

Diplomarbeit

Abstraktion verteilter Produktions- maschinen in cyber-physischen Produktionssystemen

21. September 2016

Peter Heisig

Matr.-Nr.: 3521226

Betreuer

Dipl.-Medieninf. Gordon Lemme,
Dr.-Ing. Sebastian Götz

Verantwortl. Hochschullehrer

Prof. Dr. Uwe Aßmann

Erklärung

Hiermit erkläre ich, dass ich die vorliegende Arbeit mit dem Titel

Abstraktion verteilter Produktionsmaschinen in cyber-physischen Produktionssystemen

unter Angabe aller Zitate und nur unter Verwendung der angegebenen Literatur
und Hilfsmittel selbstständig angefertigt habe.

Dresden, den 21. September 2016

Peter Heisig

Inhaltsverzeichnis

| | | |
|----------|---|-----------|
| 1 | Einleitung | 1 |
| 1.1 | Motivation | 1 |
| 1.2 | Zielsetzung | 2 |
| 1.3 | Methode und Aufbau | 5 |
| 2 | Grundlagen | 7 |
| 2.1 | Fertigung und Automatisierung | 7 |
| 2.1.1 | Kommunikationssysteme | 9 |
| 2.1.2 | Numerische Steuerung | 11 |
| 2.1.3 | Speicherprogrammierbare Steuerung | 13 |
| 2.2 | OPC Unified Architecture | 15 |
| 2.3 | Cyber-physische Produktionssysteme | 18 |
| 3 | Anforderungen | 21 |
| 3.1 | Überwachung | 21 |
| 3.2 | Steuerung | 22 |
| 3.3 | Standardisierung | 22 |
| 3.4 | Lokalität | 23 |
| 4 | Forschungsstand | 25 |
| 4.1 | Netzwerkarchitektur für CPPS | 25 |
| 4.2 | Steuerung und Überwachung aus der Ferne | 26 |
| 4.3 | Zusammenfassung | 28 |
| 5 | Konzeption | 41 |
| 5.1 | Zusammenfassung | 43 |
| 6 | Implementation | 45 |
| 6.1 | Zusammenfassung | 45 |
| 7 | Evaluation | 47 |
| 7.1 | Zusammenfassung | 47 |

| | |
|--------------------------------|------------|
| 8 Zusammenfassung | i |
| 8.1 Schlussfolgerung | i |
| 8.2 Ausblick | i |
| A Anhang | i |
| Abbildungen | iii |
| Programmcode | v |
| Tabellen | vii |
| Literatur | ix |

1 Einleitung

Seit der Mitte des 18. Jahrhunderts ist die industrielle Fertigung in stetigem Wandel. Mit der Entwicklung dampfgetriebener Arbeits- und Kraftmaschinen um 1750 wurde die erste industrielle Revolution eingeleitet. Zum Ende des 19. Jahrhunderts ermöglichten die Einführung arbeitsteiliger Massenproduktion und wissenschaftlicher Betriebsführung das erste Transportband in der fleischverarbeitenden Industrie. Knapp einhundert Jahre nach dieser zweiten Revolution, wurden 1969 erste speicherprogrammierbare Steuerungen (SPS) zur variantenreichen Serienproduktion eingesetzt. Informations- und Kommunikationstechnologie sind seither der Grundstein automatisierungsgetriebener Rationalisierungen. Im Jahr 2011 wurde ausgehend von *Lean Production* der Begriff *Industrie 4.0* geprägt, der die vierte Industrielle Revolution beschreibt (vgl. zu diesem Abschnitt [Gau+14]). Sie zeichnet sich durch neue Ansätze wie das *Internet of Things* (IoT) und cyber-physische Systeme (CPS) im Kontext industrieller Produktion aus [SG16].

1.1 Motivation

Vor der vierten Revolution war klassische Produktionssteuerung zentralisiert und Steuerungstechnik monolithisch strukturiert. Zukünftig wird die Fertigung in cyber-physische Systeme von Systemen zerlegt und mit offenen Standards dezentral betrieben (vgl. zu diesem Absatz [Mil14]).

Moderne Produktionseinrichtungen beherbergen jedoch Maschinen jeden Alters, die zu einem gemeinsamen System verwachsen müssen. Die Technologie zur numerische Kontrolle von Werkzeugmaschinen existiert bereits seit den frühen 1950er Jahren [LHL04]. Gerade diese älteren Anlagen besitzen häufig keine Möglichkeit der Integration in die IT-Systeme einer künftigen Fertigungsstrecke [Wan+04]. Das schlichte Ersetzen dieser Altmaschinen ist aufgrund hoher Kosten meist keine Lösung [Fra16b].

Jedoch behindern diese vorrangig die nahtlose Machine-To-Machine (M2M) Kommunikation durch fehlende Infrastrukturanbindung, womit die Kette von Bearbeitungsschritten für ein Produkt zahlreiche manuelle Eingriffe erfordert.

Vor einigen Jahren wurden bis zu 60% der Arbeitszeit eines Werkers auf die Über-

tragung des Entwurfs eines Fertigungsschrittes in die Umsetzung an der Maschine verwendet [GML00]. So besitzt eine Altmaschine als Teil des Fertigungsprozesses keine Möglichkeit externer Kommunikation und kein *Application Programming Interface* (API) [DP11]. Bei jüngeren Konstruktionen treten Integrationsschwierigkeiten an anderer Stelle auf. So sind selbst bei bestehender Netzwerkfähigkeit geschlossene Soft- und Hardwarearchitekturen und fehlende Schnittstellen verantwortlich für eingeschränkte Überwachung und Steuerung, respektive für die Verhinderung von ökonomisch sinnvoller Automatisierung (vgl. zu diesem Absatz [DP11; FC07]). Weiterhin erschweren die unzureichende Umsetzung von Industriestandards und -normen die Integration der Maschinen [Wan+04; Hop14].

Technische Komponenten, wie eine Netzwerkanbindung, sind nicht die einzigen Barrieren moderner Produktionsautomatisierung. Fehlerbehaftete Kommunikationsmechanismen, sowie die Gefahr der Veräußerung betriebsinterner Daten, sind Probleme die heute gelöst werden können. Auch erfordern sinkende Losgrößen und steigende Produktvariabilität eine flexible Automatisierung von Echtzeitüberwachung und -kontrolle verteilter, rekonfigurierbarer Fertigungssysteme (vgl. zu diesem Absatz [Wan+04; LS01]).

Produktionseinrichtungen basierten bisher auf dem manuellen Sammeln und Verteilen von Daten für Überwachung, Steuerung und Wartung der Maschinen. Doch gegenüber hohen Kosten, menschlichen Fehlern, dem teilweise schlechten Zugang zur Anlage und Aspekten der Datensicherheit, sind Automatisierungslösungen heute günstig, sicher und attraktiv für die Fertigungsindustrie (vgl. zu diesem Absatz [DP11]).

1.2 Zielsetzung

Nach der Motivation und der damit einhergehenden Identifikation des Kernproblems, werden nun die Ziele dieser Arbeit beschrieben. Den Schwierigkeiten in der industriellen Praxis wird wie folgt begegnet:

- Durch die entfernte Kontrolle einer Altmaschine werden manuelle Tätigkeiten wie das Übertragen eines Maschinenprogramms gemindert. Der operative Einsatz einer vormals nicht integrierten Anlage kann damit stärker automatisiert werden und beschleunigt den übergeordneten Produktionsablauf.
- Die zentrale Auswertung von Prozessdaten ermöglicht einen gesamtheitlichen Einblick in die Produktion, wobei jene Daten nicht notwendigerweise zentral zu persistieren sind. Diagnosen geschehen damit nicht mehr vor Ort, wodurch Wartungszyklen besser überprüft und eingehalten werden können. In der Kon-

sequent wird außerdem die Planung der Fertigung vereinfacht und die Zeit bis zur Produktion gesenkt. Weiterhin sollen Störfälle wie Werkzeugbruch und -wechsel ad hoc an Verantwortliche kommuniziert werden.

Im Kontext dieser Arbeit gilt eine Anlage als *integriert*, wenn die infrastrukturelle Einbettung in ein cyber-physisches Gesamtsystem den Anforderungen (vgl. Kapitel 3) genügt. Neben den praktisch orientierten Vorgaben wird die Forschung zur Anlagenmodernisierung für die Industrie 4.0 durch weitere Ziele unterstützt:

- Eine dezentrale Informations- und Kommunikationsarchitektur verbessert die Resilienz, Produktionsstabilität und Skalierbarkeit von verteilten Fertigungssystemen.
- Kommunikationskanäle zwischen einzelnen Maschinen werden aufgrund durchgängig offener Schnittstellen nicht mehr unterbrochen. Durch damit einheitlich mögliche *Machine-To-Machine* (M2M) Kommunikation wird die Kontrolle und Überwachung hierarchisiert und dezentralisiert.
- Die Modellierung von Komponenten und Funktionalität einer Maschine wird durch Standardentwicklungswerkzeuge und -austauschformate vereinfacht.
- Das Optimierungspotential der Gesamtanlage kann durch statistische Auswertung der anfallenden Daten zu Maschinenoperation und -auslastung ausgeschöpft werden.

Die Hierarchisierung von Kontrolle und Überwachung bezieht sich auf das Beispiel der flexiblen Fertigungszelle in denen ein Verbund von Maschinen eine gemeinsame Aufgabe bearbeitet (vgl. [Gro08]). Innerhalb eines solchen Verbunds wird zunehmend über Ethernet kommuniziert, wodurch eine Basis für die TCP/IP Protokollfamilie zur Verfügung steht.

„Aktuell sind nach einer Studie der Fachhochschule Südwestfalen 86% der SPS-Systeme über Ethernet angebunden, wobei von den verbleibenden 14% der Befragten 6% angaben, Ethernet wahrscheinlich in Zukunft einzusetzen.“ [WJN15]¹

Um nun die Ziele im Rahmen dieser Arbeit effektiv erreichen zu können, unterliegen Konzept und Implementierung verschiedenen Einschränkungen und Voraussetzungen. Eine Ethernet-basierte Netzwerkinfrastruktur erlaubt das Einbinden eines virtuellen Maschinenabbilds in die Fertigungsstrecke. Zugang zur Anlage, regelungstechnische Modifikationen und das Anbringen von Sensorik und Aktua-

¹ ursprüngliche Quelle: M. Rothhöft, „Marktstudie: Industrielle Kommunikation“, VDMA, 2013.

toren sind gegeben. Die zu modernisierende Werkzeugmaschine wird durch rechnergestützte numerische Steuerung (CNC) kontrolliert. Automatisierte Maschinenelemente, wie Einspannvorrichtungen oder Schutztüren, sind an eine Speicherprogrammierbare Steuerung (SPS) gekoppelt. Einplatinencomputer sind ausreichend leistungsfähig für die Steuerung und Überwachung von CNC-Maschinen [GM16].

Auch ist das vorgestellte Konzept der Anlagenmodernisierung auf diskrete Fertigung mit bestehender Netzwerkinfrastruktur beschränkt. Unter Berücksichtigung der besprochenen Ziele und Einschränkungen, wird eine konzeptuelle und prototypische Lösung durch die folgenden Schritte erreicht.

1. Ermitteln der Anforderungen für eine Integration von Altmaschinen in moderne, verteilte Produktionsumgebungen – im Folgenden als Retrofitting bezeichnet.
2. Recherchen zum heutigen Stand der Technik und die Einbeziehung vorhandener Systeme.
3. Konzeption einer virtuellen Repräsentation als Schnittstelle der zu integrierenden Anlage.
4. Ermöglichen von dezentraler Kontrolle und Überwachung im Hinblick auf cyber-physische Produktionssysteme.
 - Transfer und Ausführung von Maschinenprogrammen.
 - Erfassen von Produktionsdaten durch angeschlossene Sensoren.
 - Verifikation automatischer Aktionen durch Rückkopplung.
5. Vorstellung eines skalierenden, erweiterbaren Frameworks.
6. Eine prototypische Implementierung belegt die prinzipielle Durchführbarkeit.

Retrofitting ist nicht nur die Integration von Altmaschinen. Im Rahmen dieser Arbeit gilt die Definition von Bergweiler, nach der Retrofitting die Erweiterung des Equipments einer Anlage durch zusätzliche Hardware meint. Der funktionale Umfang einer Maschine wird durch neue Module für die Übertragung und verteilte Verarbeitung der Daten ausgebaut. Dadurch wird die Kommunikation zwischen individuellen Geräten und Produkten der Fertigung ermöglicht, bis die Fabrik den künftigen Standards, Direktiven und Prinzipien der Industrie 4.0 genügt [Ber15].

Nach Klärung der Ziele, Beschränkung des Konzepts und dem Aufzeigen eines groben Lösungswegs werden in dieser Arbeit folgende Fragen zu beantworten sein.

Welchen softwaretechnologischen Konzepten muss die Modernisierung und der infrastrukturelle Kontext einer Altmaschine unterliegen, um eine ganzheitliche Integration in cyber-physische Produktionssysteme (CPPS) gewährleisten zu können?

1. Welche System- und Softwarearchitektur ist für ein flexibles Retrofitting zur Steuerung und Überwachung veralteter CNC-Maschinen im Kontext von CPPS geeignet?
2. Wie und wo werden Informationen zu Maschinenzustand und -operation erfasst, verarbeitet, persistiert und Fremdsystemen zur Verfügung gestellt?
3. Welche standardisierten Protokolle und Datenstrukturen eignen sich für M2M-Kommunikation in einem CPPS?

1.3 Methode und Aufbau

Angelehnt an die *Design Science Research Methodology* (DSRM) wurden bisher grundlegende Probleme identifiziert und die Arbeit motiviert [Gee11]. Durch die folgenden Grundlagen (Kapitel 2) werden essentielle Technologien und Konzepte beschrieben. Die sich anschließenden Anforderungen (Kapitel 3) spezifizieren die Zielvorgaben der darauf entwickelten Lösungskonzepte (Kapitel 4) für die Abstraktion von Maschinen in cyber-physischen Produktionssystemen. Durch die prototypische Implementation (Kapitel 5) des Frameworks und das virtuelle Maschinenabbild wird die prinzipielle Durchführbarkeit des Vorhabens belegt. Die Evaluation (Kapitel 6) hat eine qualitative und quantitative Bewertung von Konzept und Implementation des Prototyps zum Ziel. Schlussendlich werden in der Zusammenfassung ein Fazit und Ausblick (Kapitel 7) auf weitere Forschung gegeben.

