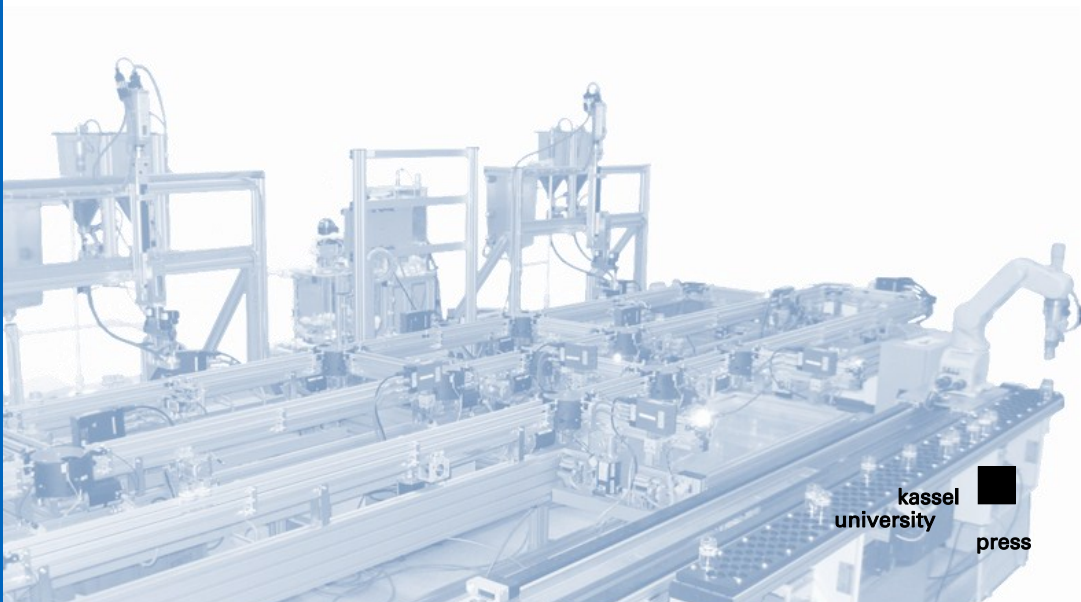




B. Vogel-Heuser (Hrsg.)

# Erhöhte Verfügbarkeit und transparente Produktion



# **Embedded Systems**

I Tagungen und Berichte 2

Herausgegeben von Prof. Dr.-Ing. Birgit Vogel-Heuser,  
Technische Universität München



---

# **Erhöhte Verfügbarkeit und transparente Produktion**

---

Herausgegeben von  
Prof. Dr.-Ing. Birgit Vogel-Heuser

Bibliographic information published by Deutsche Nationalbibliothek  
The Deutsche Nationalbibliothek lists this publication in the Deutsche  
Nationalbibliografie; detailed bibliographic data is available in the Internet at  
<http://dnb.d-nb.de>.

ISBN print: 978-3-86219-178-9

ISBN online: 978-3-86219-179-6

URN: <http://nbn-resolving.de/urn:nbn:de:0002-31799>

2011, kassel university press GmbH, Kassel  
[www.uni-kassel.de/upress](http://www.uni-kassel.de/upress)

Printed in Germany

# Inhaltsverzeichnis

<b>1</b>	<b>Cyber Physical Systems – Herausforderung für die Produktion? .....</b>	<b>1</b>
	<i>Birgit Vogel-Heuser</i>	
1.1	Herausforderungen für die Produktion .....	1
1.2	Referenzen .....	5
<b>2</b>	<b>Agenda CPS- Szenario smart factory .....</b>	<b>6</b>
	<i>Birgit Vogel-Heuser, Gülden Bayrak, Ursula Frank</i>	
2.1	Einführung und Begriffsklärung .....	6
2.2	Szenario: kundenspezifische Produktion einer Küche.....	8
2.2.1	Teilszenario 1: Auftragsänderung.....	9
2.2.2	Teilszenario 2: Qualitätsverlust während der Produktion.....	10
2.2.3	Teilszenario 3: Ausfall einer Komponente .....	11
2.3	Eigenschaften von CPS Produktionseinheiten.....	12
2.4	Herausforderungen für das Engineering und die Realisierung von ProCPS.....	13
2.4.1	Horizontale und vertikale Vernetzung .....	14
2.4.2	Integrative disziplinübergreifende Entwicklung von Produkt und Produktionssystem .....	16
2.4.3	Modularisierung.....	16
2.4.4	Modellgetriebene Entwicklung (Model Driven Design) .....	16
2.4.5	Durchgängige Engineering Werkzeugkette .....	17
2.4.6	Vorgehensweisen, Methoden, Beschreibungsmittel und Softwarewerkzeuge.....	18
2.5	Zusammenfassung/Ausblick.....	20
2.6	Referenzen .....	20

**3      Potentiale und Herausforderungen für die Steuerungs-  
programmierung im Maschinen- und Anlagenbau von  
morgen ..... 22**

*Ursula Frank, Josef Papenfort, Martin Obermeier,  
Birgit Vogel-Heuser*

3.1	Einleitung.....	22
3.2	Potentiale .....	23
3.2.2	Anwendungsszenarien .....	25
3.3	Expertenevaluation und Interpretation.....	27
3.4	Zusammenfassung und Ausblick .....	30
3.5	Danksagung .....	31
3.6	Referenzen .....	31

**4      Verfügbarkeitssteigerung von Produktionsanlagen mit  
Softwareagenten – Entwicklung eines durchgängigen  
Werkzeugs ..... 33**

*Daniel Schütz, Birgit Vogel-Heuser*

4.1	Motivation und Einleitung.....	33
4.2	Ausgangssituation und Zielstellung.....	35
4.3	Domänenspezifische Sichten eines Agentensystems.....	36
4.4	Vorgehen bei der Entwicklung von Agentensystemen.....	40
4.5	Anwendungsbeispiel – Hybrides Prozessmodell.....	42
4.5.2	Komposition der Steuerungsagenten .....	43
4.5.3	Beschreibung des Prozesses und der Anforderungen .....	44
4.5.4	Systemeinschränkungen und Softsensoren .....	46
4.6	Zusammenfassung und Ausblick .....	49
4.7	Danksagung .....	51
4.8	Referenzen .....	51

**5 Verteilt oder Zentral? Unterstützung für den Entwurf der Automatisierung durch das Projekt FAVA..... 54**

*Alexander Fay, Birgit Vogel-Heuser, Christian Diedrich*

5.1 Warum dieses Projekt? ..... 54  
 5.2 Die Aufgabenstellung ..... 55  
 5.3 Das Konzept von FAVA..... 56  
 5.4 Zentrale und verteilte Struktur von Automatisierungssystemen.... 58  
 5.5 Ausblick ..... 61  
 5.6 Referenzen ..... 61

**6 Überwachung der Lieferkette zur flexiblen Fertigungssteuerung auf MES-Ebene..... 62**

*Raffaello Lepratti, Christoph Legat, Georg Heinecke, Steffen Lamparter*

6.1 Einleitung ..... 62  
 6.2 Standardisierte Systemarchitektur ..... 63  
 6.2.1 Überbetriebliche Informationsplattform ..... 64  
 6.2.2 Produktionsassistenzsystem (PAS)..... 66  
 6.3 Identifikation kritischer Ereignisse/Störungen ..... 67  
 6.4 Kompensation von Störungen..... 69  
 6.5 Prototypische Umsetzung ..... 70  
 6.6 Zusammenfassung/Ausblick ..... 71  
 6.7 Referenzen ..... 72

**7 MES Projekte effizient spezifizieren - Anforderungen an ein einheitliches Beschreibungsmittel für den Praxisalltag ..... 74**

*Steffen Himstedt, Maria Witsch*

7.1 Manufacturing Execution Systems als Schnittstelle zwischen IT und Produktion..... 74  
 7.1.1 State of the art der MES IT-Landschaft..... 75  
 7.1.2 State-of-the-Art in MES-Projekten und Business Decisions bei MES-Anwendern ..... 75



7.2	Herausforderungen in MES-Projekten für MES Anbieter.....	76
7.3	Anforderungen an eine Beschreibungssprache für den Praxisalltag.....	78
7.4	Zukünftige Herausforderungen bei der Spezifikation von MES ...	79
<b>8</b>	<b>MES-Modeling Language – Eine Beschreibungssprache für die interdisziplinäre Anforderungserhebung und Spezifikation von MES .....</b>	<b>80</b>
	<i>Maria Witsch</i>	
8.1	Einleitung.....	80
8.2	Drei Sichten für die Spezifikation von MES .....	81
8.3	Modell des technischen Systems .....	83
8.3.1	Aufgabe des Modells in der MES-ML .....	83
8.3.2	Modellierungselemente des Modells des technischen Systems.....	83
8.4	Modell des MES .....	85
8.4.1	Aufgabe des Modells in der MES-ML .....	85
8.5	Beschreibung der Elemente des MES Modells.....	85
8.5.2	Ordnungsobjekte .....	86
8.5.3	Verbindungsobjekte .....	87
8.5.4	Verhaltensobjekte .....	88
8.5.5	Datenobjekte .....	91
8.5.6	Artefakte .....	93
8.6	Produktionsprozessmodell .....	93
8.6.1	Aufgabe des Modells in der MES-ML .....	93
8.6.2	Modellierungselemente des Produktionsprozessmodells .....	94
8.7	Strukturbaum .....	94
8.8	Verknüpfungen zwischen den Modellen .....	95
8.8.2	Datenaustausch .....	96
8.8.3	Ausführung .....	97
8.8.4	Entsprechung - Einbindung von Stellvertretern in das Produktionsprozess- oder MES-Modell.....	97
8.9	SpeziMES Editor .....	98
8.10	Evaluation der Beschreibungssprache .....	99
8.11	Danksagung .....	100
8.12	Referenzen .....	100

<b>9</b>	<b>Spezifikation eines generischen MES-Clients mit der MES-ML .....</b>	<b>102</b>
	<i>Jürgen Dendorfer, Maria Witsch</i>	
9.1	Einleitung .....	102
9.2	Vorstellung des Evaluationsprojekts .....	103
9.3	Modellierung des generischen Clients und Abgleich des Produktionsprozesses .....	104
9.3.1	Modellierungsworkshop mit der MES-ML .....	104
9.3.2	Eigenständige Überarbeitung der Modelle .....	106
9.3.3	Analyse der ausgewählten Funktionalitäten anhand der Pilotlinie .....	107
9.4	Ergebnisse der Evaluation .....	107

## Abbildungsverzeichnis

Abb. 1.1 – Herausforderungen: Horizontale und Vertikale Integration, sowie Engineering-Lifecycle .....	2
Abb. 1.2 – Varianten des Moduls Band in einem Modulbaukasten .....	4
Abb. 2.1 – CPS Factory und die beteiligten Stakeholder.....	12
Abb. 3.1 – Eigenschaften von Agenten.....	24
Abb. 3.2 – Anforderungen an die Unterstützung bei Optimierungen .....	28
Abb. 3.3 – Anforderungen an Features des Entwicklungswerkzeugs.....	29
Abb. 4.1 – Struktur und Sichten eines Agentensystems .....	37
Abb. 4.2 – Vorgehensmodell zur Systementwicklung.....	40
Abb. 4.3 – R&I-Fließbild aus dem Hybriden Prozessmodell .....	42
Abb. 4.4 – Prozessanforderung gebunden an eine Aktivität.....	45
Abb. 4.5 – Parameterdiagramm der Workstation .....	46
Abb. 4.6 – Implementierung von Softsensoren für die Workstation .....	48
Abb. 5.1 – Strukturen von Automatisierungssystemen nach IEC 62390.....	59
Abb. 6.1 – Überbetriebliche Systemarchitektur .....	64
Abb. 6.2 – Zuordnung von Ereignisklassen im Bezug auf Durchlaufzeit ...	68
Abb. 6.3 – Architektur des Hardware-in-the-Loop Demonstrators .....	71
Abb. 8.1 – Grafische Modelle der MES-ML .....	82
Abb. 8.2 – Gliederung des Baumdiagramms zum technischen System.....	84
Abb. 8.3 – Notation des MES-Modells.....	85
Abb. 8.4 – Modellierung einer Schleifenaktivität mit exklusivem Gateway (links) und einer sequentiellen Mehrfachaktivität (rechts).	89
Abb. 8.5 – Zwei Prozesse greifen auf ein und denselben Datenspeicher zu	91
Abb. 8.6 – Daten- und Nachrichtenfluss mit über Assoziationen angehefteten Datenobjekten .....	92
Abb. 8.7 – Notation des Produktionsprozessmodells .....	94
Abb. 8.8 – Einbindung einer Stellvertreter-Produktionsprozessaktivität in das MES Modell .....	97
Abb. 9.1 – Produktionsprozessmodell grob .....	104
Abb. 9.2 – Ausschnitt aus dem MES Modell.....	106

## **Tabellenverzeichnis**

Tab. 4.1 – Abbildungsregeln im Blockdefinitionsdiagramm.....	43
Tab. 4.2 – Elemente des Parameterdiagramms und Abbildung auf IEC 61131 .....	47
Tab. 8.1 – Verknüpfungsmöglichkeiten zwischen den Modellen.....	96



# 1 Cyber Physical Systems – Herausforderung für die Produktion?

Birgit Vogel-Heuser

*Lehrstuhl für Automatisierung und Informationssysteme (AIS), TU München*

## 1.1 Herausforderungen für die Produktion

Cyber Physical Systems (CPS) sind die konsequente Weiterentwicklung eingebetteter Systeme aus Sicht der Informatik. Aus Sicht der Automatisierungstechnik sind CPS die Einbeziehung der unternehmensübergreifenden globalen Vernetzung und der sich selbst adaptierenden Supply-Chains. Die Herausforderung die betriebliche Informationstechnik, wie ERP und MES, mit der Produktionsautomatisierung zu verknüpfen, ist seit einigen Jahren offensichtlich und wird mit einer Vielzahl von Aktivitäten unter dem Stichwort vertikale Integration erarbeitet. Wenn diese Lücke geschlossen wird, können die Daten (Solldaten) von den Ebenen der betrieblichen IT-Systeme als Vorgaben über zu produzierende Stückzahlen sowie Varianten automatisch in der Produktion umgesetzt werden und die Ist-Daten (z.B. Qualitätsdaten, Störungen, Ausfälle) automatisch aus der Produktion an die überlagerten IT-Ebenen gemeldet werden. Damit wird ein überlagerter Unternehmensregelkreis geschlossen (automatische Anpassung von Anlagenkonstellationen, Tracking und Tracing der Produkte und deren Qualitätsdaten).

Bei Cyber Physical Systems ist dieser Regelkreis deutlich weiter gespannt, er beinhaltet andere Anlagen eines Betreibers, Produktionsstätten anderer Betreiber, die Lieferkette der verschiedenen Betreiber sowie die deutlich stärkere Integration des Kunden und Endkunden.

Die Herausforderung ist jedoch mehrdimensional (Abb. 1.1):

- über die **vertikale** Integration als 1. Dimension hinaus, bedarf es, unter Berücksichtigung modularer Produktionseinheiten eines oder mehrerer Hersteller,
- der **horizontalen** Integration (2. Dimension). Die Produktionseinheiten sind mittels eines Informationsmodells zu beschreiben und werden über standardisierte Schnittstellen (vgl. OMAC für die Verpackungsindustrie) miteinander verbunden bzw. können sich selbst verbinden. Diese modularen Produktionseinheiten oder Teile von Produktionsein-

# 1 Cyber Physical Systems – Herausforderung für die Produktion?

heiten integrieren Mechanik, Elektrotechnik/Elektronik und Softwaretechnik. Für eine konkrete Ausprägung einer Anlage ist dies eine bestimmte Variante und Version einer solchen Produktionseinheit, eines produktionstechnischen Moduls.

- Die **dritte Dimension** ergibt sich aus dem Fortschritt im Lebenszyklus des Engineering (Z-Achse): die Module entwickeln sich von der Grobplanung über die Feinplanung hin zur Implementierung und zum Betrieb. Es werden die Varianten und Versionen (korrigierte Varianten) der Module ausgewählt bzw. konstruiert und diese müssen dann, optimaler Weise automatisch im Sinne der automatischen Synthese, zusammengefügt werden. Die Veränderung des Engineering hin zu einem Modellbasierten Entwurf ist selbstverständlich. Die Erfahrungen aus dem Maschinen- und Anlagenbau zeigten, dass sich während des Engineering die Modulschnittstellen und auch die Frage durch welches Gewerk eine Funktion erbracht wird verändern können.
- Der Aspekt des globalen, verteilten Engineering in großen Projekten fügt einen **Schwierigkeitsgrad in dieser Dimension** hinzu.

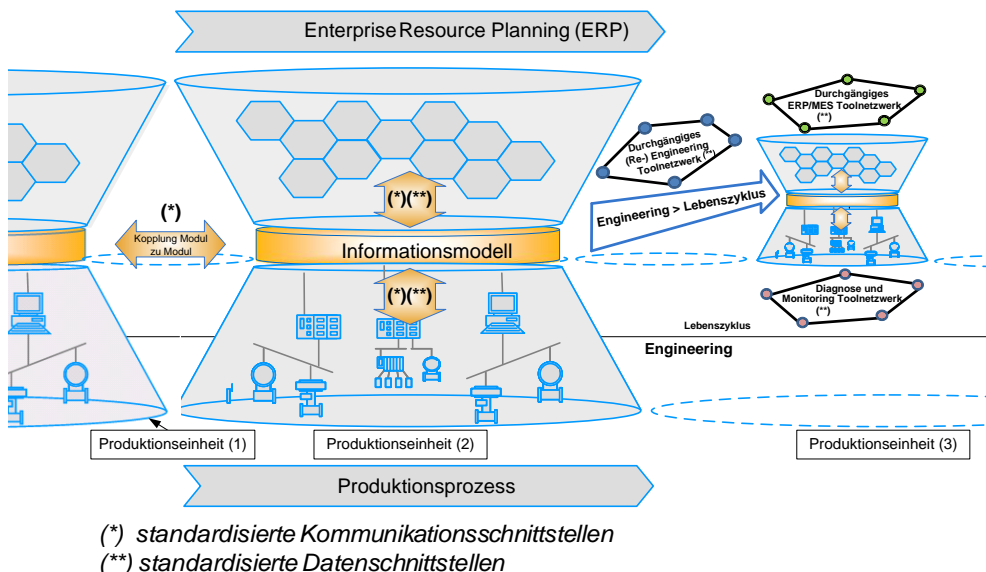


Abb. 1.1 – Herausforderungen: Horizontale und Vertikale Integration, sowie Engineering-Lifecycle (weiterentwickelt aus [1])

Die Frage wie eine geeignete Unterstützung dieser Entwicklung im Engineering mit der Detaillierung, der Änderung der Realisierung, dem iterativen Vorgehen und der gewünschten Werkzeugkette ohne Brüche (siehe „Durchgängiges Engineering Toolnetzwerk“ in Abb. 1.1) und über Unternehmensgrenzen hinweg im Sinne eines Concurrent Engineering realisiert werden kann, ist die Herausforderungen der nächsten 5 Jahre. Damit einhergehende Herausforderungen sind auch für die unter dem Stichwort Cyber Physical System zu entwickelnden Methoden, Vorgehensweisen und Plattformen.

Die Anforderungen an die Software- und Plattformeolution resultieren maßgeblich aus den langen Lebensdauern der Maschinen- und Anlagen von 15 bis 30 Jahren und der daraus resultierenden Notwendigkeit einerseits die Software, die Plattformen und zum Teil auch die Mechanik während der Laufzeit (Betrieb der Produktionssysteme) zu erneuern und andererseits dem Wunsch beim Entwurf neuer Anlagen auf bereits entwickelte Produktionssystem (-teile) zurück greifen zu können. Diese bereits entwickelten Teilsysteme (bestehend aus Software, Automatisierungsplattform und Mechanik) sollen angepasst und für weitere Anlagen genutzt werden.

Häufigste Auslöser für Softwareevolutionsschritte resultieren aus

- der Änderung der Mechanik (Umbau einer Maschine oder eines Apparates),
- der Änderung der Software-Plattform (der Änderung des Betriebssystems, Änderung der Visualisierungs- bzw. Automatisierungs-Systemsoftware, der Integration neuer Sensorik/Aktorik zur Laufzeit),
- der Änderung der Hardware-Plattform (Kommunikationssystem, durch aktualisierte Version von Bussystem, Gerätearchitektur, Rechensystem) und
- der Migration von Plattformen, sowie
- der eigentlichen Softwareevolution infolge der funktionalen Erweiterung der Applikationssoftware (Visualisierung und Steuerung bzw. Regelung).

Die Implementierung der weiterentwickelten Applikationssoftware und der geänderten Plattformen ist soweit möglich während des Anlagenbetriebs notwendig, um keinen bzw. einen minimalen Stillstand des Produktionssystems zu garantieren.



# 1 Cyber Physical Systems – Herausforderung für die Produktion?

Aufgrund der langen Betriebszeiten und den während dieser Zeit auftretenden Funktionsverbesserungen ergibt sich die Notwendigkeit Softwarevarianten eines Moduls in verschiedenen Altanlage einzubauen ohne die Funktion der Gesamtanlage zu gefährden. Die in diesem Zusammenhang auftretende Herausforderung ergibt sich aus der Entwicklung von Varianten und deren Korrektur (Abb. 1.2). Es existieren beispielsweise drei Varianten der Software eines Förderbandes. Wenn sich während der Inbetriebnahme einer von mehreren Anlagen weltweit die Notwendigkeit zu einer Softwareänderung in Variante 1 ergibt, stellt sich die Frage, ob diese Änderungen auch in den Varianten 2 und 3 eingebunden werden müssen. Wenn dies der Fall ist, dann ergibt sich eine praktische Herausforderung, wenn zwischenzeitlich bereits in den Varianten 2 und 3 andere Änderungen durchgeführt wurden. Es ist sicherzustellen, dass durch das Übernehmen der Änderung aus Variante 1 keine wichtigen Änderungen in den Varianten 2 und 3 überschrieben werden bzw. ungültige, nicht lauffähige Software entsteht. Diese Problematik gibt es in vergleichbarer Art in den verschiedenen Gewerken. Diese Herausforderung ist bisher nicht gelöst und erschwert die Verbreitung der Modularität und des Model Driven Engineering in der Automatisierungstechnik.

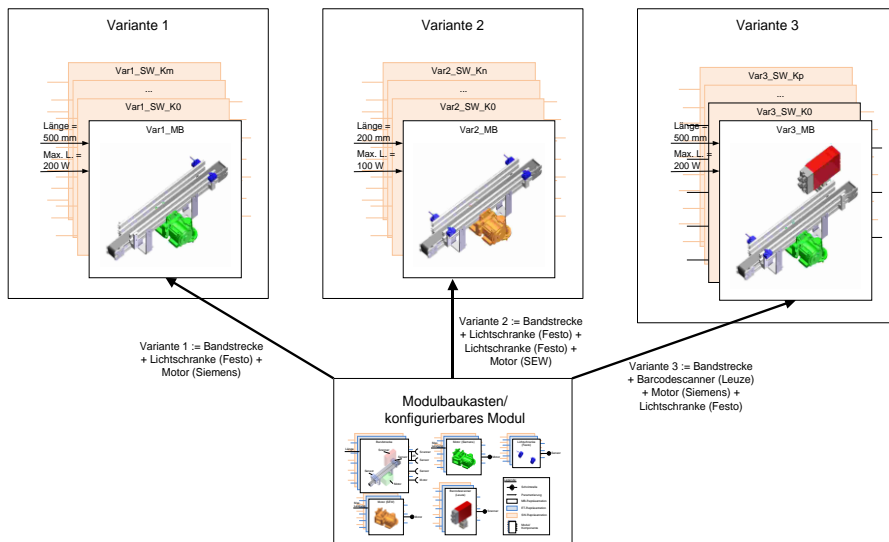


Abb. 1.2 – Varianten des Moduls Band in einem Modulbaukasten

Dabei ist hervorzuheben, dass im Anlagenbau in der Regel nicht von einer „modularen Maschine“ ausgegangen werden kann. Unter modularer Maschine wird die Modulbildung über alle beteiligten Gewerke verstanden bei der die Modulgrenzen aller drei Gewerke gleich sind. Anschaulich formuliert bedeutet dies, dass alles was zum Modul Band gehört, Elektrotechnik/Elektronik, Mechanik oder Software, ist in diesem Modul beinhaltet ist. Die Lösung dieser Herausforderung und die Modularisierung an sich ist eine Voraussetzung um Agentenorientierte Ansätze zur Flexibilitätssteigerung erfolgreich einsetzen zu können.

Um die notwendige Flexibilität und Agilität von Produktionsanlagen im Betrieb (also zur Laufzeit) zu erreichen, werden agentenorientierte Ansätze mit Methoden zur Unterstützung von Verhandlungen wie Mehrzieloptimierung, Entscheidungstheorie, Spieltheorie, Methoden zur Realisierung von context bzw. situation awareness benötigt. Ontologien sind ein Schlüssel zur Beschreibung der Schnittstellen zwischen Produktionseinheiten und der Produktionseinheiten selbst, um die Adaption während des Betriebs zu erlauben.

Eine Vision wie Cyber Physical Systems die Produktion verändern können und welchen Nutzen sie für die Produktion erbringen können, wird im folgenden Beitrag anhand eines Beispiels erläutert.

### 1.2 Referenzen

- [1] Vogel-Heuser, B.; Kegel, G.; Bender, K.; Wucherer, K.: Global Information Architecture for Industrial Automation. In: Automatisierungstechnische Praxis (atp), Oldenbourg-Verlag, München, Jahrgang 51, Heft 1, 2009, S. 108-115.

## 2 Agenda CPS- Szenario smart factory

Birgit Vogel-Heuser, Gülden Bayrak<sup>1</sup>

*Lehrstuhl für Automatisierung und Informationssysteme (AIS), TU München*

Ursula Frank

*Beckhoff Automation GmbH*

### 2.1 Einführung und Begriffsklärung

Die deutsche Investitionsgüterindustrie hat einen hervorragenden Ruf gerade in den Bereichen Automatisierung und Produktionsoptimierung. Darüber hinaus sind der Maschinen- und Anlagenbau (Prozessindustrie und Fertigungsindustrie) und die Automatisierungstechnik wichtige Industrie-sektoren in Deutschland. Noch behauptet sich Deutschland in diesem Bereich als führender Exporteur mit einem Welthandelsanteil von 20% [9]. Um weiterhin erfolgreich am Weltmarkt zu agieren und dem guten Ruf auch zukünftig gerecht werden zu können, müssen Anwendungssysteme und Produktionssystem zusammen wachsen. Die vertikale und horizontale Integration und die Schaffung von Cyber Physical Systems (CPS) (definiert nach [13]) im Bereich der Produktion sind die Trends der Zukunft.

---

#### <sup>1</sup> Unter Mitarbeit von:

Ralf Ackermann (SAP AG)

Dr. Geisberger (fortiss GmbH)

Gerd Hoppe (Beckhoff Automation GmbH)

Bernd Kärcher (Festo AG&Co.KG)

Dr. Josef Papenfort (Beckhoff Automation GmbH)

Stefan Ziegler (Bitkom e.V., Federal

Association for Information Technology,

Telecommunications and New Media)

#### Gereviewt von:

Dr. Oliver Frager (teamtechnik)

Dr. Christine Maul / Dr. Stefan Ochs (Bayer Technology Services GmbH)

Dr. Carolin Theobald / Dr. Stephan Gurke (ZVEI)

Rainer Glatz/ Claus Oetter (VDMA)

Dr. Rainer Stetter (itq)

Andrea Cato / Frank Lafos (Intel GmbH)

### Begriffsklärung **Cyber Physical Systems** [13]

**Cyber-Physical Systems** umfassen typischerweise Eingebettete Systeme (als Teil von Geräten, Gebäuden, Verkehrsmitteln, Verkehrswegen, Produktionsanlagen, Logistik- und Managementprozessen, etc.), die

- mittels Sensoren und Aktuatoren unmittelbar physikalische Daten erfassen und auf physikalische Vorgänge einwirken,
- mit digitalen Netzen verbunden sind (drahtlos, drahtgebunden, lokal, global),
- weltweit verfügbare Daten und Dienste nutzen

und über eine Reihe multimodaler Mensch-Maschine-Schnittstellen (dediziert in Geräten, unspezifisch etwa über Browser, etc.) verfügen.

Wichtige Voraussetzungen sind die Interoperabilität zwischen Hardware und Software der Internettechnologien und den Produktionseinheiten sowie die Einhaltung aller relevanten Safety- und Security-Anforderungen. Die adaptive Herstellung eines Produkts erfordert eine stärkere Integration von Produkt- und Produktionsgestaltung. Insbesondere die Überführung der Produktdaten in die Steuerung der Produktionssysteme führt zu großen Effizienzsteigerungen [1], so können durch automatische Datenübertragung zwischen Modellen Zeit und Fehler eingespart werden. Neben den technischen Grundvoraussetzungen sind neue, flexiblere und übergreifende Organisations-, Management- und Kooperationskonzepte sowie ein vertrauenswürdiges Marktmodell erforderlich. Diese Voraussetzungen zur Realisierung der Vision „*ProCPS- Production CPS*“ können nur durch die konzertierte Aktion von Anlagenbetreibern, Unternehmen des Maschinen- und Anlagenbaus, Herstellern von Automatisierungstechnik, der Informations- und Kommunikationstechnologie (IKT)-Industrie geschaffen werden.

Bestehende Ansätze aus den Bereichen Smart Factory [8], F<sup>3</sup> Factory [10], Digitale Fabrik [4], Integrative Produkt- und Prozessgestaltung, wandlungsfähige Produktionssysteme [11], Life-Cycle-Management [3] und Konfigurationsmanagement reichen zur Bewältigung der Herausforderungen nicht aus. Notwendig sind ProCPS Lösungen, die einen großen Spielraum für Varianten schaffen um den weltweiten Zugriff auf Ressourcen und bisher unbekannte Produktionseinheiten zu ermöglichen, die einen hohen Grad an Flexibilität in den Wertschöpfungsketten über die Unternehmens-

grenzen hinweg und in den Produktions- und Prozesssystemen ermöglichen, die Lieferzeiten minimieren und ein großes Maß an Nachhaltigkeit garantieren.

Die folgenden Kapitel konkretisieren die Vision der ProCPS und diskutieren die Herausforderungen sowie den Handlungsbedarf zur Umsetzung der Vision. Die in verschiedenen Industriebereichen existierenden Herausforderungen wurden der Anschaulichkeit halber in ein Szenario „Kauf einer Küche“ zusammengeführt. Das Szenario wurde als Basis für die Ableitung der Fähigkeiten und der Herausforderungen der ProCPS genutzt.

