unbekannt.)
- 3 | Eine Fähigkeit beschreibt das Potenzial einer konkreten technischen Anlage in einer konkreten Situation (wie der Verarbeitung eines bestimmten Produkttyps) eine bestimmte Tätigkeit durchzuführen. (Subjekt und Objekt sind bekannt.)

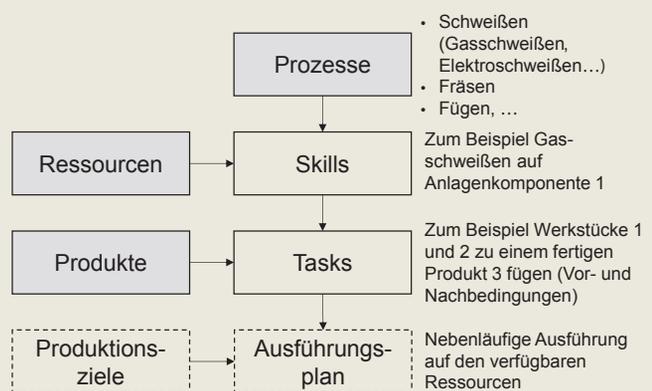
Diese Aufteilung fließt im Folgenden in die Entwicklung des Produkt, Prozess, Ressource und Skill-Frameworks (PPRS) ein. Beispiele und weitere Erläuterungen zu jeder Sichtweise finden sich bei den Definitionen von Prozess, Skill und Task im folgenden Abschnitt.

### 3. DAS PPRS-FRAMEWORK

Das PPRS-Framework wurde mit dem Ziel entwickelt, nicht eine weitere Definition von Fähigkeiten zur Liste existierender Ansätze hinzuzufügen, sondern die Unterschiede zwischen den drei Sichtweisen bezüglich Fähigkeiten explizit zu machen und in ein gemeinsames Modell zu integrieren. Dies geschieht über die Verbindungen zu den etablierten Konzepten Produkt, Prozess und Ressour-



**BILD 1:** Das Konzept von Produkt, Prozess und Ressource in AutomationML [7]



**BILD 2:** Fähigkeiten von Produktionsanlagen in Relation zum Produkt-Prozess-Ressource-Konzept

ce. Im Folgenden werden die Begrifflichkeiten von PPRS definiert. Bild 2 zeigt dazu einen grafischen Überblick.

**Produkt (product), Ressource:** behalten ihre ursprüngliche Bedeutung.

**Prozess (process):** Fertigungs- und Bearbeitungsverfahren (manufacturing processes), sowie unterstützende Vorgänge, werden in PPRS als Prozesse bezeichnet. Diese Verfahren haben noch keinen Bezug zu einem konkreten Anwendungsfall oder einer technischen Umsetzung durch eine Ressource. Prozesse können auch zu einer Hierarchie geordnet werden. Unterhalb des Prozesses *Schweißen* ließen sich so Gasschmelzschweißen und Lichtbogenschweißen ausdifferenzieren. In der DIN 8580 [24] sind gängige Verfahren aus der Fertigungstechnik bereits in ähnlicher Form hierarchisch strukturiert. Weitere relevante Themenbereiche stammen aus der Logistik, der Robotik und dem Fügen von Werkstücken.

**Fähigkeit (skill):** Ein Skill  $s \in S$  beschreibt das technische Potenzial einer Ressource  $r \in R$ , einen bestimmten Prozess  $\pi \in \Pi$  umzusetzen.

$$S \subseteq \Pi \times R \quad (1)$$

Angenommen eine Werkzeugmaschine stellt den Prozess *Fräsen* als Fähigkeit zur Verfügung. Das bedeutet nicht, dass jeder Fräsvorgang automatisch auf dieser Ressource ausführbar sein muss. Im Rahmen des Beispiels würde dies von der Größe des Werkstücks und dem verwendeten Werkstoff abhängen. Beschränkungen bezüglich der Anwendbarkeit sind als Metadaten mit dem Skill verknüpft.

**Ausführungsschritt (task):** Als Task wird die Anwendung eines Skills in der Herstellung eines Produktes (auf einer bestimmten Ressource) bezeichnet. Dazu enthalten Tasks alle Informationen, die für die Automatisierung der konkreten Aufgabe benötigt werden. Nach außen stellen sich Tasks als Blackbox dar. Bekannt sind nur

- Vorbedingungen und Nachbedingungen der Ausführung (wie der ein- und ausgehende Produkttyp und die Konfiguration beteiligter Ressourcen), sowie die
- zeitlichen Eigenschaften (die Ausführungsdauer und dadurch notwendiges Blockieren beteiligter Ressourcen; wird in diesem Beitrag zunächst nicht betrachtet).