2 Grundlagen

Für diese Arbeit relevante Technologien und Konzepte werden in folgendem Kapitel erläutert. Grundlegende Eigenschaften von Fertigung und Automatisierung sind die Basis für das Verständnis von semantischen Informationsmodellen, respektive dem virtuellen Abbild der Realität. Durch dieses Modell ist ein System in der Lage die Produktion und deren Schritte zu überwachen, Differenzen zu erwartetem Verhalten festzustellen und autonom darauf zu reagieren.

2.1 Fertigung und Automatisierung

Fertigung, als Unterbegriff der Produktion, beschreibt Verfahren zur Umwandlung oder Erzeugung von Stoffen mit Hilfe von Energie und Informationen innerhalb eines Prozesses [Lin15]. Automatisierung, nach der DIN 19233, ist „das Ausrüsten einer Einrichtung, so dass sie ganz oder teilweise ohne Mitwirkung des Menschen bestimmungsgemäß arbeitet“. Eine Verknüpfung dieser beiden Konzepte ist in Abbildung 2.1 dargestellt. Die Rückkopplung von Prozessdaten in eine Automatisierungseinrichtung befähigt das System, unter Berücksichtigung von Zielen, steuernd auf die Fertigung zu wirken. Direkte Eingriffe in den Prozess, sowie dessen Beobachtung durch den Menschen, werden verringert - im ökonomischen Zusammenhang rationalisiert (vgl. zu diesem Absatz [Lin15]).

Um die verschiedenen Bereiche der klassischen¹ Automatisierung darzustellen, wird eine Schichtenorganisation herangezogen. Die Struktur eines Unternehmens entspricht dann einer Automatisierungspyramide. Unterschieden werden diese Ebenen aufgrund der unterschiedlichen Anforderungen an Datendurchsatz und Übertragungsgeschwindigkeit (vgl. zu diesem Absatz [Lin15]). Die Ebenen des Beispiels der Abbildung 2.2 erläuterte Linke wie folgt:

- **Unternehmensleitebene**

Marktrelevante Unternehmensprozesse, wie Unternehmensführung, Personal-,

¹ Automatisierung vor der vierten industriellen Revolution.



Abbildung 2.1: Automatisierte Fertigung aus [Lin15]

Investitions- und Produktionsplanung, finden unter Zuhilfenahme eines Systems für Enterprise Resource Planning (ERP) befinden sich auf dieser Ebene.

- **Betriebsleitebene**

Prozesse wie Auftragsbearbeitung und Produktionsplanung, Terminüberwachung und Kostenanalyse sichern auf dieser Ebene den täglichen Fertigungsbetrieb eines Unternehmens. Die softwareseitige Unterstützung übernimmt beispielsweise ein Manufacturing Execution System (MES).

- **Prozessleitebene**

Je nach Größe der Anlage wird in detaillierteren Ebenen die Steuerung und Regelung von Produktionsprozessen und deren Überwachung vollzogen. Verbindungen, wie die zwischen einzelnen Fertigungszellen, werden auf dieser Ebene abgebildet.

- **Steuerungs- & Feldebene**

Aktuatoren und Sensoren und deren Schnittstelle zu Ein- und Ausgangssignalen befinden sich auf der Feldebene. Die von diesen Komponenten konsumierten, beziehungsweise erfassten Daten, werden auf der Steuerungsebene mit dem Prozess verbunden. Daraus entstehende Information werden in die Reaktion des Systems einbezogen, wodurch minimale Latenzen in der Übertragung notwendig sind.

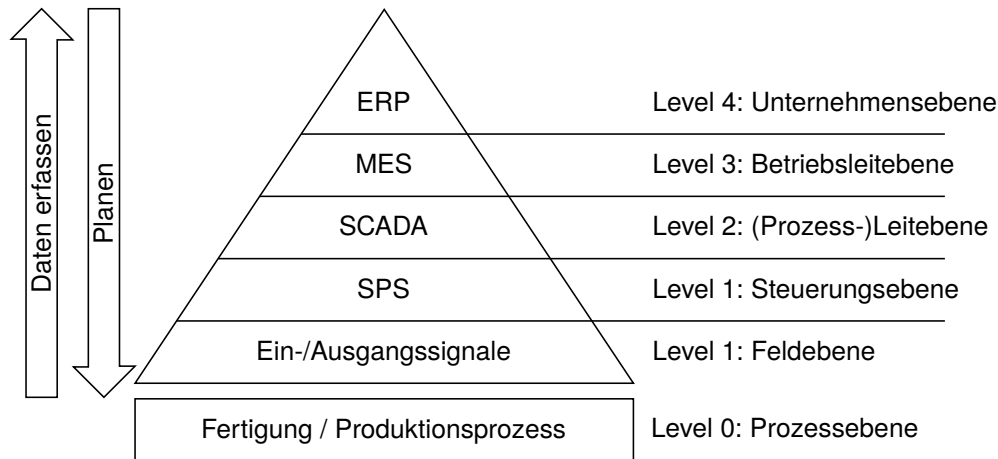


Abbildung 2.2: Beispiel einer klassischen Automatisierungspyramide¹

Auf der Prozessebene (Level 0) des Beispiels geschieht die physikalische Fertigung des Produkts. Dabei fallen große Mengen von Rohdaten an, die in jeder folgenden, höheren Schicht zu abstrakteren Informationen verarbeitet werden. Die Toleranz bezüglich der Übertragungsgeschwindigkeit ist auf diesen unteren Ebenen am geringsten. Speicherprogrammierbare Steuerungen (SPS, vgl. Abschnitt 2.1.3) und numerische Kontrolle (NC, vgl. Abschnitt 2.1.2) erhalten Befehle in Echtzeit und automatisieren den Produktionsablauf. Über einen Feldbus (vgl. Abschnitt 2.1.1) werden diese Instruktionen und Messdaten an ein Supervisory Control and Data Acquisition (SCADA) System gekoppelt. Derlei Systeme sind verantwortlich für die Überwachung und Steuerung technischer Prozesse und kontrollieren die übergeordnete Fertigungszelle, respektive Verbünde von Werkzeugmaschinen, Robotern und automatisierten Komponenten [Lin15]. Ein MES, beziehungsweise Fertigungsmanagementsystem bildet dann die Schnittstelle zum Ressourcenmanagementsystem (ERP) der Unternehmensebene.

2.1.1 Kommunikationssysteme

In der industriellen Fertigung werden Feldbusse als Kommunikationskanal in Feld- und Steuerungsebene genutzt. Neben dem Feldbus existieren weitere, teils veraltete Kommunikationskanäle, die zu der in Abbildung 2.3 dargestellten, heterogenen Automationsstruktur führen.

Hammertringl und Reinhart fanden vier Klassen von Kommunikationssystemen in dieser durch eine Befragung ermittelten Struktur. Sensoren und Aktuatoren der

¹ Darstellung durch Wikipedia-Nutzer UlrichAAB.



Abbildung 2.3: Repräsentative Automationsstruktur nach [HR15]

ersten Klasse sind direkt verbunden und übermitteln binäre oder analoge Informationen rein physikalischer Natur, wie Strom mit 24V oder Druckluft. Sie stellen keine standardisierte, digitale Beschreibung ihrer Funktion bereit, wodurch dem angeschlossenen Gerät diese händisch mitgeteilt werden muss.

Feldgeräte mit *IO-Link*¹-Fähigkeit können durch die IO Device Description (IO-DD) beschrieben werden und sind Teil einer zweiten Klasse mit Direktverbindung. Innerhalb dieser können minimale Protokolle beschrieben und die Identifikation der Komponenten durchgeführt werden. Dadurch kann innerhalb dieser Klasse extern parametrisiert und eine maschinenlesbare Beschreibung übertragen werden. Letzteres wird in der Praxis jedoch kaum genutzt. Ein weiterer, stark verbreiteter Repräsentant ist die serielle Schnittstelle RS-232.

Die dritte Klasse von Kommunikationssystemen verbindet Bus-basierte Geräte. Traditionelle Bussysteme und Real-Time Ethernet (RTE) sind Stand der Technik und erlauben die Definition der Strukturen durch einen Bus-Master. Bildverarbeitungssysteme, ihre Protokolle (z.B. GigE Vision²) und Beschreibungssprachen (z.B. GenICam³) sind hier verbreitet. Für Konfiguration und Überwachung der Systeme wird eine Mensch-Maschine-Schnittstelle (MMS) eingesetzt (vgl. zu diesem Absatz [HR15]).

Feldbusse sind digitale bidirektionale, serielle Kommunikationsnetzwerke für echtzeitfähige, verteilte Kontrolle von Instrumenten, Steuerungseinheiten und Aktuatoren [Mah03]. Trotz der Standardisierungsbemühungen durch IEC 61158, existieren

¹ Implementierung der IEC TR 61131-9

² Schnittstellenstandard industrieller Bildverarbeitung

³ Schnittstellenstandard für industriell eingesetzte Kameras

unterschiedliche Feldbusse wie CAN, ProfiBUS oder EtherCAT. Jeder Hersteller von Maschinen, Robotern und automatisierten Komponenten stellt einen anderen Busstandard, weshalb die Kommunikation der Geräte nicht garantiert werden kann. Für deren Verbindung mit unterschiedlichen Systemen wird ein Adapter benötigt, wodurch der Aufwand bezüglich Bereitstellung und Konfiguration steigt (vgl. zu diesem Absatz [Pau+13]).

Auf den höheren Ebenen der Automatisierungspyramide (vgl. Abbildung 2.2) etablierte sich das nicht Echtzeit-fähige Ethernet. Die Variante des RTE verbreitet sich jedoch zunehmend auch auf den unteren Ebenen (vgl. Abschnitt 1.2) und erlaubt Kommunikation mit Remote Procedure Calls (RPC), TCP/IP-Sockets und OPC (ursprünglich OLE¹ for Process Control, vgl. Abschnitt 2.2) [Pau+13]. Die Homogenisierung der Infrastruktur, vom Ressourcenmanagement im ERP über die Speicherprogrammierbare Steuerung bis zum einzelnen Sensor auf der Feldebene, vereinfacht den Informationsaustausch und trägt zur Flexibilisierung des Gesamtsystems bei. Weiterhin stehen damit die Daten aller Schichten für jeden anderen Netzwerkteilnehmer zur Verfügung.

Diese Form der Kommunikations- und Informationsstruktur ist nach Hammerstingl und Reinhart in einer vierten Klasse zu finden. Abbildung 2.3 zeigt OPC und OPC UA (vgl. Abschnitt 2.2) als Standard für den Datenaustausch zwischen dem Produktionsplanungssystem (PPS) und den speicherprogrammierbaren Steuerungen (SPS, vgl. Abschnitt 2.1.3). Mit dieser Technologie stellen Geräte aktiv ihre virtuelle Beschreibung bereit, was durch die Hersteller unterstützt und vorangetrieben wird (vgl. zu diesem Absatz [HR15]). Auch mit numerischer Steuerung (CNC, vgl. Abschnitt 2.1.2) kontrollierte Werkzeugmaschinen können so Informationen zu Zustand und Produktionsfortschritt bereitstellen.

2.1.2 Numerische Steuerung

Für die Fertigung eines Produkts werden Bauteile benötigt, die durch Werkzeugmaschinen entstehen. In der DIN 69651 ist eine Werkzeugmaschine definiert als eine „mechanisierte und mehr oder weniger automatisierte Fertigungseinrichtung, die durch relative Bewegung zwischen Werkstück und Werkzeug eine vorgegebene Form am Werkstück oder eine Veränderung einer vorgegebenen Form an einem Werkstück erzeugt“ (Zitat aus [Hir00]). Ein Werkstück ist dabei ein Rohling eines bestimmten Materials, welcher auf eine bestimmte Art durch Bearbeitung verändert wird. Die aus diesem entstehende Form wird mit Software für Computer-Aided Design (CAD) entworfen, wobei eine zwei- oder dreidimensionale Visualisierung des

¹ Object Linking and Embedding

Modells den Konstrukteur unterstützt. In einem zweiten Schritt werden die so entstandenen Konstruktionspläne in Bewegungsmuster umgewandelt. Der etablierte Kodierungsstandard für die Steuerungsinformationen zu diesen Mustern ist durch die DIN 66025, beziehungsweise ISO 6983 normiert und als G-Code bekannt. Eine vollständige Kompatibilität der Befehle zwischen den Anlagen wird aufgrund spezifischer Werkzeug- und Maschinenparameter, wie Drehzahlen oder Begrenzungskordinaten der Arbeitsfläche, verhindert. Durch Präprozessoren und manuelle Anpassungen werden Steuerungsinformationen des zweiten Schritts auf die Maschine angepasst.

Wurde der vorverarbeitete G-Code auf eine Werkzeugmaschine mit Computer Numerical Control (CNC) übertragen, kann diese mit dem eigentlichen Fertigungsschritt beginnen. Derlei Maschinen besitzen mehrere durch Schrittmotoren angetriebene Bearbeitungsachsen, welche die Position des Werkzeugs relativ zum Werkstück durch Translation und Rotation (Hilfsachsen) verändern. Mit dieser relativen Bewegung wird sukzessiv Material entfernt, wodurch die im CAD entworfenen Bauteile physisch entstehen. Für das Entfernen von Material werden verschiedene Typen von Werkzeugmaschinen eingesetzt. Drehmaschinen und Fräsen sind hier die prominentesten Repräsentanten, wobei zum Beispiel auch spezielle Roboter mit Befehlen der CNC gesteuert werden können.

