

Digitalisierung der Produktion in KMU

Ein Retrofit-Konzept für die Realisierung der I4.0 Fabrik

Norbert Gronau, Sander Lass

1. Einleitung

Beeinflusst von industriellen Trends und der fortschreitenden technologischen Entwicklung (Westkämper 2013), sind produzierende Unternehmen mit einer wachsenden Komplexität innerhalb der Fabrik konfrontiert. Schnelle Marktveränderungen und eine starke Individualisierung der Produkte schaffen die Notwendigkeit einer hohen Agilität des Produktionsmanagements und der Fabrikgestaltung (Kagermann et al. 2013). Daher ist die Umsetzung von Wandlungsfähigkeit (Andresen/Gronau 2005) unerlässlich, um diese Agilität effizient und zuverlässig zu erreichen.

1.1. Komplexitätsreduzierung durch Dezentralisierung

Die Dezentralisierung durch intelligente und autonome Einheiten - im Gegensatz zu herkömmlichen Kontrollstrukturen in zentraler monolithischer Form - ist ein geeigneter Ansatz, um die Komplexität in Fertigungsszenarien zu reduzieren (Barbosa et al. 2015). Die Ausstattung des Produktionssystems, welches ein komplexes sozio-technisches System (vgl. Neumann et al. 2012) darstellt, mit dezentralen und autonomen Produktionseinheiten (Windt/Jeken 2009) ermöglicht unter anderem ein hohes Maß an Rekonfigurierbarkeit (ElMaraghy 2005).

Cyber-physische Systeme (CPS) schaffen die technologische Basis für selbststeuernde Methoden in der Produktion (Gronau/Theuer 2016, acatech 2011). Sie integrieren Fabrikelemente und ganze Anlagen zu cyber-physischen Produktionssystemen (CPPS). Die Vision sieht CPPS als wesentlichen Bestandteil der modernen Fabrik zur Sicherstellung der Wettbewerbsfähigkeit produzierender Unternehmen. Wandlungsfähigkeit und Komplexitätsbeherrschung sind u. a. Potenziale dieser neuen Generation des Produktionsmanagements. Mit Hilfe von CPPS ist eine hohe Anpassungsfähigkeit an veränderte Umgebungsbedingungen durch Flexibilität und Autonomie in der Fabrik möglich (acatech 2012).

Zwar sind aus Perspektive der Anbieter bereits Produkte zu dessen Realisierung verfügbar, jedoch mit Blick auf den bisherigen Umsetzungsgrad innerhalb der Praxis scheinen die potenziellen Anwender diesen Optimismus nicht zu teilen: der Einsatz der neuen Problemlöser erfolgt nur zögerlich.

1.2. Forschungsgegenstand

Die Transformation des theoretischen Konstrukts CPPS in die praktische Realisierung kann allerdings nur unter Einbezug der den Anwendungskontext Fabrik prägenden Rahmenbedingungen erfolgen. Vor allem ist nicht davon auszugehen, dass vorhandene Maschinen und Anlagen ohne Weiteres durch neue Exemplare ersetzt werden können. Dies impliziert als Lösungsansatz, Vorhandenes in geeigneter Form zu befähigen, als Teil eines CPPS zu agieren.

Als Forschungsgegenstand resultiert die Frage nach einem Konzept, welches den Brownfield-Charakter aufgreift und die CPS-Erweiterung bestehender Systeme gestattet:

- Wie kann die Integration geschlossener Legacy-Systemen und Alt-Anlagen realisiert bzw. deren Einbindung in CPPS erfolgen, um dessen Vorteile effektiv nachhaltig zu nutzen?
- Wie können die Autonomie und lokale Informationsverarbeitung sowie die notwendige Vernetzung und Kommunikation der Elemente praxistauglich implementiert werden?

Die Beantwortung der formulierten Forschungsfragen liefern eine Basis für die Realisierung von CPPS-Elementen innerhalb bestehender (Produktions-)Systeme bzw. die Findung einer geeigneten Lösung.

1.3. Methodik und erwartete Ergebnisse

Auf Grund der Zielstellung nutzt diese Forschungsarbeit ein Vorgehensmodell aus dem Portfolio der Design Science für den Prozess der Lösungsfindung. Die Generierung des angestrebten Artefakts orientiert sich deshalb am Prozess der Design Science Research Methodology (DSRM) nach Peffers (Peffers et al. 2007). Insbesondere werden die Phasen Problemanalyse, Anforderungsanalyse, Konzeption und prototypische Implementierung sowie Validierung durchlaufen.

Die Problemanalyse arbeitet die allgemeinen Rahmenbedingungen des Diskursbereichs Fabrik heraus. In der Phase der Anforderungsanalyse liefert die Systemanalyse (vgl. Krallmann et al. 2013) einer realen Produktion weitere Erkenntnisse hinsichtlich praxisrelevanter Bedarfe und zeigt die Erweiterungsbedarf der klassischen Automatisierungstechnik auf. Die Entwicklung des Lösungskonzepts beginnt mit der Aufgabenstrukturierung durch ein Schichtenmodell und korrelierten Geräteklassen. Die diesbezügliche Zuordnung von Speicherprogrammierbaren Steuerungen (SPS) und CPS zeigt die Notwendigkeit der Funktionserweiterung des klassischen Automatisierungsgerätes und gibt Auskunft über die ergänzend notwendigen Funktionen. Die anschließende Konkretisierung der Lösungs idee

umfasst sowohl ein Hardware- als auch ein Softwarekonzept. Das Ergebnis ist eine CPS-Komponente (I4.0-Box) inklusive Architekturmodell (Fabrikbetriebsystem) für dessen Betrieb zur Erweiterung bestehender Anlagen.