Diese Kapselung von Funktionalität nimmt Anleihen an den formalisierten Prozessbeschreibungen (FBP) nach VDI/VDE 3682, siehe auch [25, 26]. Eine wichtige Erweiterung gegenüber FBP ist, dass ein Task auch Auswirkungen auf die Ressource(n) selbst haben kann. Das automatisierte Wechseln eines Werkzeugs (Rüsten) oder das Verfahren mit einem mobilen Roboter ist also ebenfalls als Task beschreibbar. Ebenso wie FBPs können Tasks hierarchisch aufgebaut sein.

- **Grundlegende Tasks:** Grundlegende Tasks können über eine Steuereinheit ohne weitere Vorverarbeitung auf

der Ressource ausgeführt werden. Die Implementierung grundlegender Tasks kann auf verschiedene Arten erfolgen:

- 1 | Automatisierte Abläufe sind mit klassischen Werkzeugen etwa basierend auf IEC 61131-3 oder DIN 66025/ISO 6983, auf den Anlagenkomponenten programmiert. Der entsprechende Task enthält dann eine Referenz, über die ein solches Programm angesteuert werden kann.
- 2 | Gleichsam kann nach dem Teach-in eines Roboters der generierte Arbeitsgang mit einer eindeutigen Identifikation versehen und über einen Task zur Ausführung zugänglich gemacht werden.
- 3 | Ausgehend von den modellierten Skills und detaillierten Daten zum Produkt wird ein ausführbarer Task automatisch generiert [10, 13].

- **Zusammengesetzte Tasks:** Ein zusammengesetzter Task besteht aus Untertasks, grundlegend oder wiederum zusammengesetzt, sowie Einschränkungen der Ausführungsreihenfolge, siehe auch Bild 3. Die Untertasks können sich auf verschiedene Skills beziehen. Zum Beispiel *Objekt erkennen – Objekt greifen – Arm bewegen*. Jeder Blattknoten der Hierarchie aus Tasks muss dabei ein grundlegender Task sein, um die Ausführbarkeit der gesamten Hierarchie zu gewährleisten. Einschränkungen der Ausführungsreihenfolge und die zeitlichen Relationen von Untertasks können zum Beispiel mittels der Intervallalgebra von Allen [27] beschrieben werden. Ein Beispiel aus der Lebensmittelindustrie: Der Task *Teigmasse auf konstanter Temperatur halten* muss vor dem Vorgang *Rühren* anlaufen und darf erst nach Beendigung von *Rühren* abgeschlossen werden.

Formal definiert sind die Tasks  $\tau \in T$  ein Übergang zwischen den Zuständen der beteiligten Ressourcen. Der Zustand  $\rho \in \mathcal{R}_r$  einer Ressource  $r \in R$  bezieht sich dabei a) auf die Konfiguration der Ressource  $c \in C_r$ , wie der Rüstzustand einer Maschine oder die Position eines mobilen Roboters, und b) auf den Zwischen-Produkttyp  $p \in P$ , der aktuell in der Ressource *enthalten* ist. Es kann auch kein Produkt, gekennzeichnet durch die leere Menge, in der Ressource enthalten sein. Die Menge  $\mathcal{R}^{init}$  enthält den initialen Zustand aller Ressourcen des Gesamtsystems.

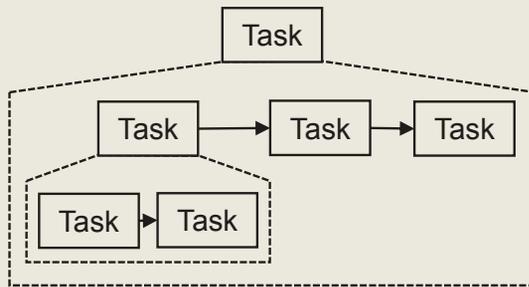
$$\mathcal{R}_r \subseteq C_r \times (P \cup \emptyset), \mathcal{R}^{init} \in \prod_{r \in R} \mathcal{R}_r \quad (2)$$