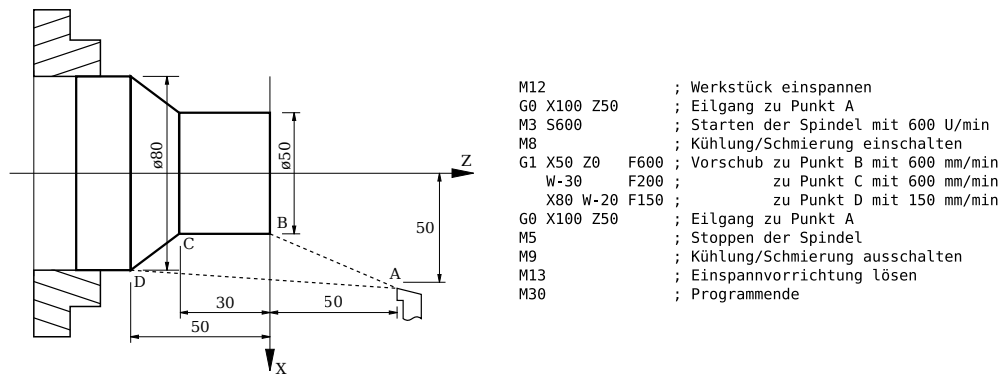


Abbildung 2.4: Beispielkonstruktion und G-Code für eine Drehbank

G-Code teilt sich in zwei Gruppen von Instruktionen, die wiederum herstellerspezifisch beziehungsweise generisch sind. Abbildung 2.4 zeigt das Beispiel einer Konstruktion und dessen Programm mit Bewegungsinstruktionen (G-Befehle) und sonstige Anweisungen (M-Befehle) für die Herstellung auf einer Drehbank¹. Das Einspannen des Rohlings durch M12 schließt die Klammern - in der Konstruktion als schraffierte Blöcke dargestellt. Ein Eilgang, wie durch G0 X100 Z50, richtet das Werkzeug ohne das Entfernen von Material an einem bestimmten Punkt aus, wobei sich A im Beispiel an Position (X = 100, Z = 50) befindet. Der Befehl G1 wird für den

¹ nach www.helmancnc.com/cnc-lathe-simple-g-code-example-g-code-programming-for-beginners vom 07.10.2016

eigentlichen Fräsvorgang verwendet und lässt das Werkzeug mit einer Geschwindigkeit (Vorschub) von 600 mm/min durch das Material laufen.

Die Vorteile der Fertigung mit CNC liegen in der Wiederholbarkeit und Genauigkeit der Operation. Weiterhin wird die Rüstzeit, jene zum Einstellen der Maschine, verringert und damit die Produktivität erhöht (vgl. zu diesem Absatz [Smi08]).

Dennoch sind auch moderne CNC-Anlagen in ihrer Funktion limitiert, da der verwendete G-Code lediglich Instruktionen und prozedurale Daten abbilden kann, wodurch ein Großteil der Konstruktionsinformationen verloren geht. Zwei neue Standards, namentlich STEP-NC (ISO 10303-238) und Function Blocks (IEC 61499), etablieren sich aus diesem Grund. Durch diese sollen CNC-Maschinen mit mehr Informationen, für intelligenteren Fertigung und bessere Interoperabilität, ausgestattet werden. Beispielsweise wird durch STEP-NC die Abhängigkeit von Parametern reduziert. Neuberechnungen zur Laufzeit beziehen unter anderem Verformungen durch Erhitzen des Werkstücks in die Fahrtenplanung ein. Function Blocks dagegen, sind Teil eines Standards für verteilte industrielle Prozesse und Kontrollsysteme. Sie kapseln Maschinendaten, wie Werkzeugeigenschaften oder Algorithmen, für CNC. (vgl. zu diesem Absatz [XLY06]).

Darüber hinaus besitzen Anlagen spezifische, automatisierte Maschinen- und Werkzeugkomponenten, die nicht durch CNC steuerbar sind. Schließmechanismen, Abluftsysteme oder Materialzufuhr werden von zusätzlichen Systemen in die Fertigung integriert.

2.1.3 Speicherprogrammierbare Steuerung

Daten und Zustände automatisierter Maschinen- Werkzeug- und Fertigungsprozesskomponenten werden durch speicherprogrammierbare Steuerungen (SPS) aufgenommen, verarbeitet und verändert. Durch der DIN IEC 60050-351 7-2013 sind sie definiert als eine „rechnergestützte programmierte Steuerung, deren logischer Ablauf über eine Programmierereinrichtung, zum Beispiel ein Bedienfeld, einen Hilfsrechner oder ein tragbares Terminal, veränderbar ist“ (Zitat aus [HLG15]). Nach Heinrich et al. sind folgende Komponenten dafür notwendig. Ein Hardwaresystem stellt die Verbindung zum Fertigungsprozess und weiteren automatisierten Anlagen her. Programm- und Datenspeicher sind für die Persistenz des Anwenderprogramms, beziehungsweise der Prozessdaten verantwortlich und werden durch die Verarbeitung verändert. Dafür sind ein Betriebssystem, sowie ein Anwenderprogramm zuständig.

Der prinzipielle Aufbau einer SPS umfasst Stromversorgungs-, Signalverarbeitungs- und vier Schnittstellenfunktionen für den Datenaustausch. Eine Mensch-Maschine-

Schnittstelle ermöglicht dem Bediener den operativen Betrieb, z.B. durch Statusanzeigen, zu überwachen. Die Programmier- und Test-Schnittstelle wird durch einen Programmierer, neben der Implementierung von Instruktionen, auch zur Fehleranalyse genutzt. Kommunikationsfunktionen erlauben die Anbindung an externe Systeme, automatisierte Komponenten und Datenquellen. Aktuatoren und Sensoren werden über die Prozessschnittstelle an eine SPS gekoppelt. Der konzeptuelle Aufbau wird durch ein, in Abbildung 2.5 dargestelltes, Hardwaresystem implementiert. Die Stromversorgungseinheit liefert die für den Betrieb notwendige Energie und kann an die jeweilige Quelle angepasst werden. Eine Zentraleinheit mit CPU, Speicher, Verarbeitungseinheit und Anschlüssen für Programmiergeräte und Netzwerk bildet die Steuerungslogik ab. Die Kopplung an digitale und analoge Datenquellen und -empfänger erfolgt über Ein-/Ausgabe-, beziehungsweise Signaleinheiten.

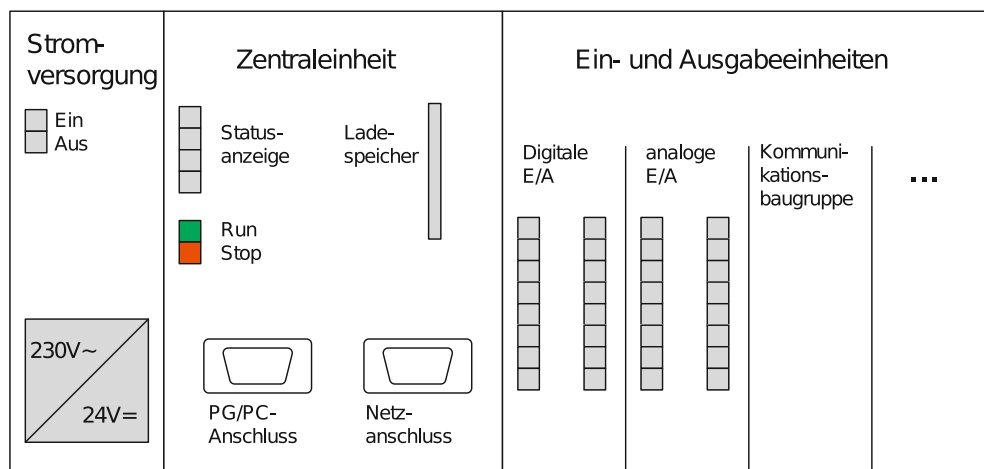


Abbildung 2.5: Grundlegender Hardwareaufbau einer SPS nach [HLG15]

Signalverarbeitungsfunktionen im Anwenderprogramm werden innerhalb der Verarbeitungseinheit zyklisch ausgeführt. In einem ersten Schritt werden dafür die aktuellen Zustände der Eingänge erfasst, z.B. Sensordaten, und der Datenspeicher gelesen. Das Anwenderprogramm wird nach dem EVA-Prinzip (Eingabe, Verarbeitung, Ausgabe) abgearbeitet, woraufhin Ausgangssignale, z.B. Befehle für Aktuatoren, erzeugt werden. Diese Zyklen verbrauchen Zeit, die kritische Aktionen verhindern können.

So existieren neben den zyklusorientierten auch unterbrechungsfähige SPS. Das aktuell laufende Programm kann dabei durch Interrupts pausiert und später wieder aufgenommen werden. Programmiert werden Anwenderprogramme mit normierten Methoden der EN 61131-3. In textueller Form existieren Standards für Anweisungslisten und strukturierten Text. Grafisch wird die Logik einer SPS durch Kontaktpläne, eine Funktionsbaustein- oder Ablaufsprache implementiert.

Eine weitere Kategorie bilden ereignisorientierte Steuerungen bei denen das Anwenderprogramm erst bei Statusveränderungen der Eingangssignale abgearbeitet wird. Weiterhin unterschieden werden modulare SPS, wie die *Simatic S7*-Serie von Siemens, und kompakt-SPS, wie zum Beispiel *LOGO!*. Letztere zeichnen sich durch fehlende Erweiterbarkeit aus. Neben hardwarebasierten SPS ermöglichen Software-Steuerungen (Soft-SPS) eine weitere Stufe der Flexibilisierung. Steuerungen mit Echtzeit-Betriebssystemen, auch in eingebetteten Recheneinheiten, übernehmen hier die Überwachung und Kontrolle des Prozesses, sind jedoch weniger verbreitet.

Eine Alternative zu SPS bietet die verbindungsprogrammierte Steuerung (VPS), bei der die Komponenten der Ein- und Ausgabe festverdrahtet und die Logik vordefiniert ist. Die speicherprogrammierbare Variante hat nicht nur den Vorteil der Flexibilität. Der Funktionsumfang, die Verarbeitung analoger und digitaler Daten, sowie die geringen Betriebskosten etablierten die SPS als Standard in der industriellen Fertigungsautomatisierung.

2.2 OPC Unified Architecture

Der Austausch und die Modellierung von Daten kann in einem heterogenen Technologieraum nur durch standardisierte Beschreibungssprachen, Kommunikationsprotokolle und Modelle erreicht werden. Diese Aussage wird im Zusammenhang mit Industrie 4.0 durch eine Tendenzbefragung von BITKOM, VDMA und ZVEI aus dem Jahr 2013 gestützt. So sehen Mitarbeiter aus 278 Unternehmen, des Maschinen- und Anlagenbaus, Standardisierung als größte Herausforderung für die Umsetzung von Industrie 4.0 [KWH13].

Die OPC¹ Foundation ist ein Industriekonsortium, das für Entwicklung und Wartung solcher Standards verantwortlich ist. Sie schuf Interoperabilitätsstandards für den sicheren und zuverlässigen Austausch von Daten im Automatisierungsraum industriell produzierender Unternehmen auf Basis des *Distributed Component Object Model* (DCOM). Dieses ist ein von Microsoft definiertes System für entfernte Methodenaufrufe (Remote Procedure Calls, RPC) innerhalb eines Windows-Ökosystems, das für die heutigen heterogenen Informationssysteme ungeeignet ist. Neben der Plattformunabhängigkeit ist die Zusicherung des nahtlosen Übergangs von Informationen, zwischen Geräten unterschiedlicher Hersteller, die Hauptaufgabe querschnittlicher Spezifikationen im Kontext der *OPC Unified Architecture* (OPC UA)². Das Konsortium berücksichtigte bei der Spezifikation folgende Ziele [OPC14].

¹ Open Platform Communications

² nach opcfoundation.org/about/what-is-opc vom 23.09.2016

- sicherer, zuverlässiger Informationsaustausch
- Plattform- und Herstellerunabhängigkeit
- standardisierte Kommunikation über Internet und Firewalls
- serviceorientierte Architektur (SOA)
- Schutz vor unerlaubtem Zugriff
- Erreichbarkeit und Zuverlässigkeit
- Vereinfachung durch Vereinheitlichung

Konkret bietet die OPC UA (auch IEC 62541) einen semantischen Kommunikations- und Datenmodellierungsstandard für den Informationsaustausch [Aya+13]. Ein erweiterbares Meta-Modell spezifiziert die Grundbausteine und Regeln für ein Informationsmodell und beinhaltet verschiedene Einstiegsknoten und Basis-Typen [OPC14]. Informationsmodelle sind Repräsentationen von Konzepten, Relationen, Beschränkungen, Regeln und Operationen zur Spezifikation der Bedeutung (Semantik) von Daten innerhalb einer bestimmten Domäne [Lee99]. Diese werden von Maschinen, Baugruppen und anderen Ressourcen im Adressraum angeboten, wodurch jede Entität innerhalb eines IT-Ökosystems mit der jeweilig anderen kommunizieren kann und deren strukturelle Eigenschaften kennt.

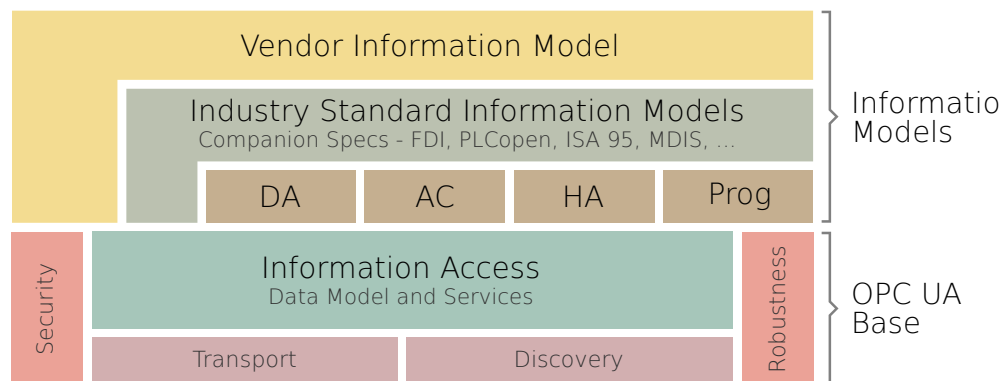


Abbildung 2.6: Spezifikationen von OPC UA

In Abbildung 2.6 ist die dafür notwendige Informationsstruktur dargestellt [OPC14]. Auf der untersten Ebene werden Transportprotokolle, das Meta-Modell und grundlegende Services beschrieben. Das bevorzugte Protokoll setzt auf TCP/IP auf und erlaubt einen performanten Austausch von beispielsweise Geräte-, Sensor-, Maschinen- und Prozessdaten innerhalb einer Client-/Server-Architektur. Eine für die Kommunikation über Firewalls taugliche Methode bietet die im Standard verankerte HTTP/XML-Schnittstelle. Durch einen Discovery-Mechanismus können Funktionen und Eigenschaften von OPC-UA-fähigen Teilnehmern in einem Subnetz be-

kannt gegeben werden. Doch auch Dienste für Ereignisregistrierung oder Sitzungsmanagement sind Teil der elementaren Definitionen. Eckpfeiler dieser Basis von OPC UA sind Fehlertoleranz und Sicherheit als zentrale Aspekte der Spezifikation. Auf Details zur Sicherheitsinfrastruktur wird in dieser Arbeit nicht eingegangen. Darauf aufbauend werden generische Informationsmodelle definiert, die unter anderem den Adressraum eines Servers repräsentieren [OPC14]:

- **Data Access (DA)**
Daten zur Automatisierung werden durch das Data Access Modell strukturiert. Sowohl analoge und diskrete Variablen, als auch Einheiten und Qualitätsattribute werden hier beschrieben. Sensoren, Steuerungen und Regler sind beispielsweise Quellen jener Variablen.
- **Alarms and Conditions (AC)**
Die Behandlung von Dialogen und Alarmen wird mit diesem Modell definiert. Events, beziehungsweise Ereignisse auf die sich Clients registrieren, werden durch Veränderungen im Zustand eines Geräts ausgelöst.
- **Historical Access (HA)**
Historische Variablenwerte und Events werden mit diesem Modell dargestellt. Die Persistenz dieser Daten ist konfigurierbar.
- **Programs**
Komplexe Aufgaben werden durch Programme repräsentiert und mit Zustandsautomaten beschrieben.