Eine prototypische Implementierung überführt diese theoretischen Artefakte in eine reale Umsetzung, die neben der Machbarkeit die Basis für die anschließende erste Konzeptevaluierung und -validierung liefert. Dies geschieht an Hand einer Fallstudie zur Prüfung der Wandlungsfähigkeit einer Anlagensteuerung, welche die klassische, zentral gesteuerte SPS-Ausführung der dezentral gestalteten Variante unter Einsatz der I4.0-Box gegenüberstellt. Diese funktionsbezogene Validierung erfolgt wegen der Limitierung der Möglichkeiten zu Experimenten am Originalsystem. Dahingehend findet das Logistiksystem des Forschungs- und Anwendungszentrum Industrie 4.0 Potsdam (Gronau/Lass 2016) Verwendung und bildet den technischen Kontext für die experimentelle Prüfung beider Ausprägungen hinsichtlich Wandlungsfähigkeit.

Ergebnisse sind neben dem proof of concept ebenfalls quantitative Aussagen zur Wirksamkeit des dezentralen Steuerungsparadigmas für diesen Anwendungsfall. Eine Verallgemeinerung ist unter der Prämisse des gleichen zu Grunde liegenden Komplexitätsmodells durch Skalierung möglich und auf weitere Systeme der Fabrik übertragbar.

2. Rahmenbedingungen des Diskursbereichs

Eine erfolgreiche Transformation in Richtung CPPS erfordert die Berücksichtigung der Bedingungen, die den Anwendungskontext einer Fabrik prägen. Die Informationstechnologie einer Fabrik besteht hauptsächlich aus Automatisierungssystemen. Im Vergleich zu den durch die Eigenschaften der Office-IT¹ charakterisierten Informationssystemen wie Manufacturing Execution Systemen (MES) oder Enterprise Resource Planning Systemen (ERP) muss die IT auf Fabrikebene (vgl. Automatisierungspyramide, z.B. Hollender 2010) weitere Rahmenbedingungen erfüllen (vgl. Gumzej 2010, Damm/Olderog 2003). Dazu gehören unter anderem (a) Echtzeitfähigkeit der Regelkreise zur Informationsverarbeitung und -übertragung sowie lange Arbeits- und Innovationszyklen (>7 Jahre), was (b) die Umsetzung innerhalb von Brownfield-Szenarien impliziert.

¹ Office-IT dient in diesem Kontext als Oberbegriff für die typischerweise durch Internet-Technologien vernetzten Informationssysteme des Büro-Bereichs und deren Backends (z. B. Datenbanken, Fileserver oder infrastrukturelle Systeme) (vgl. Stouffer et al. 2011).

(a) Die Aufgabe der Implementierung von Regelkreisen für physisch wirkende Komponenten erfordert Echtzeitfähigkeit der ausführenden Rechnersysteme. Echtzeit bedeutet, dass ein System garantiert, dass es innerhalb einer bestimmten Zeitspanne reagiert (timeliness) und es die gleichzeitige Ausführung von verschiedenen Aufgaben (concurrency) ermöglicht. Aufgrund der starken Verknüpfung mit externen technischen Prozessen muss die Verarbeitung und Kommunikation synchron mit den Vorgängen erfolgen. Damit hat die Dimension Zeit im Gegensatz zu ihrer Rolle in der allgemeinen Informationsverarbeitung der Büro-IT eine explizite Bedeutung (Halang 1989). Es gibt Zeitpunkte oder periodische Zeitvorgaben für die Ausführung von Aufgaben, d. h., das Ergebnis besitzt keinen Wert nach Ablauf der Frist (harte Echtzeit) oder nur noch ein reduzierten Wert (weiche Echtzeit) (Stankovic and Ramamritham 1990).

(b) Die typische Implementierung von CPS ist als Brownfield-Szenario zu sehen. Der Bau neuer Produktionsanlagen von Grund auf ist eher die Ausnahme als der Standard. Die typische Situation ist die Reorganisation bestehender Anlagen und Prozesse. Aufgrund von Strukturen, die sich über einen langen Zeitraum entwickelt haben, kann diese Reorganisation sehr aufwendig sein (Kühn 2006). Gegebenenfalls müssen bestehende Systeme auf Grund des Investitionsschutzes oder der mangelnden Investitionsbereitschaft einbezogen werden. Der Einsatz von CPS bei gleichzeitiger Kompletterneuerung einer Anlage ist zumeist keine realistische Option. Eine wesentliche Fragestellung für die Wissenschaft ist die angemessene und geeignete Integration von CPS in bestehende Informationssysteme (Gronau 2014).

(c) Hinzu tritt ein weiterer Aspekt. Die Programmiermöglichkeiten einer Standard-SPS erschweren die Implementierung von CPS-Fähigkeiten, da sie sich auf die Automatisierung von Geräten konzentrieren. Als potentielle Elemente eines CPPS benötigen Produktionsobjekte² ausreichende Kommunikationsfähigkeiten, um die Komplexität zu reduzieren und die Vorteile der Dezentralisierung zu nutzen. Bezogen auf Autonomie, Dezentralisierung und CPS erfassen diese Produktionseinheiten Informationen, verarbeiten sie, treffen Entscheidungen und führen sie selbstständig mit eigenen Ressourcen aus (Theuer 2012). Daher erfordern diese Subsysteme Umgebungswahrnehmung, erweiterte Speicher- und

² Der Begriff Produktionsobjekt umfasst alle möglichen Einheiten der Fertigungs- und Produktionsplanung. Sie sind die Elemente des Produktionssystems. Dabei handelt es sich in erster Linie um Maschinen und Werkzeuge, Werkstücke und Werkstückträger sowie um Logistikausrüstung wie z.B. Automobiltransporte und Pufferelemente.