### **2.2 Szenario: kundenspezifische Produktion einer Küche**

Eine kundenspezifische Küche soll mit den Anforderungen geringer Preis, Verwendung ökologisch einwandfreier Materialien und ökologische, nachhaltige Produktion hergestellt werden. Zusammen mit einem Küchenhändler wird mittels eines Küchenkonfigurators unter Nutzung von VR-Techniken (Virtual Reality) zur Erprobung der Bestellung die Küche zusammengestellt. Basierend auf den kundenspezifischen Daten: Küchendaten, Kostenrahmen (inkl. Logistikkosten), Qualitätsgrad, und den Aspekt der Nachhaltigkeit (CO2 Footprint, Ökosteuer) ermittelt die ProCPS das ideale Produktionssystem (*flexible, kontext-adaptive Kooperation über Unternehmensgrenzen (weltweit), Verhandlungsstrategien*), die Produktionskosten, mögliche Lieferzeiten und die Nachhaltigkeit für den Händler. In der ProCPS organisieren sich die Produktionseinheiten so (*Eigenständige Kommunikation mit Hilfe von Web-Services (drahtlos, drahtgebunden), vertikale Vernetzung zur Steuerung der Produktionsanlagen*), dass sie die Komponenten der Küche unter den geforderten Randbedingungen produzieren können. Zusätzlich berücksichtigen die Produktionseinheiten auch ihre Zugehörigkeit zu einem Anlagenbetreiber, ihren eigenen Standort, ihren Auslastungsgrad, die Logistikkosten, die Kompatibilität der Produktionseinheiten verschiedener Anlagenbetreiber (*horizontale Vernetzung zur Interaktion von Produktionseinheiten*) unter-einander, das Wertschöpfungsnetzwerk ausgehend vom Rohmaterial, über Zwischenstufen bis hin zur fertigen Küche als auch die vertraglichen Bedingungen der Betreiber und Lieferanten. Die Planung ergibt, dass sich alle gewählten Produktionseinheiten zur Fertigung der Möbelstücke bei dem Betreiber A in Deutschland und alle Produktionseinheiten zur Fertigung der Arbeitsplatte bei dem Betreiber B in Osteuropa befinden. Nach Auftragsvergabe durch den Endkunden über den Händler starten die Betreiber die Produktionseinheiten.

## 2.2.1 Teilszenario 1: Auftragsänderung

Der Endkunde möchte ein anderes Dekor für die Arbeitsplatte, welches eine andere Produktionsmethode (Teilszenario 1.1) sowie eine aufwendige Vorbehandlung der Roharbeitsplatte (Teilszenario 1.2) erfordert. Das erkennt das gewählte Produktionssystem und fragt bei der ProCPS nach Produktionseinheiten zur Umsetzung der erforderlichen Produktionsmethode und nach Produktionseinheiten zur Vorbehandlung der Roharbeitsplatte. Die evtl. zusätzlichen Kosten werden dem Endkunden mitgeteilt. Der Endkunde bestätigt die Auftragsänderung dem Händler und dieser bestätigt die Auftragsänderung dem Anlagenbetreiber B.

*Fähigkeiten: Auftragsänderung trotz Auftragsbestätigung und der begonnenen Fertigung der Küchenteile.*

*Anforderungen: Adaptivität der Wertschöpfungskette, Heterogenität der Produktionseinheiten (unterschiedlichste Versionen und Hersteller), Vertikale Vernetzung, Durchgängiges Toolnetzwerk (ERP/MES Ebene), Datenkopplung über Unternehmensgrenzen und verschiedene Anbieter hinweg, Methoden zur Realisierung von context/ situation awareness.*

### Teilszenario 1.1: Änderung der Produktionsmethode

Vier verschiedene Produktionseinheiten, welche sich bei Betreibern in vier verschiedenen Kontinenten befinden, bieten sich zur Realisierung der neuen Produktionsmethode an. Aufgrund der Randbedingungen bekommt die Funktionseinheit des Betreibers E in Osteuropa den Zuschlag. Nach Transport der Produktionseinheit (des Betreibers E) zum Betreiber B, integriert sich die Produktionseinheit selbständig in das Produktionssystem des Betreibers B.

*Fähigkeiten: Bewertung der Produktionseinheiten und Entscheidung anhand von Kriterien, Neue Zuordnung der Produktionseinheiten zu Anlagenbetreibern, Kommunikation/Vernetzung der Produktionseinheiten und der Anlagenbetreiber, Integration anderer, neuer Produktionseinheiten während des Betriebs.*

*Anforderungen: Datenkopplung über Unternehmensgrenzen und verschiedene Anbieter hinweg, Horizontale Vernetzung zur Interaktion von Produktionseinheiten, Interoperabilität der Produktionseinheiten von unterschiedlichen Herstellern, Methoden zur Unterstützung von Verhandlungen*

wie *Mehrzieloptimierung, Entscheidungstheorie, Dynamische Rekonfiguration zur Laufzeit, Methoden zur Konfiguration und Konsistenzanalyse.*

### **Teilszenario 1.2: Aufwendige Vorbehandlung der Roharbeitsplatte**

Die aufwendige Vorbehandlung ist mit vorhandenen Produktionseinheiten nicht realisierbar. Für eine Anpassung bietet sich ausschließlich eine Produktionseinheit an und für diese gilt die Einschränkung, dass sie weder Arbeitsplatten in der angefragten Größe vorbehandeln kann und auch nicht portabel ist, also beim Betreiber C die Vorbehandlung durchgeführt werden müsste. Die Produktionseinheit macht das Angebot eine größere Auflagefläche entwickeln zu lassen, so dass sie den Auftrag übernehmen kann. Sie erkundigt sich nach den Entwicklungs- und Fertigungskosten für eine größere Auflagefläche, berechnet die Logistikkosten für den Transport der Arbeitsplatte zu ihrem Betreiber C und erstellt einen Kostenvoranschlag. Betreiber B akzeptiert die Bedingungen und vergibt einen Unterauftrag an Betreiber C. Die Produktionseinheit für die Vorbehandlung wird erweitert. Die Produktionseinheit meldet die Daten an die Produktionseinheit des Betreibers B und diese integriert sie in den Wertschöpfungsprozess.

*Fähigkeiten: Anpassung, Umkonstruktion / Evolution von Produktionseinheiten während Produktionsauftrag bereits eingelastet.*

*Anforderungen: Skalierbarkeit von Produktionseinheiten, Durchgängige Engineering Werkzeugkette und Datenkopplung, Modellgetriebene Entwicklung, offene Plattformen, Methoden zur Unterstützung von Verhandlungen wie Mehrzieloptimierung, Entscheidungstheorie, Methoden zur automatischen Codegenerierung, Entwurf und Modellierungsmethoden.*

### **2.2.2 Teilszenario 2: Qualitätsverlust während der Produktion**

Bei der Qualitätskontrolle werden Kratzer auf der Oberfläche der Sockelleiste erkannt, daher muss sie ausgetauscht werden. Die Sockelleiste ist ein Kunststoffprofil mit Holzdekor und einseitigem Klebeband. Der Lieferant stellt die Profile in einem kontinuierlichen Kombinationsverfahren aus Heißextrusion, Beschichtung und Aufbringung des Klebebands her. Die beteiligten Produktionseinheiten erhalten die für die Optimierung der Produktqualität notwendigen Prozess- sowie Qualitätsdaten des Küchenherstellers. Die Daten werden selbstständig analysiert und die Produktions-

einheiten passen auf Basis dieser die Prozessparameter an, um den Verschleiß zu kompensieren und derartige Beschädigungen zu vermeiden.

*Fähigkeiten: Regelkreis von Qualitätsdaten und Produktionsdaten, Vorgabe geänderter Prozessparameter.*

*Anforderungen: Vertikale Vernetzung (automatische Anpassung von Tracking und Tracing der Produkte der aktuellen Anlagenkonstellation und der Qualitätsdaten), Durchgängige Engineering Werkzeugkette und Datenkopplung, Methoden zum Testen, Simulieren, Optimieren und Validieren.*

### **2.2.3 Teilszenario 3: Ausfall einer Komponente**

Das virtuelle Plattenmodell, welches durch das Tracking der Aufbereitung und Formstraße realisiert wurde, stellt als Stellvertreter der zu produzierenden Platte fest, dass für die in zwei Stunden zu sägende Platte die Hochleistungssäge bei dem Betreiber A ausfallen wird. Das virtuelle Plattenmodell meldet sich bei der ProCPS und fragt nach einer gleichwertigen Hochleistungssäge an. Alle gleichwertigen Hochleistungssägen sind besetzt, anstelle dessen melden sich zwei Sägen mit geringerer Leistung. Eigenständig simuliert das Produktionssystem den Einsatz der Sägen und ermittelt, dass die beiden Sägen gemeinsam die geforderte Leistung erbringen. Aufgrund unterschiedlicher Protokolle können die Sägen nicht direkt in die Steuerung des Produktionssystems eingebunden werden. Ein Schnittstellen-Agent erkennt dieses Defizit, bietet eine Schnittstelle zu einem akzeptablen Preis für den Agenten der virtuellen Platte an, entwickelt diese und stellt sie dem Betreiber zur Verfügung. Der Agent der virtuellen Platte veranlasst, dass das Produktionssystem während des Betriebs zum korrekten Zeitpunkt die Hochleistungssäge durch die zwei Sägen mit geringerer Leistung austauscht und produziert ohne Unterbrechung und ohne Effizienz- sowie Qualitätseinbußen die Möbelteile.

*Fähigkeiten: Vorschlag von Alternativlösungen um das Produkt zeitnah herzustellen, Simulation von Produktionseinheiten und des herzustellendes Produkt, Systemkopplung.*

*Anforderungen: Selbstkonfiguration, Flexibilität, Modularität der Produktionseinheiten, Skalierbarkeit, Adaptivität, Zuverlässigkeit, Heterogenität und Interoperabilität der Produktionseinheiten, Agenten-orientierte Methoden, Methoden zur Konfiguration und Konsistenzanalyse, Methoden zur Realisierung von context/ situation awareness, Methoden zur Unterstützung*



von Verhandlungen wie *Mehrzieloptimierung*, *Entscheidungstheorie*, *Methoden zur automatischen Codegenerierung*.

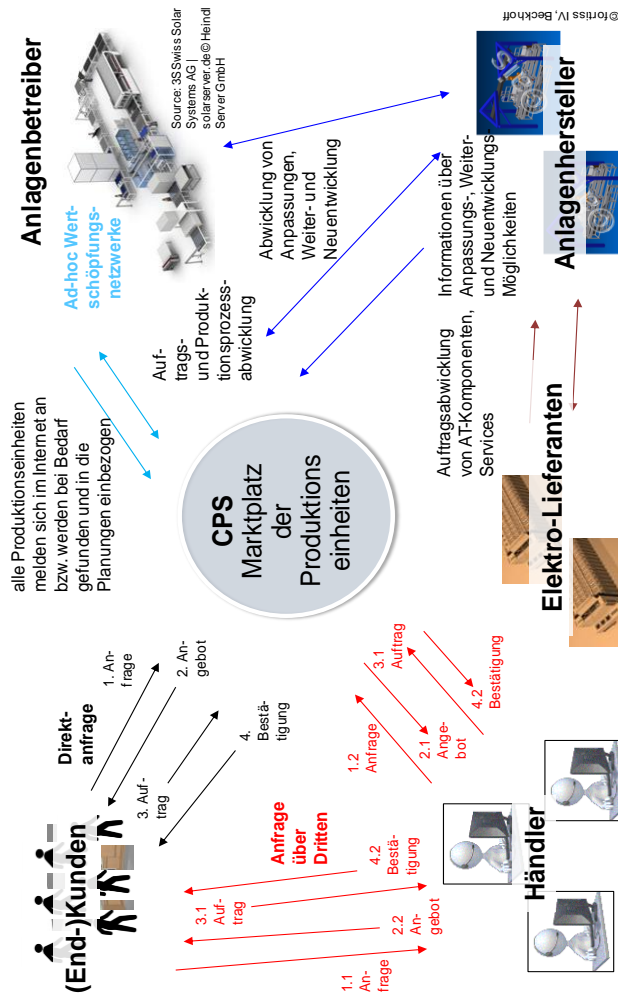


Abb. 2.1– CPS Factory und die beteiligten Stakeholder

### 2.3 Eigenschaften von CPS Produktionseinheiten

Damit das Potential der ProCPS voll ausgeschöpft werden kann, müssen bestimmte Eigenschaften realisiert werden. Die Produktionseinheiten müssen beispielsweise weitgehend ortsunabhängig sein und müssen bei Bedarf über eine CPS Logistik ihren Standort bzw. Anlagenbetreiber wechseln können. Für die neue Planung von Produktionseinheiten bzw. des Produk-

tionsprozesses aufgrund einer neuen Kundenanforderung/Auftragsänderung, wie im Szenario beschrieben, müssen Produktionseinheiten kontextspezifisch sein. Neue Planung von Produktionseinheiten sowie Änderung der Produktionseinheiten bzw. des Produktionsprozesses aufgrund eines Ausfalls oder einer Kapazitäts-, Auftrags- oder Qualitätsoptimierung erfordert die Eigenschaft Adaptivität. Zudem müssen Produktionseinheiten autonom, multifunktional, weltweit verteilt und vernetzt sein. So können sich die Produktionseinheiten selbständig nach definierten Kriterien neu organisieren.

### **2.4 Herausforderungen für das Engineering und die Realisierung von ProCPS**

Die Modularisierung, durchgängige Engineering Toolketten, und modellgetriebene Entwicklung (Model Driven Design) sind entscheidende Herausforderungen. Außerdem ist die Integration der Produktentwicklung, bei der Entwicklung von Produktionsanlagen umzusetzen, um schnell auf geänderte Anforderungen an Produkte reagieren zu können. Wissenschaftlich ist dazu die heute existierende Trennung zwischen Produktentwicklung und Entwicklung von Produktionsanlagen sowie Produktentwicklern, Qualitätsmanagement und Produktionsplanung und deren Automatisierung zu überwinden ganz zu schweigen von den dazu eingesetzten Werkzeugen und der fehlenden Datenintegration. Vor diesem Hintergrund ist ein Ansatz für das Concurrent Engineering von Produkt und Produktionsanlage inklusive der jeweils unterschiedlichen Gewerke zu entwickeln, um solche neuen und gekoppelten Prozesse zu unterstützen und um mit größtmöglicher Schnelligkeit auf die Anforderungen nach Innovationen oder Reengineering reagieren zu können. Wesentlich ist es bereits bei der Produktentwicklung die möglichen Grenzen oder gegenläufigen Kostenfunktionen der Produktionsanlagen sowie der Wartungsnotwendigkeiten von Produkt und Produktionsanlage als Optimierungskriterium für das Produkt mit einzubeziehen. Für die Komplexitätsbeherrschung und Wiederverwendung ist eine der Herausforderungen dabei, die disziplinspezifischen Module (Mechanik, Elektrik/Elektronik und Software) zu identifizieren und diese integriert zu modellieren ([6]). Die disziplinspezifischen Modellierungssichten müssen dabei abstrakt aber dennoch dazu in der Lage sein, aus der Modellierung automatisch einen lauffähigen Code zu erzeugen. Dabei müssen die spezifischen Systemeigenschaften wie Safety, Security, Interoperabilität, Adaptivität und Autonomie beschrieben, modelliert und nach Code-Übersetzung sichergestellt werden. Da die manuelle Datentransformation zwischen ver-

schiedenen Disziplinen fehleranfällig und zeitaufwändig ist, müssen einheitliche Schnittstellen realisiert und Standards entwickelt werden. Die Werkzeugkopplung entlang des gesamten Lifecycle aus dem Bereich Maschinenbau ist eine weitere Herausforderung um eine durchgängige Datenbearbeitung bzw. durchgängiges Engineering zu ermöglichen. Für das Engineering und die Realisierung der CPS Factory sind also folgende Herausforderungen zu bewältigen:

- horizontale und vertikale Vernetzung
- die integrative disziplinübergreifende Entwicklung von Produkt und Produktionssystem
- die Modularisierung der Produktionssysteme zu Produktionseinheiten
- eine modellgetriebene Entwicklung (Model Driven Design)
- eine durchgängige Engineering Werkzeugkette
- Vorgehensweisen, Methoden, Beschreibungsmittel, Softwarewerkzeuge

In den nächsten Unterkapiteln werden diese Herausforderungen näher erläutert.

### **2.4.1 Horizontale und vertikale Vernetzung**

Für den Datenaustausch und die Interaktion zwischen Stakeholdern und (Teil-)Systemen ist die Vernetzung eine wesentliche Eigenschaft, deren Ausbau und Durchdringung der Produktionssysteme weit über den bekannten Stand der Technik hinausgehen soll. Die weltweite ideale Zusammenstellung und Organisation von Produktionseinheiten, die Auswirkung von Anforderungsänderungen und Komponentenausfällen auf das Wertschöpfungsnetzwerk und die Umstrukturierung von Produktionseinheiten ist nur durch eine umfassende Vernetzung erreichbar.

Innerhalb einer CPS Factory steuert ein unternehmensinternes CPS, als durchgängiges, flexibles IT System, die Produktion auf den durch das CPS System ausgewählten Produktionseinheiten. Es deckt die Funktionen eines heutigen ERP, MES und Leitsystems ab und kommuniziert einerseits über standardisierte Schnittstellen mit den Produktionseinheiten des Anlagenbetreibers (standortübergreifend) und andererseits mit der CPS Factory (Produktionseinheiten anderer Anlagenbetreiber, weitere Kunden und Zulieferer). Das unternehmensinterne CPS plant längerfristig die vom CPS angeforderten Produktionseinheiten und sorgt durch eine optimierte Angebots-

abgabe für eine Volllastung der Produktionseinheiten. Dabei berücksichtigt es die von den Produktionseinheiten selbstständig vorausschauend auf Basis ihres Verschleißes geplanten Wartungen. Es sendet über die Standardschnittstelle die von der CPS Factory erhaltenen Produktionsaufträge, ggf. zusammen mit den benötigten Rezepten, an die ausgewählten Produktionseinheiten, erhält durch diese eine Rückmeldung ihrer aktuellen Eigenschaften und antwortet zur Initialisierung der Anlage mit den Konfigurationsdaten. Hierfür müssen entweder Herstellenweisungen und Rezepte auf einzelne Produktionseinheiten zugeschnitten werden oder die Produktionseinheiten müssen in der Lage sein, selbst die für sie relevanten Teile zu identifizieren.

Das unternehmensinterne CPS muss seine, heute MES Systemen zuzuordnenden Funktionalitäten wie das Tracking und Tracing der Produkte automatisch der aktuellen Anlagenkonstellation anpassen können. Es soll selbstständig die relevanten Kennzahlen beispielsweise in Abhängigkeit der Art und des Produktes oder der vorgegebenen Fehlertoleranz für die Produktion berechnen, die Ergebnisse zentral speichern und entsprechende Maßnahmen ergreifen.

Die standardisierte Schnittstelle zu den Produktionseinheiten muss somit nicht nur die Übertragung der Produktionsaufträge, Rezepte und aktuellen Status erlauben, sondern auch die Übermittlung und Interpretation aller für die Kennzahlenberechnung und längerfristige Produktionssteuerung notwendigen Daten. Der Teilverfolgung durch die Anlage kommt eine sehr zentrale Rolle zu. Aufgrund der gesetzlichen Anforderungen ist es immer wichtig genau zu wissen, wie welches Produkt gefertigt wurde. Die Speicherung aller relevanten Daten für ein individuelles Produkt muss zentral erfolgen.

Auf intelligenten Sensoren und Aktoren von Produktionseinheiten können sowohl Vorverarbeitungen aber auch im Falle eines Ausfalls einer Rechnerkomponente Rechenaufgaben übernommen werden und damit die Verfügbarkeit der Anlage erhöht werden. Des Weiteren ist eine geeignete aufgabenabhängige, zeitlich veränderliche Verteilung allgemeiner, steuerungstechnischer Aufgaben auf die Rechenknoten notwendig.

### **2.4.2 Integrative disziplinübergreifende Entwicklung von Produkt und Produktionssystem**

Zukünftig bieten die Produktionseinheiten ihre Dienste eigenständig an und bilden in Abhängigkeit vom zu fertigenden Produkt eigenständig Verfahren- und Technologieketten. Das erfordert die integrative Zusammenarbeit bei der Planung und Entwicklung von Produkt und Produktionssystem aller Entwickler der Bereiche Produkt-, Produktions- und Betriebsmittelentwicklung und der dort involvierten Domänen

### **2.4.3 Modularisierung**

Produkte sowie Produktionssysteme und Produktionseinheiten zur Herstellung der Produkte werden immer komplexer. Modularisierung ist ein Werkzeug diese Komplexität zu beherrschen, gleichzeitig ermöglicht sie die Wiederverwendung von bewährten Lösungen. Im Rahmen der CPS Factory können Produktionssysteme und Produktionseinheiten als Modul interpretiert werden. Eine Herausforderung ist die Festlegung der Module. Es müssen sinnvolle Einheiten als Module identifiziert und deren Systemgrenzen sowie Schnittstellen festgelegt werden. Die Module der Produktionseinheiten werden in der Regel disziplinübergreifend sein, aus ihnen müssen die disziplinspezifischen Systemgrößen und Verhaltensweisen ableitbar sein. Zur Umsetzung der Vision CPS Factory sind geeignete Definitionen von disziplinübergreifenden und disziplinspezifischen Modulen, Modularisierungsstrategien und -regeln zu entwickeln und zu standardisieren.

### **2.4.4 Modellgetriebene Entwicklung (Model Driven Design)**

Ziel ist eine Modell-basierte Entwicklung von Produkt und Produktionssystem über den gesamten Lebenszyklus. Eine Modellgetriebene Entwicklung reduziert die Durchlaufzeiten, erhöht die Qualität und unterstützt die Wiederverwendung bewährter Lösungen sowie die Nachverfolgbarkeit. Die Modelle müssen die Realität hinreichend genau abbilden und simulationsfähig sein, so dass Aussagen zu Kosten, Qualität, Funktionsfähigkeit, Zuverlässigkeit, Wartungsbedarf und Ressourcenverbrauch für das Produkt und das Produktionssystem schon frühzeitig getroffen werden können und bei Bedarf Optimierungen durchgeführt werden können. Im Rahmen von ProCPS werden hohe Anforderungen an die Modellbildung gestellt: Die Modelle sollen das Produktionssystem und das zu fertigende Produkt sowie die Wechselbeziehungen abbilden und aus den Modellen soll automatisch lauffähiger Code erzeugt werden. Zur Erreichung dieser Ziele müssen die

Modelle auf unterschiedlichen Abstraktionsniveaus, mit unterschiedlichen Sichten sowie disziplinübergreifend und disziplinspezifisch modelliert werden. Eine große Herausforderung ist dabei die disziplinspezifischen Module (Mechanik, Elektrik/Elektronik und Software) zu identifizieren und diese integriert zu modellieren [2], [6]. Die Modellierungen müssen abstrakt sein, aber es muss dennoch die Möglichkeit bestehen, aus der Modellierung automatisch einen lauffähigen Code zu erzeugen. Dabei müssen die spezifischen Systemeigenschaften wie Safety, Security, Interoperabilität, Adaptivität und Autonomie beschrieben, modelliert und nach Code-Übersetzung sichergestellt werden.

### 2.4.5 Durchgängige Engineering Werkzeugkette

Sowohl die Entwickler als auch die Produktionseinheiten nutzen diverse Software-Werkzeuge zum Planen, Entwickeln, Modellieren, Simulieren, Optimieren und Implementieren. Eine Entwicklung, Weiterentwicklung oder Anpassungsentwicklung kann losgelöst oder abhängig von einem aktuellen Produktionsprozess erfolgen. Ebenso kann die Inbetriebnahme der neuen Lösung vor dem Start eines Produktionsauftrags oder automatisch während der Betriebsphase eines Produktionssystems erfolgen (Plug and Produce). Grundsätzlich entsteht eine Vielzahl von Daten, die von den nachfolgenden Werkzeugen genutzt werden müssen. Da die manuelle Datentransformation fehleranfällig und zeitaufwändig ist, müssen einheitliche Schnittstellen realisiert und Standards entwickelt werden. Es ist eine Werkzeugkopplung entlang des gesamten Lebenszyklus erforderlich, um eine durchgängige Datenbearbeitung bzw. ein durchgängiges Engineering zu ermöglichen. Zudem ist eine Kopplung zwischen Entwicklungswerkzeugen und der Laufzeitumgebung zu schaffen, so dass auch während des Betriebs durch die Produktionseinheiten parallel Entwicklungen durchgeführt und implementiert werden können. Im Rahmen der ProCPS ist weiterhin eine Kopplung der Engineering Werkzeuge mit den überlagerten MES- und ERP-Systemen sowie der ProCPS erforderlich, so dass aufgrund geänderter Auftragsdaten die Produktionseinheiten automatisch Weiter- und Anpassungsentwicklungen initiieren und umsetzen können.

Wichtige Themen sind hier Konfigurations-, Verhandlungs- und Entscheidungsmethoden. Auf der **Unternehmens- und Betriebsebene** geht es um die Auftragsabwicklung, Produktionsplanung und Logistik, durchgängiges Engineering, Methodenkopplung und Werkzeugkopplung im Sinne der Co-Simulation. Auf der **Feldebene** - d.h. an den Produktionssystemen - stehen

im Vordergrund fehlerfreies, effizientes Produzieren und Methoden sowie Optimierungstechniken zur Erreichung dieses Ziels.

### **2.4.6 Vorgehensweisen, Methoden, Beschreibungsmittel und Softwarewerkzeuge**

Für Klassen von Produktionssystemen entstehen Leitfäden, die das Vorgehen bei der integrativen Entwicklung beschreiben. Sie beinhalten die Hauptprozessschritte und deren Resultate sowie Handlungsempfehlungen für situationsbedingte Prozessanpassungen. Die Leitfäden sind Basis für die Realisierung des automatischen Entwurfs und die automatische Implementierung sowie Inbetriebnahme.

#### **Vorgehensweisen**

Um komplexe ProCPS zu entwerfen, zu implementieren und zu betreiben ist eine Entwicklungsmethodik erforderlich, die alle oben genannten Aspekte beinhaltet. Die Entwicklungsmethodik stellt die für den Entwurf, die Implementierung und den Betrieb notwendigen Vorgehensweisen, Methoden, Modellierungstechniken sowie Beschreibungssprachen, Softwarewerkzeuge und das erforderliche Entwurfswissen zusammen.

#### **Methoden**

Es ist ein Set an Methoden zusammenzustellen, die ein Entwickeln und Betreiben von ProCPS unterstützen. Es wird angestrebt, so weit wie möglich bewährte Methoden einzusetzen. Diese allein reichen jedoch nicht aus, so dass bestehende Methoden weiterentwickelt werden und neue Methoden hinzukommen müssen. Erforderlich sind u.a. geeignete

- Entwurfs- und Modellierungsmethoden
- Methoden zur Modularisierung, Konfiguration, Konsistenzanalyse
- Methoden zur Unterstützung von Verhandlungen wie Mehrzieloptimierung, Entscheidungstheorie, Spieltheorie
- Methoden zum Testen, Simulieren, Optimieren, Validieren
- Visualisierungsmethoden
- Methoden zur automatischen Codegenerierung
- Methoden zur Realisierung von context /situation awareness

- Agentenorientierte Methoden

### **Modellierung/Beschreibungssprachen**

Die Beschreibung von Modellen erfolgt über Beschreibungsmittel, auch Spezifikationstechniken genannt. Es müssen die quantitativen und qualitativen Eigenschaften der Systeme abgebildet werden. Wichtig ist insbesondere die Modellierung von Zeiten (diskret und kontinuierlich) und deren Synchronisation. Zur Beschreibung der ProCPS sind disziplinübergreifende und disziplinspezifische Beschreibungsmittel mit unterschiedlichem Abstraktionsniveau und zur Darstellung unterschiedlicher Sichten auf das System erforderlich. Die Beschreibungsmittel müssen zu einer ganzheitlichen Beschreibung des Systems integrierbar sein. Die aktuell existierenden Beschreibungsmittel wie beispielsweise UML, SysML und deren Erweiterungen reichen zur Spezifikation aller Aspekte der ProCPS nicht aus. Sie müssen erweitert werden und neue Beschreibungsmittel müssen entwickelt werden. Ziel ist ein standardisiertes Set an Beschreibungsmitteln.

### **Softwarewerkzeuge**

Alle für die Entwicklung und den Betrieb notwendigen Softwarewerkzeuge (Entwurfswerkzeuge, Datenverwaltungswerkzeuge, Workflow-Systeme, ERP- und MES-Systeme, Datenbanken, Assistenzsysteme) sind zu einer virtuellen Arbeitsumgebung zusammenzuführen. Diese Arbeitsumgebung muss für jeden ProCPS Anwender weltweit zur Verfügung stehen und an seine Bedürfnisse anpassbar sein. Zur Erschließung der Potentiale der ProCPS sind zudem insbesondere Erweiterungen und neue Konzepte im Bereich Varianten- und Versionsmanagement ([7]), Änderungsmanagement, Daten- und Zugriffssicherheit, Umgang mit Zeitversatz und unterschiedlichem Zeitverhalten, Autonomie und Vertraulichkeit zu entwerfen. Zusammenfassend sind eine entsprechende Architektur und standardisierte Schnittstellen zu schaffen. Die Modularisierung, Durchgängige Engineering Toolketten, und Modellgetriebene Entwicklung (Model Driven Design) sind dabei sehr wichtig. Außerdem ist die Integration der Produktentwicklung, der Entwicklung der Produktionsanlagen zu integrieren um schnell auf geänderte Anforderungen an die Produkte reagieren zu können. Wissenschaftlich ist dazu die heute existierende Trennung zwischen Produktentwicklung und Entwicklung von Produktionsanlagen sowie Produktentwicklern, Qualitätsmanagement und Produktionsplanung und deren



Automatisierung zu überwinden ganz zu schweigen von den dazu eingesetzten Werkzeugen und der fehlenden Datenintegration.

### **2.5 Zusammenfassung/Ausblick**

Cyber Physical Systems (CPS) sind die konsequente Weiterentwicklung eingebetteter Systeme aus Sicht der Informatik. Aus Sicht der Automatisierungstechnik sind sie die Einbeziehung der unternehmensübergreifenden globalen Vernetzung und der sich selbst adaptierenden Supply-Chains.