Viele bereits existierende Modelle, wie MTConnect, PLCopen, FDI und ISA95, unterscheiden sich von OPC UA durch fehlende Erweiterbarkeit. Semantische Zusammenhänge lassen sich oft nicht ohne weiteres darstellen, wie Hoppe schrieb [Hop14]:

- Wie geben sich z. B. ein „Temperatursensor“ oder eine „Ventilsteuerung“ zu erkennen?
- Welche Objekte, Methoden, Variablen und Ereignisse definieren die Schnittstelle für Konfiguration, Initialisierung, Diagnose und Laufzeit?

Die Erweiterbarkeit des Informationsmodells von OPC UA (vgl. Abbildung 2.6) ermöglicht *Companion Specifications*, die diesen Mangel ausgleichen und zusätzlich domänenspezifische Definitionen erlauben. Plattformunabhängigkeit wird durch frei verfügbare, aber auch proprietäre Implementierungen des Softwareinfrastruktur-Stacks ermöglicht. Ein API, Codegeneratoren für den Adressraum und Entwick-

lungswerkzeuge werden für die Programmiersprachen Ansi C/C++, .NET, Java und weitere durch die OPC Foundation bereitgestellt.

2.3 Cyber-physische Produktionssysteme

Die Verbindung von Überwachung und Kontrolle technischer Systeme mündet in Paradigmen, die Realität und virtuelle Umgebungen miteinander verschmelzen lassen. So wurde das Konzept cyber-physischer Systeme (CPS) 2006 durch Edward A. Lee erstmalig erläutert. Er versteht diese als Integration von Informationsverarbeitung und physischen Prozessen. Virtuelle und physische Abläufe werden durch Sensoren und Aktuatoren überwacht, beziehungsweise beeinflusst, stehen in unmittelbarer Wechselwirkung und sind durch Kontrollschleifen rückgekoppelt [Lee06]. Der historische Weg, hin zu darauf aufsetzenden Systemen, ist in Abbildung 2.7 dargestellt¹.

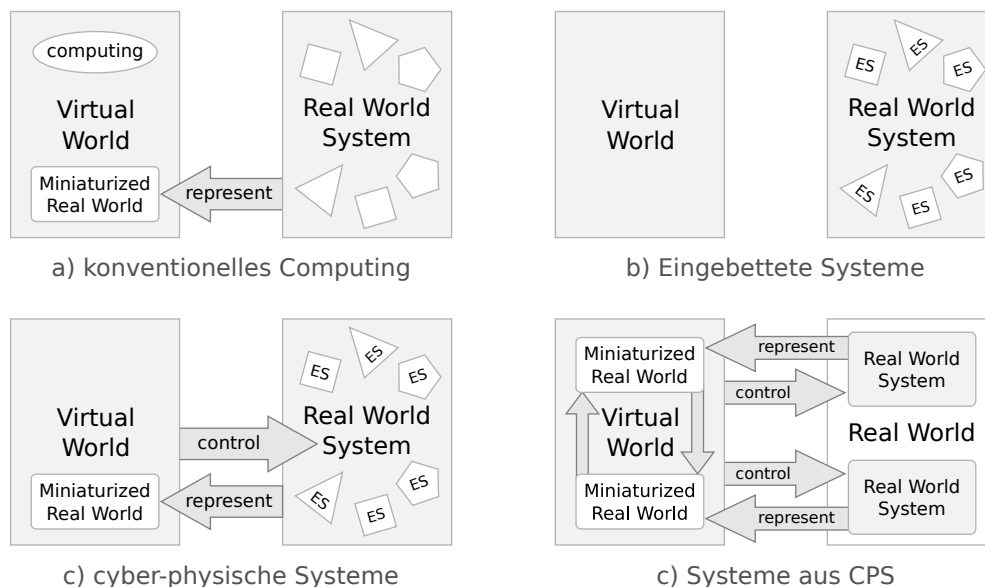


Abbildung 2.7: Der historische Weg zu CPSoS

Im konventionellen Computing (Abbildung 2.7a) sind Systeme der physischen Welt durch abstrakte Modelle repräsentiert. Berechnungen bezüglich der Realität werden in Simulationen auf diesen Modellen durchgeführt. Durch eingebettete Systeme (ES, Abbildung 2.7b) wird der Computer in das Realweltobjekt integriert, wodurch Berechnungen in die physikalischen Systeme getragen werden. Mit CPS

¹ Abbildung 2.7 und folgender Absatz nach Vortrag *Life with Cyber-Physical Systems* von Prof. Dr. Uwe Aßmann, 29. Juni 2016, Technische Universität Dresden

(Abbildung 2.7c) existiert nicht nur ein passives Modell des Realweltssystems. Das Wissen um den Zustand des Realitätsausschnitts verhilft zur Steuerung der ES, wodurch neben dem realen Objekt ein synchrones, virtuelles Modellobjekt entsteht. Diese Konzept *dualer Realität* von Objekten steht für die Kontrolle von Dingen der physischen Welt. Um die Synchronität des Modells gewährleisten zu können, müssen Rückkopplungsschleifen die Effekte physischer Prozesse auf Berechnungen und Simulationen beziehungsweise vice versa verifizieren [Lee08]. Weiterhin sollen derlei Systeme autonom auf Diskrepanzen reagieren und korrigierende Maßnahmen einleiten. In Systemen von CPS (CPSoS, Abbildung 2.7d) wird die physische Welt in Realweltssysteme gegliedert, die über ihre Modelle interagieren. CPSoS bieten Potential für die vierte industrielle Revolution und sind Grundlage cyber-physischer Produktionssysteme (CPPS).

Produkte, Maschinen und andere Ressourcen werden in diesen durch CPS repräsentiert, welche Informationen und Dienste über das Netzwerk der gesamten Produktionsstrecke teilen. CPS sind fundamentale Elemente eines CPPS, die unmittelbaren Zugriff auf relevante Informationen, Maschinenparameter, Produktionsprozesse und deren Produkte besitzen. Durch die Dezentralisierung der Produktionslogik haben CPPS, im Gegensatz zu traditionellen Produktionssystemen, wesentliche Vorteile bezüglich Transparenz, Adaptivität, Ressourceneffizienz und Flexibilität. Auf Ebene der Automatisierung werden Informationen eines CPS-Netzwerk benötigt, um den Fertigungsprozess auf Basis von strategischen Entscheidungen erfolgreich durchzuführen. Für Entscheidungsfindung und Kontrolle der Fertigung werden konsistente, kohärente Informationen über die reale Welt benötigt [Ber15].

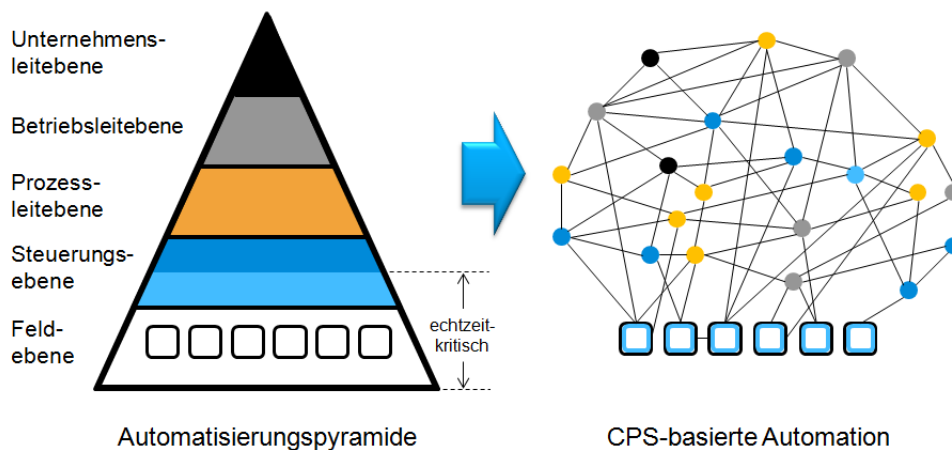


Abbildung 2.8: Auflösung der Automatisierungspyramide aus [Ver13]

Diese Informationen, Dienste und Funktionen werden an jener Stelle lokalisiert, die im Sinne einer flexiblen Entwicklung und Produktion den größten Vorteil bringt [Ver13]. Starre Strukturen, wie die der klassischen Automatisierungspyramide, sind

ungeeignet für die dezentrale Verortung der genannten Ressourcen. Die demnach notwendige Auflösung dieser Architektur zu einem vernetzten System von Systemen, beziehungsweise CPSoS, wird in Abbildung 2.8 verdeutlicht. Sowohl Hard- und Software als auch die Verarbeitung der anfallenden Daten wird nicht länger in Schichten organisiert werden [SSG13]. Damit sollen die Produktionsressourcen auf Knoten eines Netzwerks aufgeteilt und schrittweise auf ihre funktionale Struktur abstrahiert werden [Ver13]. Bis eine geeignete Architektur für CPPS andere Möglichkeiten bietet (vgl. [LBK15; Bor+14]), werden Echtzeit-Steuerungen in der Feldebene verbleiben [Ver13].

3 Anforderungen

Für die in Abschnitt 1.2 aufgestellten Ziele, werden in diesem Kapitel die spezifischen Kriterien zu deren Erfüllung erläutert.

Informationssysteme in der Produktion dienen der Verbesserung der Wettbewerbsfähigkeit und müssen Innovations- und Zeitdruck standhalten. Moderne Produktionsumgebungen helfen Arbeitsabläufe zu optimieren und vereinfachen Beteiligten die Ausführung ihrer Arbeit. Jedoch verhindern Altmaschinen aufgrund fehlender Infrastrukturanbindung und geschlossener Architekturen (vgl. [DP11]) die Vollautomatisierung dieser Arbeitsabläufe und erfordern die physische Anwesenheit einer Fachkraft [Wan+04].

3.1 Überwachung

Im Wartungs- und Störfall muss der Zustand der Anlage bekannt sein. Dieser kann bei nicht integrierten Altmaschinen nur am Terminal eingesehen werden. Ein Techniker muss die Betriebs- und Prozessdaten vor Ort erfassen um eine Diagnose stellen zu können und unter anderem das ERP-System darüber zu informieren. Weiterhin kann eine cyber-physikalische Rückkopplungsschleife nicht autonom auf den Prozess wirken, wenn die Daten in keinem virtuellen Weltmodell zur Verfügung stehen.

REQ1 Die Überwachung von Betriebs- und Prozessdaten der Altmaschine und ihrer automatisierten Maschinen- und Werkzeugkomponenten ist ortsunabhängig, so dass Zustandserfassung und Störfalldiagnose durch Subsysteme des CPPS erfolgen kann.

3.2 Steuerung

Um einen bestimmten Fertigungsschritt an einer numerisch kontrollierten (NC) Anlage durchzuführen, muss das auszuführende Programm übertragen werden. Auch Speicherprogrammierbare Steuerungen (SPS) benötigen ein oft händisch übermitteltes Anwenderprogramm. Diese Befehlsketten werden entweder mit einem Speichermedium auf den Steuerungs-PC kopiert oder direkt an dessen Terminal kodiert. Der zeitliche Aufwand und das notwendige Personal verlangsamen die Fertigung des Endprodukts und führen zu einer suboptimalen Fertigungsstrecke. Für das Retrofitting der Anlage muss die entfernte numerische Kontrolle ermöglicht werden. Weiterhin sind Produktionsmaschinen mit zusätzlichen automatisierten Komponenten wie Schließmechanismen für Schutztüren, Kühl-, Entlüftungs- oder Einspannsystemen ausgestattet. Auch die Steuerung dieser muss ortsunabhängig sein, damit ein CPPS ganzheitlich in den Produktionsprozess eingreifen kann.

REQ2 Die Steuerung der Altmaschine und ihrer automatisierten Maschinen- und Werkzeugkomponenten ist ortsunabhängig, so dass Übertragung, Ausführung und Abbruch von NC-Programmen, beziehungsweise produktionsbedingter Steuerbefehle, durch Subsysteme des CPPS erfolgen kann.

Die steigende Automatisierung zur Optimierung der Produktionsabläufe wird in einem CPPS durch Rückkopplung erreicht. Mit den Einhalten der Anforderungen zu Überwachung und Steuerung hat das System die Möglichkeit automatisch auf veränderte Bedingungen zu reagieren.

3.3 Standardisierung

Nach Ferrolho et al. entstehen auch mit Netzwerkanbindung und Programmierschnittstellen noch zu überwindende Probleme [FC07]. CNC-Maschinen basieren auf einer geschlossenen Architektur numerischer Kontrolle und sind nicht für die Integration mit anderen ausgelegt. Die Kontrolleinheit der Anlage lässt die Steuerung von einem entfernten PC nicht zu. Programmierumgebungen sind nicht ausreichend leistungsfähig um komplexe Aufgaben, wie die kollaborative Operation innerhalb einer flexiblen Fertigungszelle, zu entwickeln. Unterschiedliche Hersteller verwenden eigene Programmiersprachen und Entwicklungstools, wodurch Integration und gemeinschaftliche Produktion erschwert werden. Die sich damit ergebende Heterogenität der Anlagen einer Produktionsstrecke ist ein bereits betrachtetes Problem cyber-physikalischer Systeme (vgl. [SG16]). Im Falle proprietärer Schnittstellen und

geschlossener Architekturen muss ein Adapter die Standardisierung von Protokollen und Informationen durchsetzen [Aya+13]. Für SPS gelten in diesem Zusammenhang die gleichen Anforderungen.

REQ3 Standardisierte Informationsprotokolle und -modelle werden für die Integration heterogener Altmaschinen eingesetzt, so dass Datenaggregation und M2M-Kommunikation gesamtheitlich gewährleistet werden kann.