Es können mehrere Ressourcen  $R_\tau$  an einem Task beteiligt sein. Zum Beispiel, um im Rahmen eines Transportprozesses ein Produkt zwischen zwei Ressourcen zu übergeben. Somit gilt für die Vor- und Nachbedingungen (die Zustände der beteiligten Ressourcen)  $\tau^{pre}$  und  $\tau^{post}$ , dass

$$\tau^i \in \prod_{r \in R_\tau} \mathcal{R}_r, \forall i \in \{pre, post\}. \quad (3)$$

Die Betrachtung von Tasks als Zustandsübergänge ermöglicht die Betrachtung mittels bekannter Formalismen zur Beschreibung diskreter dynamischer Systeme, wie diskrete Ereignissysteme (DES) [28] oder den Situationskalkül [29]. Somit steht eine Vielzahl von Techniken

**BILD 3:** Zusammengesetzte Tasks (gestrichelter Rahmen) bestehen aus grundlegenden oder wiederum zusammengesetzten Tasks und einer Einschränkung der Ausführungsreihenfolge.



für das logische Schließen über mögliches Systemverhalten und zur gezielten Ablaufsteuerung zur Verfügung.

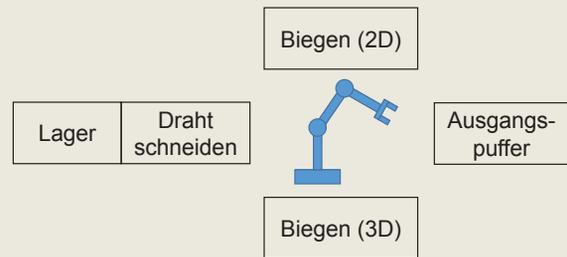
**Ausführbarer Ablaufplan:** Anhand der verfügbaren Tasks  $T$ , dem initialen Systemzustand  $\mathcal{R}^{init}$  und der Produktionsziele kann von einem entsprechenden Planungswerkzeug ein ausführbarer Ablaufplan generiert werden. Tasks können sich dabei funktional, wie Gasschmelzschweißen gegenüber Lichtbogenschweißen, und auf Ebene der Ressourcen, etwa für zwei identische Werkzeugmaschinen, substituieren. Auf der Ebene individueller Produkte kann sich dadurch die *Route* innerhalb einer Produktionsanlage unterscheiden. Effiziente Pläne werden meist einen hohen Grad an Nebenläufigkeit aufweisen, also mehrere Tasks auf den verfügbaren Ressourcen parallel ausgeführt. Für die Beachtung der zeitlichen Effizienz von Plänen muss dafür die Ausführungsdauer von Tasks vorab bekannt sein.

Nach dem Eintreten unvorhergesehener Ereignisse, wie der Ausfall einer Ressource, oder das Nachbearbeiten eines Werkstücks aufgrund von Qualitätsproblemen, wird auf Basis des aktuellen Systemzustands neu geplant, um die Ausführbarkeit und Effizienz des Ablaufplans zu gewährleisten. Damit durch die Neuplanung keine allzu langen Wartezeiten entstehen, sollten nur relevante Teile des Ablaufplans neu erzeugt werden. Es bietet sich zum Beispiel an, an den Puffern einer Produktionsanlage Teilbereiche des Plans voneinander zu entkoppeln.

Das beschriebene Vorgehen wurde prototypisch umgesetzt. Die Beschreibung der Ressourcen und der Tasks ist dazu in der Planning Domain Description Language (PDDL) [30] ausgeführt. Mittels des frei verfügbaren Solvers OPTIC [31] wird dann, ausgehend von einem initialen Zustand und den Produktionszielen, ein Ablaufplan erzeugt.

Die in PPRS definierten Begrifflichkeiten lassen sich auch in Anlagen ohne dynamische Ablaufplanung zur Laufzeit verwenden. Die Rolle von Tasks haben dann die fest vordefinierten lokalen "Reaktionen" von Anlagenkomponenten auf Ereignisse. Es sind auch hybride Anlagen möglich, bei denen nur Teilabläufe zur Laufzeit geplant werden. Der Übergang hin zu einer dynamischen Ablaufplanung kann demnach fließend erfolgen.

**BILD 4:** Topologie der Beispiel-Produktionsanlage



**Beispiel einer einfachen Produktionsanlage**

Das Anwendungsbeispiel ist Teil einer größeren Produktionsanlage, mit der Drahtkörbe gefertigt werden. Durch die hohe Anzahl an Varianten und niedrige Losgrößen musste die Anlage in der Vergangenheit oft umgerüstet werden. Weiterhin gab es eine Reihe von Arbeitsschritten, die durch die häufigen Produktwechsel nicht wirtschaftlich zu automatisieren waren.