Kommunikationsmöglichkeiten, um einen umfassenden Informationsaustausch und eine autonome Aufgabenausführung zu ermöglichen (Freitag et al., 2004).

Diese Anforderungen an die lokale Informationsverarbeitung und Entscheidungsausführung erfordern die Implementierung geeigneter Algorithmen. Diese Aufgabe stößt schnell an ihre Grenzen, wenn es um die Umsetzung mit den klassischen Programmierverfahren der Automatisierungssysteme nach IEC-61131 (IEC 2013) geht. Der Versuch, Szenarien zur schnellen und möglichst automatischen Rekonfiguration von Anlagen (Änderbarkeit) sowie zur lokalen Bereitstellung von aggregierten Funktionen und Systemzuständen (Komplexitätsreduzierung) mittels klassischer speicherprogrammierbarer Steuerungen (SPS) zu realisieren, hat keine geeignete und zufriedenstellende Lösung gefunden. Die Implementierung der erforderlichen Algorithmen, basierend auf der gängigen Schrittkettenprogrammierung, beeinträchtigt die Echtzeitfähigkeit (Zykluszeit) des Systems oder führte zu einer schwer zu wartenden Software. Die Notwendigkeit einer adäquaten lokalen Umsetzung komplexer Algorithmen ist daher neben den Echtzeitfähigkeiten und dem Brownfield-Szenario eine weitere Grundvoraussetzung.

Die Kombination der Prämissen (a), (b) und (c) führt zu der Erkenntnis, dass die Umsetzung der CPS-Fähigkeiten eine adäquate Erweiterung der bestehenden Fertigungseinheiten erfordert. Diese Erweiterungen müssen sowohl Echtzeitfähigkeiten als auch die mögliche Implementierung komplexer Algorithmen berücksichtigen. Dies bedeutet insbesondere die Vermeidung von Einschränkungen der klassischen Automatisierungsparadigmen.

3. CPS-Befähigung bestehender Produktionsobjekte

Im Hinblick auf die Brownfield-Situation ist es notwendig, bestehende Komponenten zu befähigen, als Teil eines CPPS zu agieren. Insbesondere die Integration von geschlossenen Altsystemen ist ein typischer Anwendungsfall. Um dieser Herausforderung gerecht zu werden, wurde Gerät entwickelt, welches die Nachrüstung der geforderten Eigenschaften ermöglicht und eine Produktionsanlage mit CPS-Fähigkeiten ausstattet. Das Gerät ermöglicht die Umsetzung verschiedener Middleware-Konzepte wie den theoretischen Ansatz des Referenzarchitekturmodells Industrie 4.0 (RAMI 4.0) mit seinem konzeptionellen Begriff Administrationsschale (Adolphs/Eppele 2015).

3.1. Funktionsprinzip

Nach dem Prinzip der CPS, mit der umfassenden Wahrnehmung ihrer Umgebung durch Sensoren und ihrer Interaktion durch Aktoren (vgl. Lee/Seshia 2014), gibt es zwei grundlegende Aufgaben der Weiterentwicklung bestehender

Systeme: die Erfassung von Umweltinformationen als passive Stufe und die Beeinflussung der physikalischen Umgebung als aktive Stufe. Die passive Stufe realisiert die Datenerfassung durch vorhandene Sensoren oder durch den Einsatz von Zusatzgeräten. Er greift nur lesend auf das System zu. Analog zur Unterscheidung in Lese- und Schreiboperationen bedeutet dies, dass es keinen kritischen Eingriff in das System gibt und die Berücksichtigung von Nebenwirkungen und die Regelung von Schreiboperationen nicht notwendig ist. Die aktive Stufe ist dagegen ein Eingriff in das Systemverhalten, z.B. durch Regelkreise mit Aktoren. Abhängig von den vorhandenen Eigenschaften der Antriebsregler (z.B. elektrischer Anschluss, Grundlogik, ggf. Zustandsrückmeldung) ist die geeignete Synchronisation der externen Signale mit der vorhandenen Steuerung (z.B. SPS und deren Programm) eine wichtige Aufgabe.

Information node	L4	Aggregation, Funktionsaufrufe zur Komplexitätsreduzierung	vorverarbeitete/aggregierte Information	SPS	CPS		
	Gateway	L3	semantische Interpretation			komplexe Datentypen, Modelle	
Connector		L2	Basisdatentypen			Number, String, ...	
	L1	physische Verbindung	Bits				
Zugriffsebene		Ziel	Übertragungsobjekte				

Abbildung 1: Vergleich von SPS und CPS über Schichtenmodell und Geräteklassen

Durch Kombination der Pyramide der Weisheit (DIKW-Hierarchie - Data, Information, Knowledge and Wisdom; vgl. Ackoff 1989) und der Schichtenarchitektur von Referenzmodellen für Netzwerkprotokolle (vgl. ISO 2011) ergeben sich die Zugriffsarten wie folgt: Signal-, Daten- und Informations- sowie Aggregationsebene. Der Signalpegel bezieht sich auf die physikalische Verbindung. Sie deckt hauptsächlich die Spezifikationen für elektrische Signale ab. Übertragungselemente sind typischerweise Bits oder Analogwerte. Basierend auf diesen Elementen definiert die Datenebene grundlegende Datentypen, z.B. numerische Werte. In Verbindung mit Protokollen und Datenmodellen bietet die Informationsebene eine Erweiterung um semantische Aspekte oder komplexe Datenstrukturen. Die Aggregationsebene erlaubt sowohl eine Vorverarbeitung als auch eine Aggregation von Daten oder Informationen. Sie gestattet Funktionen auf hohem Abstraktionsniveau für die Kommunikation und Betrieb, d. h., ein au dieser

Ebene operierendes System stellt Funktionsaufrufe bereit, die komplexe Vorgänge kapseln und die Kommunikation vereinfachen.