3.4 Lokalität

CPPS müssen in geringstmöglicher Zeit Betriebs- und Prozessdaten der Maschine analysieren, bewerten und in den Produktionsprozess eingreifen können. Die Synchronisation des virtuellen Modells der Realität wird jedoch durch stetig wachsende Datenvolumina aufgrund steigender Geräteanzahl erschwert. Damit verlangsamt sich die Verarbeitung der Daten mit der geografischen Entfernung zwischen Gerät und System (vgl. [Bon+12]). Bei der Integration von Altmaschinen muss demnach die Datenanalyse, -persistenz und Historie, sowie die Reaktion auf dadurch erkannte Veränderungen möglichst nahe an der Anlage geschehen. Läuft eine Rückkopplungsschleife direkt an der Maschine, muss außerdem nur ein Teil der anfallenden Daten veräußert und die Kontrolle nur teilweise an hierarchisch übergeordnete Systeme abgegeben werden [Bon+12]. Durch den verminderten Austausch zwischen den Systemen werden die Sicherheit der Daten verbessert und Kommunikationsfehler minimiert (vgl. [Wan+04]).

REQ4 Die Erfassung und Persistierung anfallender Betriebs- und Prozessdaten, sowie die Interpretation von Maschinenbefehlen geschieht geografisch nahe der Anlage, wodurch zeitliche Latenzen, Kommunikationsaufwände und -fehler minimiert werden.

Auch wenn die Zeit für die Kommunikation von Steuerbefehlen und Sensordaten durch die Nähe zur Maschine minimiert wird, ist weder harte noch weiche Echtzeit ein Kriterium. Es wird davon ausgegangen, dass die Interpretation und Ausführung der Maschinenbefehle, sowie die Aggregation der Daten, direkt an der Maschine geschieht. Um in adäquater Zeit reagieren zu können, unterliegen die für CPPS erforderlichen Kontrollschleifen damit ebenfalls dem Lokalkriterium [Bon+12].

4 Forschungsstand

Nach der Spezifikation der Zielvorgaben werden in diesem Kapitel der aktuelle Stand der Technik, sowie bereits bestehende Forschungsarbeiten zum Thema erläutert und mit den aufgestellten Kriterien für eine Lösung abgeglichen.

4.1 Netzwerkarchitektur für CPPS

Monolithische Architekturen sind ungeeignet für cyber-physische Systeme (CPS). Durch die Aufteilung eines skalierenden ganzheitlichen Systems in Subsysteme entsteht ein *System of Systems* (SoS). Abbildung 4.1 verbildlicht das Konzept der Systemkapselung und Verbindung einzelner Produktionskomponenten in einer Architektur für CPPS.

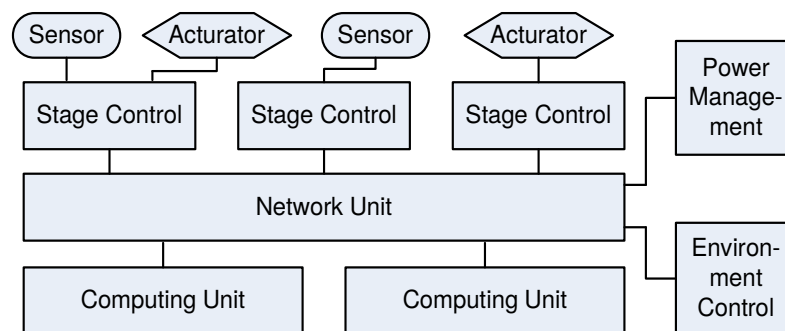


Abbildung 4.1: CPS-Architektur nach Wang et al. [WVG08]

Diese von Wang et al. vorgestellte heterogene Struktur für Steuerungsnetzwerke besteht aus verschiedenen, miteinander verbundenen CPS-Einheiten welche sich wiederum aus Kontrolleinheiten für Subsysteme, Sensoren, Kameras und Aktuatoren zusammensetzen. Einige der Komponenten kommunizieren nicht direkt mit Sensoren oder Aktuatoren, sondern stellen die Rechenleistung für die Ausführung der Anfragen anderer Systeme. Dem gegenüber stehen unmittelbar mit dem Netzwerk verbundene Sensoren und Aktuatoren. Der zentrale Aspekt dieser Architektur ist die *heterogene Netzwerkeinheit* (HE). Durch sie werden zeitliche und den Ausfall

betreffende Probleme der Komponenten abstrahiert.

Damit reduziert sich das Problem der Komplexität des Designs von CPS auf den Entwurf der HE, deren Verhalten und Latenz bezüglich notwendiger Kommunikationskriterien garantiert werden muss.

Im Falle der Störung oder des Ausfalls einer HE und des damit verbundenen Bearbeitungsschritts der Fertigungsstrecke, kommt die Produktion nicht vollständig zum Stillstand. Werkstücke oder Produkte können über alternative, autark agierenden Strecken umgeleitet werden [WVG08].

Mit Fokus auf Garantien zu zeitlichem Verhalten und Zuverlässigkeit der Komponenten eines CPS, beschreiben Wang et al. eine generische Netzwerkarchitektur für moderne Produktionssysteme. Jede zu steuernde Phase (*Stage* in Abbildung 4.1) des Fertigungsprozesses ist mit einem eingebetteten Kontrollsystem versehen und kann damit autonom agieren. Sensoren als Teil einer möglichen Rückkopplungsschleife sind vorgesehen, nicht aber Schwerpunkt des vorgestellten Konzepts. Für Anwendungen der Industrie 4.0 muss die HE zur virtuellen Repräsentation abstrahiert werden.

Die Anforderung zur ortsunabhängigen Steuerung (REQ1) kann mit dieser Netzwerkarchitektur durch eine Erweiterung erfüllt werden. Arbeitsteilig können die *Computing Units* und *Stages* (vgl. Abbildung 4.1) die Interpretation und Ausführung von NC-Programmen und automatisierten Komponenten übernehmen. Die ortsunabhängige Überwachung der Maschine (REQ2), deren virtuelle Repräsentation als *Stage* fungiert, ist durch die Einbindung Sensoren vorbereitet, nicht aber erfüllt. Um diesen Anforderungen vollends zu entsprechen muss ein lösungsorientiertes Konzept noch entwickelt werden. Es werden keinerlei Aussagen zu standardisierten Kommunikationsprotokollen oder Informationsmodellen getroffen, weshalb REQ3 nicht erfüllt wird. Da auf die Persistenz von Betriebs- und Prozessdaten der Stages nicht eingegangen wird, ein eingebettetes Kontrollsystem aber Bestandteil der jeweiligen Phase ist, wird der Anforderung der Lokalität nicht vollständig entsprochen (REQ4). Zusammenfassend kann das Konzept von Stages und Computing Units für die Lösung des Retrofitting-Problems übernommen werden.

4.2 Steuerung und Überwachung aus der Ferne

Auf einer Netzwerkarchitektur wie in Abschnitt 4.1, können konkrete Mechanismen für die Überwachung und Kontrolle von Anlagen aufgebaut werden. Nach den Anforderungen REQ1 und REQ2 (vgl. Abschnitte 3.1, 3.2) muss die Interaktion mit CPPS-Subsystemen und Menschen fernab vom Terminal gewährleistet werden. Wang et al. entwickelte 2008 eine offene Architektur für die Echtzeitüberwachung

und -kontrolle von im Netzwerk befindlichen CNC-Maschinen über eine grafische Schnittstelle mit 3D Repräsentation [Wan+04].

Ein Web-basierter Thin-Client des *Wise-ShopFloor* ermöglicht die Kontrolle und Überwachung der Maschinen über ein dreidimensionales Modell der Fertigungsstrecke. Das darunterliegende Framework basiert auf einer Client/Server-Architekturstil und verwendet seitens des Servers das MVC-Entwurfsmuster. Maschinen werden über das Fabriknetzwerk mit dem Server verbunden und sind somit vom Internet getrennt. Bei der Verwendung mehrerer Clients wird für das Routing ein Publish/Subscribe Mechanismus über HTTP-Streaming eingesetzt. Mit Hilfe dessen wird das Verhalten des auf Java 3D basierenden Visualisierungsmodells durch Sensorik an den Maschinen beeinflusst. In der von Wang et al. durchgeführten Case Study wurde unter Verwendung einer CNC-Fräsmaschine die Tauglichkeit des Konzepts verifiziert. Die Schnittstelle zwischen Server und Maschine wurde durch einen *Open Architecture Controller*¹ bereitgestellt. Für die Kontrolle der Fräse kann zwischen einem manuellen und einem automatischen Modus gewählt werden, wobei letzterer die direkte Übertragung von G-Code ermöglicht.

Das Internet ist ein zentraler Aspekt verteilter Produktion. Jedoch sind damit Sicherheitslücken fatal für interne Daten und vertrauliche Informationen der Organisation. Die gezielte Verbreitung dieser stellt ein erhöhtes wirtschaftliches Risiko dar und muss in besonderem Maße geschützt werden. Weiterhin sind Systemfehler auf Maschinenebene im Bezug auf Personen- und Materialschäden untragbar. Daher muss die reibungslose Kommunikation von Steuerungsbefehlen zu jeder Zeit gewährleistet sein.

Standards für die Kommunikation von Sensor- und Steuerungsinformationen sind notwendig um Effizienz und Integration der Systeme zu vereinfachen. So müssen globale Schnittstellen definiert und durch die Komponenten des Systems implementiert werden. Durch die Verwendung eines zuverlässigen NC-Befehlsinterpreters ist die verteilte Echtzeitsteuerung von CNC-Maschinen nach Wang et al. praktisch möglich. Jedoch setzt dieses System eine bestehende Anbindung an die Steuerungsebene voraus.

Die direkte Verbindung des Clients zu einer Maschine ist mit der verwendeten Technologie nicht möglich. Sowohl die Java Sicherheitsinfrastruktur, als auch die Überwindung von Firewalls stellen zukünftig zu lösende Probleme dar. Für künftige Maschinen ist daher das Einbetten eines dedizierten Web-Services in die Kontrolleinheit notwendig.

Ein wichtiger Aspekt des Konzepts von Wang et al. ist die technische Umsetzung auf der Java-Plattform. Mit dieser werden Sicherheitsinfrastrukturmerkmale wie byte-code-Verifikation und Rechteverwaltung direkt unterstützt. Die Indirektion des Kontrollflusses über den Server der Architektur zu den Maschinen verhilft zur Einhaltung.

¹ Steuerungskomponente, die Modifikationen über das API hinaus zulässt [Yon05]

Durch Verteilung von Steuerung und Überwachung der Maschine auf im Netzwerk befindliche Clients, sowie die browserbasierte Nutzungsschnittstelle, werden die Anforderungen REQ1 und REQ2 (vgl. Abschnitte 3.1, 3.2) vollständig erfüllt. Ein Standard wird mit der Kommunikation via HTTP verwendet, während Informationsprotokoll und -modell nicht näher erläutert wird. Damit ist REQ3 nur ansatzweise erfüllt. Auch mit der Konzeption einer Netzwerkarchitektur für CPS (vgl. Abschnitt 4.1) wurde die Notwendigkeit des Einbettens der Steuerung erkannt. Da aber Persistenz von Prozess- und Betriebsdaten von einem dedizierten Datenserver übernommen wird, ist das Lokalkriterium (REQ4) nur teilweise erfüllt. Grundsätzlich wurden Sicherheitsaspekte im Konzept beachtet und in den vertikalen Prototypen integriert. Die Verwendung von HTTP für die Kommunikation über Firewalls hinaus ermöglicht skalierenden, hierarchischen Zugriff.

4.3 Zusammenfassung

Die Gegenüberstellung von Anforderungen und bestehenden Forschungsarbeiten ist in Tabelle 4.1 zusammengefasst. Wobei ● die Erfüllung, ◐ die eingeschränkte oder teilweise Erfüllung und ○ die Nichterfüllung symbolisiert.

Tabelle 4.1: Anforderungen bzgl. bestehender Forschungsarbeiten

| | Überwachung | Steuerung | Standards | Lokalität |
|----------|-------------|-----------|-----------|-----------|
| [WVG08] | ◐ | ◐ | ○ | ◐ |
| [Wan+04] | ● | ● | ◐ | ◐ |

- Control software for industrial CNC machines [FCL05; FC07]
 - kein Standard bei Kommunikation
- Prototypischer OPC UA Server für Fernsteuerung [Aya+13]
- Unterschied RetroNet [Fra16b] (TODO)
 - keine zentrale Datenhaltung

Problem: Datenerfassung (=> Prozessüberwachung) bei Altmaschinen nicht vorhanden

Lösung: unabhängige minimalinvasive Sensorik

Purpose. Ziel von Deshpande et al. war eine nicht-invasive Methode der Echtzeitüberwachung von Energieverbrauch und weiteren Parametern bei Legacy-Maschinen.

Design/Methodology/Approach. Durch das Abgreifen des Stromverbrauchs über eine *Universal Power Cell* (UPC), dem Sensor an der Maschine, können Informationen via TCP und UDP an eine externe Komponente übergeben und ausgewertet werden. Die in Kilowatt eingehenden Verbrauchsdaten wurden durch an Bedingungen gekoppelte Schwellwerte in Status (an, aus, Leerlauf), Energieverbrauch, Werkzeugwechsel und Werkstückdurchsatz unterschieden. Für die Case Study und einen anschließenden Vergleich hatten Deshpande et al. auch moderne Maschinen mit der UPC ausgestattet. Verglichen wurde die zeitabhängige Auslastung von drei unterschiedlichen Modellen.

Findings. Mit einer Genauigkeit von 95% für den Status und 99% für Werkzeugwechsel und Durchsatz wurde das Konzept erfolgreich getestet.

Research Limitations/Implications. Das Konzept nutzt ausschließlich den Eingangsstrom der Maschine, wodurch Genauigkeit und Umfang der Daten begrenzt werden.

Practical Implications.

Für den Einsatz in einer realen Produktionsumgebung fehlen UI und API. Anforderungen an die Energieversorgung, sowie der Verbrauch von Teilsystemen der Produktionskette werden messbar. Die periodische Planung der Wartung kann aufgrund tatsächlicher Nutzung und Auslastung geschehen. Außerdem werden Prinzipien wie VSM echtzeitfähig und ermöglichen eine höhere Produktivität.

Originality/Value. Die minimal-invasive Methode ist unabhängig von Hard- Software und erlaubt die autonome Aggregation von Informationen unzugänglicher Altmaschinen.