### **2.6 Referenzen**

- [1] Jovane, F.: "Leadership of European Manufacturing Industry" in The ManuFuture Road. Towards Competitive and Sustainable High-Adding-Value Manufacturing, F. Jovane, E. Westkämper, D. Williams, Springer-Verlag: Berlin Heidelberg, 2009, S. 5-29.
- [2] Gausemeier, J.; Frank, U.; Donoth, J.; Kahl, S.: Specification technique for the description of self-optimizing mechatronic systems. In: Research in Engineering Design, Springer London, 2009.
- [3] Life-Cycle-Management für Produkte und Systeme der Automation, ein Leitfaden des Arbeitskreises Systemaspekte im ZVEI Fachverband Automation, ZVEI-Zentralverband Elektrotechnik- und Elektronikindustrie e.V. Fachverband Automation, Frankfurt, 2010.
- [4] Kuhn, W.: "Digital Factory - Simulation Enhancing the Product and Production Engineering Process," Simulation Conference, 2006. WSC 06. Proceedings of the Winter, 2006.
- [5] Zuehlke, D.: SmartFactory – from Vision to Reality in Factory Technologies In: Proceedings of the 17th World Congress, The International Federation of Automatic Control, Seoul, Korea, 2008.
- [6] Sim, T. Y.; Li, F.; Vogel-Heuser, B.: Benefits of an Interdisciplinary Modular Concept in Automation of Machine and Plant Manufacturing. In: 13th IFAC Symposium on Information Control Problems in Manufacturing (INCOM), Moskau, 2009, S. 898-903.

- [7] Sim, T.Y.; Li, F.; Vogel-Heuser, B.: Modules, version and variability management in automation engineering of machine and plant manufacturing. In: Emerging Technologies and Factory Automation (ETFAs), Hamburg, 2008, S. 46-49.
- [8] Technologieinitiative SmartFactory KL e.V., 2011 [online]. <http://www.smartfactory-kl.de/>
- [9] VDMA -Verband Deutscher Maschinen- und Anlagenbau e.V. 2007 [online]. [www.vdma.org](http://www.vdma.org).
- [10] Hessel, V.; Cortese, B.; de Croon, M.H.J.M.: Novel process windows - Concept, proposition and evaluation methodology, and intensified superheated processing, Chemical Engineering Science, Ausgabe 66, Microfluidic Engineering, 2011.
- [11] Wandlungsfähigkeit durch modulare Produktionssysteme (WAMOPRO). 2011 [online]. <http://www.wamopro.de>
- [12] Vogel-Heuser, B.; Kegel, G.; Bender, K.; Wucherer, K.: Global Information Architecture for Industrial Automation. In: Automatisierungstechnische Praxis (atp), Oldenbourg-Verlag, München, Jahrgang 51, Heft 1, 2009, S. 108-115.
- [13] Broy, M (Hrsg.): Cyber-Physical Systems. Innovation durch Software-Intensive Eingebettete Systeme. In: acatech diskutiert, Springer-Verlag, Berlin, Heidelberg, 2010.
- [14] Hertzberg, J.; Beetz, M.; Englert, R.; Buss, M.; Wollherr, D.: Cognitive Technical Systems — What Is the Role of Artificial Intelligence? In: KI 2007: Advances in Artificial Intelligence, Lecture Notes in Computer Science, Springer Berlin / Heidelberg, 2007.

# 3 Potentiale und Herausforderungen für die Steuerungsprogrammierung im Maschinen- und Anlagenbau von morgen

Ursula Frank, Josef Papenfort  
*Beckhoff Automation GmbH, Verl*

Martin Obermeier, Birgit Vogel-Heuser  
*Lehrstuhl für Automatisierung und Informationssysteme(AIS), TU München*

**Zusammenfassung:** Die Softwareentwicklung im Maschinen- und Anlagenbau befindet sich im Umbruch. Neben den klassischen, prozeduralen SPS-Programmiersprachen der IEC 61131-3 kommen zunehmend objektorientierte Elemente zum Einsatz, wie sie auch in der objektorientierten Erweiterung der IEC 61131-3 eingeführt werden. Dieser Artikel erläutert zunächst an Hand von Szenarien ausgewählte Potentiale neuer, im speziellen agentenbasierte Ansätze für automatisierungstechnische Anlagen. Im Anschluss werden in einer Expertenevaluation erhobene Anforderungen an Entwicklungsmethoden und Werkzeuge aufgezeigt und interpretiert. Abschließend werden noch zu bewältigende Herausforderungen abgeschätzt.

## 3.1 Einleitung

Die Notwendigkeit des Einsatzes dezentraler, hochgradig flexibler und rekonfigurierbarer Systeme wird angesichts des steigenden Kostendrucks, kürzerer Entwicklungszyklen und der zunehmend an den individuellen Kundenwünschen ausgerichteten Produktion größer [1]. Dementsprechend steigt das Interesse auf Seiten der Industrie an neuen flexiblen Lösungen, welche zudem die steigende Komplexität moderner Anlagen handhabbar machen. Obwohl in der konventionellen Softwaretechnik längst Objektorientierung und Modularisierung Standard sind und sogar Agenten im Bereich des Internets eine weite Verbreitung haben, haben diese Paradigmen in der Anlagenautomatisierung kaum Einzug erhalten. Mit den Programmier- und Laufzeitumgebungen CoDeSys 3 [2] und TwinCAT 3 [3] bieten nur die Unternehmen 3S bzw. Beckhoff Automation unter den bedeutenden Softwareherstellern von IEC 61131-3-Umgebungen eine am

Markt verfügbare Lösung an, die einen objektorientierten Softwareentwurf auf Basis der Norm IEC 61131-3 ermöglicht. Die objektorientierte Erweiterung des etablierten Standards und damit auch die Möglichkeit weiterhin konventionell in IEC 61131-3 programmieren zu können, ist ein Hinweis darauf, wie wichtig industriellen Anwendern Kontinuität ist. Nichts desto trotz sind alle namhaften Hersteller von Steuerungssystemen in zahlreichen Forschungsprojekten an der Entwicklung zukünftiger Systeme beteiligt, die weit über die Objektorientierung hinaus, über serviceorientierte Systeme bis hin zu agentenorientierten Systemen reichen.

Beckhoff war als eines von zwei CoDeSys-Partnerunternehmen im Projekt „Steigerung der Effizienz und Qualität im Software-Engineering der Automatisierungstechnik für die Domäne des Maschinen- und Anlagenbaus“ beteiligt und arbeitete wesentlich an der Anforderungs- und Konzeptentwicklung der im Projekt realisierten Entwicklungswerkzeuge mit [4]. Im DFG Transferprojekt Konzeption, Realisierung und Evaluation einer werkzeugunterstützten Vorgehensweise für die Entwicklung von Agentensystemen in der Automatisierungstechnik unter Berücksichtigung der Usability (KREAagentuse) wurden ein neuer Ansatz für die Entwicklung und Realisierung für agentenbasierte Steuerungssysteme erforscht und u.a. eine Expertenbefragung zur Anforderungsbewertung durchgeführt. Der Ansatz von KREAagentuse fokussiert auf den Einsatz der Agenten in der Feldebene für technische Prozesse der Fertigungs- und der Verfahrenstechnik und auf marktüblichen Automatisierungsgeräten (Speicherprogrammierbare Steuerung - SPS).

Im Folgenden werden zunächst die Potentiale des gewählten Ansatzes und mögliche Anwendungsfälle diskutiert und im Anschluss auf die Ergebnisse der Expertenbefragung eingegangen.

## 3.2 Potentiale

In KREAagentuse wird die Steigerung der Verfügbarkeit eines Prozessautomatisierungssystems angestrebt. Dazu sind effiziente und durch Applikationsingenieure gut anwendbare Entwicklungsmethoden für das Engineering mit Werkzeugunterstützung bis zur Laufzeitumgebung notwendig. In der Domäne der Automatisierungstechnik ist es für die Akzeptanz in der Industrie entscheidend, dass neuartige Ansätze wie

### 3 Potentiale und Herausforderungen für die Steuerungsprogrammierung im Maschinen- und Anlagenbau von morgen

---

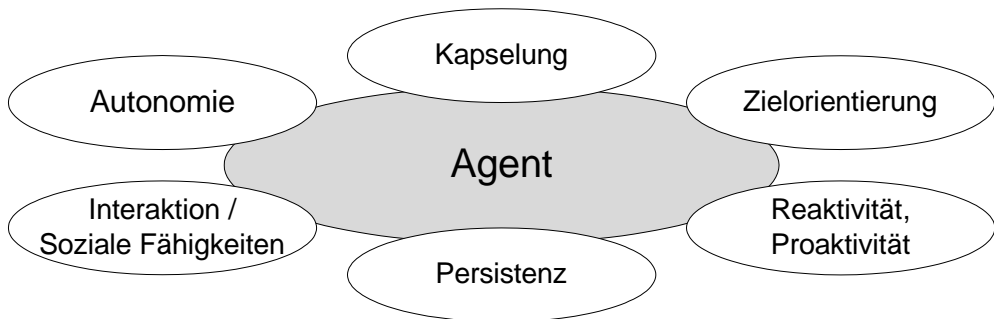


Abb. 3.1 – Eigenschaften von Agenten nach [5]

Agenten ohne Brüche in marktübliche Systeme integriert werden. Aufgrund der langen Lebensdauer der betriebenen Anlagen und deren Investitionsschutz wird diese Integration nur dann akzeptiert, wenn sie von einem auf dem Markt etablierten Systemhaus selbst unterstützt und weitergeführt wird, was Beckhoff als innovatives Unternehmen im Projekt KREAagentuse leistet. Die agentenorientierte Softwareentwicklung (AOSE) eignet sich besonders für die Entwicklung von dezentralen, komplexen Systemen mit vielfältiger Funktionalität und Interaktionen der einzelnen Systemelemente. Dabei wird ein System als eine Menge von autonomen Agenten betrachtet, die selbstständig innerhalb ihres Entscheidungsrahmens handeln und dabei vorgegebene Ziele verfolgen. In diesem Kontext verwenden wir die folgende Definition von Agenten.

**Definition:** (technischer) Agent nach VDI 2653 [6]:

Ein Agent ist eine abgrenzbare (Hardware-oder/und Software-) Einheit mit definierten Zielen. Ein Agent ist bestrebt, diese Ziele durch selbstständiges Verhalten zu erreichen und interagiert dabei mit seiner Umgebung und anderen Agenten. Agenten sind ein Modellierungskonzept zur Lösung von technischen Aufgabenstellungen unabhängig von einer bestimmten Realisierungsform [6].

In der agentenorientierten Denkweise wird eine Problemstellung mit Hilfe der Grundkonzepte Kapselung, Autonomie, Interaktion, Reaktivität, Zielorientierung, Proaktivität und Persistenz (Abb. 3.1) in einzelne Agenten abstrahiert, um so z.B. verteilte Prozesse beschreiben zu können, vgl. [7]. Die AOSE ist beispielsweise für Systeme geeignet, die eine logische Verteilung aufweisen, strukturellen Änderungen zur Laufzeit unterworfen sind oder komplexe Abläufe bzw. Verhalten aufweisen. Auf Basis eines agenten-

orientierten Lösungsansatzes können flexible, anpassungsfähige Softwaresysteme entwickelt werden, welche sowohl die Verteilung von Informationen, Aufgaben, Ressourcen oder Entscheidungsprozessen widerspiegeln als auch verschiedener Sichtweisen oder gegensätzliche Interessen der realen Problemstellung im Softwaresystem repräsentieren. Somit ist es bei der agentenorientierten Softwareentwicklung möglich, die gewünschte Flexibilität des Systems – bezüglich Struktur oder Verhalten – gezielt zu entwerfen. Der AOSE-Ansatz aus KREAagentuse stellt hierfür Konzepte, Methoden, Vorgehensweisen und Werkzeuge zur Verfügung [8]. Im Anschluss werden verschiedene potentielle Anwendungsszenarien für agentenbasierte Steuerungssysteme im Maschinen- und Anlagenbau aufgezeigt.

#### 3.2.2 Anwendungsszenarien

Wie in den vorherigen Kapiteln bereits erwähnt, besitzen Softwareagenten ein großes Potenzial für die Verbesserung der Steuerung von Anlagen. Im Folgenden wird dazu beispielhaft anhand von zwei unterschiedlichen Anwendungsszenarien der mögliche Einsatz von Agenten und ihr Nutzen erläutert.

##### **Szenario 1: Agent zur Kompensation von Sensorverschmutzung**

Für einen Prozess mit hoher Staub- oder Brüdenbelastung ist die Zuverlässigkeit von Binärsensoren, wie Lichtschranken, problematisch, da diese nach kurzer Zeit so stark verschmutzen, dass sie ihrer Detektorfunktion nicht mehr nachkommen. Um den dadurch entstehenden Ausschuss bzw. die Stillstandzeiten der Anlage zu vermeiden, kann mit einem agentenbasierten Ansatz angesetzt werden. Binärsensoren, welche ab einem festgelegten Schwellwert ihr Signal ändern, könnten z.B. durch Sensoragenten in Kombination mit je einem entsprechenden Analogsensor ersetzt werden oder das analoge Signal der Lichtschranke direkt analysiert werden. Durch die Analogsensoren liegt immer ein Signal an, welches vom Sensoragenten verarbeitet werden kann. Durch seine Wissensbasis, die unter anderem verschiedene Verschmutzungsgrade sowie Alterungsprozesse berücksichtigen kann, ist es dem Agenten nun möglich zu unterscheiden, ob eine Signaländerung von einem zu detektierten Material oder nur von der zunehmenden Verschmutzung/Alterung herrührt und er kann die Intensitätsänderung des Signals kompensieren. Ist der Verschmutzungsgrad zu hoch oder die Lebensdauer des Sensors zu 90% erreicht, könnte bspw. eine Warnung für den betroffenen Sensor ausgegeben werden. Somit wäre eine

### 3 Potentiale und Herausforderungen für die Steuerungsprogrammierung im Maschinen- und Anlagenbau von morgen

---

erhöhte Genauigkeit bei der Bearbeitung und eine einfacher zu koordinierende Wartung/Reinigung der Anlage zu erreichen.

#### **Szenario 2: Agent für Plug&Produce zur Zusammenstellung neuer Produktionseinheiten**

Das zweite Szenario beschreibt die Produktion einer individuell geplanten Küche, wobei während der Produktion Anpassungen des Produktionssystems erforderlich werden. Hierfür sind flexible, adaptive, sich selbst organisierende Produktionseinheiten im Bereich der Verfahrens- und Fertigungstechnik nötig. Die Produktionseinheiten können durch den Einsatz von AOSE zu intelligenten autonomen Objekten werden, die über ein gemeinsames Netzwerk Organisationsaufgaben selbst übernehmen und untereinander regeln.

Die Produktionseinheiten sind intelligent. Sie kennen ihre Funktionalität, die Rahmenbedingungen unter denen sie zuverlässig funktionieren, ihren Standort, ihren Verbrauch an Ressourcen, entstehende Betriebskosten und ihre Konfigurationsmöglichkeiten - d.h. mit welchen anderen Produktionseinheiten sie verkoppelt werden können. Zudem haben sie einen Überblick über ihre aktuelle und geplante Auslastung. Die Produktionseinheiten bieten ihre Produktionsleistungen auf der Anlagenebene an, erkennen Produktionsleistungen auch für sie bisher unbekannte Produktionseinheiten und kommunizieren miteinander. Erhält die Anlage eine Produktionsanfrage (z.B. Produktion einer kundenindividuellen Küche) prüft jede Produktionseinheit ob sie hierzu einen Beitrag leisten kann. Die relevanten Produktionseinheiten melden sich und verhandeln über Rahmenbedingungen unter denen sie gemeinsam produzieren können. Diese Rahmenbedingungen können sich je nach Bedarf z.B. auf Schnittstellen, Datenaustausch, Logistik, einzuhaltende Richtlinien, Verfügbarkeit, Ressourcenbedarf und auf Standortfragen erstrecken. Das Ergebnis der Verhandlungen ist eine optimale Konfiguration zur Erfüllung des Produktionsauftrags, wobei die Produktionseinheiten die Verhandlungen komplett über einen Planungssagenten durchführen können. Dabei werden die für die Planung grundlegenden Informationen wie Kosten, Verfügbarkeit und Produktionszeiten für die Verhandlung zur Verfügung gestellt.

Wie so ein solches System in der Produktion arbeiten könnte wird im Folgenden aufgezeigt.

Werden beispielsweise bei der Qualitätskontrolle Kratzer auf der Oberfläche der Abschlussleiste erkannt, muss sie ausgetauscht werden. Die Abschlussleiste ist ein Kunststoffprofil mit Holzdekor und doppelseitigem Klebeband. Die Profile werden in einem kontinuierlichen Kombinationsverfahren aus Heißextrusion, Beschichtung und Aufbringung des Klebebands hergestellt. Die beteiligten Produktionseinheiten bzw. deren Prozessagenten erhalten die für die Optimierung der Produktqualität notwendigen Prozessdiagnosedaten der Qualitätskontrolle des Küchenherstellers, analysieren diese anhand ihrer Wissensbasis und werten sie aus. Basierend auf diesen Auswertungen passen die Prozessagenten die Prozessparameter (z.B. Temperatur beim Extrudieren, Anpressdruck beim Beschichten, Geschwindigkeit der Prozessführung) so an, dass der Verschleiß kompensiert wird und derartige Beschädigungen nicht mehr auftreten.

Unerwartet meldet nun die Hochleistungssäge, dass sie in ca. 2 Stunden ausfallen wird. Die Einzelteile der Möbel können dann nicht mehr zugeschnitten werden. Das Produktionssystem meldet sich auf Anlagenebene und fragt eine gleichwertige Hochleistungssäge an. Keine gleichwertige Hochleistungssäge ist frei, stattdessen melden sich zwei Sägen mit geringerer Leistung. Eigenständig prüft das Produktionssystem den Einsatz der zwei vorgeschlagenen Sägen anstelle der einen Hochleistungssäge und kommt zu dem Ergebnis, dass die beiden Sägen gemeinsam die geforderte Leistung erbringen. Das Produktionssystem tauscht eigenständig während des Betriebs die Hochleistungssäge durch die zwei Sägen mit geringerer Leistung aus und produziert ohne Unterbrechung und ohne Effizienz- sowie Qualitätseinbußen die Möbelteile.

Wie wichtig die u.A. in den Szenarien beschriebenen Eigenschaften im Maschinenbau sind, wird nun an Hand der im Rahmen von KREAagentuse durchgeführten Expertenevaluation aufgezeigt.

### **3.3 Expertenevaluation und Interpretation**

Die Expertenevaluation wurde als geführtes Interview mit sieben ausgewählten Kunden aus dem Bereich des Maschinenbaus der Beckhoff GmbH im Rahmen von KREAagentuse durchgeführt. Die Bedeutung der einzelnen Kriterien wurde den verschiedenen Anforderungen an moderne Automatisierungskonzepte zugeordnet und mit sehr wichtig bis zu unwichtig bewertet. Zudem konnten eigene Kommentare zu den Fragen formuliert werden. Im Fokus sind hierbei die abgeleiteten Anforderungen aus den Projekten



### 3 Potentiale und Herausforderungen für die Steuerungsprogrammierung im Maschinen- und Anlagenbau von morgen

„Evolvable Ultra-Precision Assembly SystemS“ (EUPASS) [9] und „Agenten für flexible und verlässliche eingebettete Echtzeitsysteme“ (AVE) [6] bezüglich der Steigerung der Verfügbarkeit und die Wissensbasis dazu. Die Anforderungen betreffen alle Aspekte des Konzeptes, also insbesondere auch Anforderungen an die Modellinhalte eines Agentenentwurfes, an seine Darstellungen und Werkzeuganforderungen. Im Folgenden wird ein Ausschnitt der Ergebnisse in Abb. 3.2 bzw. Abb. 3.3 gezeigt und diskutiert.

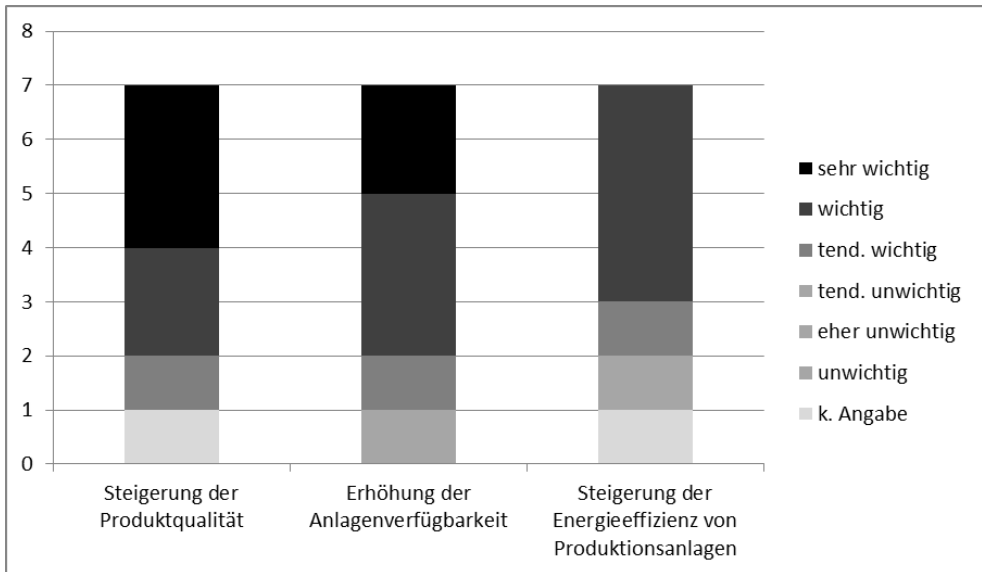


Abb. 3.2 – Anforderungen an die Unterstützung bei Optimierungen

Die Experten bewerteten die Steigerung der Produktqualität als wichtigste abgefragte Anforderung (siehe Abb. 3.2). Ihre Bedeutung überwiegt knapp die der Erhöhung der Anlagenverfügbarkeit. Die Steigerung der Energieeffizienz wird als wichtig, aber nicht sehr wichtig bewertet.

Um diese Anforderungen zu erfüllen, wurden zudem die Bedeutungen von Modellinhalten und Werkzeuganforderungen evaluiert (siehe Abb. 3.3). Hier maßen die Experten der Online-Debugging Funktion die höchste Bedeutung zu, knapp gefolgt von der Anwendung von Konzepten für verteilte Systeme und der Unterstützung bei der modularen Strukturierung der Steuerungssoftware. Danach folgen eine einfache Beschreibung der Einschränkungen und eine variable Möglichkeit zur Kopplung einzelner

### 3 Potentiale und Herausforderungen für die Steuerungsprogrammierung im Maschinen- und Anlagenbau von morgen

Module der Steuerungssoftware, die von der Mehrheitlich wichtig bis sehr wichtig eingestuft wurden.

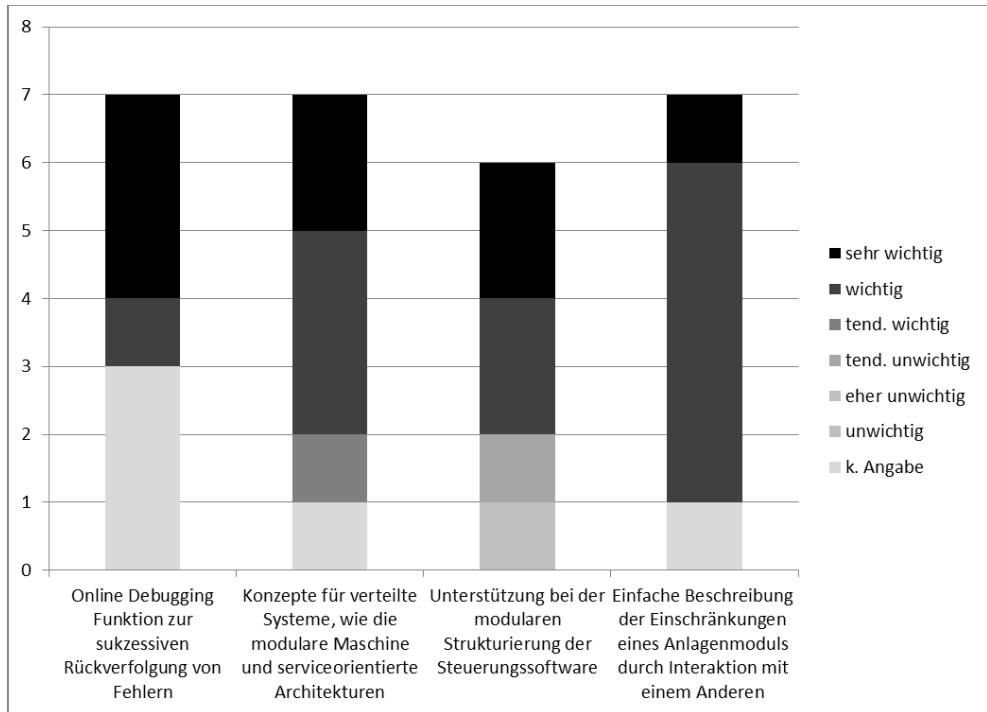


Abb. 3.3 – Anforderungen an Features des Entwicklungswerkzeugs

Vergleichsweise uneinig waren sich die Experten über die Kompensation von Fehlern zur Laufzeit durch vorgegebene Funktionsbausteine bzw. Modelle. Hier wurde vor allem bei kontinuierlichen Prozessen, die bei einem Ausfall größere Probleme zu erwarten haben, eine wichtige Rolle zugesprochen. Für andere Prozesse spielte die Kompensation zur Laufzeit nur eine untergeordnete Spezialrolle, weil generell in der Entwurfsphase festgelegt wird, wie auf Sensor-Ausfälle reagiert werden soll. Allgemein werden Sensorausfälle nicht kompensiert, sondern nur in ausgewählten Fällen (z.B. Bandwaage mit Gewichtsermittlung rechts/links) wird ein solcher Fehler unter bestimmten Bedingungen (z.B. Sicherheitsrelevanz) kompensiert. In ihrer Gesamtheit bestätigen die Ergebnisse der Experten-evaluation die Kundenrelevanz agentenbasierter Systeme und Ansätze im

### 3 Potentiale und Herausforderungen für die Steuerungsprogrammierung im Maschinen- und Anlagenbau von morgen

---

Maschinen- und Anlagenbau, wie für die Anwendungsszenarien aufgezeigt und im Projekt KREAagentuse entwickelt [8] wurden.

Neben der Bedeutung der Anforderungen wurden auch Herausforderungen bei der Realisierung neuer Ansätze erhoben. Hier wurde von Seite der Befragten erläutert, dass aktuelle Systeme eine eigene Diagnoseauswertung und herstellereigenspezifische Benutzerführung benötigen. Für die Realisierung eines durchgängigen Debugging Konzepts in Steuerungssystemen von Morgen sehen sie eine Standardisierung über alle Baugruppen aller Hersteller als Voraussetzung. Beklagt wurde, dass aktuell Web-Diagnosen der Geräte z. B. kaum nutzbar sind, weil jedes Gerät / jeder Hersteller andere Konzepte verfolgt. Das ist für die Endanwender ihrer Aussage nach nicht zumutbar. Hier könnte ein Agent die Integration übernehmen.

Bei den Konzepten für verteilte Systeme und variable Möglichkeiten zur Kopplung einzelner Module der Steuerungssoftware zu unterstützen, wurden Defizite heutiger Methoden bei der Strukturierung des Programmcodes angemerkt. Hier sei eine Herausforderung für die Zukunft zu sehen, da neue Technologien und Möglichkeiten ihrer Meinung nach nur effektiv genutzt werden können, wenn deutliche Fortschritte in diesem Punkt erzielt werden. Die Modularität die früher mit eigenständigen Geräten (in Hardware) möglich war, muss in Zukunft auch mit vielen unabhängigen Applikationsmodulen auf einer Hardwareplattform möglich werden. Für die Unterstützung bei der modularen Strukturierung der Steuerungssoftware wurden integrierte Tools zum Softwaredesign, wie z.B. Tools der UML oder SysML gefordert, da heute nur durch festgelegte Designregeln für den Softwareentwickler modulare Strukturierung gefördert werde. Wichtig sei bei neuen, integrierten Tools eine durchgängige Unterstützung des gesamten Lebenszyklus einer Software. Beispielsweise muss aus einem UML/SysML-Diagramm Code generiert werden können der dann auch online im UML/SysML-Diagramm gedebugged und gemonitort werden kann.

Die angeführten Herausforderungen wurden soweit möglich in dem Projekt KREAagentuse berücksichtigt und die Werkzeugunterstützung bereits prototypisch realisiert [8].

### 3.4 Zusammenfassung und Ausblick

In diesem Artikel wurden Potentiale und Herausforderungen für neue, agentenbasierte Steuerungsprogramme für den Maschinen- und Anlagenbau,

an Hand von Szenarien und einer Expertenevaluation im Rahmen des gemeinsamen Projekts KREAagentuse behandelt. Zur Entwicklung eines methodischen, toolbasierten Vorgehens zur Entwicklung agentenbasierter Steuerungsprogramme wurden Kundenanforderungen und Herausforderungen erhoben, welche in die Projektarbeit einfließen. Die Korrelation zwischen den Kundenanforderungen und den identifizierten Potentialen für AOSE weisen auf eine steigende Bedeutung solcher Systeme in der Zukunft hin. Um hier detailliertere Aussagen zu der Usability des in KREAagentuse entwickelten Ansatzes treffen zu können, planen die Autoren abschließende Usability-Untersuchungen im Labor.

### 3.5 Danksagung

Die Autoren bedanken sich für die finanzielle Unterstützung der Deutschen Forschungsgemeinschaft (DFG) für das Projekt „Konzeption, Realisierung und Evaluation einer werkzeugunterstützten Vorgehensweise für die Entwicklung von Agentensystemen in der Automatisierungstechnik unter Berücksichtigung der Usability“ (KREAagentuse, VO 937/8-1).

### 3.6 Referenzen

- [1] Harrison, R.; Colombo, A. W.: Service-oriented architectures for collaborative automation. In: Industrial Electronics Society 31th Annual Conference of the IEEE Industrial Electronics Society, 2005.
- [2] Controller Development System (CoDeSys): [www.3s-software.com](http://www.3s-software.com).
- [3] TwinCAT 3 eXtended Automation Technology: <http://www.beckhoff.com>.
- [4] Witsch, D.; Vogel-Heuser, B.: UML für die Steuerungsprogrammierung. In: *PC- Control*, 2004.
- [5] Jennings, N. R. et al.: A Roadmap of Agent Research and Development. In: *Autonomous Agents and Multi-Agent Systems*, Ausgabe 1, Heft 1, Springer Niederlande, 1998, S. 7-38.
- [6] Agentensysteme in der Automatisierungstechnik – Grundlagen, VDI/VDE 2653 Blatt 1, 2010.