Mit Bezug auf dieses Schichtmodell ergeben sich drei Geräteklassen:

- Ein *Connector* realisiert eine Verbindung auf physikalischer und Datenebene.
- Ein *Gateway* stellt Verbindungen auf Informationsebene her.
- Ein *Information node* arbeitet auf der Aggregationsebene.

Abbildung 1 zeigt die drei Klassen von Steuergeräten und ordnet SPS und CPS hinsichtlich Ebenen, Objekten und Übertragungselementen zu. Die Elemente der Übertragung stellen die Objekte der Kommunikation dar. Eine SPS deckt bestenfalls die beiden unteren Ebenen ab. Wechselwirkungen mit den Ebenen L3 und L4 bedürfen der Ergänzung, z. B. nach dem CPS-Konzept. CPS bieten durch ihre Systemeigenschaft der erweiterten Informationsverarbeitung die Möglichkeit, Daten semantisch zu interpretieren und zu aggregieren sowie die Komplexität durch höhere Funktionsaufrufe zu reduzieren. Somit bedeutet die CPS-Befähigung eines Objektes die Aufgabe, Informationsknoten zu implementieren.

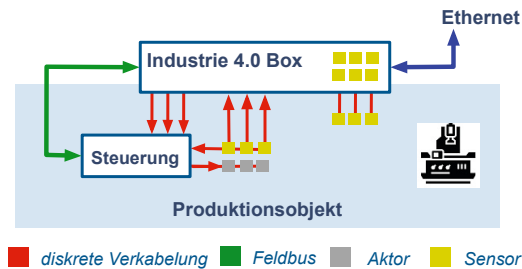


Abbildung 2: Funktionsprinzip der I4.0-Box

Eine reale Implementierung eines solchen Informationsknotens ist ein Gerät, das ein CPS darstellt und eine bestehende Anlage ergänzt. Basierend auf dem Begriff "Industrie 4.0" wird dieses Gerät als hier I4.0-Box bezeichnet. Abbildung 2 fasst das Funktionsprinzip zusammen. Das Gerät erhält Zugriff auf die eingebauten Sensoren und Aktoren von Produktionsobjekten. Bei Bedarf ergänzt es zusätzliche Erfassungsmechanismen. Das CPS-basierte Gerät verbindet Sensoren über eine diskrete Verdrahtung oder vorhandene Feldbusnetzwerke der Steuerung (z.B. SPS). Zusätzlich verwendet es unabhängige Sensoren, die in einer geeigneten Position installiert sind. Für den Zugriff auf Aktoren nutzt die Box ein Feldbusnetzwerk in Kombination mit zusätzlichen Eingangsmodulen der SPS oder in

Sonderfällen über eine direkte Verdrahtung zu den darunterliegenden Steuergeräten (z.B. Motorsteuerung).

3.2. Perspektive Software - Fabrikbetriebssystem

Hinsichtlich der Betriebssysteme und ihrer Funktion - Abstraktion von der zugrunde liegenden Hardware durch die Verwaltung der Hardware-Ressourcen (Tanenbaum 2009) - werden die Software-Komponenten der Boxen mit dem Begriff FabOS (Fabrikbetriebssystem) zusammengefasst (Lass 2017). Das FabOS basiert auf der in Abbildung 3 dargestellten Architektur. Integrale Elemente sind die Laufzeitumgebung, der Connection Service, der Monitoring Service sowie der KPI Service und das Control Center.

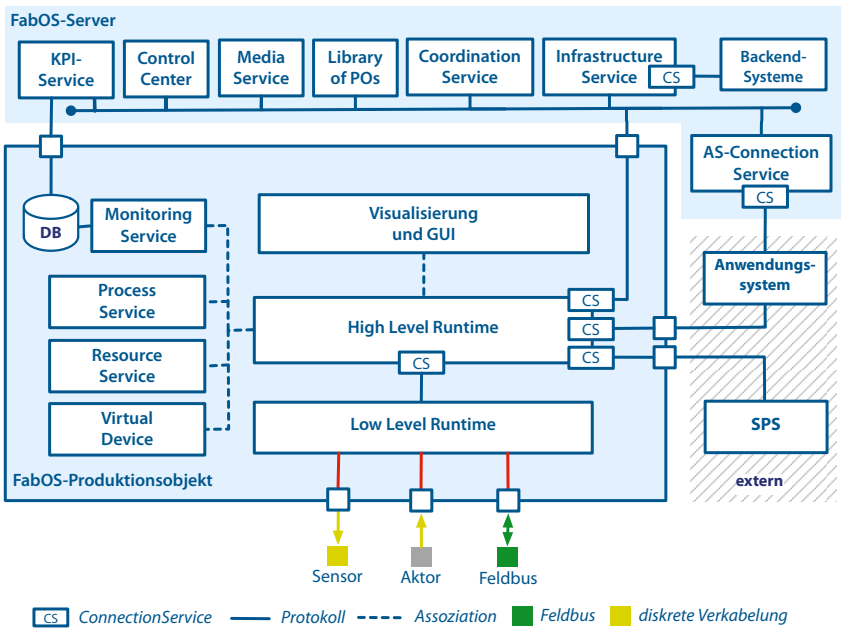


Abbildung 3: Architektur und Komponenten des FabOS (Lass 2017)