Problem: ein Sensor => zu wenig Infos

Lösung: verschiedene Sensoren (auch für Fusion) einsetzen

- Purpose
 - Online Tool Condition Monitoring (TCM) bei CNC-Drehmaschinen mittels

- * Signal-Dekomposition
 - * statischer Datenanalysen
 - * neuronaler Netze
 - Untersuchung vieler versch. Netzwerkstrukturen => Optimierung für TCM
 - Schwerpunkt auf Analyse indirekter Messungen von Maschinensignalen
- Design/Methodology/Approach
 - Sensordaten durch Wavelet-Transformation in versch. Komponenten aufgeteilt, Quellen:
 - * dreidimensionales Accelerometer
 - * Schallemissionssensor
 - statistische Elimination maschinenspezifischer Effekte auf Sensordaten
=> Filter
 - statistische Signifikanzprüfung der Komponenten
 - Entwurf zweier Systeme basierend auf signifikanten Komponenten
 1. multipler linearer Regression
 2. künstlichen neuronalen Netzen
 - Fallstudie
- Findings
 - 12,6%ige Verbesserung der KNN-Vorhersagen durch Filter
 - 90% Genauigkeit bei multipler Regression
 - 97% Genauigkeit bei KNN
- Research Limitations/Implications
 - ausschließliche Betrachtung von TCM
- Practical Implications
 - Reduktion von Maschinenstillstand durch verbesserte Vorhersage des Werkzeugausfalls
 - Minimierung von Ausschuss
 - Kostenreduktion
- Originality/Value
 - Umfassende Untersuchung aktueller Forschung zu MDE (Tabelle 2.1) und Entscheidungssystemen

Problem: ausschließliche Betrachtung von TCM (Wartung) => Aber was wird wie bearbeitet?

Lösung: Werkstückidentifikation/-erkennung für Verknüpfung operativer Schritte

- Purpose
 -
- Design/Methodology/Approach
 -
- Findings
 -
- Research Limitations/Implications
 -
- Practical Implications
 -
- Originality/Value
 -

Ohne Carrier?

Data-Collection/Reasoning Stuff [Dow+16]

Anfallende Daten zu * Prozess * Technologie (CAD/CAM) * Werkzeug * Werkstück
* Peripherie (*loading door*)

Kommunikation der Informationen via OPC UA, MTConnect, MQTT ansprechen

- Umfangreiche Recherchen zu bestehender Forschung durch [Tet+10]
 - Werkzeugzustand
 - Spaneigenschaften
 - Prozesszustand
 - Werkstückbeschaffenheit

Problem: Wissen über operative Schritte und Werkstück verfügbar - Umsetzung der Schritte?

Lösung: Kontrolle der Maschine

Problem: Kontrolle der Maschine durch SPS mit CNC auf Terminal-PC inflexibel und teuer

Lösung: Portierbarkeit des CNC-Kernels auf andere Systeme

- Purpose
 - Untersuchung von CNC mit ARM-Computern
 - * Portierung eines CNC-Kernels auf Pi 2 durch Virtualisierung mit gemeinsamen Bibliotheken (Cross-Compile)
 - * Kontrolle mehrerer paralleler Kanäle mit Schrittmotoren und Spindel
 - * Kommunikation über EtherCAT Feldbus (Echtzeit)
- Design/Methodology/Approach
 - Soft-SPS, RT-Linux auf ARM-Computer
 - ARM/PC-Anbindung
 - * Spindel- und Motortreiber via EtherCAT
 - * Feldbus-Koppler via EtherCAT
 - * NC-Terminal via TCP
 - Fallstudie mit einer Dreh-/Fräsmaschine
 - Synch. der Kanäle durch High-Level-Funktionen (Load, Run, Start, Stop, Reset, Wait, Sleep)
- Findings
 - Ressourcen eines Einplatinencomputers sind ausreichend
 - parallele Portierung PC/ARM von NC-Software bei guter Kernel-Arch. mgl.
 - lediglich individuelle Konfiguration der Werkzeuge/Maschinen notwendig
 - CNC-Kernel braucht idle 20% ARM-CPU / 3% PC-CPU => steigt mit #Kanäle #Achsen
- Research Limitations/Implications
 - weitere Forschung auf Basis dessen

- * Verarbeitungspräzision/-Stabilität (Precision Engineering)
- * Adaptive Kontrolle
- * Diagnose/Prognose
- Practical Implications
 - EtherCAT Cycle Time¹ < 2ms, perspektivisch < 1ms => beeinfl. Anzahl paralleler Kontroll-Kanäle
 - künftig: Einplatinencomputer kontrollieren > 12 Achsen
- Originality/Value
 - Bewertung mit *Technology Readiness Level*² 6
- SOA [Mey10; FWW16]
- MAS, Holonic [Lei09; Fal16]
- Blackboard [MVK06; Pau+13]
- RAMI4.0 [AE15]

Problem: Betrachtungen des Retrofitting bzgl. Kontrolle und Überwachung einzelner Maschine, Integration?

Lösung: flexibel konfigurierbarer Verbund miteinander kommunizierender Maschinen => FFZ

- Purpose
 - Montage und Konfiguration von FFZ bzgl. Informationsebene vereinfachen
 - Design von Informationsmodell und zentralem Daten-Cache
 - Design einer Kommunikationsarchitektur für die Komponenten einer FFZ
 - Rekonfiguration einer FFZ zur Laufzeit (Losgrößenproblematik)
 - wissensch. Experimente mit rekonfigurierbaren Fertigungssystemen
- Design/Methodology/Approach
 - VDMA 34180 als Arch.-Ref. für FFZ betrachtet
 - * beschreibt Kommunikationsarchitekturen für *Robotized Manufacturing Cells* (RMC)

¹ „[...] die Zeit, die ein Teilnehmer (slave) warten muß, bis er wieder ‚dran‘ ist.“

² www.nasa.gov/directorates/heo/scan/engineering/technology/txt_accordion1.html

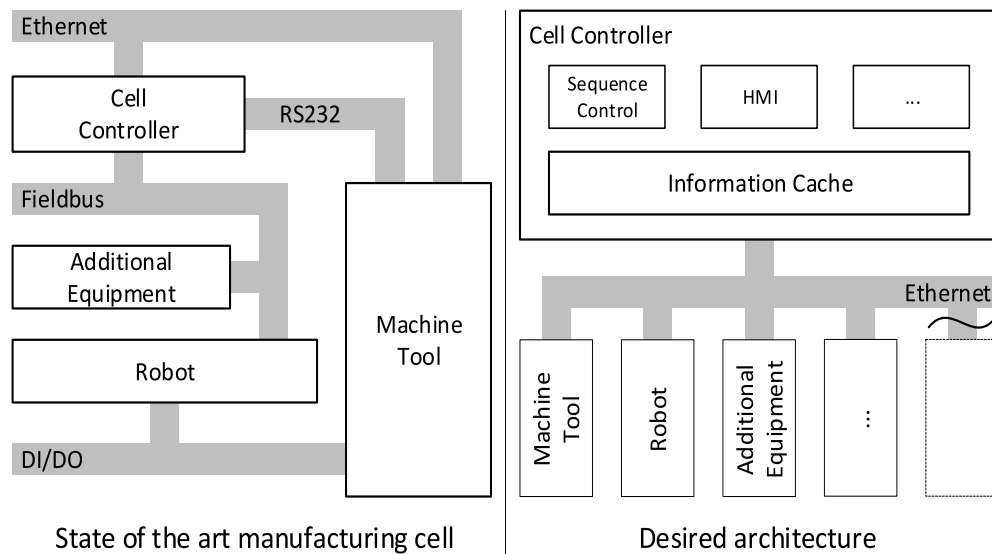


Abbildung 4.2: Vergleich der Architekturen von Fertigungszellen aus [Pau+13]

- * Schwerpunkt auf Werkstücklogistik mit Robotern
- Blackboard-Architektur für Kommunikation
 - * zentrale Datenbasis
 - * keine Kontrolleinheit (vgl. Blackboard in intelligenten Systemen)
 - * XML als Datenformat
 - Parameter für Kommunikation (IP, Port)
 - Maschinenstatus
 - Programm (Ident., Status)
 - Informationen zu angeschlossenen Geräten
 - * Konsistenz durch R/W-Locks
 - * Sequenzkontrollkomponente legt Handlungsweisungen auf Blackboard ab
 - * *Microsoft Visual Programming Language* (VPL) als Sequenzdefinitionsformat (Datenflussdiagramm)
 - * Maschinen und Sequenzkontrolle als aktive *Knowledge Sources* holen sich diese Instruktionen
- Master/Slave-Prinzip für Workflow und Job-Delegation (Master-Worker-Pattern)
- Ethernet-basierte Kommunikationsschnittstellen

- Adapter zu den Maschinen-Controllern (via Socket an Cell-Controller)
- Fallstudie durch FFZ mit zwei Fräsen und einem Roboter
- Findings
 - Reduktion von Komplexität der Topologie => geringerer Konfigurationsaufwand
 - VDMA 34180 nutzt DI/DO für Kommunikation => Konfiguration auf HW-Ebene => zu inflexibel für rekonfigurierbare RMCs
 - Fallstudie zeigt die Eignung der Architektur für FFZ
 - Blackboard geeignet bei notwendiger Austauschbarkeit der Komponenten
- Research Limitations/Implications
 - Schwerpunkt auf Blackboard-Informationscache
 - Eignung von VPL für Datenflussdefinition zu untersuchen
 - Performanz-Probleme mit XML als Blackboard-Backend (SQLite wird diskutiert)
 - * R/W-Locks
 - * Parsen der Datei
 - Socket-basierte Kommunikation Problematisch
 - * proprietäre Kommunikationsprotolle => Adapter immer notwendig
 - * OPC UA Server für beteiligte Komponenten
- Practical Implications
 - Hinzufügen/löschen von HW-/SW-Komponenten ohne Überarbeiten der Informationsarch./-modelle
 - *Plug & Produce* für FFZ wird möglich
 - Einbinden einer Komponente erfordert Adapter
- Originality/Value
 - Blackboard den intelligenten Systemen entlehnt
 - Definition der Sequenz für den Cell-Controller

=> auch [Dür+14]

Problem: Socket-basierte, proprietäre Maschinenkommunikation (Adapter) **Lösung:** OPC UA Server für Komponenten einer FFZ

- Purpose
 - OPC UA Server für CNC innerhalb einer FFZ
 - semantische Kommunikationsschnittstelle
- Design/Methodology/Approach
 - Implementierung eines OPC UA Informationsmodells für CNC-Maschinen
 - Methoden des Informationsmodells für Instruktionen verwendet
 - Kommunikation zw. Server und Maschine via propriät. Direct Numerical Control (DNC)
 - Case-Study an CNC-Drehmaschine & Industrieroboter
 - C++ UA Server SDK, .NET UA Client SDK (Unified Automation)
 - OPC UA Client mit Kontrollfunktionalität
- Findings
 - OPC UA ist eine geeignete Technologie für erweiterte Maschinenanbindung
 - Problem der Granularität des Modells (Ockhams Rasiermesser, KISS), 1-zu-1 Mapping ungeeignet
- Research Limitations/Implications
 - Definition eines Standard-Informationsmodells benötigt (vgl. MTConnect-OPC UA Comp.)
 - Anforderungen (horiz./vert. Integr.) für bestehende Standards (MES)
- Practical Implications
- Originality/Value
 - OPC UA Methoden als Steuerungsschnittstelle
 - dynamische (Runtime) Werkzeugrepräsentation im OPC UA Adressraum

Problem: OPC UA ist nicht alleiniger Standard für Informationsmodelle

Lösung: Informationsmodelle modellgetrieben entwickeln

- MOF?
- M0 == FFS
- FFS repräs. durch PIM, definiert in UML auf M2
=> plattformunabh. Modellierung auf M1

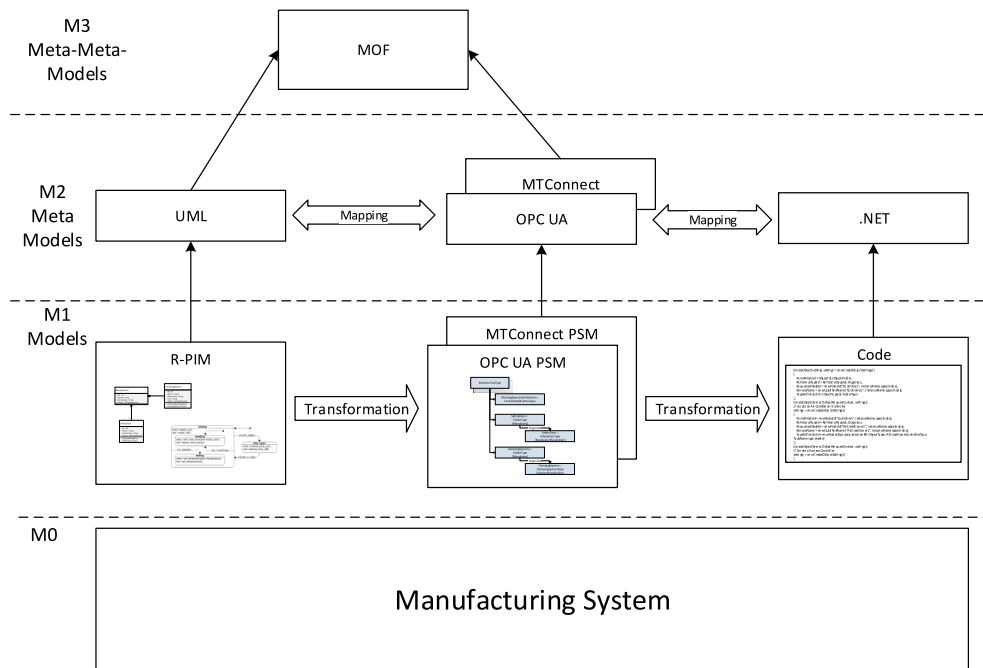


Abbildung 4.3: Transformationsprozess PIM zu PSM aus [PFK16]

- PIM => PSM: ein PIM-Elem. entspricht einem PSM-Elem. pro PSM (OPC UA, MTConnect, etc.)
- ein PSM kann dann in untersch. Sprachen (C#, C++, ...) übersetzt werden
- M3 Meta-Metamodell beschr. Sprache für spez. Metamodelle auf M2 => Interoperabilität zw. M2-Modellen

- Purpose
 - UML als einheitliche Modellierungssprache in der Softwareentwicklung, untersch. Modellierungssprachen in der Fertigungsindustrie
 - Model und Code müssen für jede Sprache separat entwickelt und gewartet werden => hoher manueller Aufwand
 - Konzept modellgetriebenen Designs (MDA) auf Fertigungsindustrie übertragen
 - * automatisiertes Ableiten plattformspezifischer Modelle von plattformunabhängigen