### 3 Potentiale und Herausforderungen für die Steuerungsprogrammierung im Maschinen- und Anlagenbau von morgen

---

- [7] Wagner, T. et al.: Softwareagenten - Einführung und Überblick über eine alternative Art der Softwareentwicklung, Teil I: Agentenorientierte Softwareentwicklung. In: *Automatisierungstechnische Praxis (atp)*, Ausgabe 45, Heft 10, 2003.
- [8] Frank, U. et al.: Real-time capable software agents on IEC 61131 systems –Developing a tool supported method. In: *IFAC World Congress 2011*, Mailand, Italien, 2011, akzeptierter Beitrag.
- [9] Papenfort, J.; Hoppe, G.: Evolvable skills for assembly systems. In: *Automation Technology in Practice, atp international*, Ausgabe 4, Heft 3, 2006, S. 27-31.

## 4 Verfügbarkeitssteigerung von Produktionsanlagen mit Softwareagenten – Entwicklung eines durchgängigen Werkzeugs

Daniel Schütz, Birgit Vogel-Heuser

*Lehrstuhl für Automatisierung und Informationssysteme(AIS), TU München*

**Zusammenfassung:** In diesem Beitrag wird ein Ansatz vorgestellt, mit dem durch die Verwendung von Softwareagenten Verfügbarkeit von sowohl verfahrenstechnischen als auch fertigungstechnischen Produktionsanlagen zu steigern. Der Ansatz baut auf Vorarbeiten des Lehrstuhls AIS auf, Softwareagenten, die über Softsensoren verfügen, direkt auf Speicherprogrammierbaren Steuerungen (SPS) in den Sprachen der IEC 61131-3 zu implementieren. Dazu wurde im Rahmen des Transferprojekts KREAagentuse ein prototypisches Werkzeug sowie eine angepasste Vorgehensweise entwickelt, die die modellbasierte Entwicklung und automatische Implementierung der Softwareagenten in einer marktüblichen Entwicklungsumgebung für IEC-Steuerungscode ermöglichen.

### 4.1 Motivation und Einleitung

Industrielle Produktionstechnik wird in der Fachliteratur untergliedert in Fertigungstechnik und Verfahrenstechnik. Während die Fertigungstechnik sich durch überwiegend diskrete Prozesse auszeichnet, kann die die Verfahrenstechnik weiter in kontinuierliche Prozesse und Batch-Prozesse unterteilt werden [1]. Sowohl in der Fertigungstechnik als auch in der Verfahrenstechnik steigen durch eine zunehmende Verkürzung der Produktlebenszyklen die Anforderungen bezüglich der Flexibilität der Produktionssysteme im Hinblick auf Änderungen ihres technischen Aufbaus sowie veränderte zu realisierende Produktionsprozesse und herzustellenden Produkten und Produktvarianten.

Zusätzlich werden an Produktionssysteme hohe Anforderungen in Bezug auf ihre Robustheit gegenüber Störungen und Ausfällen gestellt. Diese Anforderungen haben verschiedene Gründe, die zum Teil spezifisch für die jeweilige Prozessklasse sind, da die Folgen von Ausfällen und ungeplanten Unterbrechungen der Produktion verschiedene Auswirkungen in den

#### 4 Verfügbarkeitssteigerung von Produktionsanlagen mit Softwareagenten – Entwicklung eines durchgängigen Werkzeugs

---

Prozessklassen haben [2]. So sind Unterbrechungen des Produktionsprozesses in der Fertigungstechnik unerwünscht, da ein Maschinenstillstand Kosten verursacht, jedoch zumeist unkritisch, da die Produkte innerhalb der Anlage während der Unterbrechung stabil bleiben. In der Verfahrenstechnik jedoch ist es häufig der Fall, dass das Prozessmaterial (Produkt) seine Reaktion auch während Unterbrechungen des Produktionsprozesses fortsetzen kann, wodurch Gefährdungen entstehen können. Zudem sind schnelle kontinuierliche Fließprozesse schwer zu unterbrechen und stellen harte Echtzeitanforderungen an das Automatisierungssystem einer Anlage [2].

Eine der häufigsten Fehlerursachen und Gründe für Stillstände von fertigungstechnischen wie auch verfahrenstechnischen Produktionsanlagen sind defekte oder fehlerhaft arbeitende Sensoren. Nach Wannagat et al. existieren für die verschiedenen Prozessklassen drei unterschiedliche Strategien auf solche Ausfälle zu reagieren [2, 3]:

Das *Herunterfahren* eines Produktionsprozesses und die Durchführung von Wartungsarbeiten (Ersetzen defekter Sensoren) ist in der Fertigungstechnik eine kostenintensive aber mögliche Strategie. Im Gegensatz dazu kann diese Strategie innerhalb der Verfahrenstechnik ungeeignet ist, da die Prozesse dieser Klasse nicht immer gefahrlos Heruntergefahren werden können. Eine Produktionsanlage im Falle eines Sensorausfalls in einen *stabilen Zustand* zu fahren kann abhängig vom jeweiligen Prozess ebenfalls eine Möglichkeit sein, um weitere Funktionsstörungen zu vermeiden und gegebenenfalls Wartungsarbeiten vorzunehmen. Die Verwendung von *redundanten Geräten oder Informationen* und eine darauf basierende dynamische Rekonfiguration zur Laufzeit ist ein dritter Weg, um auf Sensorausfälle reagieren, ohne den jeweiligen Prozess zwingend abbrechen, herunterfahren oder in einen sicheren Zustand überführen zu müssen. Die Verwendung von redundanten Geräten würde jedoch die Kosten für Sensorik einer Anlage verdoppeln und es kann in einigen Fällen aufgrund von räumlichen Einschränkungen technisch nicht möglich sein redundante Sensoren zu installieren. Davon ausgehend wurde von Wannagat et al. ein Ansatz entwickelt, die Verfügbarkeit von Produktionsanlagen in der Fertigungsautomatisierung sowie in der Prozessautomatisierung mittels redundanter Informationen, welche innerhalb der Steuerungssoftware durch Software berechnete Sensorwerte bereitgestellt werden, zu erhöhen [2, 3, 4]. Dieser Ansatz integriert das Konzept der Redundanz durch Softsensoren in einen Ansatz für Softwareagenten, die auf handelsüblichen Speicherprogrammierbaren

Steuerungen (SPS) implementiert werden können. Die Evaluation des Ansatzes an mehreren verschiedenen Demonstratoren zeigten, dass diese Softwareagenten in der Lage sind, Fehlfunktionen von Sensoren selbstständig zu erkennen und während des laufenden Produktionsprozesses durch in Software berechnete Werte in Echtzeit zu kompensieren.

### 4.2 Ausgangssituation und Zielstellung

Die eigenen Vorarbeiten entstanden im Rahmen des von der Deutschen Forschungsgemeinschaft (DFG) geförderten Projekts „Agenten für flexible und verlässliche, eingebettete Echtzeitsysteme“ (AVE) [5] und umfassten sowohl ein Framework für die Implementierung von Softwareagenten in den Sprachen der IEC 61131-3 als auch bereits einzelne Konzepte für Beschreibungsformen und Modellierungsansätze für eine automatische, modellbasierte Implementierung der Softwareagenten. Die Modellierungsansätze und Beschreibungsformen basierten dabei auf den Notationen der Systems Modeling Language (SysML), einer objektorientierten Modellierungssprache, die auf die Beschreibung von technischen Systemen angepasst ist. Innerhalb des auf dem Projekt AVE aufbauenden Transferprojekts KREAagentuse [6] wurden zusammen mit dem Transferunternehmen Beckhoff Automation GmbH diese einzelnen Beschreibungsformen zu einem einheitlichen, konsistenten Modell eines Agentensystems integriert, aus dem automatisch der Steuerungscode für eine Speicherprogrammierbare Steuerung generiert werden kann. In Vorarbeiten des Lehrstuhls AIS im Bereich der Modellierung von Produktionsanlagen mit Hilfe der SysML konnte ein großes Potenzial dieser Modellierungssprache in Bezug auf die Integration einer Anforderungsmodellierung [7] sowie einer parametrischen Modellierung von Anlagenmodulen [8] festgestellt werden. Die Stärken der SysML wurden während der Bearbeitung des Projekts KREAagentuse sowie paralleler Projekte für ein breiteres Feld von Anwendungsfällen, wie zum Beispiel der energetischen Modellierung von Anlagen [9, 10] weiter erschlossen. Im Rahmen des Projekts KREAagentuse wurde auf Basis der im Vorgängerprojekt AVE und weiteren Arbeiten des Lehrstuhls entwickelten Beschreibungsformen, wie zum Beispiel einer gerichteten Graphen für analytische Parameterzusammenhänge eines Anlagenmoduls, eine Werkzeugunterstützung entwickelt und in eine marktübliche Entwicklungsumgebung für Steuerungssoftware nach IEC 61131-3 integriert. Das Werkzeug beinhaltet die entsprechenden Editoren zum Erstellen der Agentenmodelle, welche innerhalb der Entwicklungsumgebung direkt auf



Elemente der objektorientierten Erweiterung der IEC 61131-3 werden. Dabei wird die objektorientierte Erweiterung der IEC 61131-3 verwendet, wie sie in CoDeSys V3 der Firma 3S und TwinCAT 3 der Firma Beckhoff Automation GmbH umgesetzt wird [11, 12]. Das entwickelte Werkzeug basiert auf einem Plugin für CoDeSys V3 zur Erstellung von Steuerungscode durch verschiedene Diagramme der Unified Modeling Language (UML), das ebenfalls in Vorarbeiten des Lehrstuhls AIS entwickelt wurde [13]. Um den Applikationsingenieur weiter bei der Entwicklung von Softwareagenten für Speicherprogrammierbare Steuerungen zu unterstützen, wurde zusätzlich ein Vorgehensmodell entwickelt, das an die Struktur der Agentensysteme auf der einen Seite und an das Vorgehen bei der Entwicklung von konventionellem Steuerungscode auf der anderen Seite angepasst ist. Verschiedene domänenspezifische Sichten der Architektur trennen dabei die Entwicklung des Anlagenprozesses, z.B. eines Batch-Rezepts, von der Programmierung der einzelnen Anlagenfunktionen und der Anbindung der Software an die Sensorik und Aktorik einer Anlage. Damit wird die Komplexität der Entwicklung einer Anlagensteuerung in Abschnitte aufgeteilt, die einzelnen Gewerken zugeordnet werden können und das interdisziplinäre Arbeiten in einem gemeinsamen Softwareprojekt ermöglichen.

### 4.3 Domänenspezifische Sichten eines Agentensystems

Der verwendete Modellierungsansatz für Agentensysteme basiert auf den Diagrammen und Notationen der Systems Modeling Language (SysML). Dabei ist eine Aufteilung des Systemmodells in vier verschiedene spezifische Sichten vorgesehen, die getrennt voneinander entwickelt werden können (siehe Abb. 4.1) [14, 15]. Das aktuelle Kapitel gibt eine Übersicht über diese verschiedenen Sichten und deren Zusammenhänge. Im Anwendungsbeispiel in Kapitel 4.5 werden die Struktur eines Agentensystems, die verschiedenen Sichten sowie die darin jeweils verwendeten Diagramme und Notationen näher erläutert und dargestellt wie die einzelnen Modellteile auf Elemente der objektorientierten IEC 61131-3 abgebildet werden können.

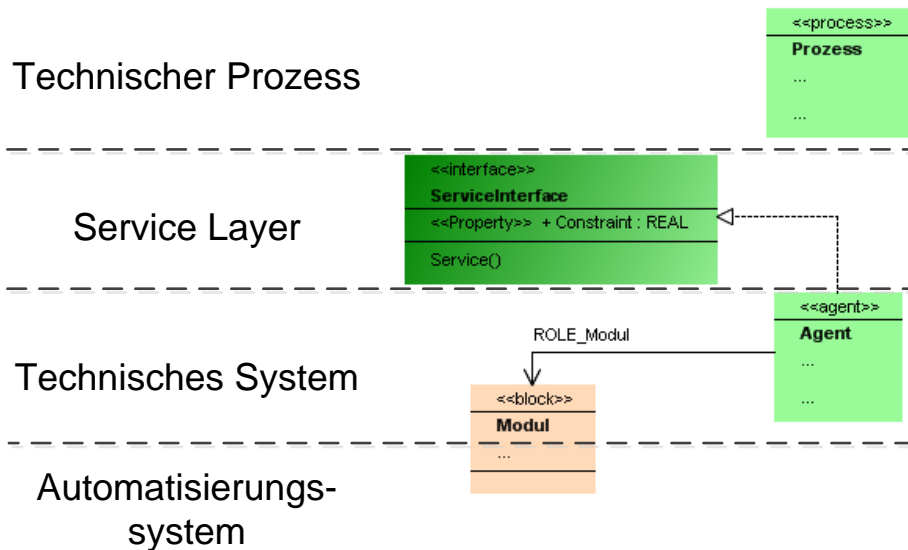


Abb. 4.1 – Struktur und Sichten eines Agentensystems [14, 15]

Die ersten drei Sichten eines Agentensystems „Technischer Prozess“, „Technisches System“ und „Automatisierungssystem“ sind an die jeweilige Definition der Begriffe in [16] abgelehnt. So beschreibt die Sicht Technisches System die maschinenbaulichen Komponenten einer Anlage, die Sicht Automatisierungssystem die Automatisierungskomponenten, wie zum Beispiel Sensoren und Aktoren einer Anlage, und die Sicht Technischer Prozess die Fertigungs- oder Verfahrensschritte zur Herstellung eines Produkts. Eine eingefügte Sicht „Service Layer“ verbindet die Sicht Technischer Prozess mit den Sichten Technisches System und Automatisierungssystem (siehe Abb. 4.1). Das Service Layer enthält alle Prozessfunktionen des Technischen Systems als Service-Interface, in denen Aktivitäten (*Activity*) von Softwareagenten für die Implementierung eines Produktionsprozesses angeboten werden. Für die Beschreibung der Service-Interfaces wird dabei das SysML-Element *Interface* verwendet. Technische Prozesse werden durch Blöcke des speziellen Typs *Process* gekapselt. Einfache Blöcke (*Block*) repräsentieren die Module des Technischen Systems einer Produktionsanlage, das heißt Einheiten des technischen Aufbaus einer Anlage. Gesteuert werden die Anlagenmodule durch die Softwareagenten (*Agent*), die zu diesem Zweck eine *Referenz* (Pointer) auf die Blöcke der Anlagenmodule besitzen. Durch diese Referenz sind die Softwareagenten in der Lage die

unterlagerten Funktionen der von ihnen gesteuerten Module aufzurufen und so die Prozessfunktion durch die Implementierung eines Service-Interfaces anzubieten.

Die Beschreibung des *Technischen Prozesses* wird als Verhalten eines *Process-Blocks* innerhalb eines Aktivitätsdiagramms der SysML durch den sequenziellen oder parallelen Aufruf von Aktivitäten (*Activity*) beschrieben, die von den Agenten als Prozessfunktionen auf dem Service Layer angeboten werden. Als zusätzliche Funktionalität des Ansatzes können Anforderungen des Prozesses an einzelnen Prozessschritten als *Requirements* an die jeweiligen Aufrufe der Aktivitäten gebunden werden. Da die Aktivitäten dabei Aufrufe von auf dem Service Layer angebotenen Service-Interfaces darstellen, kann nachvollzogen werden, mit welchem Service-Interface eine Anforderung erfüllt werden muss.

Die auf dem *Service Layer* des Agentensystems beschriebenen *Interfaces* stellen in diesem Ansatz wie im Paradigma der Objektorientierung eine abstrakte Beschreibung von Struktur und Verhalten dar, die von einem Block des Systems übernommen bzw. implementiert werden kann. Die Prozessfunktionen, modelliert durch Aktivitäten, werden innerhalb eines *Interfaces* lediglich abstrakt durch die Deklaration eines Namens und der ein- und ausgehenden Parametern beschrieben. Darüber hinaus fordert der entwickelte Modellierungsansatz, dass das *Interface*, über das eine mit Prozessanforderungen belegte Prozessfunktion (Verknüpfung eines *Requirements* mit einer Aktivität in der Prozessbeschreibung) angeboten wird, eine entsprechende Eigenschaft (*Variable*) enthält, mit der die Erfüllung der Anforderung ausgedrückt bzw. nachgewiesen werden kann. Existieren zum Beispiel Anforderungen an einen Prozessschritt, dass bei dessen Ausführung ein bestimmter Temperaturbereich einzuhalten ist, so muss das *Interface* in dem diese Prozessfunktion angeboten wird eine *Variable* enthalten, in welcher der einzuhaltende Temperaturbereich angegeben wird. Mit der Implementierung eines Service-Interfaces implementiert ein Agent sämtliche Eigenschaften, die als Teil des Service-Interfaces mit der Funktion verknüpft sind. Durch die aus den Anforderungen abgeleiteten Variablen, die ein Agent zusammen mit einem *Interface* implementiert, müssen bei der Implementierung eines Agenten auch explizit die Prozessanforderungen berücksichtigt werden. Das Service Layer stellt somit eine bidirektionale Schnittstelle zwischen dem Technischen System und dem Technischen Prozess dar, auf die von beiden Seiten aus zugegriffen werden kann. Während aus dem Technischen System

die Prozessfunktionen und ihre mögliche Parametrisierung angeboten werden kann, können vom Technischen Prozess aus konkrete Anforderungen auf diese Funktionen gelegt werden. Durch diese erweiterte Funktionalität des Service Layers wird eine getrennte Entwicklung von Prozessimplementierung und Anlagensteuerung auch durch verschiedene Personen (Applikationsingenieure) effektiv unterstützt.

Das *Technische System* beschreibt die Struktur, Parameter und das Verhalten einzelner Anlagenmodule durch Dekomposition ausgehend von den Modulen, die durch Agenten gesteuert werden. Dabei handelt es sich um eine klassische, hierarchische Zerlegung von Steuerungssoftware in einzelne Bausteine, wie sie bereits in der konventionellen Entwicklung von IEC 61131-3 Steuerungscode angewendet wird. Unterstützt wird diese durch die objektorientierten Erweiterungen der IEC 61131-3 und des UML-Plugins von CoDeSys V3 bzw. TwinCAT 3. Im Unterschied zu diesen werden aber nicht *Klassen* der UML als Repräsentanten für Funktionsbausteine verwendet, sondern das SysML-Element *Block*. Darüber hinaus existieren zusätzliche Diagramme, die zur modellbasierten Implementierung der Steuerungslogik verwendet werden können. Insbesondere das Parameterdiagramm ist hier von hoher Relevanz, da in diesem die Bewertung des Technischen Systems hinsichtlich der Anforderungen des Technischen Prozesses modelliert bzw. implementiert werden kann.

Das *Automatisierungssystem* eines Anlagenmoduls bzw. eines Agenten beschreibt die Sensoren und Stellglieder des jeweiligen technischen Moduls. Da Sensoren und Stellglieder ebenfalls technische Komponenten darstellen, werden sie bereits innerhalb des Technischen Systems bei der hierarchischen Strukturierung der Steuerungssoftware eingefügt, sodass die notwendigen Informationen bezüglich der real verbauten Sensoren und Aktoren aus dieser Sicht übernommen werden können. Die Erweiterung der Sensorik durch zusätzliche Softsensoren erfolgt durch die Modellierung der analytischen Zusammenhänge zwischen Sensorwerten innerhalb eines speziellen Parameterdiagramms. Die Semantik dieses Parameterdiagramms ist angelehnt an die des gerichteten Graphen [2, 3, 4], den Wannagat et al. zum Zweck der Modellierung von Zusammenhängen von Sensorwerten entwickelt haben. Aus diesem Parameterdiagramm können alle notwendigen Informationen für die Diagnose und Kompensation von Sensorfehlern durch einen Softwareagenten generiert werden.

#### 4.4 Vorgehen bei der Entwicklung von Agentensystemen

Das Vorgehensmodell für die Erstellung eines Agentensystems entsprechend der entwickelten Architektur unterteilt sich in sieben verschiedene Schritte und kann – wie auch die Architektur selbst – auf die verschiedenen Sichten des Agentensystems abgebildet werden (siehe Abb. 4.2) [14]. Es besitzt zwei Einstiegspunkte, die abhängig davon gewählt werden können, ob ein vollständig neues System entwickelt (A) oder lediglich ein weiterer Prozess in einem bereits entwickelten System implementiert (B) werden soll.

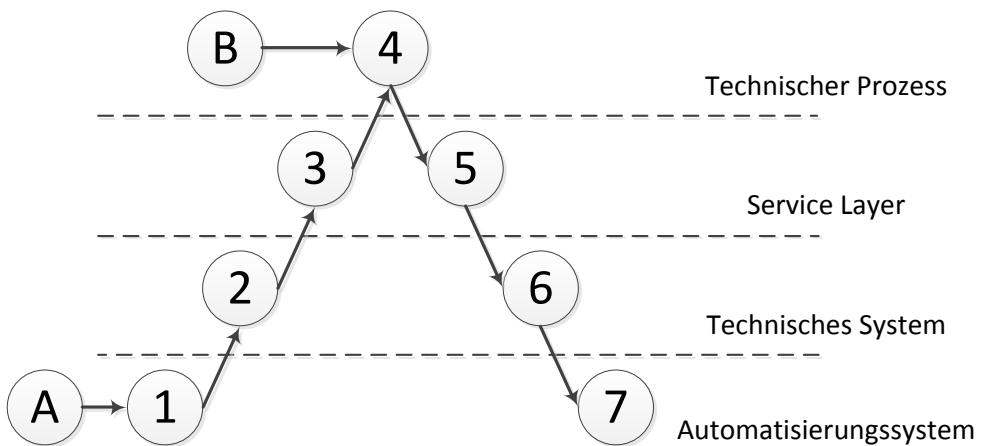


Abb. 4.2 – Vorgehensmodell zur Systementwicklung [14]

Die einzelnen Schritte des Vorgehensmodells lassen sich in drei übergeordnete Abschnitte einteilen:

- Komposition der Steuerungsagenten (Schritte 1 bis 3)
- Beschreibung des Prozesses und der Anforderungen (Schritt 4, 5)
- Systemeinschränkungen und Softsensoren (Schritt 6, 7)

Durch die verschiedenen Einstiegspunkte in das Vorgehensmodell und die Aufteilung in übergeordnete Abschnitte ist es insofern skalierbar, da nur die für den jeweiligen Anwendungsfall relevanten Teile der Architektur und der Softwareagenten erstellt werden müssen.

In den Schritten 1 bis 3 (siehe Abb. 4.2) werden ausgehend von Standardbausteinen für Sensoren und Aktoren (Automatisierungssystem) die Softwarebausteine für die Komponenten des Technischen Systems hierarchisch zusammengestellt bis hin zu einer Hierarchieebene, die durch Agenten gesteuert eine Prozessfunktion als Aktivität in einem Interface auf dem Service Layer bereitstellt. Sind diese Schritte für alle Module einer Anlage abgeschlossen, können auf der Grundlage der Softwareagenten und deren Prozessfunktionen verschiedene Prozesse implementiert werden, ohne in das Agentensystem und die Implementierung der Prozessfunktionen selbst eingreifen zu müssen.

In den Schritten 4 und 5 (siehe Abb. 4.2) wird der Technische Prozess (der Produktionsprozess) durch ein Aktivitätsdiagramm beschrieben. Mit der Verknüpfung einer Anforderung an eine Prozessfunktion wird das Interface, über das diese Funktion bereitgestellt wird, und damit auch der ausführende Softwareagent, um die textuelle Beschreibung dieser Anforderung sowie einem charakteristischen Parameter erweitert. Dieses Feature des Ansatzes ermöglicht es dem Service Layer eine Schnittstelle bei dem Softwareentwicklungsprozess darzustellen, die nicht nur von einer Sicht aus bereitgestellt wird, sondern als Vermittlerschicht zwischen der Entwicklung eines Anlagenprozesses und der Implementierung der einzelnen Funktionen dienen kann.

In den Schritten 6 und 7 wird zum einen das Technische System eines Moduls hinsichtlich der an es gestellten Anforderung bewertet. Dazu werden die Parameter eines Agenten, die er aufgrund der Bereitstellung einer mit Anforderungen belegten Prozessfunktion gezwungen ist zu implementieren, auf Grundlage der Parameter des Technischen Systems berechnet. Damit kann berechnet werden ob eine Prozessanforderung durch das jeweilige Anlagenmodul erfüllt werden kann. Zum anderen werden in diesem Abschnitt Softsensoren zum Automatisierungssystem eines Anlagenmoduls hinzugefügt und durch die Verknüpfung mit den realen Sensoren beschrieben, aus welchen Quellinformationen die Werte in der Software berechnet werden können. Für unkritische Pfade (Module) innerhalb einer Anlage, in denen eine Erhöhung der Verfügbarkeit durch Softsensoren nicht zwingend erforderlich ist und keine speziellen Prozessanforderungen an die Funktionen des Anlagenteils gestellt werden, können diese Schritte des Vorgehensmodells vernachlässigt werden.

## 4.5 Anwendungsbeispiel – Hybrides Prozessmodell

Das Demonstrationsbeispiel verwendet den verfahrenstechnischen Anlagenteil des hybriden Prozessmodells, einer Laboranlage des Lehrstuhls AIS, die sowohl verfahrenstechnische als auch fertigungstechnische Anlagenteile enthält. Der verfahrenstechnische Teil bietet die Möglichkeit in einer MPS Kompaktwerkstation [Festo Didaktik] verschiedene Prozessschritte mit einer Flüssigkeit auszuführen, die anschließend an zwei Abfüllstationen zusammen mit Granulat in Flaschen abgefüllt werden kann.

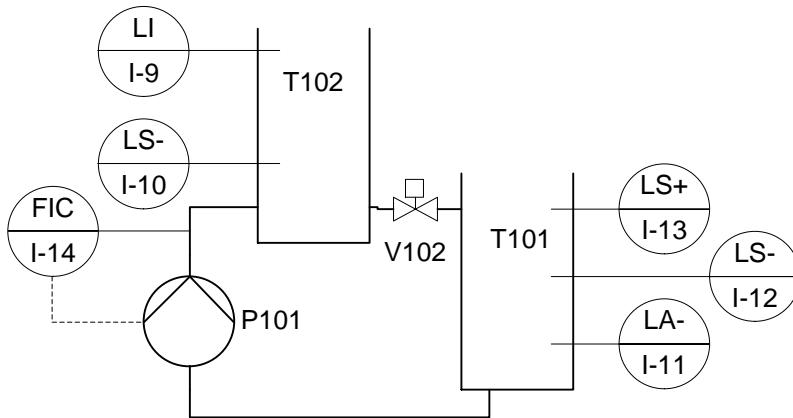


Abb. 4.3 – R&I-Fließbild aus dem Hybriden Prozessmodell

Im Anwendungsbeispiel wird ein einzelner Softwareagent entwickelt, der den in Abb. 4.3 dargestellten Ausschnitt des verfahrenstechnischen Anlagenteils der Laboranlage steuert und als Prozessfunktion das Umwälzen der Flüssigkeit zwischen den beiden Tanks T101 und T102 anbietet. Die Prozessfunktion besteht im Detail aus dem Abpumpen der Flüssigkeit aus Tank T101 bis der untere Füllstand (LS-) erreicht ist, woraufhin der Tank T102 wieder über das Ventil V102 in den Tank T101 entleert wird, bis dessen oberer Füllstand (LS+) erreicht ist. Um auch bei einem Sensorausfall der Füllstandssensoren des Tanks T101 die Prozessfunktion noch ausführen zu können, werden innerhalb des Agenten Softsensoren implementiert, die zur Kompensation von Ausfällen der realen Sensoren (I-12, I-13) verwendet werden können. Im Folgenden wird für diesen Modellprozesses gezeigt, wie ein Agentensystem nach der hier vorgestellten Architektur mithilfe der im

Projekt KREAagentuse entwickelten Vorgehensweise und dem Werkzeug auf einem Laufzeitsystem nach IEC 61131-3 implementiert werden kann.

## 4.5.2 Komposition der Steuerungsagenten

Die modulare Komposition des Anlagenmoduls wird im Blockdefinitionsdiagramm (BDD) durch Blöcke und Interfaces direkt in der IEC 61131 Entwicklungsumgebung durchgeführt. Die Abbildungsregeln des BDD basieren auf denen des Klassendiagramms, das in früheren Arbeiten des Lehrstuhls AIS in die Entwicklungsumgebung integriert wurde [13]. Im Unterschied zum Klassendiagramm werden hier jedoch Blöcke der SysML und nicht *Klassen* der UML auf Funktionsbausteine abgebildet (siehe Tab. 4.1). Des Weiteren wurden zusätzlich zu den Elementen des Klassendiagramms noch das Element *Constraint-Block* der SysML in das Diagramm integriert, das auf Funktionen der IEC 61131-3 abgebildet werden kann. Als Implementierungssprache für die Blöcke, Constraint-Blöcke und weitere Objekte der objektorientierten IEC 61131-3 [11] stehen die Standardsprachen der IEC 61131-3 zur Verfügung sowie das Parameterdiagramm der SysML für das im Rahmen des Projekts ein prototypischer Codegenerator realisiert wurde.

Tab. 4.1 – Abbildungsregeln im Blockdefinitionsdiagramm

SysML Element	Spezialfälle	Abbildung auf IEC
Block	Agent Process Sensor Actuator	Function Block
Constraint		Function
Interface		Interface
Property		Property
Attribute		Variable (VAR)
Method	Activity	Method

Sensor- sowie Aktorblöcke, die zur Anbindung der Software an die Signale genutzt werden, bietet das Werkzeug vorbereitete Spezialfälle des Elements Block (siehe Tab. 4.1) in einer Toolbar an, die in das Diagramm eingefügt,



parametriert und in einem übergeordneten Block instanziiert werden können. Die Sensorbausteine bieten eine erweiterte Funktionalität an. So besitzen sie die zwei verschiedenen Modi „Hardwaremodus“ und „Softwaremodus“, zwischen denen sie umgeschaltet werden können. Während im Hardwaremodus der übergeordneten Steuerung, die über die Klemmen eingelesenen Werte weitergeleitet werden, kann im Softwaremodus ein durch einen Softsensor berechneter Wert zur Weiterleitung an die übergeordnete Steuerung übergeben werden. Darüber hinaus liefern die Sensorbausteine zusätzlich zu dem eigentlichen Sensorwert auch ein Maß für dessen aktuelle Präzision und einen Vertrauenswert gegenüber anderen verfügbaren Sensoren.