Die Laufzeitumgebung gliedert sich in zwei Funktionsbereiche. Die Low-Level-Runtime (LLR) ermöglicht die Implementierung von echtzeitkritischen Funktionen. Die Realisierung erfolgt über SPS oder echtzeitfähige Mikrocontroller, die mit typischen Entwicklungsumgebungen der Automatisierung (z.B. CODESYS) programmiert werden. Experimente hinsichtlich der praktischen Umsetzung haben gezeigt, dass eine Implementierung von Wandlungsfähigkeit, die diesem Programmierstil folgt, schnell an ihre Grenzen stößt. Der Aufbau eines frei kon-

figurierbaren Transportsystems³ mit einer zentralen SPS als Steuerungseinheit erwies sich mit sequentiellen Steuerungsparadigmen als schwierig zu handhaben. Daher beinhaltet das Konzept eine weitere Umsetzungsmöglichkeit. Die High-Level-Runtime (HLR) ermöglicht den Einsatz von Hochsprachen und Programmierparadigmen, die schnelle und einfache Erweiterungen für komplexe Informationsverarbeitung ermöglichen. Abhängig von dem Produktionsobjekt und der beabsichtigten Anwendung wird im Betrieb die LLR, die HLR-Implementierung oder eine Kombination aus beidem verwendet.

Der Connection Service ermöglicht die Kommunikation der Komponenten. Ähnlich dem Treiberkonzept von Betriebssystemen abstrahiert er von den technischen Details und ermöglicht den Zugriff auf Funktionen des FabOS. Des Weiteren erlaubt er die Implementierung einer Gateway-Funktion zwischen der internen Kommunikation der Systemkomponenten und den Protokollen externer Komponenten oder Geräten. Dies kann mit Hilfe von Standards (z.B. OPC-UA bei der Kopplung von LLR und HLR) oder auf die vom Zulieferer explizit definierte Weise geschehen. Instanzen des Connection Service können als zentraler Service oder dezentral in den jeweiligen Komponenten implementiert werden.

Wesentliche Eigenschaft eines CPS ist die Umgebungswahrnehmung. Zuständige lokales Modul ist der MonitoringService. Die Laufzeitumgebung jedes Produktionsobjekts nimmt das Loggen und Abspeichern von Ereignissen (Events) vor. Dies umfasst u. a. Kommunikationsvorgänge und Veränderungen von Umgebungsdaten, Mengen und Eigenschaften, die das interne Modell abbildet, sowie Benutzerinteraktionen. Ebenfalls ist die Protokollierung von Wertverläufen in bestimmten Zeitintervallen vorgesehen, die sich über zeitbasierte Events in die ereignisorientierte Verfahrensweise einreicht.

Ergänzend zu der lokalen Persistierung von Ereignissen ist dem allgemeinen Paradigma Inversion of Control (vgl. Sommerville 2011) folgend mit der Verwendung von Listnern eine weitere Möglichkeit gegeben, Informationen zielgerichtet zu kommunizieren bzw. abzurufen. Adressierter Anwendungsfall ist vorrangig die Bereitstellung von Echtzeitdaten und die Vermeidung von ineffizientem Polling. In diesem Zusammenhang bezieht sich der Begriff Echtzeit auf die Aktualität, d. h., die Werte spiegeln den zum Betrachtungszeitpunkt tatsächlich vorliegenden Wert wider.

³ Die Auswahl des Szenario Transportsystem erfolgt auf Grund seines Einsatzes als typisches Element der Fabrik und seines übergreifenden Charakters als prozessbildendes System.

Der `ProcessService` koordiniert zeitliche Abläufe und ist für Zeitsynchronisation der Produktionsobjekte zuständig. Mit dem `ResourceService` werden Abhängigkeiten bezüglich notwendiger Betriebsmittel bei der Ausführung abgebildet. Dies sind insbesondere benötigtes Material und Werkzeuge, Personal, Baugruppen oder Halbzeuge. Ein `VirtualDevice` bildet ggf. ein Modell eines bestimmten Elements oder Subsystems ab (Digitaler Zwilling).

4. Proof of Concept

Die prototypische Realisierung von Boxen und deren Anwendung innerhalb der Testplattform des Forschungs- und Applikationszentrums Industrie 4.0 Potsdam (vgl. Lass 2017) ist Teil der Validierung. Der erste Schritt der Konzeptvalidierung ist die prototypische Umsetzung der Box selbst. Es umfasst sowohl die Hardware als auch die Anwendung des werkseitigen Betriebssystems als Softwareteil. Die Box-Prototypen basieren auf dem Raspberry Pi. Zusätzlicher Bestandteil ist eine I/O-Karte, die an werksspezifische Anwendungen angepasst ist. Dieses Board bietet nicht nur industrietypische Steckverbinder (nach Level 1 und 2), sondern auch die Möglichkeit, weitere Schnittstellen durch ein internes Backbone zu erweitern. Zur Demonstration des Konzepts sind die Boxen dieser Prototypen Teil der Anwendungsfälle innerhalb der Testplattform des Forschungs- und Anwendungszentrums.

Einer dieser Anwendungsfälle ist ein modulares Transportsystem. Die Nachrüstung dieser bestehenden Anlage mit Hilfe der Box-Prototypen stellt den zweiten Teil der Validierung dar. Ziel ist es, den diskutierten theoretischen Ansatz im Hinblick auf seine praktische Umsetzbarkeit und den Grad der Zielerreichung zu bewerten. Die daraus resultierende Aufgabe ist die Implementierung eines dezentralen Steuerungssystems, das die vorhandenen Komponenten integriert. Insofern zeichnet sich das betrachtete System durch seine Flexibilität aus. Das heißt, der Change Agent befindet sich außerhalb des Systems (Ross et al. 2008). Bei der Operationalisierung erfolgt dies durch Ermittlung des Aufwands für extern verursachte Änderungen am Transportsystem, die eine Neukonfiguration erfordern. Die für die Rekonfiguration der Anlage benötigte Zeit ist der verwendete Messwert. Es geht also darum, die Flexibilität bestehender Anlagen zu erhöhen, indem der Aufwand für Systemänderungen durch den beschriebenen Ansatz reduziert wird.