Im betrachteten Ausschnitt wird Metalldraht aus dem Eingangslager entnommen und zunächst in Stücke der geforderten Länge geschnitten. Die Drahtstücke können anschließend von zwei Anlagenkomponenten weiterverarbeitet werden, die den Draht in die gewünschte Form biegen. Die Maschinen beherrschen dabei entweder 2D- oder 3D-Biegen. Der Transport wird von einem Roboterarm bewerkstelligt, der auch die fertig gebogenen Drahtstücke in einen Ausgangspuffer legt. Die Topologie der Anlage ist in Bild 4 dargestellt.

Prozesse:  $\Pi = \{schneiden, biegen, transport\}$   
 Ressourcen:  $R = \{ds, b2D, b3D, ra, puffer\}$

Die Benennung der Ressourcen ist aus Platzgründen abgekürzt, vergleiche Bild 4. Der Roboterarm  $ra$  kann sich in einer von vier Konfigurationen (Positionen) befinden, jeweils dem Drahtschneider, Biegen (2D/3D) oder dem Ausgangspuffer zugewandt  $C_{ra} = \{pos_{ds}, \dots\}$ . Die restlichen Ressourcen haben nur eine mögliche Konfiguration (standard). Alle Ressourcen können egal welches Produkt oder auch kein Produkt enthalten.

Produkte:  $P = \{p^1, \dots, p^4, p^{21}, \dots, p^{23}\}$

- Die vier finalen Produkte sind  $\{p^1, p^2, p^3, p^4\}$ . Von ihnen sind  $p^1$  und  $p^2$  2D-gebogen,  $p^3$  und  $p^4$  3D-gebogen. Die Länge des Drahts von  $p^1$  und  $p^3$  ist jeweils identisch. Alle weiteren Produkte haben unterschiedliche Längen.
- Die Zwischenprodukte, geschnittene, aber noch ungebogene Drahtstücke, sind demnach  $\{p^{21}, p^{22}, p^{23}\}$ . Das Zwischenprodukt  $p^{21}$  kann durch unter-

schiedliches Biegen in  $p^1$  oder  $p^3$  weiter verarbeitet werden.

Skills:  $S = \{ \text{drahtschneiden}, \text{biegen2D}, \text{biegen3D}, \text{bewegen}, \text{entnehmen} \}$

- $\text{drahtschneiden} = (\text{schneiden}, ds)$
- $\text{biegen2D} = (\text{biegen}, b2D)$
- $\text{biegen3D} = (\text{biegen}, b3D)$
- $\text{bewegen} = (\text{transport}, ra)$
- $\text{entnehmen} = (\text{transport}, puffer)$

Die Metadaten, die einen Skill soweit beschreiben, dass ein ausführbarer Task (bezogen auf ein konkretes Produkt) automatisch generiert werden kann, werden in diesem Beispiel nicht betrachtet.

Tasks:  $T = \{ \text{schneiden}_1, \text{bewegen}_1, \text{bewegen}_2, \text{biegen}_1, \text{entnehmen}_1 \}$

Die Auflistung der Tasks ist an dieser Stelle nicht vollständig, sondern in erster Linie repräsentativ. Tasks sind zur Identifikation benannt und durchnummeriert, dies hat jedoch keine funktionale Bedeutung.

- $\text{schneiden}_1: R_{\text{schneiden}_1} = \{ds\}$ 
  - $\text{schneiden}_1^{\text{pre}} = (\text{standard}, \emptyset)$
  - $\text{schneiden}_1^{\text{post}} = (\text{standard}, p^1)$
- $\text{bewegen}_1: R_{\text{bewegen}_1} = \{ds, ra, b2D\}$ 
  - $\text{bewegen}_1^{\text{pre}} = ((\text{standard}, p^1), (pos_{ds}, \emptyset), (\text{standard}, \emptyset))$
  - $\text{bewegen}_1^{\text{post}} = ((\text{standard}, \emptyset), (pos_{b2D}, \emptyset), (\text{standard}, p^1))$
- $\text{biegen}_1: R_{\text{biegen}_1} = \{b2D\}$ 
  - $\text{biegen}_1^{\text{pre}} = (\text{standard}, p^1)$
  - $\text{biegen}_1^{\text{post}} = (\text{standard}, p^1)$
- $\text{bewegen}_2: R_{\text{bewegen}_2} = \{b2D, ra, puffer\}$ 
  - $\text{bewegen}_2^{\text{pre}} = ((\text{standard}, p^1), (pos_{b2D}, \emptyset), (\text{standard}, \emptyset))$
  - $\text{bewegen}_2^{\text{post}} = ((\text{standard}, \emptyset), (pos_{puffer}, \emptyset), (\text{standard}, p^1))$
- $\text{entnehmen}_1: R_{\text{entnehmen}_1} = \{puffer\}$ 
  - $\text{entnehmen}_1^{\text{pre}} = (\text{standard}, p^1)$
  - $\text{entnehmen}_1^{\text{post}} = (\text{standard}, \emptyset)$