Nachdem auf diese Art ein übergeordneter Block beschrieben wurde, der für die Implementierung der für das im Anwendungsbeispiel betrachtete Anlagenmodul verwendet werden soll, kann ihm ein Agent zugeordnet werden, der die Steuerung dieses Moduls sowie die Diagnose und Kompensation von Sensordefekten übernimmt [14, 15]. Für die Agenten existiert dabei ebenso wie für die Sensorbausteine ein Standardbaustein, der direkt mittels der Toolbar des Editors erstellt werden und anschließend in den späteren Schritten applikationsspezifisch erweitert werden kann. Die Verknüpfung beider Bausteine erfolgt mittels einer Referenz (Pointer) des Agenten auf den Block des Moduls [14, 15]. Über diese Referenz hat der Agent Zugriff auf die unterlagerten Funktionalitäten des Moduls, wie die Ansteuerung der Pumpe und des Ventils, sodass der Agent in der Lage ist ein Service-Interface zu implementieren, indem die Prozessfunktion „Umwälzen“ als *Activity* enthalten ist.

### 4.5.3 Beschreibung des Prozesses und der Anforderungen

Wie Kapitel 4.3 einleitend beschrieben werden Produktionsprozesse als Verhalten eines *Process*-Blocks innerhalb eines Aktivitätsdiagramms beschrieben. Dieser Prozessblock stellt eine weitere spezialisierte Form des einfachen Blocks (siehe Tab. 4.1) dar, der in der Toolbar des Editors für Blockdefinitionsdiagramme verfügbar ist, und ebenfalls auf einen Funktionsbaustein abgebildet wird. Der Editor für Aktivitätsdiagramme des entwickelten Werkzeugs basiert auf dem des UML-Plugins [12]. In ihm kann durch den sequenziellen oder parallelen Aufruf von Aktivitäten (*Activity*) als Prozessschritte der jeweilige applikationsspezifische Prozess als Implementierung des Funktionsbausteins modelliert werden. Bei den

## 4 Verfügbarkeitssteigerung von Produktionsanlagen mit Softwareagenten – Entwicklung eines durchgängigen Werkzeugs

*Activities*, die in den einzelnen Prozessschritten aufgerufen werden, handelt es sich um jene Aktivitäten, die von den Softwareagenten über Service-Interfaces bereitgestellt werden.

Über den Editor aus dem UML-Plugin hinaus bietet das hier vorgestellte Werkzeug die Funktionalität, Anforderungen in Form von *Requirements* an aufgerufene Aktivitäten zu binden. Diese Requirements enthalten eine textuelle Beschreibung der Anforderung sowie einen Variablennamen und –typ mit dem die Erfüllung der Anforderung bewertet werden soll. Im Anwendungsbeispiel kann die Einhaltung eines definierten Bereichs für das umgewälzte Flüssigkeitsvolumen eine mögliche Prozessanforderung sein, die vom Technischen System eingehalten werden muss (siehe Abb. 4.4).

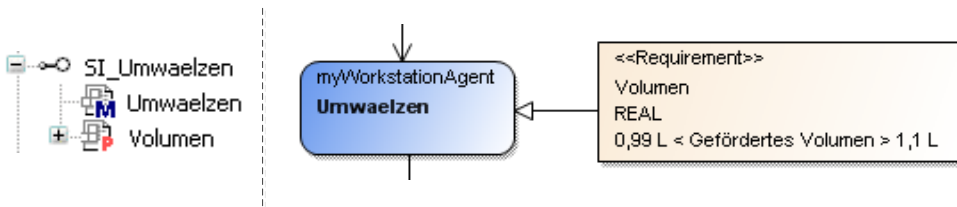


Abb. 4.4 – Prozessanforderung gebunden an eine Aktivität

Wird innerhalb des Aktivitätsdiagramms ein Requirement an den Aufruf einer Prozessfunktion gebunden, fügt das Werkzeug automatisch eine Property zu den Funktionsbausteinen der Agenten hinzu, welche mittels einem Interface die Ausführung der jeweiligen Prozessfunktion anbieten. Diese besitzt den Namen und den Datentyp, die in der Anforderung spezifiziert wurden, sowie die textuelle Beschreibung der Anforderung als Kommentar. Dadurch ist der Entwickler des Agenten der Workstation in der Pflicht, das Modul auf Grundlage von dessen Parametern hinsichtlich der Anforderung zu bewerten.

Das Element Requirement kann nicht direkt auf ein Element objekt-orientierten IEC 61131-3 abgebildet werden und stellt daher ein erweitertes Element dar, das innerhalb des Aktivitätsdiagramms existiert. Jedoch wird durch die vorgegebene Architektur jedes Requirement auf die Property eines Interfaces abgebildet, sodass für jedes Requirement im SysML-Modell des Agentensystems eine Property im korrespondierenden IEC 61131-3 Code existiert.

#### 4.5.4 Systemeinschränkungen und Softsensoren

Nach der Implementierung eines Prozesses, in dem Anforderungen an die Aktivitäten der Agenten gestellt werden, müssen *Properties*, die durch Anforderungen zu den Interfaces hinzugekommen sind, in den Agenten implementiert werden. Dies kann bei vollem Durchlauf des Vorgehensmodells der Fall sein oder dann, wenn nur ein neuer Prozess für ein bestehendes Agentensystem implementiert wurde.

Der hier vorgestellte Ansatz sieht die Implementierung dieser *Properties* durch die Modellierung eines Parameterdiagramms in den jeweiligen Agenten vor. In diesen Parameterdiagrammen muss der Zusammenhang modelliert werden, der zwischen den Parametern eines Moduls und den an seine Funktionen gestellten Prozessanforderungen besteht.

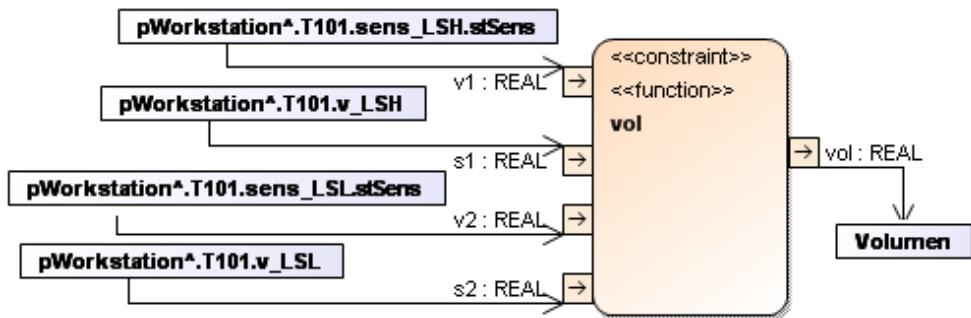


Abb. 4.5 – Parameterdiagramm der Workstation


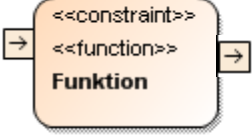


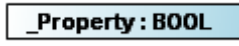


So muss für die beispielhafte Anforderung an den Prozessschritt „Umwälzen“ der Workstation auf Grundlage von deren physikalischen Parametern (Fläche und Höhe der Tanks, Position der angebrachten Sensoren) sowie der Präzision (Messungenauigkeit) der Sensoren der Volumenbereich berechnet werden, der durch die Workstation umgewälzt werden kann (siehe Abb. 4.5).

Parameterdiagramme stellen im Gegensatz zu Blockdefinitionsdiagrammen, die der Strukturierung der Software dienen, eine Implementierungssprache für Funktionsblöcke, Funktionen und Methoden dar. Die verwendeten

## 4 Verfügbarkeitssteigerung von Produktionsanlagen mit Softwareagenten – Entwicklung eines durchgängigen Werkzeugs

Diagrammelemente können teilweise auf Elemente der IEC 61131-3 abgebildet werden (siehe Tab. 4.2).

Tab. 4.2 – *Elemente des Parameterdiagramms und Abbildung auf IEC 61131*

SysML Element	Beispiel	Abbildung auf IEC
Block		Aufruf der Instanz eines Funktionsbausteins
Constraint		Funktionsaufruf
Port_In / Port_Out		Variable (VAR_IN / VAR_OUT)
Attribute		Variable (VAR)
Property		Property
BindingConnection		-
FunctionalDependency		-

Lediglich die *BindingConnections* und *FunctionalDependencies* (Verbindungs Pfeile zwischen den Elementen) werden nicht auf Elemente der IEC 61131-3 abgebildet sondern dienen als Informationen für die Codegenerierung.

## 4 Verfügbarkeitssteigerung von Produktionsanlagen mit Softwareagenten – Entwicklung eines durchgängigen Werkzeugs

---

Die Implementierung von Softsensoren für ein Anlagenmodul wird ebenfalls im Parameterdiagramm von dessen Agenten durchgeführt. Die Generierung des Steuerungscode für diese spezielle Anwendung ist der allgemeinen Generierung nachgeschaltet. In diesen Parameterdiagrammen können Softsensorbausteine aus einer Bibliothek heraus instanziiert (siehe Abb. 4.6 „VS\_T101 : VS\_Fuellst“) und aufgerufen werden. Die Bausteine werden darin ebenfalls mit den Quellinformationen (u.a. Werte realer Sensoren) verknüpft, die für die Berechnung eines Softsensor-Wertes benötigt werden.

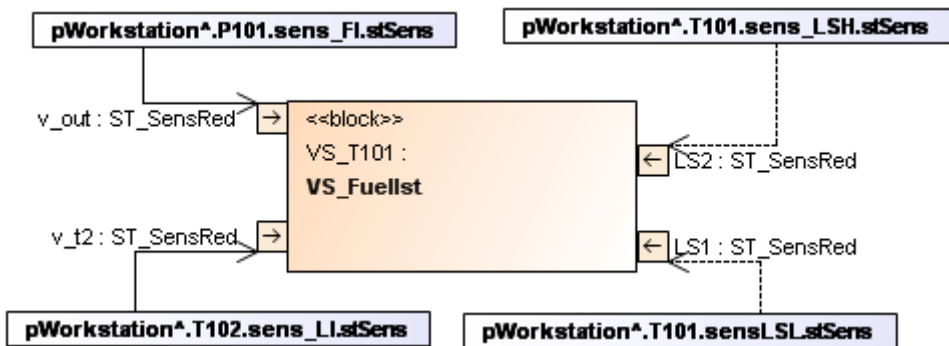


Abb. 4.6 – Implementierung von Softsensoren für die Workstation

Die Semantik dieser Parameterdiagramme ist angelehnt an den von Wannagat et al. eingeführten gerichteten Graphen [2, 3, 4], mit dem Softsensorbausteine auf Grundlage des analytischen Zusammenhangs der realen Sensoren und Parameter modelliert werden können. Die zusätzliche Codegenerierung für diese Art von Parameterdiagrammen generiert über den reinen Aufruf der Softsensoren hinaus den Teil des Codes der Softwareagenten, der applikationsspezifisch für die Diagnose und Kompensation der realen Sensoren verantwortlich ist. Die Information, welche realen Sensorwerte durch die als Blöcke dargestellten Softsensoren ersetzt werden können, können dabei aus den im Diagramm verwendeten *Functional-Dependencies* (gestrichelte Verbindungspfeile) generiert werden. Im vorgestellten Applikationsbeispiel der Workstation kann aus den Informationen über die Volumenströme in und aus dem Tank T101 ein Softsensor für dessen Füllstandsensoren implementiert werden (siehe Abb. 4.6). Dies ist gekennzeichnet durch die Verbindung der Werte der realen Sensoren mit Ports des Blocks, der den Softsensor implementiert.

## 4.6 Zusammenfassung und Ausblick

In diesem Beitrag wurde die Verwendung von Softwareagenten zur Steigerung der Verfügbarkeit von sowohl verfahrenstechnischen als auch fertigungstechnischen Produktionsanlagen erläutert. Die Architektur der Softwareagenten baut auf Vorarbeiten des Lehrstuhls AIS auf, Softsensoren innerhalb von Softwareagenten direkt auf Speicherprogrammierbaren Steuerungen (SPS) zu implementieren. Im Rahmen des Transferprojekts KREAagentuse [6] wurde eine werkzeugunterstützte Vorgehensweise entwickelt und ein prototypisches Werkzeug erstellt. Die modellbasierte Entwicklung und automatische Implementierung der Softwareagenten erfolgt innerhalb einer marktüblichen Entwicklungsumgebung für IEC-Steuerungscode.

Um bei der Modellierung auch die Komplexität von größeren Produktionssystemen handhaben zu können, sieht der Entwicklungsansatz eine Aufteilung des Modells in verschiedene domänenspezifische Sichten vor. Eine Besonderheit ist dabei, dass Prozessanforderungen explizit innerhalb der Modellierung berücksichtigt werden und auf die jeweils verantwortlichen Elemente des Produktionssystems zurückgeführt werden.

Das entwickelte und vom Werkzeug unterstützte Vorgehensmodell deckt sich in großen Teilen mit den konventionellen Vorgehensweisen bei der Entwicklung von Steuerungscode, sodass eine Integration der Vorgehensweise in bestehende Entwicklungsabläufe vereinfacht durchgeführt werden kann. Zudem unterstützt die Skalierbarkeit des Vorgehensmodells eine Anpassung der Architektur des Agentensystems an den jeweiligen Anwendungsfall und reduziert damit den Einstiegsaufwand. Funktionalitäten wie zum Beispiel die Verfügbarkeitssteigerung mittels Softsensoren können durch die Skalierbarkeit auch nur für kritische Pfade innerhalb eines Produktionssystems implementiert werden.

Das entwickelte Werkzeug, die vorgeschlagene Vorgehensweise und die Architektur für einzelne Softwareagenten wie auch Agentensysteme wurden an verschiedenen Demonstratoren evaluiert. Für diesen Beitrag wurde ein Anwendungsbeispiel aus der verfahrenstechnischen Domäne ausgewählt. Weiter wurden Werkzeug und Vorgehensweise in Expertenevaluierungen bei zwei Unternehmen aus dem Maschinen- und Anlagenbau und einem Unternehmen aus der Anlagentechnik durchgeführt. Es wurde das industrielle Anwendungsbeispiel (Abb. 4.3) als Referenzszenario verwendet

#### 4 Verfügbarkeitssteigerung von Produktionsanlagen mit Softwareagenten – Entwicklung eines durchgängigen Werkzeugs

---

um die Projektergebnisse vorzustellen. In den anschließend geführten Interviews ergab sich positives wie auch negatives Feedback zu einzelnen Aspekten des Gesamtkonzepts:

1. Hauptkritikpunkt war das Thema Datenfluss und Zugriff des Softwareagenten auf die Steuerung der Anlagenmodule. Der Zugriff des Agenten auf die Steuerung mittels eines Pointers wurde als kritisch bewertet, da die Transparenz der Datenzugriffe verloren gehen kann.
2. Welche Funktionen ein soll ein Agent innerhalb der Steuerungssoftware einer Anlage übernehmen:
  - a. Vollständige Übernahme der Funktionen eines Anlagenmoduls durch einen Agenten wurde kritisch bewertet.
  - b. Konzept der Softwareagenten kann einfacher in den praktizierten Softwareentwurf integriert werden, wenn die Agenten lediglich die Diagnose und Kompensation von Sensorausfällen übernehmen.
3. Positiv wurde die Entwicklung und Strukturierung der Steuerungssoftware innerhalb von Blockdefinitionsdiagrammen bewertet. Der grafisch repräsentierte Steuerungscode ist einfacher verständlich, wodurch die Wartung des Codes durch Personen, die nicht die ursprünglichen Entwickler des Steuerungscode sind, vereinfacht wird.
4. Die Modellsichten geben die Möglichkeit zur Trennung des Anlagenprozesses von der Implementierung der Modulfunktionen, Hierin wurde ein Potenzial für ein interdisziplinäres (Prozesstechnologie und Anlagenbauer) Arbeiten innerhalb eines gemeinsamen Projekts gesehen.
5. Mit dem entwickelten Konzept und Werkzeug ist es möglich, Anforderungen innerhalb des Prozesses zu beschreiben und auf einzelne Anlagenmodule zu projizieren. Die Angabe von Parametern mit konkreten Datentypen durch Prozesstechnologen ist jedoch nicht praktikabel. Die umgesetzte natürlichsprachliche Beschreibung einer Anforderung innerhalb eines Requirements ist sinnvoller.

An das Projekt KREAagentuse anschließende Arbeiten werden sich zum einen mit der Einarbeitung der Evaluationsergebnisse in das Konzept und prototypische Werkzeug beschäftigen. Zum anderen wird die Integration weiterer Konzepte in die Agenten beschäftigen. So existieren bereits Vorarbeiten des Lehrstuhls wie auch gemeinsame Vorarbeiten mit Beckhoff Automation GmbH, um modellbasiert eine energieoptimale Auslegung von Produktionsanlagen zu unterstützen.

Nach Expertenmeinung werden einzelne Agentenkonzepte wie die Diagnose und Kompensation von Sensorfehlern zwar als einsetzbar bewertet, eine Steuerung der Modulfunktionen durch Softwareagenten jedoch kritisch gesehen. Dies kann insofern interpretiert werden, dass seitens der industriellen Anwender noch wenig Vertrauen in die agentenbasierte Steuerung besteht. Daher wird in weiteren Arbeiten untersucht werden, wie dem Bediener einer Anlage mithilfe einer Benutzerschnittstelle das Monitoring und die Transparenz des Verhaltens der Agenten verdeutlicht werden kann, um Vertrauen in die Agenten aufzubauen und das „Ausschalten“ der Agentenbasierten Steuerung zu vermeiden.

Ein weiterführendes Projekt „CogniONT“, das im Rahmen des Exzellenzclusters CoTeSys durchgeführt wird, untersucht die Kopplung der auf einer SPS implementierten Agenten an andere Module eines Produktionssystems wie zum Beispiel mobile Roboter und Vision-Systeme auf Basis einer gemeinsamen Ontologie.

### 4.7 Danksagung

Die Autoren bedanken sich für die finanzielle Unterstützung der Deutschen Forschungsgemeinschaft (DFG) für das Projekt KREAagentuse (Konzeption, Realisierung und Evaluation einer werkzeugunterstützten Vorgehensweise für die Entwicklung von Agentensystemen in der Automatisierungstechnik unter Berücksichtigung der Usability, VO 937/8-1).

### 4.8 Referenzen

- [1] Vogel-Heuser, B.: Automation in wood and paper industry. In: Handbook of Automation, Nof, S.Y. (Ed.), 2008.
- [2] Wannagat, A.; Vogel-Heuser, B.: Increasing flexibility and availability of manufacturing systems – Dynamic reconfiguration of automation software at runtime on sensor faults. In 9th IEEE Workshop on Intelligent Manufacturing Systems (IMS), Stettin, 2008.
- [3] Wannagat, A.; Vogel-Heuser, B.: Agent oriented software-development for networked embedded systems with real time and dependability requirements the domain of automation. In: 17th IFAC World Congress, Seoul, Korea, 2008.



#### 4 Verfügbarkeitssteigerung von Produktionsanlagen mit Softwareagenten – Entwicklung eines durchgängigen Werkzeugs

---

- [4] Wannagat, A.: Entwicklung und Evaluation agentenorientierter Automatisierungssysteme zur Erhöhung der Flexibilität und Zuverlässigkeit von Produktionsanlagen, Ph.D. Dissertation, Technische Universität München, 2010.
- [5] DFG-Projekt „Agenten für flexible und verlässliche eingebettete Echtzeitsysteme (AVE)“, VO 937/5-1.
- [6] DFG-Transferprojekt „Konzeption, Realisierung und Evaluation einer werkzeugunterstützten Vorgehensweise für die Entwicklung von Agentensystemen in der Automatisierungstechnik unter Berücksichtigung der Usability (KREAagentuse)“, VO 937/8-1.
- [7] Schütz, D.; Wannagat, A.: Domänenspezifische Modellierung für automatisierungstechnische Anlagen mit Hilfe der SysML. In: Automatisierungstechnische Praxis (atp), Ausgabe 51, Heft 3, 2009, S. 54-62.
- [8] Schütz, D.; Vogel-Heuser, B.; et al.: Architekturmodelle zur Bewertung von energetischen Optimierungskriterien - MDA als Bindeglied domänenspezifischer Diagnose. In: Mechatronik 2009, Wiesloch, 2009, S. 69-76.
- [9] Schütz, D.; Vogel-Heuser, B.: Modellierung der Verhaltensaspekte automatisierungstechnischer Module von Produktionsanlagen unter Berücksichtigung der energetischen Zusammenhänge. In: 11. Fachtagung Entwurf komplexer Automatisierungssysteme (EKA), Magdeburg, 2010, S. 85-93.
- [10] Schütz, D.; Vogel-Heuser, B.: Modellintegration von Verhaltens- und energetischen Aspekten für mechatronische Module. In: Automatisierungstechnik (at), Ausgabe 59, Heft 1, Januar, 2010, S. 33-42.
- [11] Wannagat, A.; Vogel-Heuser, B.: Wiederverwendung und modulares Engineering mit CoDeSys V3, Oldenbourg Industrieverlag, München, 2008.
- [12] Vogel-Heuser, B. (Hrsg.): Automation and Embedded Systems – Effizienzsteigerung im Engineering, Kassel, Kassel University Press, 2009.

- [13] Witsch, D.: Einsatz von UML-Diagrammen in der Steuerungsprogrammierung. In: Vogel-Heuser, B.: Automation and Embedded Systems – Effizienzsteigerung im Engineering, Kassel, Kassel University Press, 2009, S. 30-77.
- [14] Frank, U.; Papenfort, J.; et al.: Real-time capable software agents on IEC 61131 systems – Developing a tool supported method. In: 18th IFAC World Congress, Mailand, Italien, 2011, akzeptierter Beitrag.
- [15] Schütz, D.; Vogel-Heuser, B.: Flexible, echtzeitfähige Steuerungssoftware durch Agenten auf IEC 61131 Systemen. In: SPS/IPC/Drives, Nürnberg, 2010, S. 411-421.
- [16] Lauber, R.; Göhner, P.: Prozessautomatisierung 1, Springer, Berlin, 1999.

## 5 Verteilt oder Zentral? Unterstützung für den Entwurf der Automatisierung durch das Projekt FAVA

Alexander Fay

*Institut für Automatisierungstechnik, Helmut-Schmidt-Universität*

Birgit Vogel-Heuser

*Lehrstuhl für Automatisierung und Informationssysteme (AIS), TU München*

Christian Diedrich

*Institut für Automatisierungstechnik (IFAT), Otto-von-Guericke-Universität Magdeburg*

**Zusammenfassung:** Im Forschungsprojekt FAVA (Funktionaler Anwendungsentwurf für verteilte Automatisierungssysteme) arbeiten der Lehrstuhl für Automatisierung und Informationssysteme der TU München unter der Leitung von Frau Prof. Dr.-Ing. Birgit Vogel-Heuser, der Lehrstuhl für Integrierte Automation am Institut für Automatisierungstechnik der Otto-von-Guericke-Universität Magdeburg unter der Leitung von Prof. Dr.-Ing. Christian Diedrich und das Institut für Automatisierungstechnik der Helmut-Schmidt-Universität / Universität der Bundeswehr Hamburg gemeinsam an Methoden, Beschreibungsmitteln und Werkzeugen zur Unterstützung derjenigen, die verteilte Automatisierungssysteme planen und realisieren.

### 5.1 Warum dieses Projekt?

Betreiber automatisierter Produktionsanlagen streben danach, ihre Anlage auch bei wechselnden Randbedingungen möglichst optimal zu betreiben. Dazu müssen vielerlei Informationen erfasst und im Automatisierungssystem berücksichtigt werden: schwankende Bedarfe, Produkte mit unterschiedlichen Merkmalen in kleinen Losgrößen, wechselnde Qualitäten der Rohmaterialien, Diagnoseinformationen von Maschinen und Produktionsschritten u.v.a.m., Früher getrennt, d.h. jeweils für sich durchführbare Regelungs- und Steuerungsaufgaben werden durch diese Informationsflüsse miteinander vernetzt. Je mehr zusätzliche Informationen dabei einbezogen werden, desto dichter am Optimum kann die Produktionsanlage „gefahren“ werden – zumindest theoretisch. Bedingt durch die räumliche Ausdehnung

der Anlagen führt dies zu verteilten Automatisierungssystemen, in denen verschiedene Komponenten Automatisierungsaufgaben ausführen und miteinander über Kommunikationssysteme vernetzt sind.

Herkömmliche Entwurfsmethoden für Automatisierungssysteme fokussieren im Allgemeinen auf zentralistische Strukturen. Es fehlt bisher eine Methode für den systematischen Entwurf solcher verteilter Automatisierungssysteme. „Systematisch“ bedeutet dabei, einerseits die besonderen verschiedenen Anforderungen zu berücksichtigen, die sich aus der Verteilung der Automatisierungsfunktionen auf mehrere (verschiedene) Komponenten und aus den Kommunikationserfordernissen zwischen diesen Komponenten ergeben, und andererseits Wiederverwendung von guten Lösungen zu fördern. Im Forschungsprojekt FAVA („Funktionaler Anwendungsentwurf für verteilte Automatisierungssysteme“) soll eine solche Methode entwickelt werden.

### 5.2 Die Aufgabenstellung

Zur optimalen Steuerung und Regelung technischer Prozesse müssen Informationen von verschiedenen, oft räumlich weit auseinander liegenden, aber doch funktional zusammengehörigen Informationsquellen (Sensoren, Bedieneingriffen, übergeordneten Zielvorgaben, ...) verknüpft werden, gemeinsam verarbeitet werden und dann an verschiedene Informationsempfänger (Aktoren, Schnittstellen zum Bediener und zu anderen informationsverarbeitenden Einrichtungen, zum Beispiel MES und ERP-Systemen) übermittelt werden. Die dafür erforderlichen verteilten Automatisierungssysteme werden aus verschiedenen Komponenten zusammengesetzt, die jeweils aus Hard- und Software bestehen und meist hinsichtlich der Funktionen, die sie ausführen können, anpassbar sind. Zu diesen Komponenten zählen nicht nur Speicherprogrammierbare Steuerungen, konfigurierbare Sicherheitsrelais und parametrierbare Buskoppler, sondern vermehrt auch so genannte „intelligente“ Sensoren und „intelligente“ Antriebe mit eigener Datenverarbeitungsfähigkeit.

Solche Automatisierungssysteme sind komplex hinsichtlich der Art und Anzahl der Komponenten, deren Eigenschaften und Verknüpfungen, jeweils beschrieben durch eine große Zahl von Merkmalen. Der Entwurf dieser verteilten Automatisierungssysteme, bei dem Verhalten und Verteilung der Funktionen auf die Komponenten und ihre Kommunikation untereinander festzulegen sind, stellt somit ein komplexes Optimierungsproblem dar.

Üblicherweise löst man in der Praxis diese Aufgabe dadurch, dass man für bestimmte automatisierungstechnische Teilaufgaben bestimmte Technologien und Komponenten festlegt, zum Beispiel ein bestimmtes digitales Kommunikationssystem (beispielsweise ein Feldbus oder eine Variante des Industrial Ethernet) und bestimmte Sensoren und Aktoren. Dabei orientiert man sich an Erfahrungen („betriebsbewährt“), an Produkten bestimmter Hersteller, an der Kompatibilität zu anderen Automatisierungslösungen im gleichen Betrieb, an einem bekannten Technologien oder an anderen, durchaus auch nicht-technischen Vorgaben. Dadurch reduziert sich der Lösungsraum, so dass man im Allgemeinen zu einer handhabbaren Menge von alternativen Lösungen kommt, aus der man dann eine auswählt. Möglicherweise ist diese unter den gegebenen Umständen die Beste, möglicherweise gibt es aber andere Lösungen, die aus Unkenntnis nicht in Betracht gezogen wurden und die Vorteile aufweisen könnten – zumindest hinsichtlich bestimmter Kriterien. Durch die Festlegungen auf bestimmte Technologien und Komponenten erhöht sich im Allgemeinen der Aufwand, die gegebenen Anforderungen an das Automatisierungssystem zu erfüllen, insbesondere, wenn sich – wie heutzutage vielfach der Fall – die Anforderungen häufig ändern. Dies führt zum Verharren bei zunehmend suboptimaler werdenden Lösungen oder zu einer großen Varianz der Lösungen und damit zu verminderten Möglichkeiten der Wiederverwendung. Darunter leiden wiederum Effizienz und Qualität des Entwurfs von Automatisierungssystemen.

Hilfreich könnte für den Planer der Automatisierungstechnik sein, im Planungsprozess Vorschläge zu erhalten, wie die jeweiligen Anforderungen gut erfüllt werden könnten. Aufgrund des schnellen Wandels der Technik müssten die Vorschläge eher grundsätzlicher Natur sein, im Sinne einer Lenkung hin zu geeigneten Lösungen.

### **5.3 Das Konzept von FAVA**

Damit Vorschläge Anforderungen erfüllen, müssen die Anforderungen zunächst spezifiziert werden, d.h. eindeutig formuliert werden. Unter „Spezifikation der Anforderungen an ein Automatisierungssystem“ denkt man im Allgemeinen an die Spezifikation der Funktion, die das System ausführen soll, also zum Beispiel „Regelung des Füllstands“ oder „Abschalten der Pumpe“. Für die Spezifikation automatisierungstechnischer Funktionen gibt es eine Vielzahl von Beschreibungsmitteln, zum Beispiel

Statecharts, Petrinetze, Taktzeitdiagramme, Gantt-Charts, Funktionspläne usw. (eine bewertende Übersicht gibt [1]). Bei dieser Spezifikation der Funktion vermeidet man zunächst die Festlegung auf eine Technologie oder auf bestimmte Komponenten der Realisierung – dies folgt dann im nächsten Arbeitsschritt. Dabei müssen die anderen Anforderungen, die sich nicht auf das „WAS“ der automatisierungstechnischen Funktion beziehen, sondern auf das „WIE“ ihrer Umsetzung und ihres späteren Einsatzes, berücksichtigt werden. Diese so genannten nicht-funktionalen Anforderungen sind sehr vielfältig. Einen Eindruck davon gibt die VDI/VDE-Richtlinie 3694 [2], die die Fragen aufzeigt, die man hinsichtlich der nicht-funktionalen Anforderungen stellen muss – aber keine Antworten dazu gibt.

Im Projekt FAVA wird nicht angestrebt, darauf eindeutige oder endgültige Antworten zu geben, sondern vielmehr, unter Berücksichtigung spezifizierter nicht-funktionaler Anforderungen Hinweise darauf zu geben, wo die Antworten zu suchen sind und wie sie prinzipiell aussehen könnten.

Ein verteiltes Automatisierungssystem stellt spezifische funktionale und nicht-funktionale Anforderungen hinsichtlich seiner Komponenten, an deren Verhalten, der Funktionsverteilung der Gesamtanwendung auf die Komponenten und der dazwischen stattfindenden Kommunikation, die zusätzlich zu denen für nicht verteilte Automatisierungssysteme geeignet darzustellen und zu erfüllen sind.

- Im Projekt FAVA wird auf technische nicht-funktionale Anforderungen fokussiert, nicht-technische (z.B. kommerzielle) Anforderungen werden nicht betrachtet.
- Im Projekt FAVA werden insbesondere diejenigen funktionalen und nicht-funktionalen Anforderungen betrachtet, die mit Hilfe der Verteilung der Komponenten erfüllt werden bzw. die erst aus der Verteilung der Komponenten resultieren.