4.1. Aufbau der Fallstudie

Ausgangspunkt bildet eine konventionelle Rollenbahn als Repräsentant für bereits bestehende Anlagen. Sie besteht aus verschiedenen Arten von Transportmodulen, die innerhalb des Produktionslayouts zu Förderstrecken kombiniert wer-

den. Jedes Modul hat eine Motorsteuerung, deren Aktivitäten durch einfache 24 Volt Signale bestimmt werden. Die verwendeten Module sind in Abbildung 4 dargestellt.

Basismodul	Modul					Summen			
	Sensoren pro Modul	Aktoren pro Modul	Konfiguration Ports	Konfiguration Positionen	Anzahl	Sensoren	Aktoren	Ports	Positionen
Transfer	4	3	3	1	3	12	9	9	3
Gerade kurz	2	3	2	2	5	10	15	10	10
Gerade lang	4	6	2	4	3	12	18	6	12
Abschnitt gekrümmt	6	6	2	2	2	12	12	4	4
Dispatcher	5	5	6	1	1	5	5	6	1
						51	59	35	30
						Gesamtsumme:			175

Abbildung 4: Komplexitätsmodell mit verwendeten Modulen

Folgende Grundmodule stehen zur Verfügung: Transfere (3 Ports und 1 Position), gerade Module (2 Ports und 2 Positionen sowie 2 Ports und 4 Positionen) und gekrümmte Segmente (2 Ports mit 2 Positionen) sowie ein Dispatcher (6 Ports und 1 Position). Alle Module zeichnen sich jeweils durch die eingebauten Sensor- und Aktorsignale, die Anzahl der Anschlüsse für den Ein-/Auslauf und die internen Positionen aus. Ports sind die Übergabepunkte zwischen den Modulen, die als Ein- oder Ausgang dienen. Positionen stellen die Positionen der zu transportierenden Objekte innerhalb des Moduls dar. Sie werden im Wesentlichen von den Sensoren eines Moduls bestimmt. Insgesamt 110 ein- und ausgehende Signale sowie 65 Variablen, die eine Konfiguration oder einen Zustand (Ports und Positionen) beschreiben, müssen bei der Programmierung berücksichtigt werden. Die konfigurierten Routen von den jeweiligen Start- und Endpunkten erhöhen die Komplexität. Es gibt zwei Varianten der Implementierung:

- (A) eine dezentrale Steuerung der Anlagenkoordination durch intelligente Elemente nach dem CPS-Konzept unter Verwendung der I4.0-Boxen.
- (B) eine zentrale Steuerung mit diskreter Verdrahtung aller Komponenten nach dem klassischen SPS- und Schrittketten-Paradigma

Die Implementierung (A) befasst sich auch mit der Reduzierung der Komplexität durch dezentrale Steuerung und der Möglichkeit einer schnellen Rekonfiguration

des Transportsystems (Gronau et al. 2016, Theuer et al., 2013]. Das gesamte System ist in Segmente unterteilt, die über die I4.0-Boxen eine kleine Anzahl von Modulen einzeln integrieren. Alle Signale werden direkt an das Schnittstellenmodul der Box angeschlossen (siehe Abbildung 5). Auf LLR-Ebene sind die Grundfunktionen implementiert (wie Motor ein bis Lichtschranke auslöst). Die HLR abstrahiert von diesen elementaren Prozessen und stellt übergeordnete Funktionsaufrufe (wie z.B. Transit etwas von Position A nach Position B) zur Verfügung, die auch für externe Entitäten zugänglich sind. Darüber hinaus löst die HLR - eventuell in Zusammenarbeit mit anderen CPS - komplexe Aufgaben (wie Routing oder Konfliktlösung). Die Lösungsalgorithmen hierfür werden mit Hilfe von Hochsprachen implementiert (in diesem Fall Python).

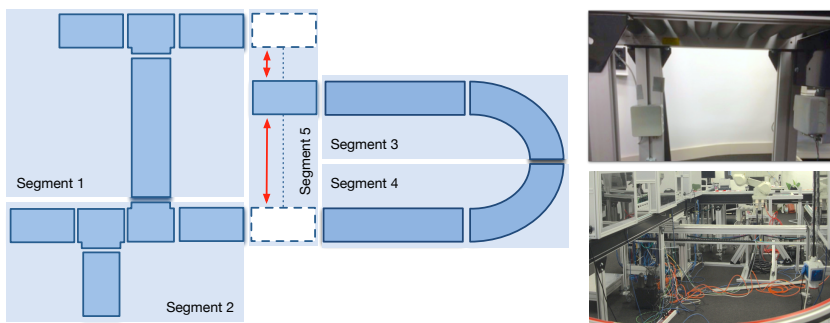


Abbildung 5: Segmentierung, I4.0-Box und Versuchsaufbau

Die Implementierung (B) stellt den heutigen typischen Kontrollansatz dar. Durch die Verwendung einer Standard-SPS als zentrale Steuerungseinheit wird der Materialfluss sichergestellt. Diese Art der Realisierung verbindet alle Sensoren und Aktoren über eine diskrete Verdrahtung mit einem Gerät. Die Programmerstellung entspricht den SPS-typischen Schrittketten und basiert auf Anweisungslisten und teilweise auf Funktionsbausteinen.