Werden die modellierten Tasks in der Reihenfolge der Auflistung ausgeführt, wird zunächst ein Produkt vom Typ  $p^1$  erzeugt, in den Ausgangspuffer gelegt und dort für eine Weiterverwendung wieder entnommen. Dabei wird ein initialer Zustand  $\mathcal{R}^{\text{init}}$  angenommen, in dem der Roboterarm dem Drahtschneider zugewandt ist und keine der Ressourcen ein Produkt enthält.

Die Tasks  $\text{bewegen}_1$  und  $\text{bewegen}_2$  lassen sich in mehrere Teilschritte *Produkt aufnehmen*, *Roboterarm bewegen*, *Produkt absetzen* mit je zwei kooperierenden Ressourcen unterteilen. Diese können dann in der Definition zusammengesetzter Tasks wiederverwendet werden. Andern-

falls würden sie über die Ablaufplanung ad-hoc aneinander gefügt. Die erzeugten Pläne wären dadurch aber entsprechend länger – mit Auswirkungen auf die Laufzeit der Planungsalgorithmen. Siehe [32] für eine Diskussion der Vor- und Nachteile feingranularer Modellierung gegenüber umfangreicheren, monolithischen Fähigkeiten.

### Limitierungen von Tasks und Themen für zukünftige Erweiterungen

Falls an einem Task mehrere Ressourcen beteiligt sind, könnten eine oder mehrere Ressourcen noch beschäftigt sein, während die restlichen Ressourcen bereits auf die Ausführung des nächsten Tasks warten. In diesem Beitrag ist die zeitliche Dauer von Tasks zunächst nicht berücksichtigt. Dadurch wurde implizit die Annahme getroffen, dass jede Ressource undefiniert lange auf eine Synchronisation mit beschäftigten Ressourcen warten kann. Diese Annahme muss aber nicht unbedingt zutreffen. Zum Beispiel in der Verarbeitung von erhitztem Metall oder in logistischen Ablaufketten. In diesem Fall müsste für die Erzeugung ausführbarer Pläne die Ausführungsdauer von Tasks bekannt sein und den Ressourcen ein Zeitstempel zugeordnet werden.

Eine weitere Limitierung der aktuellen Modellierung ist, dass Ressourcen nur einen Produkttyp enthalten können. Dies kann durch die Vereinigung mehrerer Produkte zu einem kombinierten Produkttyp umgangen werden. Ein solches Vorgehen führt aber in den meisten Fällen zu einer Explosion der Anzahl von Produkttypen, die innerhalb einer Anlage auftreten können. Etwa bei der Modellierung großer Puffer oder Lager.

## 4. NEUE MÖGLICHKEITEN IM PLUG-AND-PRODUCE

Im Zusammenhang mit Plug-and-produce wird oft der Universal Serial Bus-Standard (USB) [33] aus der PC-Welt als Vorbild genannt. USB definiert die mechanische Verbindung, die Kommunikation und die Energieversorgung über den Bus, sowie eine Reihe grundlegender Geräteklassen. Die PC-Betriebssysteme enthalten passende Treiber, die für fast jede Peripherie Plug-and-play, die Integration ohne manuelle Konfiguration, und Hot-plugging, das Ein- und Ausstecken von Peripherie an ein laufendes System, ermöglichen.