- Spezifikation eines Prozesses mit dem virtuelle Repräsentationen von I4.0-Komponenten generiert werden können
 - * Diagramme und Tools für Modellierung und Code-Generierung
 - * OPC UA Informationsmodelle von Systemanforderungen zu Code-generierung und Deployment
- RAMI4.0 Konformität
- Design/Methodology/Approach
 - erfassen statischer und dynamischer (Verhalten) Aspekte von Produktionssystemen
 - systematischer Ansatz für Modelltransformation zu OPC UA Informationsmodelle (hier: Zieltechnologie)
 - wichtigste plattformspez. Technologien untersucht => MDA Workflow definiert => Restriktionen für PIM definiert
=> Restricted Platform Independent Model (R-PIM)
 - * z.B.: Mehrfachvererbung in UML + nicht mgl. mit C#
 - * z.B.: notwendige Attribute der OPC UA Knoten (NodeId, NodeClass, BrowseName, DisplayName)
 - * untersch. zu MDA liegt in diesen Restriktionen
 - generisch bzgl. Zielmodell (OPC UA, MTConnect, etc.) <= Systemanalyse und -design mit der UML
 1. Komponenten- und Anwendungsfalldiagramme zur Beschreibung der Domäne
 - * manuell, da Wissen von Domänenexperten/Systemnutzern benötigt
 - * Komponentendiagramm: Systemgrenzen und Subsysteme identifizieren
 - * Anwendungsfalldiagramme: abstrakte Systemfunktionalität
 2. Domänenwissen => R-PIM
 - * Klassendiagramme
 - * Zustandsdiagramme
 3. R-PIM => Zieltechnologie
 - * Adressraummodell
 4. Zieltechnologie => Code-Gerüst
 5. Implementierung der Logik in Komponenten (API)
 6. Deployment eines ausführbaren Artefakts

- Findings
 - fehlende APIs/Tools => noch viel manueller Aufwand
- Research Limitations/Implications
 - Definition eines Meta-Metamodells für Restriktionen steht noch aus
 - manuelle R-PIM => PSM Transformation (exist. Konzept erlaubt nur statischer Aspekte des Address Space)
 - TU Wien arbeitet an Transf. dynamischer Aspekte => Guarded State Machines aus OPC UA Spec
 - vorerst nur OPC UA
 - Verbesserung der CIM-Definition
 - Ontologien als Wissensbasen statt Experten
- Practical Implications
 - Adressraummodellierung, Logikimplementierung zum Großteil noch händisch (fehlende Tools und APIs)
- Originality/Value
 - MDA-Prozess für generische Informationsmodelle durch UML

Die Integration bestehender Hardware in die intelligente Steuerung einer Fabrik ist Thema des *RetroNet*-Projekts. Das Fraunhofer IPK entwickelt mit Industriepartnern physische und logische Adapter für die Anbindung von bestehenden Anlagen an eine Steuerungsplattform. Maschinen-, Anlagen und Produktionsdaten werden zu diesem Zweck zentral erfasst und gespeichert. Weiterhin soll eine Middleware im Client-Server-Architekturstil Dienste und zugrunde liegende Teilsysteme miteinander verbinden und eine vermittelnde Rolle im Gesamtsystem einnehmen [Fra16b].

Forschung im Bereich Cloud-basierter Industriesteuerung wird in Zusammenarbeit von Fraunhofer, der TU Berlin und Industriepartnern betrieben. Im Projekt *pI-CASSO* werden die Auslagerung von Steuerungsdiensten in die Cloud und Möglichkeiten einer Verteilung und Modularisierung herkömmlicher Kontrollsysteme auf CPS-Komponenten untersucht [Fra16a].

Im Projekt *OPC4Factory* der TU Wien, wurden generische OPC UA Informationsmodelle entwickelt. Diese verbessern die Konnektivität von NC-Maschinen, Industrierobotern und anderen Komponenten innerhalb einer flexibel automatisierten

Fertigungszelle. Die Orchestrierung der Fertigungsoperationen, sowie die Konfiguration der Komponenten soll durch die Lösung der Schnittstellenproblematik vereinfacht werden¹.

[Sch+15; Sch+14; Vic+15]

- TODO Projekte zusammenfassen & gegeneinander abgrenzen
- Schnittstellenproblematik immer Teil des Problems
- Entwicklung von Adaptern meist Standardlösung
- Kosten => Remote Maintenance?

¹ www.ift.at/forschung/forschungsprojekte/opc4factory vom 04.10.2016 see [Aya+13; Pau+13; Pau14]

5 Konzeption

- Regelbasierte Rückkopplung => anlernen dessen ist Engineering, nicht Forschung (kann man immer noch machen => Ausblick)
- Sensorwert-Thresholds für Anomaly Detection
- Distributed NC!
- Laufzeitkonfiguration des Surrogate?
- Surrogate in bestehende Netzwerkinfrastruktur einbinden?
 1. direkte SG-Kommunikation (Wifi-Direct, BT, ...) zur Konfiguration der Netzwerkanbindung via Mobile device
 2. Auswahl des Access-Points
 3. Festlegen der OPC UA Discovery-Service Adresse
 4. SG mit Infrastr. verbunden + Registrierung bei Discovery-Service
 5. OPC UA Systemevent => neues Surrogate registriert
 6. Konfiguration des Adressraums via HTTP/WebApp
- Controlling nicht aus der Cloud @... sondern an der Maschine
 - CNC-Kernel auf dediziertem Controller
 - Kernel muss nicht portiert werden (vgl. [GM16])
 - Echtzeit kein Problem (OPC UA kann's eh nicht)
 - kein Feldbus, keine Koppler, kein DI/DO (vgl. [GM16])
 - Ethernet-basierte Kommunikation
- Konzept eines *Cell Controller* als Basis (vgl. [Aya+13; FWW16])
- kein Maschinenspez. NC-Terminal => verteiltes System => entfernte Mensch-Maschine-Schnittstelle (vgl. [GM16])
- Rekonfigurierbare FFZ [Pau+13; Dür+14]
- Bisher OPC UA Server als Adapter zu proprietären Maschinenprotokollen

- Server <-> Maschine => Server <-> Adapter <-> Maschine ?
 - Smoothieboard/Embedded kompakte(soft-)SPS als Adapter zu Maschine
- Wiederverwendung [Aya+13]
 - des Informationsmodells
 - des Flow-Charts für die Server-Logik (teilweise)
- Persistenzkonzept: Blackboard? [Pau+13]
- Kontrolle der Arbeitssequenz? (PROtEUS, BPMN/Activiti)
- Framework
 - Definition der Bindings von Extension Points in OPC UA Modell
- SPS siehe [WJN15] Bild 3
- Surrogate+OPCUA-Modell auf für SPS (Programm übertragen, starten, reset, etc.)

Unlike web services, OPC UA is currently integrated in a large number of PLCs on the market. The IEC standardization commission recommends OPC UA as a standard for the implementation of a smart factory [23]. For this reason, OPC UA is used as server standard for NGDs. However, OPC UA does not allow real-time transmission, which is why a real-time communication channel must still exist. – [HR15]

Nach Rücksprache

- OPC UA zur Metamodellierung bzgl. der Maschine/SPS
- But not hard real-time (not yet) => nicht geeignet für direkt Bewegungskontrolle [Pau14]
- OPC UA Modell synchron mit Realität => Laufzeitmodell
- Zu erwartendes Verhalten des physischen Systems über FB-Loop (MAPE-K?) kontrollierbar => Modellierung/Sprache der *Regeln*?
- Elemente eines Frameworks mit Schichtenarch. im Client/Server-Stil
- Microkernel-Ansatz (Plugins für OPC UA Typen, Sensoren und Aktuatoren)
- FB-Loop intern oder extern?
- „zentral“ erfassen durch RAMI4.0 Verwaltungsschale
- „zentral“ auswerten mit Cloud-Analytics (Big Data)

OPC4Factory:

OPC UA Server und ihre Informationsmodelle repräsentieren alle für die Automatisierungsaufgaben erforderlichen Komponenten der angeschlossenen Maschinen und Roboter (Ladetüren, Spannmittel, Werkzeuge, NC-Programme etc.) mit ihren Attributen, Ereignissen und Methoden. Die Kommunikation auf dieser Ebene **erfordert keine Echtzeitfähigkeit**, da Steuerungsaufgaben mit Echtzeitanforderungen ausschließlich innerhalb der Maschinen- bzw. Robotersteuerung abgewickelt werden.

- Statusreport: Referenzarchitekturmodell Industrie 4.0 (RAMI4.0) => **Abb. 9 I4.0-Komponente** [AE15]
- Kruchten 4+1?
- arc42?
- Service-/Protokolllayer (OPC UA)
- CPS-Layer, Hardwareanbindung
- Einbetten eines dedizierten Web-Services [Wan+04]
- Stage braucht eingebettetes Kontrollsystem [WVG08]

[Pau+13] Komponentengranularität bis zur Achse (intelligente Maschinenteile)

5.1 Zusammenfassung

6 Implementation

- Smoothieboard als Maschinen-Adapter
 - Nachteil: Beobachten des Prozessfortschritts langsam (*progress*) => kann nicht in online-FB einbezogen werden
- open62541 oder ähnliche OPC UA Stack-Implementierungen für Server auf Pi

6.1 Zusammenfassung

7 Evaluation

These/Behauptung?

- Steigerung des Automatisierungsgrads durch Feedback Loop
- physische Anwesenheit des Werkers technisch überwinden (Remote-Control/-Programming)
 - „Echtzeitanalyse“ durch Werker auch entfernt mgl.
- Laufzeitmodell für *online*-Monitoring

Umsetzung?

- Proof of concept
- Case-Study mgl.?
- HIL-Simulation?

Blocking Factors/mögliche Kritik?

- Leistung von embedded computing devices => siehe [GM16]
- Energieverbrauch

7.1 Zusammenfassung

8 Zusammenfassung

8.1 Schlussfolgerung

8.2 Ausblick

- Steuerungsalternative *Programs*
- MDSD mit [PFK16]
- CNC ersetzen durch STEP-NC? [Suh+03; XN06; Xu06; XLY06]
- Prozessmodell für Abstrakte Leitebene
- AutomationML und OPC UA [OPC14]
- Wo läuft die Logik für orchestrierende Steuerung? (OPC UA Clients)
- Surrogate als reaktiver Agent => Einbindung in MAS denkbar
- Echtzeitfähigkeit des Konzepts überprüfen
- Möglichkeiten des Nutzens der Daten
 - Welcher G-Code Befehl korreliert auf welche Weise mit welchen gemessenen Werten?
 - Automatische Erkennung von Gut-/Schlechtteilen

A Anhang

Abbildungen

| | | |
|-----|---|----|
| 2.1 | Automatisierte Fertigung aus [Lin15] | 8 |
| 2.2 | Beispiel einer klassischen Automatisierungspyramide | 9 |
| 2.3 | Repräsentative Automationsstruktur nach [HR15] | 10 |
| 2.4 | Beispielkonstruktion und G-Code für eine Drehbank | 12 |
| 2.5 | Grundlegender Hardwareaufbau einer SPS nach [HLG15] | 14 |
| 2.6 | Spezifikationen von OPC UA | 16 |
| 2.7 | Der historische Weg zu CPSoS | 18 |
| 2.8 | Auflösung der Automatisierungspyramide aus [Ver13] | 19 |
| 4.1 | CPS-Architektur nach Wang et al. [WVG08] | 25 |
| 4.2 | Vergleich der Architekturen von Fertigungszellen aus [Pau+13] | 34 |
| 4.3 | Transformationsprozess PIM zu PSM aus [PFK16] | 37 |

Programmcode

Tabellen

| | | |
|-----|--|----|
| 4.1 | Anforderungen bzgl. bestehender Forschungsarbeiten | 28 |
|-----|--|----|

Literatur

- [AE15] Peter Adolphs und Ulrich Epple. *Statusreport: Referenzarchitekturmodell Industrie 4.0 (RAMI4.0)*. Technischer Bericht April. 2015 (siehe Seiten 33, 43).
- [Aya+13] Iman Ayatollahi, Burkhard Kittl, Florian Pauker u. a. „Prototype OPC UA Server for Remote Control of Machine Tools“. In: *International Conference on Innovative Technologies*. Band 1009. 12. 2013, Seiten 73–76 (siehe Seiten 16, 23, 28, 40–42).
- [Ber15] Simon Bergweiler. „Intelligent Manufacturing based on Self-Monitoring Cyber-Physical Systems“. In: *International Conference on Mobile Ubiquitous Computing, Systems, Services and Technologies c* (2015), Seiten 108–113. DOI: 10.13140/RG.2.1.2762.9281 (siehe Seiten 4, 19).
- [Bon+12] Flavio Bonomi, Rodolfo Milito, Jiang Zhu u. a. „Fog Computing and Its Role in the Internet of Things“. In: *Proceedings of the first edition of the MCC workshop on Mobile cloud computing*. 2012, Seiten 13–16. DOI: 10.1145/2342509.2342513 (siehe Seite 23).
- [Bor+14] Theodor Borangiu, Silviu Raileanu, Damien Trentesaux u. a. „Distributed manufacturing control with extended CNP interaction of intelligent products“. In: *Journal of Intelligent Manufacturing* 25.5 (2014), Seiten 1065–1075. DOI: 10.1007/s10845-013-0740-3 (siehe Seite 20).
- [Dow+16] Jonathan Downey, Denis O’Sullivan, Mirosław Nejmen u. a. „Real Time Monitoring of the CNC Process in a Production Environment- the Data Collection & Analysis Phase“. In: *Procedia CIRP* 41 (2016), Seiten 920–926. DOI: 10.1016/j.procir.2015.12.008 (siehe Seite 31).
- [DP11] Amit Deshpande und Ron Pieper. „Legacy Machine Monitoring Using Power Signal Analysis“. In: *ASME 2011 International Manufacturing Science and Engineering Conference, Volume 2*. ASME, 2011, Seiten 207–214. DOI: 10.1115/MSEC2011-50019 (siehe Seiten 2, 21).
- [Dür+14] Lars Dürkop, Henning Trsek, Jens Otto u. a. „A field level architecture for reconfigurable real-time automation systems“. In: *IEEE International Workshop on Factory Communication Systems - Proceedings, WFCS* (2014). DOI: 10.1109/WFCS.2014.6837601 (siehe Seiten 35, 41).