Die Systemelemente (= Lösungselemente) "Komponente", "Struktur" und "Kommunikationssystem" werden dafür durch Merkmale beschrieben, aus denen die Eigenschaften des verteilten Automatisierungssystems abzuleiten sind. Daraus ergibt sich zum Einen die Forschungsfrage, welche Merkmale dafür einerseits geeignet und andererseits relevant sind. Zum Anderen muss eine Möglichkeit gefunden werden, mit der Planer ihre nicht-funktionalen Anforderungen (in Ergänzung zu den funktionalen Anforderungen)

formulieren können (als Text? Oder grafisch? Oder in Kombination?). Das könnte durch die Erweiterung eines bewährten Beschreibungsmittels für funktionale Anforderungen erfolgen.

Und schließlich muss ein Beschreibungsmittel gefunden bzw. erarbeitet werden, welches einen Zusammenhang herstellen kann zwischen den so spezifizierten Anforderungen einerseits und den prinzipiellen, durch die Merkmale beschriebenen Lösungen andererseits. Diese Lösungen sollen in Form von „Mustern“ angegeben werden. Da es um eine Unterstützung in der Planung geht, werden also automatisierungstechnische „Entwurfsmuster“ gesucht. Der Begriff der „Entwurfsmuster“ (engl. design pattern) stammt aus der Software-Technik, wo sich diese bewährt haben, um prinzipielle Lösungen für prinzipielle Entwurfsprobleme zu beschreiben. Die Eignung des Ansatzes, mit Entwurfsmustern zwischen diesen Merkmalen und den gestellten Anforderungen eine Verbindung herzustellen und damit den Entwurfsprozess zielgerichtet zu leiten, soll im Projekt FAVA erforscht werden. Dafür werden die Systemelemente „Komponente“, „Struktur“ und „Kommunikation“ durch Merkmale beschrieben. Diese werden mit funktionalen und nicht-funktionalen Anforderungen, die in geeignet erweiterten Beschreibungsmitteln dokumentiert werden, in Beziehung gesetzt. Dem Ziel der Wiederverwendung soll dadurch Rechnung getragen werden, dass mögliche Automatisierungslösungen in Form von Entwurfsmustern dargestellt werden, die einerseits auf diese Merkmale referenzieren und andererseits diese erweiterten Beschreibungsmittel verwenden. Die erarbeitete Methode wird als Funktionsmuster in einem Software-Werkzeug implementiert und auf dieser Basis durch Anwender evaluiert werden.

### **5.4 Zentrale und verteilte Struktur von Automatisierungssystemen**

Für die Konzeption von Automatisierungssystemen haben sich drei Architekturparadigmen, die auf der Nutzung von Funktionsblocktechnologien basieren, etabliert (siehe Abb. 5.1).

Die in der Automatisierungstechnik klassische Systemarchitektur ist als zentrales System konzipiert, in dem verschiedensprachliche Notationen nach IEC 61131-3 auf prozessnahen Steuerungskomponenten verwendet

## 5 Verteilt oder Zentral? Unterstützung für den Entwurf der Automatisierung durch das Projekt FAVA

---

werden. Eingriffe in den technologischen Prozess werden durch zentrale Ein-/Ausgabebaugruppen oder Remote-IOs mit Feldbuskommunikation zur Steuerseite implementiert. Die Ein- Ausgabekomponenten kommunizieren über analoge Schnittstellen, wie z.B. das 4-20mA-Einheitssignal mit den funktional einfachen Feldgeräten.

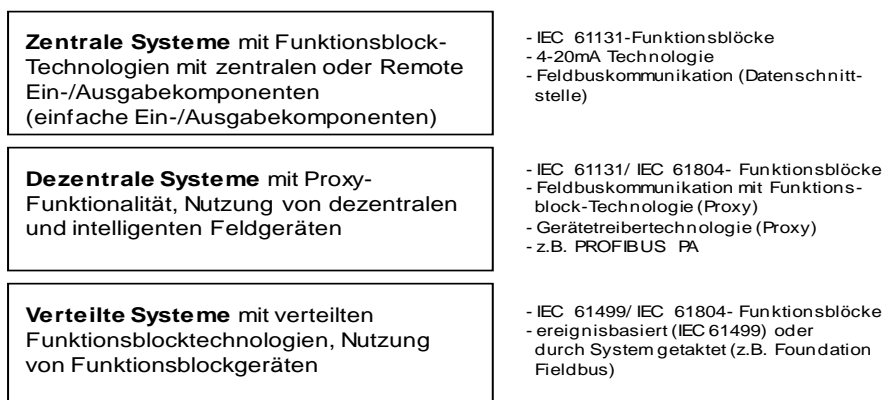


Abb. 5.1 – Strukturen von Automatisierungssystemen nach IEC 62390

Ein stark anwachsender Anteil implementierter Automatisierungssysteme ist mittlerweile mit dezentraler Architektur unter Nutzung so genannter „intelligenter“ Feldgeräte konzipiert. Die Kommunikation erfolgt via digitaler Technologien unter Einsatz von Proxy-Funktionsblöcken, die eine „Stellvertreterfunktion“ dezentraler Komponenten bei der Abarbeitung der Steuerungsalgorithmen repräsentieren. Die Abarbeitung der Algorithmen wird in dezentralen Systemen zentral durch die verarbeitende prozessnahe Komponente realisiert.

Ein im Vergleich zu zentraler und dezentraler Struktur neues Architekturparadigma stellt der Entwurf von verteilten Systemen dar, der in FAVA untersucht werden soll. Erweiternd zu den vorher behandelten Lösungen werden die Steuerungsalgorithmen in verteilten Systemen auf mehrere Komponenten aufgeteilt, wodurch insbesondere neue Anforderungen an den Entwurf und bei der Koordination der Abarbeitung der Einzelfunktionen entstehen.

Das in FAVA zu betrachtende System ist eine Zusammenstellung von räumlich getrennten Automatisierungsgeräten (Komponenten) mit vorzugs-



## 5 Verteilt oder Zentral? Unterstützung für den Entwurf der Automatisierung durch das Projekt FAVA

---

weise digitaler Funktionsabarbeitung, die gemeinschaftlich Prozessbeobachtung, Steuerungs-, Regelungs- und Prozessbeeinflussungsaufgaben erfüllen. Dieses System wird als verteiltes Automatisierungssystem bezeichnet.

Verteilte Automatisierungssysteme (AT-Systeme) haben folgende spezifische Randbedingungen:

- Verteilte AT-Systeme sind heterogene Systeme bezogen auf die verfügbare Funktionalität ihrer Komponenten, d.h. AT-Systeme sind keine homogenen Systeme, wie z.B. PC-Cluster zur Abarbeitung komplexer Berechnungsaufgaben.
- Verteilte AT-Systeme bestehen aus reaktiven Komponenten, d.h. die Komponenten generieren basierend auf ihrem Verhalten selbsttätig Meldungen bzw. reagieren auf Ansprache mit einzelnen oder Folgen von Reaktionen.
- Verteilte AT-Systeme unterliegen Echtzeitanforderungen, d.h. ihre selbsttätigen Aktionen oder Reaktionen unterliegen zeitlichen Restriktionen.
- Verteilte AT-Systeme sind in gewissem Maße zuverlässig (beschrieben durch die Aspekte Reliability, Availability, Maintainability und Safety - RAMS), d.h. die Komponenten sind unter den gegebenen Einsatzbedingungen mit der geforderten Wahrscheinlichkeit funktionstüchtig.
- Verteilte AT-Systeme unterliegen zusätzlich zu den aus der Kommunikationstechnik bekannten QoS-Anforderungen nicht-funktionalen Qualitätskriterien, die im Entwurf berücksichtigt und, wo nicht vereinbar, gegeneinander abgewogen werden müssen.
- Verteilte AT-Systeme enthalten frei programmierbare, konfigurierbare oder parametrierbare Geräte, wie z.B. speicherprogrammierbare Steuerungen, Frequenzumrichter, Feldbusknoten, Sensoren und Aktoren. Diese Geräte können automatisierungstechnische Funktionen ausführen. Ihre Funktionalität wird mit Gerätemodellen (Parameterlistenmodellen, Parametergruppenmodellen, Funktionsbausteinmodellen oder Objektmodellen) beschrieben.

Die verteilten Automatisierungssysteme im Sinne von FAVA werden auf der Basis dieser Typen von Automatisierungsgeräten entworfen und betrie-

ben. FAVA zielt nicht auf die Veränderung der existierenden Gerätebasis ab, sondern auf eine geeignete Methode, verteilte Systeme mit den heute vorhandenen Komponenten zu realisieren.

### 5.5 Ausblick

Das Projekt FAVA fokussiert auf die Besonderheiten verteilter Automatisierungssysteme. Daher kommt es im Projekt zunächst darauf an, zu identifizieren, welche nicht-funktionalen Anforderungen in verteilten Systemen von besonderer Bedeutung sind – vielleicht, weil sie durch verteilte Systeme besonders gut erfüllt werden können, vielleicht aber auch, weil diese Anforderungen in verteilten Systemen schwieriger zu erfüllen sind als in zentralen oder dezentralen Systemen. Sind diese Kategorien von Anforderungen identifiziert, kann zunächst exemplarisch erarbeitet werden, welche Merkmale Komponenten und Kommunikation in einem verteilten System haben müssen, um diesen Anforderungen zu entsprechen. Diese Erkenntnisse sollen dann verallgemeinert werden, um zu Mustern zu gelangen, welche möglichst universell, d.h. nicht nur auf ein Teilanwendungsgebiet der Automatisierungstechnik beschränkt, eingesetzt werden können.

### 5.6 Referenzen

- [1] Einordnung und Bewertung von Beschreibungsmitteln, VDI/VDE-Richtlinie 3681, Oktober 2005.
- [2] Lastenheft / Pflichtenheft für den Einsatz von Automatisierungssystemen, VDI/VDE-Richtlinie 3694, Juni 2005.

# 6 Überwachung der Lieferkette zur flexiblen Fertigungssteuerung auf MES-Ebene

Raffaello Lepratti  
*Siemens AG – Industry Sectors*

Christoph Legat, Georg Heinecke, Steffen Lamparter  
*Siemens AG – Corporate Technology*

**Zusammenfassung:** Moderne Lieferketten sind aufgrund hoher Komplexität und enger Kopplung anfällig gegenüber Störereignissen. Präventive Maßnahmen zu deren Vermeidung sind dabei nicht in der Lage, deren Auftreten völlig auszuschließen. Daher besteht ein Bedarf an Systemen, die Störereignisse frühzeitig erkennen und unmittelbare Reaktionen einleiten, um negative Auswirkungen auf die Produktion zu minimieren bzw. im Idealfall zu vermeiden. In diesem Beitrag wird zu diesem Zweck das Konzept einer reaktiven Fertigungsanlage eingeführt, welche sich durch eine erhöhte Reaktivität auf Störeinflüsse in der Lieferkette im Gegensatz zu gängigen Lösungen auszeichnet. Dies wird zum Einen durch eine direkte Integration überbetrieblicher Objektverfolgungsdaten in das Manufacturing Execution System (MES) der Fertigungsanlage erreicht. Um in der Produktion effektiv auf Störungen reagieren zu können, muss zum Anderen die Produktionsanlage flexible gestaltet sein. Dies gilt sowohl bezüglich der Planung der Fertigungsaufträge als auch im Hinblick auf die Durchführung der Produktionsprozesse selbst. Im folgenden Beitrag werden Arbeiten der Siemens AG im Forschungsprojekt RAN<sup>2</sup> vorgestellt, welche zum Ziel haben, die beschriebenen Funktionalitäten auf MES-Ebene umzusetzen.

## 6.1 Einleitung

Lieferketten sind seit Jahren zwei wesentlichen Entwicklungen unterlegen [1]: Zum Einen führt die zunehmende globalisierte Vernetzung im Zusammenspiel mit der engen Kopplung der Unternehmen zu hoher, stetig steigender Komplexität. Dies führt zu immer höheren Koordinations-

---

<sup>2</sup> Das Projekt RFID-based Automotive Network (RAN) wird vom BMWi gefördert. Ausführlichere Informationen unter <http://www.auran.de>.

aufwänden innerhalb einer Lieferkette. Zum Anderen weisen Lieferketten durch Einführung von Prinzipien wie Lean Management und Lean Production eine zunehmende Verschlankung auf. Dies führt zu einer drastischen Reduktion von Zeit- und Bestandspuffern, und dadurch im Zusammenspiel mit der eingangs beschriebenen Komplexitätssteigerung der Lieferkette zu einer erhöhten Wahrscheinlichkeit, auftretende Störungen nicht mehr kompensieren zu können [2,3].

Folglich werden Systeme und Konzepte benötigt, die Störungen innerhalb der Lieferkette frühzeitig erkennen und deren Auswirkungen durch gezielte Eingriffe in Produktionsplanung und –durchführung<sup>3</sup> reduzieren [3]. Hierfür ist sowohl eine unternehmensübergreifende Auftragsverfolgung als auch ein flexibles Produktionssystem, das noch kurzfristig korrigierende Eingriffe erlaubt, notwendig. Daher wird im Rahmen dieses Beitrags eine Systemarchitektur vorgestellt, welche Objektverfolgungsdaten aus der Lieferkette zur Verfügung stellt und eine reaktive Produktionssteuerung ermöglicht.

### 6.2 Standardisierte Systemarchitektur

Die bisherigen Ausführungen haben aufgezeigt, dass die termingerechte, deterministische Planung, wie sie in Unternehmen angestrebt wird, durch stochastisch auftretende Störungen in der Lieferkette stark beeinflusst werden kann.

Deshalb wird in diesem Abschnitt zuerst die im Forschungsprojekt RAN entwickelte, auf etablierten Standards basierende Architektur zur überbetrieblichen Verfolgung von Objekten vorgestellt. Hierauf aufbauend wird die Verwendung der zusätzlich zur Verfügung stehenden Informationen zur Umsetzung einer reaktiven, auf Störungen dynamisch reagierenden Fertigungsanlage präsentiert. Die Realisierung erfolgt mittels eines Produktionsassistenzsystems (PAS), welches der Integration logistischer Überwachungsfunktionen in die Fertigungsleitebene dient. Ziel hierbei ist die frühzeitige Erkennung von Zulieferstörungen auf Basis der Objektverfolgungsdaten, der produktionsbezogenen Bewertung der Störungen, sowie die effektive Behandlung der Störungen durch Anpassung der Auftragsplanung bzw. der Produktionsprozesse.

---

<sup>3</sup> Im vorliegenden Beitrag wird mit Produktion das Beispiel der Automobilproduktion (Rohbau, Lackiererei, und Montage) gemeint.

## 6.2.1 Überbetriebliche Informationsplattform

Für die Erkennung von Zulieferstörungen ist ein hohes Maß an Transparenz in den Logistikprozessen erforderlich, die durch eine möglichst lückenlose Objektverfolgung gewährleistet werden kann; d.h. für jedes Produkt in der Lieferkette soll zu jedem Zeitpunkt dessen Ort und Zustand bekannt sein. Eine automatisierte Verfolgung wird durch die Kennzeichnung der Objekte mit einer eindeutigen Identifikationsnummer moderner Erfassungstechnologien ermöglicht. Um den Aufbau einer entsprechenden Infrastruktur zu erleichtern, werden im Projekt RAN Standard-Erfassungsklassen auf Basis innovativer RFID-Technologien definiert.

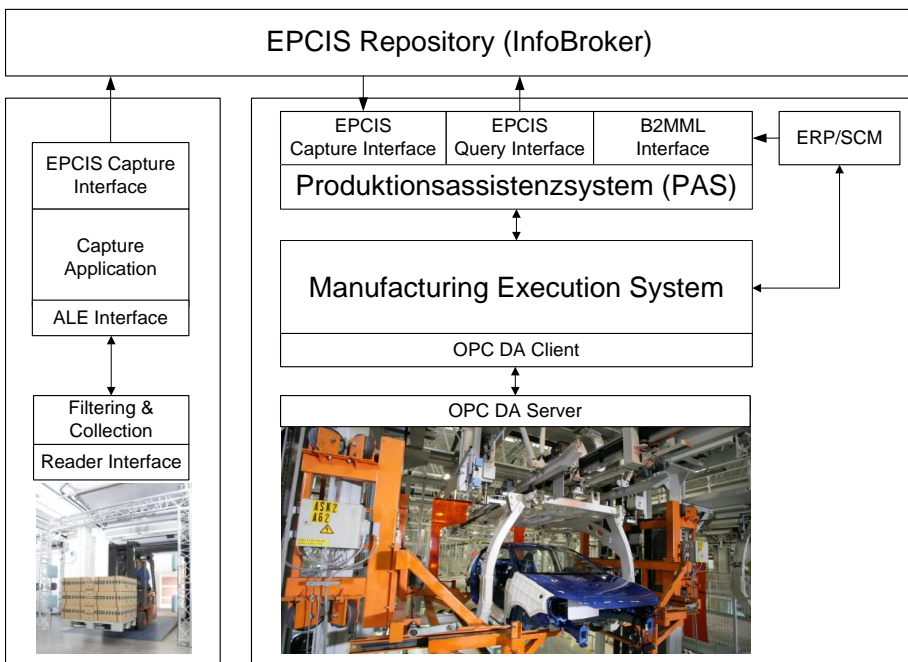


Abb. 6.1 – Überbetriebliche Systemarchitektur

Um die erfassten Objektverfolgungsdaten den in einer Lieferkette vertretenen Unternehmen zur Verfügung zu stellen, wurde zudem im Projekt RAN das Konzept des InfoBrokers entwickelt. Der InfoBroker bietet standardisierte Schnittstellen und Protokolle für den unternehmensübergreifenden Austausch von Objektverfolgungsdaten. Als Industriestandard für den Austausch solcher Daten hat sich in den letzten Jahren das

EPC Network von EPCglobal herauskristallisiert [4]. Die Berücksichtigung existierender Standards ist essentiell, um die Integration von neuen sowie bereits in der Lieferkette vertretenen Logistik- und Produktionssystemen einfach bewerkstelligen zu können. Die Standards des EPC Networks reichen von einem eindeutigen Nummerierungssystem für zu verfolgende physische Objekte, dem Electronic Product Code (EPC), über die für den Datenaustausch notwendigen Schnittstellen bis hin zu unternehmensübergreifenden Datenstrukturen. Letztere sind im EPC Information System (EPCIS) Standard [5] definiert und bilden die Grundlage für die in Abb. 6.1 dargestellte Architektur.

Grundsätzlich lässt sich die Architektur in drei Komponenten einteilen:

- InfoBroker: Der InfoBroker besteht typischerweise aus mehreren lokalen EPCIS konformen Datenspeichern, sogenannten EPCIS Repositories. Zudem sind zentrale Dienste vorhanden, die den Austausch der Daten zwischen den lokalen EPCIS Repositories koordinieren sowie Funktionalitäten wie Berechtigungs- und Zugangsmanagement bereitstellen. Für den Zugriff auf EPCIS Repositories stehen standardisierte Schnittstellen wie das EPCIS Capture Interface für das Hinzufügen von Ereignissen und das EPCIS Query Interface für die Datenabfrage zur Verfügung.
- Logistische Systeme: Hierunter fallen Komponenten für die physische Erfassung der Objektverfolgungsdaten (Capture Application in Abb. 6.1) sowie Systeme für die unternehmensübergreifende Überwachung und Steuerung der logistischen Prozesse unter Verwendung des InfoBrokers, sogenannte RAN Logistikassistenzsysteme.
- Produktionstechnische Systeme: Analog zu den logistischen Systemen können Objektverfolgungsdaten auch aus der Produktion erzeugt und mit Hilfe des InfoBroker zugänglich gemacht werden. Um die Verwendung von Objektverfolgungsdaten aus der Logistik in der Produktion für eine effizientere Steuerung verwenden zu können, wird im Projekt RAN die Entwicklung von Produktionsassistenzsystemen (PAS) vorangetrieben.

Im Folgenden soll der Fokus des vorliegenden Beitrags auf den Aufbau und die Funktionalität eines PAS gelegt werden, da die Integration logistischer

Information in die Fertigungssteuerung weder aus wissenschaftlicher Sicht noch in der praktischen Umsetzung ausreichend untersucht wurde.

### 6.2.2 Produktionsassistenzsystem (PAS)

Der EPCIS Standard adressiert ausschließlich die Versorgung der an einem Geschäftsvorgang beteiligten Unternehmen mit entsprechenden Objektverfolgungsdaten, jedoch nicht die anwendungsspezifische Verwendung dieser Daten. Letzteres erfolgt je nach Anwendung mittels Logistik- oder Produktionsassistenzsystemen. In diesem Beitrag werden Anwendungsszenarien eines Produktionsassistenzsystems (PAS) betrachtet, welches Objektbewegungsdaten für das Produktionsmanagement sowie die Produktionssteuerung nutzt.

Um eine Kopplung von Logistik und Fertigung zu erreichen, müssen unabhängig arbeitende Systeme und Standards in Beziehung gesetzt werden. Dies geschieht durch das PAS, welches, wie in Abb. 6.1 gezeigt, die EPCIS Query und Capture Schnittstellen implementiert und die so verfügbaren logistischen Daten für eine weitere Verwendung in der Produktion auswertet. Das PAS erweitert somit die Funktionalität eines Manufacturing Execution Systems (MES) gemäß ISA 95 bzw. IEC 62264 [6] um lieferkettenübergreifende Koordinationsaufgaben. Das MES dient als Schnittstelle zwischen Unternehmens- und Steuerungsebene, wodurch es u.a. Kundenaufträge aus den übergeordneten ERP-Systemen entgegennimmt und die entsprechenden Fertigungsaufträge auf die bestehenden Automatisierungsanlagen verteilt. Im Gegensatz zur ERP-Ebene, sind dadurch auf der MES-Ebene oft noch kurzfristig Anpassungen der Planung der Fertigungsaufträge möglich. Die Ansiedlung des PAS in dieser Ebene erlaubt somit auch noch Eingriffe bei Störungen kurz vor Produktionsbeginn. Wie groß die kurzfristigen Eingriffsmöglichkeiten sind, hängt hierbei maßgeblich von der Flexibilität der Automatisierungskomponenten sowie der Komplexität der Fertigung ab.

Es ergeben sich zwei zentrale Fragestellungen bei der Entwicklung eines PAS, die in den nächsten zwei Abschnitten adressiert werden:

1. Wie können für einen gegebenen Produktionsplan kritische Störungen möglichst frühzeitig erkannt werden?
2. Wie kann die Flexibilität der Produktion erhöht werden, um so auch kurzfristig auftretende Störungen noch kompensieren zu können?

### 6.3 Identifikation kritischer Ereignisse/Störungen

Zukünftig wird es ein Ziel sein, in zunehmendem Maße den gesamten Materialfluss der Lieferkette realitätsgetreu (d.h. möglichst in Echtzeit) über die Informationssysteme der jeweiligen Unternehmen abzubilden und die sich ergebenden Datenströme zu Überwachen.

Dieses Ziel ist jedoch in weiter Ferne. Transparenz der Lieferkette ist heute eher die Ausnahme als die Regel [7]. Somit klaffen heute große Lücken in den Überwachungsmöglichkeiten, sodass von einer durchgehenden Transparenz von Lieferketten noch nicht die Rede sein kann. Neben der heute noch problematischen Bereitstellung von Objektverfolgungsdaten durch die Lieferkettenpartner ist dieser Umstand insbesondere auf die anhaltende Skepsis der Praxis gegenüber einer durchgängigen Einführung (z.B. standortübergreifend) von AutoID-Technologien (z.B. RFID) zurückzuführen [7]. Gründe hierfür sind vielfältig, wie z.B. Datensicherheit und -besitz sowie finanzieller und operativer Mehrwert [8]. Dennoch wird die Technologie sich in Zukunft stärker durchsetzen, wodurch große Mengen an Objektverfolgungsdaten auf relevante Ereignisse hin überwacht werden müssen. Nur diese sollten einem Produktionsplaner zur Verfügung gestellt werden. Events gelten dabei dann als relevant, wenn sie Störungen beschreiben, die die Produktion maßgeblich negativ beeinflussen. Demnach können sie in drei Klassen eingeteilt werden [9]: Abweichungen, Störungen und Katastrophen (siehe Abb. 6.2). Auswirkungen können im Bezug auf den hier betrachteten Anwendungsfall der Zulieferung beispielsweise Qualitäts- (z.B. Unterschreitung von Toleranzen), Mengen- (z.B. Fehlteile) oder Terminabweichungen (z.B. Lieferverspätung) darstellen. Letztere werden häufig betrachtet, da sich die anderen Aspekte oft auf temporalen Aspekt zurückführen lassen. Abb. 6.2 stellt diese Tatsache im Bezug auf die eingangs beschriebenen Ereignisklassen dar, wobei als zeitlicher Aspekt die Durchlaufzeit von Prozessen gewählt wurde. Die verschiedenen Ereignisse resultieren aus unterschiedlich großen Abweichungen von Sollwerten einer Kenngröße. Dabei treten Abweichungen unmittelbar um den Sollwert auf, während sich Katastrophen in den höchstmöglichen Ausprägungen einer Kenngröße manifestieren.

Unabhängig von diesen Ereignisklassen, entsteht durch die registrierten Abweichungen eine Häufigkeitsverteilung der entsprechenden Prozesskenngröße, wie sie schematisch in Abb. 6.2 dargestellt ist. Auch wenn nach



## 6 Überwachung der Lieferkette zur flexiblen Fertigungssteuerung auf MES-Ebene

Möglichkeit jede Abweichung vom Soll adressiert werden sollte, um so nah wie möglich an die Planvorgaben für die Produktion heranzukommen, wäre ein solches Vorgehen dennoch fragwürdig. Zum Einen würden unverhältnismäßig viele Ressourcen gebunden werden und zum Anderen würde dies zu einer hohen Instabilität des Planungsbildes führen (d.h. häufige Planänderungen). Daher gilt, dass bei der Adressierung von Störungen Aufwand und Nutzen in einem angemessenen Verhältnis stehen müssen.

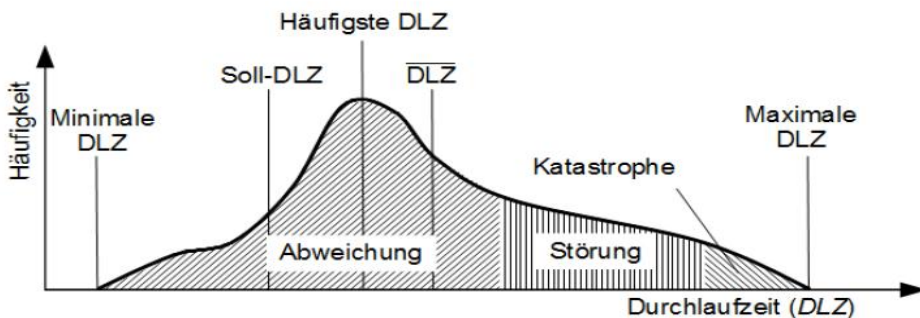


Abb. 6.2 – Zuordnung von Ereignisklassen im Bezug auf Durchlaufzeit [10]

Hierfür gibt es zwei unterschiedliche Ansätze. Erstens kann während der Umplanung die Relevanz von Abweichungen berücksichtigt werden um etwaige Überreaktionen zu vermeiden (siehe hierzu Abschnitt 6.4). Zweitens kann während der Überwachung eine frühzeitige Identifikation relevanter Abweichungen erfolgen. Dies geschieht in der Regel durch die Definition von Schwellwerten [10]. Dabei handelt es sich um einen vom Produktionsplaner definierten Toleranzbereich, innerhalb dessen eine Abweichung als unkritisch angesehen wird. Liegt allerdings die Abweichung über dem Schwellwert so impliziert dies schwerwiegende Auswirkungen und eine Handlung für deren Kompensation ist zwingend erforderlich. Eine derartige Handlung könnte zum Beispiel die Umplanung der Fertigungsaufträge sein, um einen Liefertermin noch einzuhalten.

In [10] konnte gezeigt werden, dass eine intelligente Bestimmung der Schwellwerte messbare Vorteile erzielen kann. Dabei wurde ein Verfahren vorgestellt, welches Schwellwerte auf Basis der Korrelation zwischen den Ist-Abweichungen an den jeweiligen Erfassungspunkten eines Systems mit den Ist-Abweichungen am Prozessende in Relation setzt. Eine Simulation zeigte, dass dieser Ansatz im Vergleich zur Referenzmethode ein deutlich

verbessertes Verhältnis von Fehlalarmen zu richtigen Identifikationen von kritischen Situationen erzielen kann. Vorteile des Verfahrens neben dessen Intuitivität sind: i) Anpassungsfähigkeit an Risikopräferenzen des Produktionsplaners und ii) Schwellwertverfeinerung sobald mehr Objektverfolgungsdaten vorliegen.

Im Falle, dass eine kritische logistische Störung aufgetreten und erkannt ist, kann ein PAS produktionsseitig Gegenmaßnahmen einleiten. Dies wird im folgenden Abschnitt erläutert.

### 6.4 Kompensation von Störungen

Während in der wissenschaftlichen Literatur eine Vielzahl komplexer Verfahren zur Zeitablaufplanung vorgeschlagen werden, basieren nur wenige auf realistischen Annahmen, die direkt in die Praxis umgesetzt werden können [10]. Grundsätzlich wird zwischen Offline- und Online-Ansätzen zur Ablaufplanung unterschieden [11], da sie die Unsicherheit von Störungen sowie deren Effekte auf verschiedene Arten berücksichtigen.

Ziel von Offline-Ansätzen ist im Allgemeinen die vorausschauende Berücksichtigung möglicher Probleme bereits während der Planung. Dies führt jedoch zumeist zu erhöhten Durchlaufzeiten auch ohne Störung, wie beispielsweise bei der Planung mit Schlupf. Während Unsicherheit durch präventive Maßnahmen berücksichtigt werden kann, können kritische Effekte von Störungen, wie z.B. Sequenzverwirbelungen, nur schwer ohne eine direkte Reaktion behandelt werden. Aus diesem Grund bieten Online-Verfahren die Möglichkeit, zur Laufzeit anpassende Eingriffe vorzunehmen. Solche reaktive Verfahren legen einen zuvor bestimmten Ablaufplan zu Grunde und passen diesem mittels unterschiedlicher Strategien sukzessive an. Dies geschieht, falls auf Grund von Störeffekten ein gegebener Fertigungsplan nicht mehr die geforderte Leistung erbringen oder in seiner geplanten Form nicht mehr durchgeführt werden kann. Somit kann negativen Effekten, welche nicht im Vorhinein berücksichtigt werden können, entgegen gewirkt werden. Die Stabilität der Fertigungspläne hängt hierbei stark davon ab, in wieweit die Unsicherheit von Störungen bereits bei der Planung berücksichtigt wurde. Neben der vorausschauenden Berücksichtigung von Störungen ist jedoch bei der Fertigungsplanung auch die Fähigkeit des genutzten reaktiven Verfahrens zu berücksichtigen, da hierdurch die Leistungsfähigkeit zur Laufzeit erheblich gesteigert werden kann. Aus diesem Grund wird in [12] ein bikriterielles Verfahren zur

Fertigungsplanung vorgeschlagen, welches neben der Termintreue, prozessbedingte Einschränkungen als Folge möglicher Eingriffe berücksichtigt, um den robustesten Fertigungsplan zu bestimmen. Dadurch können bereits bei der Umplanung vorausschauend weitere, notwendige Anpassungen Berücksichtigung finden.