Analog zu USB findet in der Idealvorstellung von Plug-and-produce die funktionale Integration von neuen oder ausgetauschten Anlagenkomponenten automatisch statt. Die Modellierung von Fähigkeiten mit PPRS könnte zur Grundlage für eine langfristige Umsetzung dieser Vision werden. Der Ablauf eines Plug-and-produce-Vorgangs wäre dann wie folgt:

- 1 | Physische Kopplung der Komponente mit der Gesamtanlage (manuell)
- 2 | Aufbau der Kommunikation und Bekanntmachen der Komponente innerhalb der Gesamtanlage (automatisiert)
- 3 | Kommunikation der Komponenteneigenschaften an die Anlagensteuerung (automatisiert über Mechanismen der Selbstbeschreibung)
- 4 | Definition der neu verfügbaren Tasks ausgehend von den zu fertigenden Produkten

- (entweder manuell programmiert oder automatisiert aus Skill-Beschreibungen abgeleitet)
- 5 | Aktualisierung des Ablaufplans (wird nach dem Eintreten eines unerwarteten Ereignisses, wie einer Änderung der Anlagentopologie, automatisch neu erzeugt)
  - 6 | Ausführen des Ablaufplans durch die Anlagensteuerung und fortlaufende Kommunikation mit den Anlagenkomponenten zur Überwachung des Betriebszustands (automatisiert)

Natürlich können nicht alle bestehenden Anlagenkomponenten ein solches Vorgehen nativ unterstützen. Ihnen würde ein Adapter, rein softwarebasiert oder als physi-

sche Hardware, vorgeschaltet werden. Eine einheitliche Schnittstelle, die ein solcher Adapter nach außen anbietet, könnte auf Grundlage der OPC Unified Architecture (OPC UA) [34] definiert sein.

## ZUSAMMENFASSUNG

Im Beitrag wurde gezeigt, wie sich die Fähigkeiten von Komponenten einer Produktionsanlage anhand ihrer Beziehung zu Produkt, Prozess und Ressource (PPR) beschreiben lassen. Anschließend illustrierten die Autoren die Modellierung von Fähigkeiten im resultierenden PPRS-Framework anhand eines Beispiels. Weiter ging es darum,