4.2. Ergebnisse

Der Vergleich erfolgt mit Hilfe von drei Szenarien: Teiländerungen am Layout (Ausfall eines Moduls, Hinzufügen eines Elements) und größere Änderungen am Layout (Rekonfiguration von mehr als der Hälfte der Module). Abbildung 6 zeigt den Aufwand für die Umsetzung (A) und (B). Diese gliedern sich in Hardware-Setup (vor allem Verdrahtung), Codierung und Einsatz mit abschließendem Test der gesamten Anlage. Der unabhängigen Anpassung im Fehlerfall mit einem kurzen Test der Anlage von (A) steht ein wesentlich höherer Aufwand für die Anpassung des Quellcodes gegenüber. von (B). Insbesondere der Aufwand für die

Codierung wird durch den Einsatz von CPS reduziert. Im Gegensatz dazu erfordert das Hardware-Setup nur minimalen Mehraufwand für die Variante (B).

	Variante (A): CPS			
Szenario	Hardware-setup	Programm-erstellung	Inbetrieb-nahme	Gesamt-aufwand
(1) Ausfall eines Moduls	0h 5m	0h 0m	0h 30m	0h 35m
(2) Hinzufügen eines Elements	0h 35m	0h 0m	1h 0m	1h 35m
(3) Änderung des Layouts	2h 0m	12h 0m	2h 0m	16h 0m

	Variante (B): zentrale SPS			
Szenario	Hardware-setup	Programm-erstellung	Inbetrieb-nahme	Gesamt-aufwand
(1) Ausfall eines Moduls	0h 5m	3h 40m	0h 50m	4h 35m
(2) Hinzufügen eines Elements	50m	5h 10m	1h 55m	7h 55m
(3) Änderung des Layouts	3h 20m	16h 0m	4h 35m	23h 55m

Abbildung 6: Benötigte Aufwände

5. Fazit

Das vorgestellte Konzept adressiert vor allem die typische industrielle Brownfield-Situation. Neben der komplexitätsreduzierenden Wirkung des dezentralen Steuerungskonzeptes gegenüber der zentralen Version zeigt die oben beschriebene Fallstudie, dass diese Art der Steuerung auch mit bestehenden Nicht-I4.0-Systemen möglich ist. Durch die entsprechende Erweiterung bestehender Produktionseinheiten zu I4.0-Informationsknoten können die Potenziale einer dezentralen Organisation für bestehende Systeme genutzt werden. Die erste Validierung bezieht sich insbesondere auf den Aufwand bei Systemänderungen. Unter der Prämisse der zunehmenden Komplexität der Informationsverarbeitung und des Rekonfigurationsaufwands in Abhängigkeit von der zu verarbeitenden Datenmenge skalieren die Ergebnisse progressiv. Dies bedeutet, dass mit zunehmender Größe des Systems die in der Fallstudie identifizierten positiven Effekte zunehmen werden.

Die Ergebnisse zeigen, dass durch entsprechende Segmentierung der Steuerungsaufgabe eine Reduzierung des Rekonfigurationsaufwands erreicht werden kann, wenn die erforderlichen Fähigkeiten - in diesem Fall durch die Erweiterung bestehender Systeme - mit geringem Aufwand installiert werden können. Das vorgestellte Konzept geht in diese Richtung und deckt sowohl die Software- als auch

die Hardware-Ebene ab. Die I4.0-Box erweitert bestehende Systeme und schafft die Voraussetzungen für die Integration. Das Fabrikbetriebssystem bietet eine flexible Softwarearchitektur für die einfache Anwendung der Box und ermöglicht die Kombination von klassischer SPS-Programmierung und der Implementierung komplexer Algorithmen.

Neben der Erhöhung der Rekonfigurationseffizienz bestehender Anlagen gehören auch die maschinenübergreifende Prozessüberwachung in heterogenen Systemlandschaften und das Design von Realzeit-Regelkreisen⁴, d. h. die Auswertung der Daten mit Informationsgenerierung erfolgt im operativen Betrieb. Beide Fälle sind Gegenstand weiterer Forschungsarbeiten.

Literatur

- acatech (2011). acatech POSITION: Cyber-Physical Systems: Innovationsmotor für Mobilität, Gesundheit, Energie und Produktion. Deutsche Akademie der Technikwissenschaften, Springer Berlin Heidelberg.
- acatech (2012). acatech Studie: Integrierte Forschungsagenda Cyber-Physical Systems. Deutsche Akademie der Technikwissenschaften Berlin.
- Ackoff, R. L. (1989). From data to wisdom. *Journal of Applied Systems Analysis*, 16, S. 3–9.
- Adolphs, P. and Epple, U. (2015). Statusreport Referenzarchitektur Industrie 4.0 (RAMI4.0). Verein Deutscher Ingenieure e.V. (VDI) und Zentralverband Elektrotechnik- und Elektronikindustrie e.V. (ZVEI).
- Andresen, K., Gronau, N., (2005) Adaptability concepts for enterprise resource planning systems-a component framework. In: *AMCIS 2005 Proceedings*, S. 150.
- Barbosa, J., Leitão, P., Adam, E., and Trentesaux, D. (2015). Dynamic self- organization in holonic multi-agent manufacturing systems: The adacor evolution. *Computers in industry*, 66:99–111.
- Damm, W. and Olderog, E. (2003). Formal Techniques in Real-Time and Fault-Tolerant Systems: 7th International Symposium, FTRTFT 2002, Co-sponsored by IFIP WG 2.2, Oldenburg,

⁴ Im Gegensatz zur der technischen Echtzeit aus dem Automatisierungsbereich adressiert der Begriff Realzeitregelkreis die nutzbringende Rückkopplung von Erkenntnissen in das operative Prozessmanagement.