Die Leistungsfähigkeit von Ablaufplanungsverfahren ist in hohem Maße von der Flexibilität der Produktionsanlage und somit direkt von der Adaptivität der Feldebene abhängig. Neben der Anzahl unterschiedlicher Betriebsmodi der Feldebene, ist es von wesentlicher Bedeutung, für auftretende Störungen adäquate Operationen zur Kompensation der jeweiligen Störung bereitzustellen. Aus diesem Grund müssen Verfahren entwickelt werden, um die Steuerung der Feldebene einerseits reaktiver zu gestalten und andererseits deren Fähigkeiten in adäquater Weise zu beschreiben. Dies ermöglicht dem Planungssystem eine verbesserte Nutzung verfügbarer Ressourcen. Hierfür wurde in [13] ein Verfahren vorgeschlagen um semantische Modelle der Feldebene zu verwalten und eine Integration in Produktionsassistenzsysteme zu ermöglichen.

Die Integration von Objektverfolgungsdaten der Lieferkette hat jedoch auch negative Effekte auf die Reaktivität der Produktionsplanung, da die Erkennung von Abweichungen bzw. Störungen eine Änderung der Planungsgrundlage zur Folge hat (siehe hierzu auch Abschnitt 6.3). Treten solche Veränderung, z.B. durch stetige Nachkorrekturen von Lieferterminen, sehr häufig auf, weisen viele reaktive Verfahren ein nervöses Verhalten auf. Dies manifestiert sich in einer stetigen Anpassung des Fertigungsplans. Häufig ist jedoch eine geringe Anzahl an Plananpassungen ausreichend um die Leistungsfähigkeit des Fertigungsplans wieder herzustellen. Hierzu wird in [14] ein bikriterielles, reaktives Verfahren vorgeschlagen, bei dem neben der Termintreue auch der zuvor gültige Schedule berücksichtigt wird. Erste Testergebnisse haben gezeigt, dass die Anzahl der Anpassungen deutlich reduziert werden kann, während nur ein geringer Leistungsverlust zu verzeichnen ist.

### 6.5 Prototypische Umsetzung

Abb. 6.3 zeigt eine prototypische Umsetzung der Ansätze anhand des Hardware-in-the-Loop Konzepts. Dieses beschreibt die Kopplung von realen mit simulierten Systemkomponenten; in diesem Fall von einer realen Testanlage mit einer simulierten Lieferkette. Ein derartiges System erlaubt es, Objekt-

verfolgungsdaten in der Lieferkette zu generieren, die durch ein PAS auf Störungen überwacht werden können (vgl. Abschnitt 6.3). Gleichzeitig ermöglicht die Einbindung einer realen Anlage, eine Reaktion auf die frühzeitig identifizierte Störung (vgl. Abschnitt 6.4). Hierdurch lassen sich die Funktionalitäten eines PAS unter realistischen Produktionsbedingungen und Hardwarevoraussetzungen untersuchen.

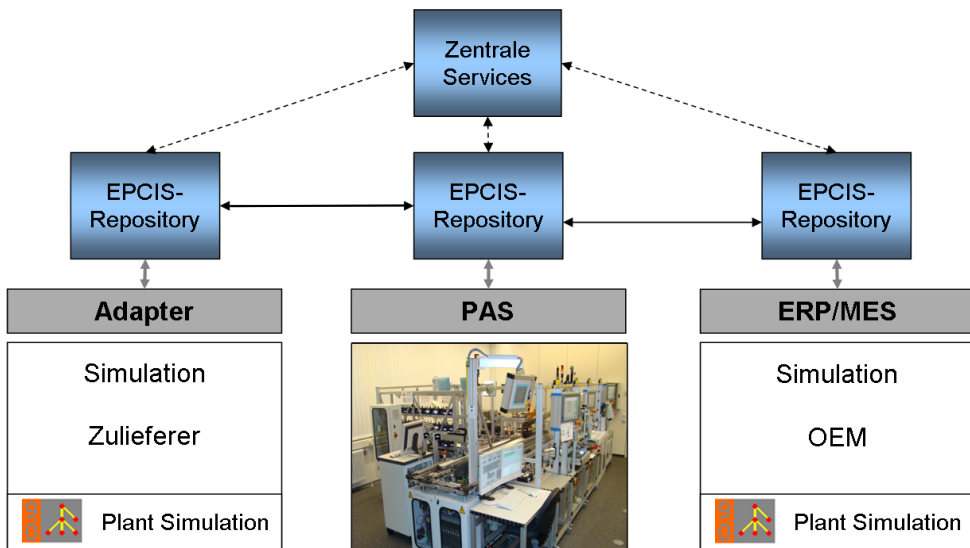


Abb. 6.3 – Architektur des Hardware-in-the-Loop Demonstrators

### 6.6 Zusammenfassung/Ausblick

In diesem Beitrag wurde die Notwendigkeit einer reaktiven Fertigungssteuerung auf MES-Ebene aufgezeigt um unternehmensinterne wie -externe Störungen und deren Effekte weitestgehend kompensieren zu können. Um Störungen innerhalb der Lieferkette frühzeitig erkennen zu können, wurde eine standardisierte Systemarchitektur vorgestellt, welche im Rahmen des Projekts RAN2 entwickelt wird und die Integration von Objektverfolgungsdaten in ein Produktionsassistenzsystem ermöglicht. Des Weiteren wurden Herausforderungen deren potenzielle Lösungen dargelegt, um kritische Situationen zu erkennen und auf diese in adäquater Weise zu reagieren.

### 6.7 Referenzen

- [1] Wagner, S. M.; Bode, C.: An empirical investigation into supply chain vulnerability. In: *Journal of Purchasing and Supply Management*, Jahrgang 12, Heft 6, 2006, S. 301-312.
- [2] Thun, J.-H.; Hoenig, D.: An empirical analysis of supply chain risk management in the German automotive industry. In: *International Journal of Production Economics*, 2009.
- [3] Knickle, K.: Supply Chain Event Management - The next best thing to supply chain perfection, *AMR Research Outlook*, 2001.
- [4] The EPCglobal Architecture Framework, EPCglobal Final Version 1.3, 2009.
- [5] EPC Information Services (EPCIS), EPCglobal Version 1.0.1, 2007.
- [6] Enterprise-control system integration, IEC 62264, 2003.
- [7] Günthner, W.; et al.: Stand und Entwicklung des RFIDEinsatzes in der Automobillogistik: Ergebnisse einer empirischen Studie," *RFID Anwenderzentrum München*, 2010.
- [8] Bensel, P.; et al.: Supply Chain Event Management: Entwicklung eines SCEM-Frameworks, *Digitale Schriftenreihe Logistik der Technischen Universität Berlin*, 2008.
- [9] Heinecke, G.; et al.: Produktionsbasiertes Management von Störereignissen: Spezifikation und Evaluation von Kritizitätsmodellen für die präventive, RFID-basierte Prozessüberwachung. In: *Automation 2011, Baden-Baden, Deutschland*, 2011.
- [10] Reisman, A.; et al.: Flowshop scheduling/sequencing research: A statistical review of the literature. In: *IEEE Trans. on Engineering Management*, Ausgabe 44, Heft 3, 1997, S. 316-329.
- [11] Raheja, A. S.; Subramaniam, V.: Reactive recovery of job shop schedules - a review. In: *International Journal of Advanced Manufacturing Technology*, Ausgabe 19, 2002, S. 756-763.
- [12] Legat, C.; Heinecke, G.: Dynamic Reconfiguration of Manufacturing Systems using its Control Capabilities. In:

- International Conference on Production Research, Stuttgart, Germany, 2011.
- [13] Legat, C. et al.: Service-oriented Product-driven Manufacturing. In: 10th IFAC Workshop on Intelligent Manufacturing Systems, Lisabon, Portugal, 2010, S. 161-166.
- [14] S. Lamparter et al., "Event-based Reactive Production Order Scheduling for Manufacturing Execution Systems," In: IFAC World Congress 2011, Mailand, Italien, 2011, akzeptierter Beitrag.

## 7 **MES Projekte effizient spezifizieren - Anforderungen an ein einheitliches Beschreibungsmittel für den Praxisalltag**

Steffen Himstedt

*Trebing & Himstedt Prozessautomation GmbH&Co.KG*

Maria Witsch

*Lehrstuhl für Automatisierung und Informationssysteme (AIS), TU München*

**Zusammenfassung:** Manufacturing Execution Systems gewinnen in den letzten Jahren zunehmend an Bedeutung. Um MES-Projekte erfolgreich realisieren zu können, sind jedoch einige Herausforderungen zu bewältigen, die in diesem Beitrag vorgestellt werden.

### 7.1 **Manufacturing Execution Systems als Schnittstelle zwischen IT und Produktion**

Der steigende Wettbewerbsdruck und immer strengere gesetzliche Vorgaben zwingen Unternehmen zur ständigen Optimierung, Überwachung und Dokumentation ihrer Produktionsprozesse. Manufacturing Execution Systems (MES) schließen die Lücke zwischen Management und Shopfloor. MES sind prozessnah operierende IT-Systeme, die die Unternehmensleitebene mit der Prozess-/Fertigungsebene verbinden, indem sie beispielsweise einerseits aus der großen Menge der Daten des technischen Prozesses in Echtzeit die für das Warenwirtschaftssystem (ERP) und Management wichtigen Informationen aggregieren, auswerten und für Entscheidungen zur Verfügung stellen und andererseits grobe Produktionspläne in eine operative Feinplanung umsetzen und ihre Durchführung steuern. Mit diesen Echtzeit-Daten können Unternehmen Anlagenzustände überwachen und schnell auf Änderungen reagieren. Die Informationen ermöglichen Führungsteams, Verbesserungsmöglichkeiten zu erkennen und schneller Lösungen für etwaige Probleme zu finden. Dadurch können Produktivitätssteigerungen, Verbesserungen der Produktqualität, eine höhere Maschinenauslastung und Kostensenkungen erreicht werden.

Ein weiterer Grund für den Einsatz von MES sind stark gestiegene gesetzliche Anforderungen hinsichtlich der Nachverfolgbarkeit und Transparenz des Produktionsprozesses insbesondere in der Nahrungs-, Genussmittel-, Chemie- und Pharmaindustrie. Zertifizierungen und die Einhaltung von Standards können nur durch die Automatisierung der Dokumentationsprozesse sichergestellt werden.

### **7.1.1 State of the art MES IT-Landschaft**

Die heutige IT-Landschaft in Unternehmen ist noch geprägt durch selbst oder von lokalen IT-Providern entwickelten MES-Lösungen. Die Auswahl der Systeme und ihrer Funktionen wurde zudem häufig auf Basis der Funktionalität und weniger aufgrund von Prozessanforderungen durchgeführt. Dabei wurden nur die Anforderungen einzelner Abteilungen oder Werke berücksichtigt, wodurch gleiche Funktionalitäten in unterschiedlichen Abteilungen oder Werken von unterschiedlichen Systemen bereitgestellt werden. Vorhandene MES-Lösungen sind somit oftmals auf Abteilungen oder Werksebene begrenzt.

Die Integration dieser einzelnen Systeme ist noch wenig vorangeschritten. Hingegen ist der Einsatz von Papier und Excel-Tabellen zur Informationsweitergabe noch weit verbreitet. Wurden bereits automatische Schnittstellen geschaffen, so bestehen diese meist nur als Point-to-Point-Lösungen zwischen zwei Systemen (ERP-MES; ERP-PLM; MES-PLM; ERP-QS etc.), wodurch sich eine Vielzahl von Schnittstellen ergeben. Je größer die Anzahl von IT-Subsystemen in einem Unternehmen, desto mehr Schnittstellen müssen geschaffen und gepflegt werden. Die Erfahrung der letzten Jahre zeigt, dass in Produktionsstätten zwischen 10 und 50, in Unternehmen 40 bis 700 verschiedene IT-Subsysteme implementiert sind. Viele dieser Systeme greifen wiederum auf verschiedene und uneinheitliche Datenbanken zu, sodass ein hoher Aufwand getätigt werden muss, um die Konsistenz der Daten sicherzustellen.

### **7.1.2 State-of-the-Art in MES-Projekten und Business Decisions bei MES-Anwendern**

Die Entscheidungsfindung (Lieferantenauswahl) findet meist auf betrieblicher Ebene durch die Automatisierung oder den Fertigungsleiter statt. Hierdurch entsteht ein ständiger Konflikt in Bezug auf den Funktionsumfang der Systeme und der Verantwortung für die Systeme zwischen der



zentralen IT-Abteilung des Unternehmens, die die Verantwortung für das unternehmensweite ERP System trägt und den lokalen MES-Abteilungen, die mit ihren MES Systemen nicht nur auf die Plandaten der ERP Systeme zugreifen, sondern auch an diese zurückmelden und mehr und mehr auch funktionale Überschneidungen zu diesen Systemen haben.

MES Projekte sind in Unternehmen zudem häufig nebenläufige Projekte, die parallel zum Tagesgeschäft durchgeführt werden müssen. Dies bedingt einen Mangel an Ressourcen während der Ausführung der MES-Projekte mit der Folge von verlängerten Projektlaufzeiten und oft verbunden mit einer deutlichen Überschreitung des ursprünglich geplanten Budgets.

Durch die Definition einer globalen MES-Strategie und der Fokussierung auf einen globalen MES Anbieter, können die Aufwände für die MES-Einführung, den Betrieb und die Wartung gesenkt werden. Eine generisches MES Template und ein Standard Projektvorgehen können die Aufwände zusätzlich reduzieren. Um die Anbindung des MES an die Automatisierungsebene zu erleichtern, ist die Definition globaler Standardconnectoren wichtig. Neben diesen Standardisierungen müssen werkspezifische Anforderungen dennoch Berücksichtigung finden, die Mitarbeiter vor Ort in die Planungsprozesse mit einbezogen und durch Schulungen auf die Implementierung vorbereitet werden.

Eine häufige Ursache für Erfolgsminderungen ist eine mangelnde Prozess-Modellierung und Dokumentation von Beginn der Projekte an. Von der Anforderungserhebung, der Lastenhefterstellung, den Systemtests bis zur Handbucherstellung könnten durchgängige Modelle der MES Funktionalitäten, ihrer Einbindung in die IT-Landschaft des Unternehmens und ihrer Anbindung an die Prozessebene Missverständnisse vermeiden, interdisziplinäre Entscheidungen dokumentieren, Optimierungspotentiale aufzeigen, die Definition von Schnittstellen vereinfachen und Planungssicherheit für die Implementierung geben.

### **7.2 Herausforderungen in MES-Projekten für MES Anbieter**

Die wesentliche Herausforderung bei der Applikation von MES besteht in der Integration von Wissen, welches in verschiedenen Fachabteilungen unterschiedlicher Unternehmen in Form von unterschiedlichen Gedankenmodellen vorhanden ist. Die Erhebung des Prozess-Wissens und der An-

forderungen an ein MES ist ein unumgänglicher, aber heute für den MES-Anbieter langwieriger und schlecht kalkulierbarer Prozess. Wenn die Prozesse des MES-Kunden nur schlecht beschrieben oder dokumentiert sind oder sogar ganze Prozesse unklar sind, dann können die Projekte durch den MES-Anbieter verlässlich nur mit Budget nach Aufwand abgerechnet und die Kosten vorab lediglich grob abgeschätzt werden. Ungenauigkeiten/Schwankungen von +/- 20 % können dabei nicht ausgeschlossen werden. Ein Festpreis kann vorab nicht oder nur unter sehr hohem Risiko des MES-Anbieters genannt werden. Daher ist für die für beide Seiten verlässliche Angebotserstellung eine vorherige Prozessanalyse unerlässlich.

Diese Prozess- und Anforderungsanalyse beinhaltet dabei auch die eindeutige und vollständige Dokumentation der Anforderungen in einem Lastenheft.

Erhält der MES Anbieter eine unvollständige oder inkonsistente Darstellung der Lasten des MES Projektes, bleibt für ihn das Risiko, dass Lastenhefte nach Auftragsannahme häufig aufwendig detailliert werden müssen, da wichtige Informationen vergessen wurden und Prozessbeschreibungen lückenhaft sind. Insbesondere rein textuelle Spezifikationen suggerieren eine Vollständigkeit und Eindeutigkeit, die sie in der Pflichtenheft- und Umsetzungsphase eines Projektes oft nicht halten können.

Weitere Herausforderungen in MES-Projekten sind für den MES Anbieter wie für den MES-Anwender die Identifikation und Interpretation der Datenpunkte und ihrer Werte, importierten Daten sowie die Definition der benötigten Kennzahlen, die aufgrund unterschiedlicher Wünsche hinsichtlich ihrer Granularität (Maschinensicht/Teilanlagensicht etc.) auf die Prozesses des Kunden abgebildet werden müssen.

Durch die Entwicklung eines, für alle am Entwicklungsprozess beteiligten Mitarbeiter verständlichen Beschreibungsmittels, welches durch ein Modellierungswerkzeug unterstützt wird, kann die Applikation von MES zielgerichteter und schneller durchgeführt werden. Das notwendige Wissen kann transparent in einem allgemeinverständlichen Modell gesammelt und für die Entwicklung des MES ausgewertet werden. Durch die Visualisierung und gemeinsamen Betrachtung von Prozessen werden zudem unterschiedliche Ansichten der einzelnen Abteilungen oder Mitarbeiter aufgedeckt und somit Missverständnisse vermieden. Solch eine Beschreibungssprache muss jedoch aufgrund des interdisziplinären Umfelds von MES und seiner

Einbindung in die meist komplexen IT-Landschaften eines Unternehmens besondere Anforderungen erfüllen.

### **7.3 Anforderungen an eine Beschreibungssprache für den Praxisalltag**

Um die sehr verschiedenen Gruppen an der Spezifikation und damit der Diskussion vor der Implementierung des MES zu integrieren, haben sich grafische Beschreibungsmittel, wie die Business Process Modeling Notation, als besonders effizient erwiesen. Dabei ist es nicht notwendig, das Rad völlig neu zu erfinden. Es ist aber unerlässlich, die für eine vollständige Spezifikation benötigten Informationen zu identifizieren, darauf aufbauend die richtigen Beschreibungssprachen auszuwählen, die eine vollständige Abbildung dieser Informationen ermöglichen, sie für die Domäne MES anzupassen und einfach anwendbare Konzepte für eine interdisziplinäre Diskussion zu integrieren.

Da textuelle Spezifikationen in der Regel unvollständig und unpräzise sind, ist es notwendig, dass Funktionalitäten und Abhängigkeiten des MES und der anderen Systeme in ihrer Struktur und ihrem Ablauf grafisch dargestellt werden. Hierdurch wird eine Transparenz und Durchgängigkeit erreicht, die in der Umsetzungsphase der Projekte viel Zeit und Aufwand für nachträgliche Anpassungen spart.

Eine Beschreibungssprache für die Spezifikation muss interdisziplinär und intuitiv verständlich sein, damit sie bei der Anforderungserhebung im Unternehmen von allen Beteiligten verstanden und akzeptiert wird. Wichtig ist dabei auch, dass die Sprache nicht zu komplex ist, MES-Projekte meist neben dem Tagesgeschäft durchgeführt werden. Daher ist die Zeit, in der alle für die vollständige Prozessanalyse und Anforderungserhebung notwendigen Personen gemeinsam an der Spezifikation arbeiten können, sehr begrenzt. Für die Einarbeitung steht somit nur wenig Zeit zur Verfügung. In Workshops muss die Beschreibungssprache daher schnell anwendbar sein. Weiterführende Details, die eine umfangreichere Beschreibungssprache erfordern, können dann von einzelnen Experten im Verlauf der Anforderungserhebung integriert werden. Zudem müssen einzelne Diagramme schnell vom Betrachter erfasst werden können, da die Workshopteilnehmer oft durch Unterbrechungen des Tagesgeschäfts zwischenzeitlich abwesend sind und anschließend wieder in die Diskussion zurückfinden müssen.

Die Abhängigkeiten zwischen dem Produktionsprozess und den IT-Systemen müssen in der Spezifikation deutlich werden, um auch zeitliche Aspekte beispielsweise hinsichtlich der Visualisierung von Kennzahlen parallel zu bestimmten Prozessschritten vorab auf die Verfügbarkeit der benötigten Daten hin überprüfen zu können. Zudem müssen auch die Abhängigkeiten der IT-Systeme funktional wie datentechnisch abgebildet werden können, damit ein für das Unternehmen optimaler Funktionsumfang des MES in Anbetracht der vorhandenen Systeme und eine sinnvolle Einbindung in die IT-Landschaft definiert werden kann.

Bei der Modellierung der Anforderungen an das MES und der Ist-Analyse des Produktionsprozesses ist eine Unterscheidung zwischen implementierten Ist-Prozessschritten und Funktionen, neu definierten Soll- Prozessschritten und Funktionen und ausgeschlossenen Soll- Prozessschritten und Funktionen notwendig. Hierdurch wird eine direkte Dokumentation der Diskussionen im Workshop ermöglicht, die auch im späteren Verlauf des MES-Projekts noch nachvollziehbar ist. Zudem kann ein Abgleich zwischen Soll- und Ist-Prozess die Wirtschaftlichkeitsberechnung der MES-Einführung unterstützen und eine fortlaufende Pflege der Modelle den Aufwand für die Dokumentation der Systeme und Planung von Weiterentwicklungen reduzieren.

### **7.4 Zukünftige Herausforderungen bei der Spezifikation von MES**

Durch den vermehrten Einsatz von Labels und RFID-Tags in der Produktion und Logistik können mehr Informationen im Produkt selbst eingebettet werden, wodurch insbesondere die Rückverfolgbarkeit unterstützt wird. Die dadurch vermehrten und schneller vom Prozess zur Verfügung gestellten und zu verarbeitenden Daten müssen bei der Modellierung effizient abgebildet werden können. Die steigende Komplexität der Produkte und zunehmende Geschwindigkeit der Produktentwicklung verstärken die Notwendigkeit einer flexiblen Produktion und vollständigen, strukturierten, leicht verständlichen und leicht anpassbaren Dokumentation der Prozesse und Systeme.

Die in diesem Artikel angeführten Herausforderungen waren Grundlage der Entwicklung einer grafischen Beschreibungssprache in dem Projekt SpeziMES, welches im folgenden Kapitel 8 vorgestellt wird.

## 8 **MES-Modeling Language – Eine Beschreibungssprache für die interdisziplinäre Anforderungserhebung und Spezifikation von MES**

Maria Witsch,

*Lehrstuhl für Automatisierung und Informationssysteme (AIS), TU München*

**Zusammenfassung:** Die MES-Modeling Language ist eine interdisziplinär verständliche Beschreibungssprache für die Anforderungserhebung und Spezifikation. Sie wurde im Rahmen des Projekts SpeziMES entwickelt und evaluiert. In diesem Beitrag wird der Kern der Beschreibungssprache vorgestellt sowie eine kurze Zusammenfassung der Evaluationsergebnisse gegeben.

### 8.1 **Einleitung**

Im Maschinen- und Anlagenbau werden Manufacturing Execution Systems (MES) eingesetzt, um die Unternehmensleitebene mit der Prozess-/Fertigungsebene zu verbinden und somit die Anlagenproduktivität sowie die Prozessqualität zu steigern. In MES-Projekten sehen sich Anwender wie Anbieter vielen Herausforderungen gegenüber, die sich insbesondere aus der Schnittstelleneigenschaft eines MES und den daraus resultierenden Aspekten der Komplexität, Abhängigkeiten zwischen Systemen und Prozessen sowie der notwendigen Interdisziplinarität des Engineeringprozesses ergeben. Die Integration des komplementären Wissens aus der Produktion und IT zu einer gemeinsamen Wissensbasis stellt dabei eine besondere Herausforderung dar.

Für die interdisziplinäre Diskussion ist ein für alle Teilnehmer schnell verständliches und einfach anwendbares Kommunikationsmittel von besonderer Bedeutung. Diese Eigenschaft erfüllen intuitiv verständliche grafische Beschreibungsmittel am Besten. Bestehende Beschreibungssprachen sind jedoch auf eine spezielle Sicht der Anlage oder der Geschäftsprozesse beschränkt.

Um diesen Herausforderungen, wie u. A. in [1, 2] ausführlich beschrieben, zu begegnen, wurde in dem Projekt SpeziMES eine Beschreibungssprache entwickelt, die es ermöglicht, den Ist-Stand der Produktions- und Geschäftsprozesse und IT-Systeme im Unternehmen vor Beginn des MES Projektes, die Sollprozesse und Anforderungen an die IT-Systeme und das MES sowie deren Schnittstellen zu beschreiben und nach Implementierung zu dokumentieren. Diese Beschreibungssprache stellt die Systeme und Prozesse transparent dar und ist für alle am Entwicklungsprozess beteiligten Personengruppen leicht verständlich.

Das zweijährige Projekt wurde von 2009 bis 2011 in Kooperation mit der BN Automation AG und der Trebing & Himstedt Prozessautomation GmbH & Co.KG durchgeführt und durch das Bundesministerium für Bildung und Forschung (BMBF) gefördert.

Die entwickelte Beschreibungssprache, die MES-Modeling Language (MES-ML) wird im Folgenden vorgestellt und anschließend eine kurze Zusammenfassung der Ergebnisse der Evaluationsprojekte gegeben.

### 8.2 Drei Sichten für die Spezifikation von MES

Für die interdisziplinäre Prozessanalyse und Anforderungsspezifikation in MES-Projekten ist eine einerseits möglichst einfache und interdisziplinär verständliche, aber andererseits für die vollständige Beschreibung eines MES, seiner Interaktionen und Abhängigkeiten geeignete Beschreibungssprache notwendig. Sie muss die Möglichkeit bieten, alle notwendigen Informationen vollständig und dennoch übersichtlich darzustellen.

Die MES-ML bietet drei Sichten auf die Anlage (siehe Abb. 8.1) und bildet diese jeweils in einem eigenständigen Modell ab.

1. Das **Modell des technischen Systems** bildet die Struktur der Anlage hierarchisch als Baumdiagramm ab.
2. Das **Modell des Prozessablaufs** ermöglicht eine integrierte Abbildung des Produktionsprozesses, des Geschäftsprozesses und der Prozesse zur Erhaltung des Betriebszustandes.
3. Das **Modell des MES** bildet das MES, seine Funktionalitäten, die Funktionalität angrenzender IT-Systeme und die Schnittstellen zwischen den Systemen ab.

## 8 MES-Modeling Language – Eine Beschreibungssprache für die interdisziplinäre Anforderungserhebung und Spezifikation von MES

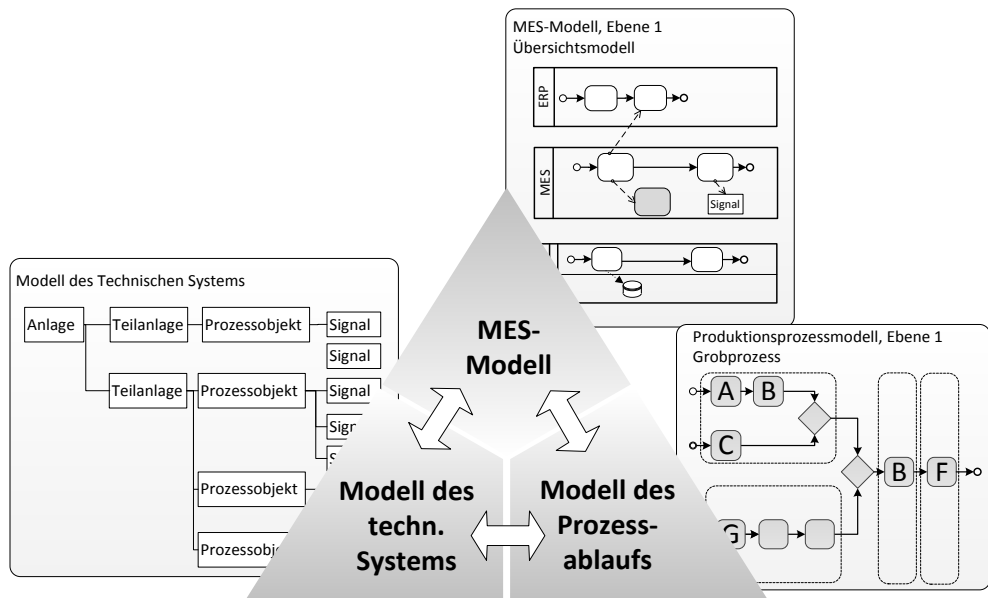


Abb. 8.1 – Grafische Modelle der MES-ML

Über sogenannte Verknüpfungen vom Typ „Datenaustausch“, „Äquivalenz“ oder „Ausführung“ können die drei Modelle miteinander verknüpft und somit die Abhängigkeiten modelliert werden.

Die Anwendung der Beschreibungssprache wird durch ein Vorgehensmodell und einen Softwareprototypen unterstützt. Die drei Modelle werden im Folgenden vorgestellt.

Jedes der im Folgenden vorgestellten Elemente besitzt Attribute, die es näher beschreibt und die in den Eigenschaften definiert sind. Die Attribute werden in den Eigenschaften der Elemente gespeichert, jedoch nur teilweise grafisch im Diagramm dargestellt. So besitzen IT-Aktivitäten beispielsweise das Attribut „Subtype“, welches angibt, ob es sich um eine „undefined“, „automatic“ oder „userinteraction“ handelt. Automatische IT-Aktivitäten („automatic“) werden durch einen Kegel und Nutzerinteraktionen („userinteraction“) durch eine Person symbolisiert (vgl. hierzu auch die Vorstellung der zentralen Elemente des MES-Modells). Eine grafische Darstellung wurde nur für die Attribute definiert, deren Ausprägung eine Aus-

wirkung auf das Verständnis des Modells hat. Wenn der Subtyp undefiniert ist („undefined“), wird somit kein zusätzliches Icon angezeigt.