## REFERENZEN

- [1] Wiendahl, H. P., ElMaraghy, H. A., Nyhuis, P., Zäh, M. F., Wiendahl, H. H., Duffie, N., Brieke, M.: Changeable manufacturing-classification, design and operation. *CIRP Annals-Manufacturing Technology* 56(2), S. 783-809, 2007
- [2] Schleipen, M.: *Adaptivität und semantische Interoperabilität von Manufacturing Execution Systemen (MES)*. KIT Scientific Publishing 2013
- [3] Arai, T., Aiyama, Y., Maeda, Y., Sugi, M., Ota, J.: Agile assembly system by "plug and produce". *CIRP Annals-Manufacturing Technology* 49(1), S. 1-4, 2000
- [4] Drath, R., Luder, A., Peschke, J., Hundt, L.: AutomationML - the glue for seamless automation engineering. In: *Proc. IEEE Conf. Emerging Technologies & Factory Automation (ETFA'08)*, S. 616-623. IEEE 2008
- [5] Maropoulos, P. G., McKay, K. R., Bramall, D. G.: Resource-Aware Aggregate Planning for the Distributed Manufacturing Enterprise. *CIRP Annals - Manufacturing Technology* 51(1), S. 363-366, 2002
- [6] Cutting-Decelle, A. F., Young, R. I. M., Michel, J. J., Grangel, R., Le Cardinal, J., Bourey, J. P.: ISO 15531 MANDATE: A Product-process-resource based Approach for Managing Modularity in Production Management. *Concurrent Engineering* 15(2), S. 217-235, 2007
- [7] Schleipen, M., Drath, R.: Three-View-Concept for modeling process or manufacturing plants with AutomationML. In: *Proc. IEEE Conf. Emerging Technologies & Factory Automation (ETFA'09)*, S. 1-4. IEEE 2009
- [8] Angelsmark, O., Malec, J., Nilsson, K., Nowaczyk, S., Prosperi, L.: Knowledge representation for reconfigurable automation systems. In: *Proc. 9th Nat. Conf. on Robotics*, S. 129-138. Wydawnictwo Komunikacji i Łączności 2006
- [9] Malec, J., Nilsson, A., Nilsson, K., Nowaczyk, S.: Knowledge-based reconfiguration of automation systems. In: *Proc. IEEE Int. Conf. Automation Science and Engineering (CASE'07)*, S. 170-175. IEEE 2007
- [10] Bengel, M.: Model-based configuration - A workpiece-centred approach. In: *Proc. ASME/IFToMM International Conference on Reconfigurable Mechanisms and Robots (ReMAR)*, S. 689-695. ASME 2009
- [11] Björkelund, A., Edstrom, L., Haage, M., Malec, J., Nilsson, K., Nugues, P., Robertz, S. G., Storkle, D., Blomdell, A., Johansson, R.: On the integration of skilled robot motions for productivity in manufacturing. In: *Proc. IEEE Int. Symp. Assembly and Manufacturing (ISAM'11)*, S. 1-9. IEEE 2011
- [12] Björkelund, A., Malec, J., Nilsson, K., Nugues, P.: Knowledge and skill representations for robotized production. In: *Proc. 18th IFAC Congress*, S. 8999-9004. IFAC 2011
- [13] Malec, J., Nilsson, K., Bruyninckx, H.: Describing assembly tasks in declarative way. In: *Workshop Semantics, Identification and Control of Robot-Human-Environment Interaction at ICRA 2013*, S. 50-53. IEEE 2013
- [14] Kainz, G., Keddiss, N., Pensky, D., Buckl, C., Zoitl, A., Pittschelis, R., Kärcher, B.: AutoPnP - Plug-and-Produce in der Automation. *atp edition - Automatisierungstechnische Praxis* 55(4), S. 42-49, 2013
- [15] Keddiss, N., Kainz, G., Buckl, C., Knoll, A.: Towards adaptable manufacturing systems. In: *Proc. IEEE Int. Conf. Industrial Technology (ICIT)*, S. 1410-1415. IEEE 2013.
- [16] Huckaby, J., Christensen, H. I.: A Taxonomic Framework for Task Modeling and Knowledge Transfer in Manufacturing Robotics. In: *Workshops at the Twenty-Sixth AAAI Conference on Artificial Intelligence*, S. 94-101. AAAI 2012
- [17] Järvenpää, E., Luostarinen, P., Lanz, M., Tuokko, R.: Presenting capabilities of resources and resource combinations to support production system adaptation. In: *2011 IEEE International Symposium on Assembly and Manufacturing (ISAM'11)*, S. 1-6. IEEE 2011
- [18] Björkelund, A., Malec, J., Nilsson, K., Nugues, P., Bruyninckx, H.: Knowledge for Intelligent Industrial Robots. In: *AAAI Spring Symposium Series: Technical Report SS-12-02*. AAAI 2012. <http://www.aaai.org/ocs/index.php/SSS/SSS12/paper/view/4340>
- [19] Angerer, S., Pooley, R., Aylett, R.: Self-Reconfiguration of Industrial Mobile Robots. In: *Proceedings of the 4th IEEE International Conference on Self-Adaptive and Self-Organizing Systems (SASO'10)*, S. 64-73. IEEE 2010
- [20] Barata, J., Onori, M., Frei, R., Leitão, P.: Evolvable production systems in a RMS context: enabling concepts and technolo-

wie sich über die Modellierung von Fähigkeiten eine funktionale Integration neuer Anlagenkomponenten im Rahmen von Plug-and-produce automatisieren lässt. Dabei verbleibt eine Reihe von Fragen. Wichtig für einen praktischen Einsatz sind insbesondere Garantien zur Einhaltung der nichtfunktionalen Anforderungen industrieller Steuerungssysteme (Echtzeitfähigkeit, Laufzeit von Planungsalgorithmen, Performanceeinbußen durch den Einsatz von Heuristiken). Und nicht zuletzt bleibt die Zusammenarbeit von Mensch und Maschine ein zentrales Thema.