- Germany, September 9-12, 2002. Proceedings. Lecture Notes in Computer Science. Springer Berlin Heidelberg.
- ElMaraghy, H. A. (2005). Flexible and reconfigurable manufacturing systems paradigms. *International journal of flexible manufacturing systems*, 17(4), S. 261–276.
- Freitag, M., Herzog, O., and Scholz-Reiter, B. (2004). Selbststeuerung logistischer Prozesse - ein Paradigmenwechsel und seine Grenzen. *Industrie Management*, 20(1), S. 23–27. GITO Verlag Berlin.
- Gronau, N. (2014). Der Einfluss von cyber-physischen Systems auf die Gestaltung von Produktionssystemen. In Kersten, W., Koller, H., and Lödding, H., *Industrie 4.0 - Wie intelligente Vernetzung und kognitive Systeme unsere Arbeit verändern*, S. 279–295, Schriftenreihe der Hochschulgruppe für Arbeits- und Betriebsorganisation e.V. (HAB), GITO Verlag Berlin.
- Gronau, N., Lass, S. (2016). Anwendungszentrum Industrie 4.0 Potsdam. *Industrie 4.0 Management*, 32(4): S. 66–67.
- Gronau, N., Theuer, H. (2016). Determination of the optimal degree of autonomy in a cyber-physical production system. *Procedia CIRP*, 57, S. 110–115.
- Gronau, N., Grum, M., and Bender, B. (2016). Determining the optimal level of autonomy in cyber-physical production systems. In *Industrial Informatics (INDIN)*, 2016 IEEE 14th International Conference on, pages 1293–1299. IEEE.
- Gumzej, R. (2010). *Real-time Systems' Quality of Service: Introducing Quality of Service Considerations in the Life Cycle of Real-time Systems*. Springer London.
- Halang, W. A. (1989). Schwerpunkte der internationalen Forschung im Bereich Echtzeitsysteme. In Henn, R. and Stieger, K., *PEARL 89 - Workshop über Echtzeitsysteme*, Informatik-Fachberichte Volume 231, S. 1–12. Springer Berlin Heidelberg.
- Hollender, M. (2010). *Collaborative Process Automation Systems*. ISA.
- IEC (2013). IEC 61131-3: Programmable controllers - Part 3: Programming languages. International Electrotechnical Commission (IEC), ed. 3.0 edition.
- ISO (2011). ISO/IEC/IEEE 42010:2011: Systems and software engineering – Architecture description. International Organization for Standardization (ISO), Genf.
- Kagermann, H., Wahlster, W., and Helbig, J. (2013). *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0 – Abschlussbericht des Arbeitskreises Industrie 4.0*. Forschungsunion im Stifterverband für die Deutsche Wissenschaft Berlin; Acatech - Deutsche Akademie der Technikwissenschaften e.V.
- Krallmann, H., Bobrik, A., and Levina, O. (2013). *Systemanalyse im Unternehmen*. Oldenbourg Verlag München.
- Kühn, W. (2006). *Digitale Fabrik*. Hanser München.
- Lass, S. (2017). *Simulationskonzept zur Nutzvalidierung cyber-physischer Systeme in komplexen Fabrikumgebungen*. Dissertation, Universität Potsdam.
- Lee, E. A. and Seshia, S. A. (2014). *Introduction to Embedded Systems - A Cyber Physical Systems Approach - Edition 1.5*. LeeSeshia.org, Berkley.

- Neumann, M., Constantinescu, C., and Westkämper, E. (2012). A method for multi- scale modeling of production systems. In *Enabling Manufacturing Competitiveness and Economic Sustainability*, pages 471–475. Springer.
- Peffer, K., Tuunanen, T., Rothenberger, M. A., and Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), S. 45–78.
- Ross, A. M., Rhodes, D. H., and Hastings, D. E. (2008). Defining changeability: Reconciling flexibility, adaptability, scalability, modifiability, and robustness for maintaining system lifecycle value. *Systems Engineering*, 11(3), S. 246–262.
- Sommerville, I. (2011). *Software Engineering*. Addison-Wesley, Boston 2011.
- Stankovic, J. A. and Ramamritham, K. (1990). What is predictability for real-time systems? *Real-Time Systems*, 2(4):S. 247–254.
- Stouffer, K. ; Falco, J. ; Scarfone, K.: *Guide to Industrial Control Systems (ICS) Security: Supervisory Control and Data Acquisition (SCADA) systems, Distributed Control Systems (DCS)*, Institute of Standards and Technology. CreateSpace Independent Publishing Platform, 2011
- Tanenbaum, A. S. (2009). *Moderne Betriebssysteme*. Pearson Education Deutschland GmbH.
- Theuer, H. (2012). Extension of value stream design for the simulation of autonomous production systems. In *Enabling Manufacturing Competitiveness and Economic Sustainability - Proceedings of the 4th International Conference on Changeable, Agile, Reconfigurable and Virtual production (CARV2011)*, Montreal, Canada, 2-5 October 2011, S. 586–591. Springer-Verlag Berlin Heidelberg.
- Theuer, H., Gronau, N., and Lass, S. (2013). The impact of autonomy on lean manufacturing systems. In Azevedo, A., *Advances in Sustainable and Competitive Manufacturing Systems*, 23rd International Conference on Flexible Automation and Intelligent Manufacturing, S. 1413–1423. Springer.
- Westkämper, E. (2013): Struktureller Wandel durch Megatrends. In: Westkämper, E., Spath, D., Constantinescu, C., Lentjes, J.: *Digitale Produktion*. Springer.
- Windt, K. and Jeken, O. (2009). Allocation flexibility - a new flexibility type as an enabler for autonomous control in production logistics. In *42nd CIRP Conference on Manufacturing Systems*. Citeseer.