- [Fal16] Solmaz Mansour Fallah. „Multi Agent based Control Architectures“. In: *26th DAAAM International Symposium on intelligent Manufacturing and Automation* (2016), Seiten 1166–1170. DOI: 10.2507/26th.daaam.proceedings.164 (siehe Seite 33).
- [FC07] Antnio Ferrolho und Manuel Crisostomo. „Intelligent Control and Integration Software for Flexible Manufacturing Cells“. In: *IEEE Transactions on Industrial Informatics* 3.1 (Feb. 2007), Seiten 3–11. DOI: 10.1109/TII.2006.890529 (siehe Seiten 2, 22, 28).
- [FCL05] António Ferrolho, Manuel Crisostomo und Mario Lima. „Intelligent control software for industrial CNC machines“. In: *2005 IEEE International Conference on Intelligent Engineering Systems, 2005. INES '05*. IEEE, 2005, Seiten 267–272. DOI: 10.1109/INES.2005.1555171 (siehe Seite 28).
- [Fra16a] Fraunhofer IPK. *Industrie 4.0 - Cloudbasierte Steuerungen*. 2016. URL: https://www.ipk.fraunhofer.de/fileadmin/user_upload/IPK/publikationen/themenblaetter/aut_pICASSO_web.pdf (siehe Seite 39).
- [Fra16b] Fraunhofer IPK. „RetroNet - Praxisnahe Brücke in die Industrie 4.0“. In: *FUTUR* (2016), Seite 8. URL: https://issuu.com/claudiaengel/docs/futur_1_2016 (siehe Seiten 1, 28, 39).
- [FWW16] Solmaz Mansour Fallah, Sabine Wolny und Manuel Wimmer. „Towards model-integrated service-oriented manufacturing execution system“. In: *2016 1st International Workshop on Cyber-Physical Production Systems (CPPS)*. IEEE, Apr. 2016, Seiten 1–5. DOI: 10.1109/CPPS.2016.7483917 (siehe Seiten 33, 41).
- [Gau+14] Jürgen Gausemeier, Roman Dumitrescu, Jürgen Jasperneite u. a. *Auf dem Weg zu Industrie 4.0: Lösungen aus dem Spitzencluster it's OWL*. Technischer Bericht. Paderborn: it's OWL Clustermanagement GmbH, 2014. URL: http://www.its-owl.de/fileadmin/PDF/Industrie_4.0/Auf_dem_Weg_zu_Industrie_4.0_-_Loesungen_aus_dem_Spitzencluster_its_OWL_RGB.pdf (siehe Seite 1).
- [Gee11] Guido L. Geerts. „A design science research methodology and its application to accounting information systems research“. In: *International Journal of Accounting Information Systems* 12.2 (Juni 2011), Seiten 142–151. DOI: 10.1016/j.accinf.2011.02.004 (siehe Seite 5).
- [GM16] Sergej N. Grigoriev und Georgi M. Martinov. „An ARM-based Multi-channel CNC Solution for Multi-tasking Turning and Milling Machines“. In: *Procedia CIRP* 46 (2016), Seiten 525–528. DOI: 10.1016/j.procir.2016.04.036 (siehe Seiten 4, 41, 47).

- [GML00] a. Gunasekaran, H. B. Marri und B. Lee. „Design and Implementation of Computer Integrated Manufacturing in Small and Medium-Sized Enterprises: A Case Study“. In: *International Journal of Advanced Manufacturing Technology* 16.1 (2000), Seiten 46–54. DOI: 10.1007/PL00013131 (siehe Seite 2).
- [Gro08] Mikell P. Groover. *Automation, production systems, and computer-integrated manufacturing*. Prentice Hall, 2008, Seite 815. ISBN: 0132393212 (siehe Seite 3).
- [Hir00] Andreas Hirsch. *Werkzeugmaschinen Grundlagen: Lehr- und Übungsbuch*. Vieweg, 2000. ISBN: 978-3528049508 (siehe Seite 11).
- [HLG15] Berthold Heinrich, Petra Linke und Michael Glöckler. *Grundlagen Automatisierung*. Springer, 2015. DOI: 10.1007/978-3-658-05961-3 (siehe Seiten 13, 14).
- [Hop14] Stefan Hoppe. „Standardisierte horizontale und vertikale Kommunikation: Status und Ausblick“. In: *Industrie 4.0 in Produktion, Automatisierung und Logistik*. Wiesbaden: Springer Fachmedien Wiesbaden, 2014, Seiten 325–341. DOI: 10.1007/978-3-658-04682-8_16 (siehe Seiten 2, 17).
- [HR15] Veit Hammerstingl und Gunther Reinhart. „Unified Plug&Produce architecture for automatic integration of field devices in industrial environments“. In: *Proceedings of the IEEE International Conference on Industrial Technology* 2015-June. June (2015), Seiten 1956–1963. DOI: 10.1109/ICIT.2015.7125383 (siehe Seiten 10, 11, 42).
- [KWH13] Henning Kargermann, Wolfgang Wahlster und Johannes Helbig. *Umsetzungsempfehlungen für das Zukunftsprojekt Industrie 4.0*. Technischer Bericht April. 2013. URL: https://www.bmbf.de/files/Umsetzungsempfehlungen_Industrie4_0.pdf (siehe Seite 15).
- [LBK15] Jay Lee, Behrad Bagheri und Hung An Kao. „A Cyber-Physical Systems architecture for Industry 4.0-based manufacturing systems“. In: *Manufacturing Letters* 3. September 2016 (2015), Seiten 18–23. DOI: 10.1016/j.mfglet.2014.12.001 (siehe Seite 20).
- [Lee06] Soo-Yen Lee. „In-process tool condition monitoring systems in CNC turning operations“. Dissertation. Iowa State University, 2006 (siehe Seite 18).
- [Lee08] Edward A. Lee. „Cyber Physical Systems: Design Challenges“. In: *2008 11th IEEE International Symposium on Object and Component-Oriented Real-Time Distributed Computing (ISORC)* (2008), Seiten 363–369. DOI: 10.1109/ISORC.2008.25 (siehe Seite 19).
- [Lee99] Tina Y. Lee. „Information modeling: From design to implementation“. In: *Proceedings of the second world manufacturing congress*. 1999, Seiten 315–321 (siehe Seite 16).

- [Lei09] Paulo Leitão. „Agent-based distributed manufacturing control: A state-of-the-art survey“. In: *Engineering Applications of Artificial Intelligence* 22.7 (2009), Seiten 979–991. DOI: 10.1016/j.engappai.2008.09.005 (siehe Seite 33).
- [LHL04] Steven Y. Liang, Rogelio L. Hecker und Robert G. Landers. „Machining Process Monitoring and Control: The State-of-the-Art“. In: *Journal of Manufacturing Science and Engineering* 126.2 (2004), Seite 297. DOI: 10.1115/1.1707035 (siehe Seite 1).
- [Lin15] Petra Linke. „Grundlagen zur Automatisierung“. In: *Grundlagen Automatisierung*. Wiesbaden: Springer Fachmedien Wiesbaden, 2015, Seiten 1–28. DOI: 10.1007/978-3-658-05961-3_1 (siehe Seiten 7–9).
- [LS01] Gunter Lay und Elna Schirrmeister. *Sackgasse Hochautomatisierung? Praxis des Abbaus von Overengineering in der Produktion*. Technischer Bericht. Karlsruhe: Fraunhofer-Institut für System- und Innovationsforschung, 2001. URL: <http://hdl.handle.net/10419/29534> (siehe Seite 2).
- [Mah03] Nitaigour Premchand Mahalik, Herausgeber. *Fieldbus Technology*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003. DOI: 10.1007/978-3-662-07219-6 (siehe Seite 10).
- [Mey10] Heiko Meyer. „Software Architecture of Manufacturing Execution Systems“. In: *Systemics, Cybernetics and Informatics* 8.2 (2010), Seiten 62–66 (siehe Seite 33).
- [Mil14] Joachim Milberg. *Trends in der fabrik*. Technischer Bericht. Institut für Produktion und Logistik, 2014. URL: http://www.ifpconsulting.de/media/pdf/ifp_ku_trends_2014sm_abstract.pdf (siehe Seite 1).
- [MVK06] L. Monostori, J. Váncza und S.R.T. Kumara. „Agent-Based Systems for Manufacturing“. In: *CIRP Annals - Manufacturing Technology* 55.2 (2006), Seiten 697–720. DOI: 10.1016/j.cirp.2006.10.004 (siehe Seite 33).
- [OPC14] OPC Foundation. *OPC Unified Architecture - Wegbereiter der 4. industriellen (R)Evolution*. Technischer Bericht. 2014. URL: https://opcfoundation.org/wp-content/uploads/2014/03/OPC_UA_I_4.0_Wegbereiter_DE_v2.pdf (siehe Seiten 15–17, i).
- [Pau+13] Florian Pauker, T Weiler, I Ayatollahi u.a. „Information Architecture for Reconfigurable production systems“. In: *DAAAM International Scientific Book 2013* January (2013), Seiten 873–886. DOI: 10.2507/daaam.scibook.2013.53 (siehe Seiten 11, 33, 34, 40–43).
- [Pau14] Florian Pauker. „OPC UA for machine tending industrial robots - Prototypic development of an OPC UA server for ABB industrial robots“. In: *October* (2014). DOI: 10.15224/978-1-63248-031-6-155 (siehe Seiten 40, 42).

- [PFK16] Florian Pauker, Thomas Frühwirth und Burkhard Kittl. „A systematic approach to OPC UA information model design“. In: *49th CIRP Conference on Manufacturing Systems*. May. 2016 (siehe Seiten 37, i).
- [Sch+14] Jan Schlechtendahl, Felix Kretschmer, Armin Lechler u. a. „Communication Mechanisms for Cloud based Machine Controls“. In: *Procedia CIRP*. Band 17. Elsevier, 2014, Seiten 830–834. DOI: 10.1016/j.procir.2014.01.074 (siehe Seite 40).
- [Sch+15] Jan Schlechtendahl, Felix Kretschmer, Zhiqian Sang u. a. „Extended study of network capability for cloud based control systems“. In: *Robotics and Computer-Integrated Manufacturing*. May 2014. 2015. DOI: 10.1016/j.rcim.2015.10.012 (siehe Seite 40).
- [SG16] David Siepmann und Norbert Graef. „Industrie 4.0 – Grundlagen und Gesamtzusammenhang“. In: *Einführung und Umsetzung von Industrie 4.0*. Berlin, Heidelberg: Springer Berlin Heidelberg, 2016, Seiten 17–82. DOI: 10.1007/978-3-662-48505-7_2 (siehe Seiten 1, 22).
- [Smi08] Peter Smid. *CNC programming Handbook*. Industrial Press, 2008. ISBN: 9780831133474 (siehe Seite 13).
- [SSG13] Jochen Schlick, Peter Stephan und Thomas Greiner. „Kontext, Dienste und Cloud Computing - Eigenschaften und Anwendungen cyber-physischer Systeme“. In: *atp edition* 55.4 (2013), Seiten 32–41 (siehe Seite 20).
- [Suh+03] S.H. Suh, B.E. Lee, D.H. Chung u. a. „Architecture and implementation of a shop-floor programming system for STEP-compliant CNC“. In: *Computer-Aided Design* 35.12 (2003), Seiten 1069–1083. DOI: 10.1016/S0010-4485(02)00179-3 (siehe Seite i).
- [Tet+10] R. Teti, K. Jemielniak, G. O'Donnell u. a. „Advanced monitoring of machining operations“. In: *CIRP Annals - Manufacturing Technology* 59.2 (2010), Seiten 717–739. DOI: 10.1016/j.cirp.2010.05.010 (siehe Seite 31).
- [Ver13] Verein Deutscher Ingenieure e.V. *Cyber-Physical Systems: Chancen und Nutzen aus Sicht der Automation*. Technischer Bericht. VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik, 2013. URL: <https://www.vdi.de/technik/fachthemen/mess-und-automatisierungstechnik/fachbereiche/thesen-und-handlungsfelder/> (siehe Seiten 19, 20).
- [Vic+15] Axel Vick, Christian Horn, Martin Rudorfer u. a. „Control of robots and machine tools with an extended factory cloud“. In: *2015 IEEE World Conference on Factory Communication Systems (WFCS)*. IEEE, Mai 2015, Seiten 1–4. DOI: 10.1109/WFCS.2015.7160575 (siehe Seite 40).

- [Wan+04] Lihui Wang, Peter Orban, Andrew Cunningham u. a. „Remote real-time CNC machining for web-based manufacturing“. In: *Robotics and Computer-Integrated Manufacturing* 20.6 (2004), Seiten 563–571. DOI: 10.1016/j.rcim.2004.07.007 (siehe Seiten 1, 2, 21, 23, 27, 28, 43).
- [WJN15] S. Windmann, F. Jungbluth und O. Niggemann. „Ansätze zur Erhöhung der Flexibilität und Vernetzbarkeit industrieller Steuerungen“. In: *Branchentreff der Mess- und Automatisierungstechnik (Automation)*. 2015 (siehe Seiten 3, 42).
- [WVG08] Yunbo Wang, Mehmet C. Vuran und Steve Goddard. „Cyber-physical systems in industrial process control“. In: *ACM SIGBED Review* 5.1 (2008), Seiten 1–2. DOI: 10.1145/1366283.1366295 (siehe Seiten 25, 26, 28, 43).
- [XLY06] X.W. Xu, Lihui Wang und Yiming Yiming Rong. „STEP-NC and function blocks for interoperable manufacturing“. In: *IEEE Transactions on Automation Science and Engineering* 3.3 (Juli 2006), Seiten 297–308. DOI: 10.1109/TASE.2005.862147 (siehe Seiten 13, i).
- [XN06] X.W. Xu und S.T. Newman. „Making CNC machine tools more open, interoperable and intelligent - a review of the technologies“. In: *Computers in Industry* 57.2 (2006), Seiten 141–152. DOI: 10.1016/j.compind.2005.06.002 (siehe Seite i).
- [Xu06] X.W. Xu. „Realization of STEP-NC enabled machining“. In: *Robotics and Computer-Integrated Manufacturing* 22.2 (2006), Seiten 144–153. DOI: 10.1016/j.rcim.2005.02.009 (siehe Seite i).
- [Yon05] Chi Yonglin. „An evaluation space for open architecture controllers“. In: *The International Journal of Advanced Manufacturing Technology* 26.4 (Aug. 2005), Seiten 351–358. DOI: 10.1007/s00170-004-2111-x (siehe Seite 27).