MANUSKRIPTEINGANG  
12.07.2013

Im Peer-Review-Verfahren begutachtet

- gies. In: Proc. 2nd Int. Conf. Changeable, Agile, Reconfigurable and Virtual Production, S. 239-244. CARV 2007
- [21] DIN EN 62264-1: Integration von Unternehmensführungs- und Leitsystemen - Teil 1: Modelle und Terminologie (IEC 62264-1:2003), 2008
- [22] Waibel, M., Beetz, M., Civera, J., D'Andrea, R., Elfring, J., Galvez-Lopez, D., et al.: Roboearth. IEEE Robotics & Automation Magazine 18(2), S. 69-82, 2011
- [23] Tenorth, M., Perzylo, A. C., Lafrenz, R., Beetz, M.: The RoboEarth language: Representing and exchanging knowledge about actions, objects, and environments. In: 2012 IEEE International Conference on Robotics and Automation (ICRA), S. 1284-1289. IEEE 2012
- [24] DIN 8580: Fertigungsverfahren - Begriffe, Einteilung, 2003
- [25] VDI/VDE 3682: Formalisierte Prozessbeschreibungen, 2005
- [26] Jäger, T., Christiansen, L., Strube, M., Fay, A.: Durchgängiges Engineering von der Anforderungserhebung bis zur Anlagenstrukturbeschreibung. at - Automatisierungstechnik 61(2), S. 92-101, 2013
- [27] Allen, J. F.: Maintaining knowledge about temporal intervals. Communications of the ACM 26(11), S. 832-843, 1983.
- [28] Ramadge, P. J., Wonham, W. M.: Supervisory Control of a Class of Discrete Event Processes. SIAM Journal on Control and Optimization 25(1), S. 206-230, 1987
- [29] Reiter, R.: Knowledge in action: logical foundations for specifying and implementing dynamical systems. MIT 2001
- [30] Fox, M., Long, D.: PDDL2. 1: An Extension to PDDL for Expressing Temporal Planning Domains. Journal of Artificial Intelligence Research (JAIR) 20, S. 61-124, 2003
- [31] Benton, J., Coles, A. J., Coles, A.: Temporal Planning with Preferences and Time-Dependent Continuous Costs. In: 22nd Int. Conf. Automated Planning and Scheduling (ICAPS), S. 2-10. AAAI 2012
- [32] Weser, M., Zhang, J.: Autonomous planning for mobile manipulation services based on multi-level robot skills. In: 2009 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS'09), S. 1999-2004. IEEE 2009
- [33] USB Implementers Forum: Universal Serial Bus 3.0 Specification, 2008. <http://www.usb.org/developers/docs>
- [34] IEC/TR 62541-1: OPC Unified Architecture - Part 1: Overview and Concepts, 2010.

## AUTOREN



**JULIUS PFROMMER** (geb. 1985) ist seit 2013 wissenschaftlicher Mitarbeiter am Lehrstuhl für Interaktive Echtzeitsysteme des Karlsruher Instituts für Technologie (KIT) und arbeitet in enger Kooperation mit dem Fraunhofer IOSB. Davor hat er am KIT, am Institut National Polytechnique de Grenoble und an der ETH Zürich studiert.

**Karlsruher Institut für Technologie,  
Lehrstuhl für Interaktive Echtzeitsysteme,  
Adenauerring 4, D-76131 Karlsruhe,  
Tel. +49 (0) 721 6 09 12 86,  
E-Mail: [julius.frommer@kit.edu](mailto:julius.frommer@kit.edu)**



Dr.-Ing. **MIRIAM SCHLEIPEN** (geb. 1983) arbeitet seit 2005 am Fraunhofer IOSB. Sie leitet die Gruppe Leitsysteme und Anlagenmodellierung. Ihr Hauptinteresse gilt der Adaptivität und Interoperabilität von Komponenten und Systemen in Produktionsanlagen speziell für die MES-Ebene.

**Fraunhofer IOSB,  
Fraunhoferstr.1, D-76131 Karlsruhe,  
Tel. +49 (0) 721 6 09 13 82,  
E-Mail: [miriam.schleipen@iosb.fraunhofer.de](mailto:miriam.schleipen@iosb.fraunhofer.de)**



Prof. Dr.-Ing. **JÜRGEN BEYERER** (geb. 1961) ist Inhaber des 2004 eingerichteten Lehrstuhls für Interaktive Echtzeitsysteme an der Fakultät für Informatik des Karlsruher Instituts für Technologie (KIT). Gleichzeitig ist er Leiter des Fraunhofer-Instituts für Optronik, Systemtechnik und Bildauswertung (IOSB) in Karlsruhe. Prof. Beyerer studierte Elektrotechnik an der Universität Karlsruhe und promovierte 1994 am Institut für Mess- und Regelungstechnik (MRT). Anschließend baute er eine Forschungsgruppe zum Thema Automatische Sichtprüfung und Bildverarbeitung am gleichen Institut auf. 1999 habilitierte er sich für das Fach Messtechnik. Von 1999 bis 2004 leitete er das Mannheimer Mittelstands-Unternehmen Hottinger Systems GmbH (heute: inspectomation GmbH) und war stellvertretender Geschäftsführer des Schwesterunternehmens Hottinger Maschinenbau GmbH.

**Fraunhofer IOSB,  
Fraunhoferstr.1, D-76131 Karlsruhe,  
E-Mail: [juergen.beyerer@iosb.fraunhofer.de](mailto:juergen.beyerer@iosb.fraunhofer.de)**