Da die Beschreibungselemente des MES- und Produktionsprozessmodells bis auf wenige Unterschiede gleich sind und sich grafisch nur durch eine unterschiedliche Hintergrundfarbe unterscheiden, werden nach der Vorstellung des Modells des technischen Systems zuerst die MES-Modell-Elemente vorgestellt und anschließend die Unterschiede im Produktionsprozessmodell erläutert, ohne alle Elemente erneut vorzustellen.

Da die Diagramme intuitiv verständlich sein sollen, soll es keine impliziten, nicht modellierten Abläufe geben. Alle für das Verständnis des Diagramms notwendigen Informationen sollen abbildbar sein.

## **8.3 Modell des technischen Systems**

### **8.3.1 Aufgabe des Modells in der MES-ML**

Das Modell des technischen Systems dient der Abbildung der Struktur der Anlage, angefangen von dem Ort der Anlage, über seine Linien, Module und Maschinen, bis hin zum einzelnen Sensor und Signal. Es informiert somit darüber, wie die Anlage aufgebaut ist, aus welchen Modulen die Anlage besteht, welche Signale durch die Anlage bereitgestellt werden und welche Signale somit von dem MES verwendet werden können.

Die Gliederung der Anlage ist hierarchisch, wie aus Anlagenkennzeichnungssystemen bekannt.

### **8.3.2 Modellierungselemente des Modells des technischen Systems**

Das Modell des technischen Systems soll eine hierarchische Struktur abbilden. Eine der einfachsten und zugänglichsten Möglichkeiten, um dies zu tun, ist die Verwendung eines Baumdiagramms. Ein Baumdiagramm besteht aus Knoten und Kanten, wobei die Kanten die verschiedenen Knoten miteinander verknüpfen. Die einzelnen Knoten sind dabei hierarchisch in mindestens vier Gliederungsstufen angeordnet. Diese Gliederungsstufen heißen Anlage, Teilanlage und Prozessobjekt (Plant, Area und Unit), wobei Anlage die oberste Stufe und größte Gliederung der Anlage und Prozessobjekt die unterste und detaillierteste Gliederung der Anlage darstellt (siehe Abb. 8.2). Alle Prozessobjekte kennen die ihnen zugehörigen



## 8 MES-Modeling Language – Eine Beschreibungssprache für die interdisziplinäre Anforderungserhebung und Spezifikation von MES

---

Signale. Auf gleicher Gliederungsstufe wie der Teilanlage kann auch die Automatisierungsarchitektur und ihre Hierarchie abgebildet werden.

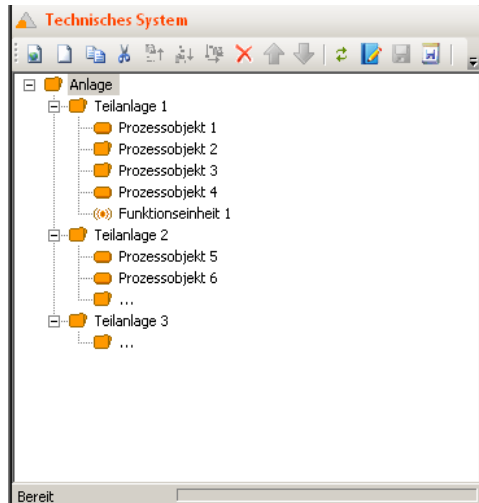


Abb. 8.2 – Gliederung des Baumdiagramms zum technischen System

Es können bei Bedarf weitere Gliederungsstufen eingefügt werden, wobei die Prozessobjekte in jedem Ast auf der gleichen Gliederungsstufe zu finden sein müssen. Gegebenenfalls werden Dummy-Gliederungsstufen eingeführt. Das Baumdiagramm wird von links nach rechts detailliert. Prozessobjekte besitzen Signale, die angeben, welche Datenpunkte zu einem Prozessobjekt gehören. Die Signale werden nicht in dem Baumdiagramm dargestellt, sondern als Eigenschaften zu den Objekten gespeichert.

Das Baumdiagramm des technischen Systems besitzt genau einen obersten Knoten vom Typ „Anlage (Plant)“. In der nächsten Gliederungsstufe können beliebig viele Knoten folgen, solange alle mit dem obersten Knoten über Kanten verbunden sind.

Neben der Benennung von Prozessobjekten mit realen Signalen können auch Funktionseinheiten (Verarbeitungsfunktionen), die mehrere reale Signale zu einem oder mehreren „virtuellen Signale/n“ kombinieren im Baumdiagramm dargestellt werden. Sie werden als Funktionseinheiten beschrieben und damit als normale Objekte behandelt. Die zu kombinierenden Signale werden unter der Eigenschaft „externe Signale“, die berechneten Werte als „interne Signale“ aufgeführt.

## 8.4 Modell des MES

### 8.4.1 Aufgabe des Modells in der MES-ML

Das MES-Modell bildet den zentralen Kern der MES-ML. Das MES-Modell stellt eine Erweiterung und gleichzeitig Selektion der Beschreibungselemente der Business Process Model and Notation (BPMN) [3] zur Abbildung produktionsnaher und technischer Informationen dar. Zu-dem wurden Modellierungskonventionen definiert, die eine unterschiedliche Modellierung gleicher Sachverhalte einschränken bzw. ausschließen.

### 8.5 Beschreibung der Elemente des MES Modells

Die folgende Abbildung zeigt die Notation für die einzelnen Modellelemente des MES Modells.

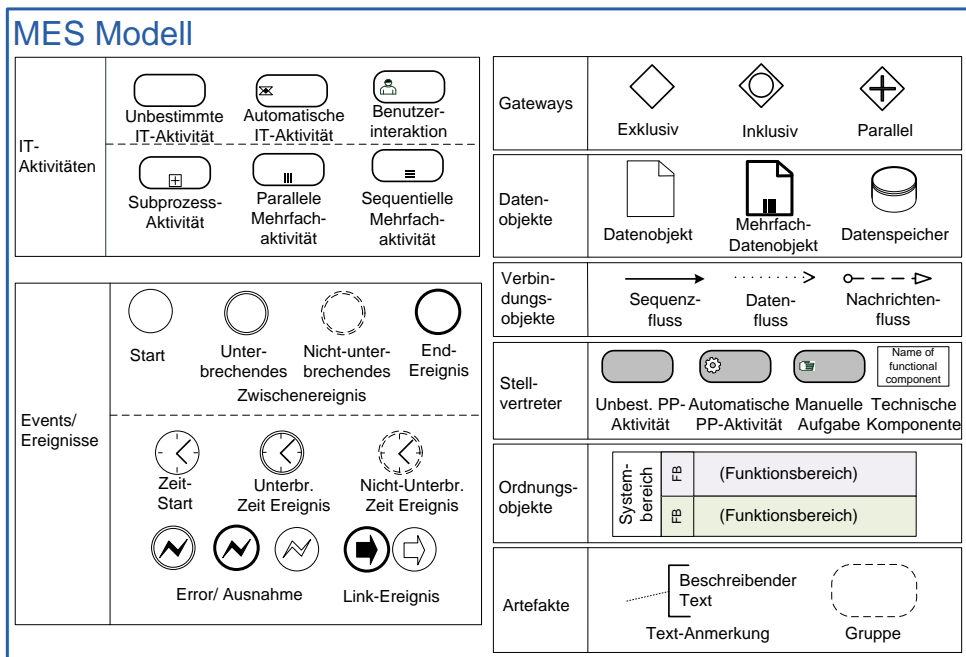


Abb. 8.3 – Notation des MES-Modells

## 8.5.2 Ordnungsobjekte

### Diagramm

Das Diagramm stellt die Zeichnungsfläche dar und besitzt keine grafische Notation. Das Diagramm ist somit ein Container, der die anderen Elemente enthalten kann. Es gibt einen Zeichnungsrahmen in A4 Größe vor, dessen Grenzen jedoch überschritten werden können.

### Systembereiche

Ein Systembereich repräsentiert ein abgrenzbares System und bestimmt somit die Verantwortlichkeiten für die modellierten Funktionalitäten. Bei der Spezifikation von MES brauchen die neben dem MES abzugrenzenden Systeme wie beispielsweise ein Enterprise Resource Planning (ERP) nur ausschnittsweise soweit relevant modelliert werden. Lediglich die Funktionalitäten oder Geschäftsprozesse, die einen Einfluss bspw. durch Datenbereitstellung auf die Ausführung des MES haben, sollen abgebildet werden.

Zwischen mehreren Pools muss sich ein Freiraum befinden, um die Eigenständigkeit der Systeme auch optisch darzustellen. Die Anordnung der Systembereiche (Über-/ Unterordnung) macht keine Aussage über die Hierarchie bzw. Wichtigkeit der Systeme. Zwar ist ein ERP einem MES übergeordnet, jedoch wird es auch Systeme geben, die übereinander abgebildet werden, obwohl sie in der Systemarchitektur auf gleicher Ebene angeordnet sind.

### Funktionsbereiche

Ein Funktionsbereich ist ein Unterbereich eines Systembereichs und stellt Teilsysteme/Module des Systems (wie beispielsweise ein Modul zur „Betriebsdatenerfassung“ für ein MES) dar. Funktionsbereiche stellen unterschiedliche Verantwortungen innerhalb eines Systembereichs dar. Sie besitzen jeweils eine andere Farbe, die jedoch für innerhalb eines Modells für die Funktionsbereiche konstant bleibt, sodass in Detaildiagrammen die Funktionsbereiche direkt wiedererkannt werden können. Funktionsbereiche füllen einen Systembereich immer vollständig aus. Funktionen, die für das gesamte MES relevant sind, werden in einen eigenen Funktionsbereich, z.B. „Infrastruktur“, eingefügt.

### 8.5.3 Verbindungsobjekte

#### Sequenzfluss

Der Sequenzfluss (Sequence Flow) verbindet Verhaltensobjekte eines Systembereichs miteinander und bestimmt damit die Reihenfolge dieser. Er darf somit Funktionsbereichs- aber keine Systemgrenzen schneiden.

#### Nachrichtenfluss

Der Nachrichtenfluss (Message Flow) stellt den Informationsfluss zwischen Systembereichen dar. Er kann nur Elemente unterschiedlicher Systembereiche, Elemente mit anderen Systembereichen oder unterschiedliche Systembereiche selbst miteinander verbinden. Dabei können jedoch ausschließlich Verhaltensobjekte, Datenspeicher oder Systembereiche Ziel oder Quelle eines Nachrichtenflusses sein. Einzel- oder Mehrfach-Datenobjekte werden durch Assoziationen dem Nachrichtenfluss angehängt. Ein Nachrichtenfluss hat selbst die Information über Sender, Empfänger sowie über das ihm assoziierte Datenobjekt über die Art der Nachricht („Request“ erwartet eine Antwort, „Response“ ist eine Antwort, „Sendung“ wird nur gesendet ohne Rückantwort zu erwarten).

#### Datenfluss

Datenflüsse stellen den Fluss von Daten in einem Prozessablauf innerhalb eines Systembereichs dar, das heißt sie sind immer mit einem Datenobjekt oder Mehrfach-Datenobjekt verknüpft. Da für ein MES und seine umliegenden Systeme fast immer getrennte Datenquellen vorhanden sind und sogar dass MES häufig unterschiedliche Datenquellen kennt, kann bei der Modellierung des MES-Modells nicht davon ausgegangen werden, dass alle Daten an jeder Stelle des Prozesses für alle Systeme zur Verfügung stehen. Daher ist die Modellierung des Datenflusses wichtig. Datenbezogene Abhängigkeiten können so erkannt werden. Auch bei einer gemeinsam genutzten Datenbasis kann so sichergestellt werden, dass im Prozessverlauf benötigte Daten vorab erzeugt werden.

#### Assoziation

Assoziationen werden entweder verwendet, um Texte/Kommentare mit Verhaltensobjekten zu verbinden oder um Einzel- oder Mehrfachdatenobjekte mit Daten- oder Nachrichtenflüssen zu verbinden.

## 8.5.4 Verhaltensobjekte

### Aktivitäten

Aktivitäten beschreiben Funktionalitäten von Systemen bzw. Schritte eines Prozessablaufs. IT-Aktivitäten bilden die gewünschten Funktionalitäten des MES und der umliegenden IT-Systeme ab.

Eine Aktivität muss immer genau einem Funktionsbereich zugeordnet sein.

Aktivitäten können durch Attribute näher beschrieben werden. So können sie durch Definition ihres Attributs „Subtype“ eine andere Bedeutung erhalten. Der Subtyp einer Aktivität wird durch ein entsprechendes Icon mittig auf der linken Seite symbolisiert.

Eine Aktivität vom Subtyp „automatische IT-Aktivität“ ist eine durch das MES ohne den Eingriff von Personen durchgeführte Aktion (bspw. automatische Kennzahlenberechnung). Eine Benutzerinteraktion ist eine Aktion, die durch Personen im Rahmen des Produktions- und Geschäftsprozesses, in Interaktion mit einem IT-System durchgeführt wird.

Subprozess-Aktivitäten sind Aktivitäten, die in einem weiteren Diagramm, einem sogenannten Subprozessdiagramm, detailliert werden (Dekomposition). Hierfür stehen innerhalb des Systembereichs die gleichen Modellierungselemente zur Verfügung wie für das Hauptdiagramm.

Subprozesse können immer nur in einem Systembereich mit seinen enthaltenen Funktionsbereichen eingesetzt werden. Dateneingänge aus anderen Funktionsbereichen und Datenausgänge in andere Funktionsbereiche/Systembereiche können dennoch als Datenflüsse oder Nachrichtenflüsse modelliert werden. Ihr Ursprung ist jedoch nur im übergeordneten Diagramm (in dem die Subprozessaktivität (Aktivität mit PLUS) modelliert ist) sichtbar.

Aktivitäten können in einem Prozess mehrfach wiederholt werden. Dabei ist einerseits die wiederholte nacheinander Ausführung einer Aktivität (Schleifenaktivität) jeweils auf Basis des Ergebnisses der vorherigen Ausführung ihrer selbst bis zur Erfüllung einer Schleifenendbedingung und andererseits die sequentielle oder die parallele Mehrfachausführung einer Aktivität (Mehrfachaktivität) möglich, beispielsweise für einzelne Stellen eines Dateninputs. (siehe Abb. 8.4) Der wesentliche Unterschied zwischen

einer Schleifenaktivität und einer Mehrfachaktivität besteht somit darin, dass bei einer Mehrfachaktivität die Häufigkeit der Ausführung der Aktivität durch die Anzahl der zu bearbeitenden Objekte im Voraus bekannt ist. Bei einer Schleifenaktivität hingegen steht im Voraus nicht fest, wie oft die Aktivität ausgeführt werden muss, bis die Schleifenbedingung erfüllt wird.

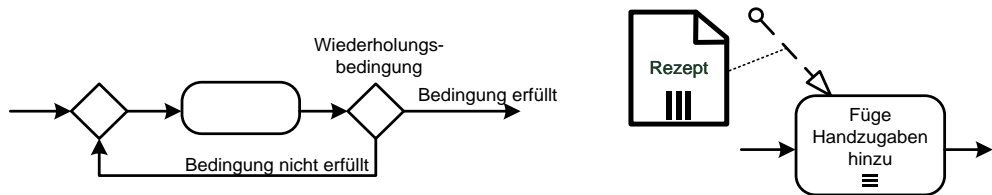


Abb. 8.4– Modellierung einer Schleifenaktivität mit exklusivem Gateway (links) und einer sequentiellen Mehrfachaktivität (rechts).

Um den Modellierungsaufwand für häufig und an verschiedenen Stellen aufgerufene Funktionen bzw. wiederkehrende Prozessschritte im Produktionsprozess zu reduzieren, können diese global als Subprozessaktivitäten definiert und über Link-Ereignisse aufgerufen werden. Diese Aufruf-Aktivitäten sind nicht in den gewöhnlichen Prozessablauf eingebunden und besitzen immer ein Subprozessdiagramm, in dem der Ablauf mit einem Link-Startereignis beginnt und einem Link-Endereignis endet.

## Ereignisse

Ein Ereignis ist etwas, das während der Ausführung einer Software oder des Ablaufs eines Prozesses passiert und den Prozess beeinflusst. Ein Ereignis hat keine Dauer, sondern stellt vielmehr das Erreichen eines Sachverhalts nach Eintritt eines Ereignisses dar. Um was für ein Ereignis es sich handelt wird durch den Namen des Ereignisses, durch die Attribute EreignisTyp (Start, Stop, (nicht) unterbrechendes Zwischenereignis) und „Verhalten“ (allgemein, Zeit, Ausnahme/Error, Link) festgelegt. Das Startereignis zeigt den Beginn eines Prozessflusses an. Das Endereignis zeigt das Ende eines Prozesses an. Zwischenereignisse treten während des Ablaufs eines Prozesses auf. Sie können an Aktivitäten angeheftet werden oder als den Sequenzfluss unterbrechende, eigenständige Elemente im Fluss auftreten. Als angeheftete Zwischenereignisse können sie entweder den aktuellen

Ablauf und damit die Durchführung der Aktivität an die sie angeheftet sind unterbrechen und einen weiteren Sequenzfluss starten oder als nicht-unterbrechende Zwischenereignisse bei Eintritt einen zusätzlichen Sequenzfluss starten, ohne die aktuell ablaufende Aktivität zu beeinflussen.

Start- und Endereignisse mit Link als Verhalten können verwendet werden, um Sequenzflüsse, die über mehrere Diagrammseiten verlaufen miteinander zu verknüpfen oder um Aufrufaktivitäten aufzurufen. Ebenso können sie anstatt langer Sequenzflussverbindungen auf einer Seite, die um zahlreiche andere Modellelemente herum gezogen werden müssten, verwendet werden.

Das zeitgebundene Ereignis kann als Start, End und Zwischenereignis eintreten und wird entsprechend als Uhrzeiger innerhalb des Basis-Ereignissymbols dargestellt.

Das außerplanmäßige Ereignis stellt eine Ausnahme vom eigentlichen Sequenzfluss dar. Es kann als unterbrechendes außerplanmäßiges Ereignis oder beendendes außerplanmäßiges Ereignis auftreten. Bei außerplanmäßigen Endereignissen wird die Fehlerbehandlung in einem extra Ablauf (gestartet durch das außerplanmäßige Starterereignis) modelliert. Diese Art der Modellierung eignet sich insbesondere, wenn es sich um wieder-kehrende Fehlerbehandlungsroutinen handelt. Außerplanmäßige Start-Ereignisse können auch nur auftreten, wenn es mindestens ein außerplanmäßiges Endereignis gibt, das auf sie zeigt in dem es namentlich auf sie verweist (bspw. „Notaus gedrückt“ auf „Fehlerbehandlung Notaus gedrückt“). Bei außerplanmäßigen Zwischenereignissen wird die Fehlerbehandlung direkt als abgehender Sequenzfluss modelliert.

An Aktivitäten können auch mehrere Ereignisse angeheftet werden, wenn sie durch mehrere Ereignisse unterbrochen oder bei deren Durchführung mehrere Ereignisse eintreten können.

### **Gateways/Entscheidungsknoten**

Um die Diagramme intuitiv verständlich zu halten und sie leicht „auf einen Blick“ erfassbar zu machen, werden Verzweigungen und Zusammenführungen durch Gateways dargestellt und somit besonders grafisch hervorgehoben. Gateways sind keine Aktivitäten. Sie stellen nur Logik dar und besitzen somit keine Zeitdauer. Gateways können somit auch keine Prüfkaktivitäten ausführen, sondern teilen lediglich den Pfad aufgrund der

Prüfung auf. Die Prüfkaktivität ist somit als eine dem Entscheidungsknoten vorgelagerte Aktivität zu modellieren. [4]

Ein exklusiver Gateway stellt die Verzweigung oder Zusammenführung alternativer Pfade dar. Er realisiert die Logik eines XOR.

Ein inklusiver Gateway realisiert die Logik eines (inklusive) ODER. Es werden mehrere aber nicht alle der aus- oder eingehenden Pfade gewählt oder zusammengeführt. Er darf jedoch in der MES-ML nur verwendet werden, wenn die Verwendung eines exklusiven oder parallelen Gateways ausgeschlossen ist.

Ein paralleler Gateway (Parallel Gateway) teilt einen Sequenzfluss in zwei oder mehr parallel laufende Pfade auf oder fasst zwei oder mehr parallel laufende Pfade zu einem Sequenzfluss zusammen (Zusammenführung). Er realisiert die Logik eines UND. Bei der Zusammenführung wartet der Prozess auf alle eingehenden Sequenzflüsse.

### 8.5.5 Datenobjekte

Datenobjekte dienen der Darstellung von im Prozess verwendeten Daten. Sie geben die Information, ob bzw. welche Daten die Aktivitäten brauchen, um ausgeführt werden zu können und welche Daten sie produzieren oder welche Daten ein Ereignis auslösen. Bei Datenobjekten werden (Einzel-) Datenobjekte und Mehrfachdatenobjekte unterschieden. Zudem gibt es die Möglichkeit, Datenspeicher abzubilden (siehe Abb. 8.5).

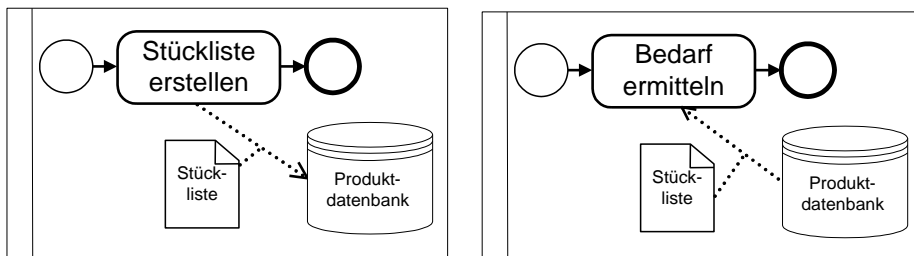


Abb. 8.5 – Zwei Prozesse greifen auf ein und denselben Datenspeicher zu

Während Einzel- oder Mehrfachdatenobjekte über Assoziationen an Daten- oder Nachrichtenflüsse angeheftet werden, stellen Datenspeicher Quelle und Ziel von Daten- und Nachrichtenflüssen dar.



## 8 MES-Modeling Language – Eine Beschreibungssprache für die interdisziplinäre Anforderungserhebung und Spezifikation von MES

---

Die Informationen, welche Daten durch Daten- oder Nachrichtenflüsse verschickt werden sollen, werden in die über Assoziationen den Flüssen angehängten Datenobjekte geschrieben (nicht in die Sende-Aktivität) (siehe Abb. 8.6). Datenobjekte können in verschiedenen Diagrammblättern mehrfach referenziert werden.

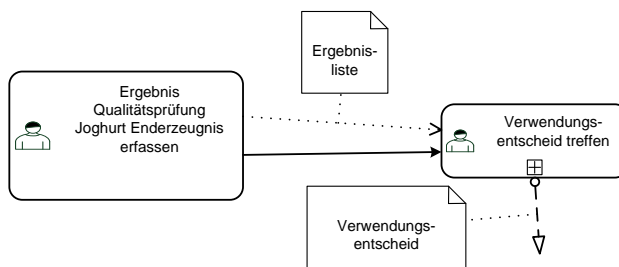


Abb. 8.6 – Daten- und Nachrichtenfluss mit über Assoziationen angehefteten Datenobjekten

Datenspeicher (Data Store) dienen sowohl der Modellierung von Datenbanken, als auch der Modellierung von internen Datenspeichern eines Systems. Datenobjekte, die über einen Prozess hinaus auch für andere Prozesse zur Verfügung stehen sollen, können in einem Datenspeicher abgelegt werden. Datenobjekte, die im MES verwaltet werden sollen, müssen einen Datenspeicher aus einem MES-Systembereich als Quelle oder Ziel haben.

Für alle Datenobjekte muss bei erstmaliger Verwendung auf einem Diagramm eine Datenquelle angegeben werden. Quelle oder Ziel können grafisch oder als Attribut des Datenflusses definiert werden. Es wird davon ausgegangen, dass Datenobjekte, die auf einem Diagramm verwendet werden, dem gesamten Prozess dieses Diagramms (auch über mehrere Folgeseiten) zur Verfügung stehen und bei der Realisierung in einem Zwischenspeicher vorgehalten werden. Das bedeutet, dass alle auf einem Diagramm definierten Funktionalitäten, die über Sequenzflüsse miteinander verbunden sind, auf den gleichen Zwischenspeicher oder die gleiche Datenbasis zugreifen können.

## 8.5.6 Artefakte

### Gruppen

Gruppen können innerhalb eines Systembereichs funktionsbereichsübergreifend Elemente visuell zusammenfassen. Eine Gruppe enthält keine Elemente, sondern referenziert diese lediglich.

### MES Funktions-Gruppen

Elemente können zu MES-Funktionsgruppen (MES-Groups) zusammengefasst werden, wenn sie logisch zu einer bspw. in der IEC 62264 definierten Funktion zugeordnet werden können und diese fast vollständig darstellen. Die Kennzeichnung erfolgt einzig über den Namen, der entsprechend der Norm gewählt wird. Das grafische Element der Gruppe bleibt gleich.

### Anmerkungen

Weitere Informationen zum Prozess können durch Anmerkungen in das Diagramm integriert werden. Sie sollten nach Möglichkeit selten verwendet werden.

## 8.6 Produktionsprozessmodell

### 8.6.1 Aufgabe des Modells in der MES-ML

Das Produktionsprozessmodell ermöglicht eine integrierte Abbildung des Produktionsprozesses, der betrieblichen Geschäftsprozesse und der Prozesse zur Erhaltung des Betriebszustandes. Im Unterschied zum Modell des technischen Systems stehen hier der Prozessgedanke und die Fragestellung „Wie funktioniert der Produktionsprozess und die ihn unterstützenden Prozesse?“ im Vordergrund. Im Gegensatz zum MES-Modell, welches die überlagerten IT-Systeme und nur die durch Software oder Softwareinteraktion realisierten Funktionen abbildet, zeigt das Produktionsprozessmodell die durch die Steuerungssoftware realisierten Automatisierungsfunktionen ebenso wie manuelle Tätigkeiten im Produktionsprozess oder den betrieblichen Abläufen.

## 8.6.2 Modellierungselemente des Produktionsprozessmodells

Im Produktionsprozessmodell stehen größtenteils die gleichen Modellierungselemente zur Verfügung wie im MES-Modell. Ausnahme sind lediglich die Verantwortungsbereiche, die im Produktionsprozessmodell nicht benötigt werden sowie die IT- Aktivitäten und Benutzer-Interaktionen, die mit der Produktionsprozessaktivität und der manuellen Aufgabe den Platz der Stellvertretung getauscht haben. Zudem haben alle Elemente des Produktionsprozesses einen grauen Hintergrund (siehe Abb. 8.7).

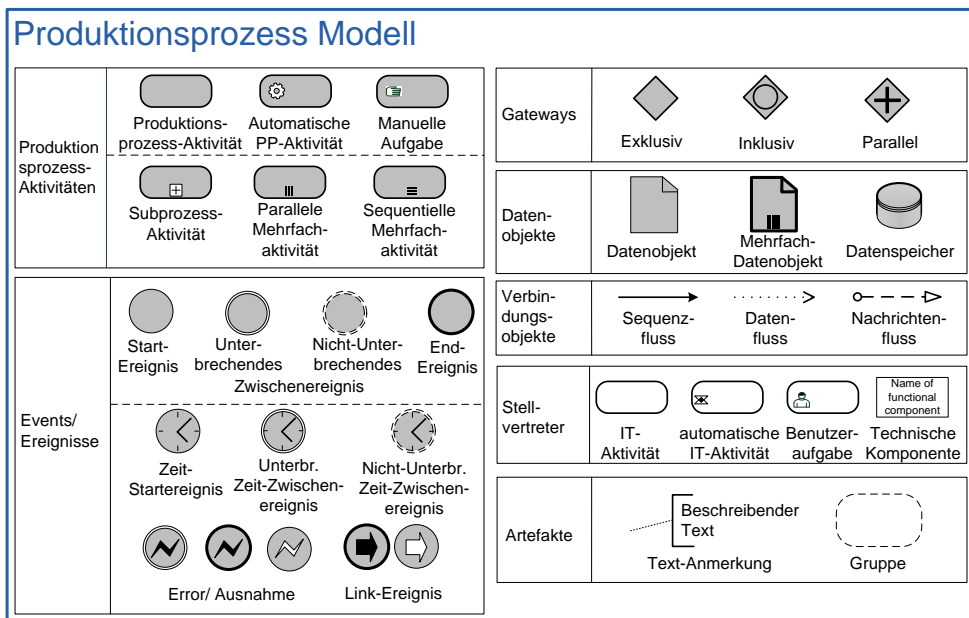


Abb. 8.7 – Notation des Produktionsprozessmodells

Die für die Elemente des MES-Modells definierten Regeln gelten vollständig auch für die Elemente des Produktionsprozessmodells.

## 8.7 Strukturbaum

Um insbesondere bei großen MES-Projekten und sehr komplexen IT-Systemen und Prozessen die Modelle transparent zu halten, existiert parallel zu den MES- und Produktionsprozessmodellen jeweils ein Strukturbaum,

der die Diagramme, ihre enthaltenen System- und Funktionsbereiche, Ereignisse, Aktivitäten, Subprozessaktivitäten und die darin enthaltenen Subprozessdiagramme mit all ihren Elementen und weiteren Subprozessdiagrammen in einer Baumstruktur hierarchisch darstellt. Diese weitere Darstellung erhöht die Transparenz über die Diagrammhierarchie innerhalb der Modelle, gibt einen schnellen Überblick über die Schwerpunkte (unterschiedlicher Detaillierungsgrad in der Ausmodellierung von Funktionen und Prozessschritten) und erleichtert im SpeziMES-Prototypen die Navigation zwischen den Diagrammen.

### 8.8 Verknüpfungen zwischen den Modellen

Besonderen Mehrwert besitzt das Beschreibungsmittel im Gegensatz zu bestehenden Beschreibungssprachen durch die Integration der drei Modelle über modellübergreifende Verknüpfungen. Es werden drei Arten von Verknüpfungen zwischen den Modellen (Cross-Diagram-Links) unterschieden:

1. Ausführung (Deployment)
2. Datenaustausch (Data transfer)
3. Entsprechung (Equivalence)

Die Verknüpfung vom Typ *Datenaustausch* ermöglicht die Modellierung der Anbindung der IT-Systeme an die Anlage bzw. Automatisierung und die Datenabhängigkeiten zwischen der IT und dem Produktionsprozess. Um die Interaktion zwischen der IT und dem Produktionsprozess auch grafisch zu verdeutlichen, können Stellvertreter ausgewählter Elemente der drei Modelle in das Produktionsprozess- und MES-Modell eingebunden und durch eine Verknüpfung vom Typ *Entsprechung* miteinander verknüpft werden.

Durch eine Verknüpfung vom Typ *Ausführung* können Abläufe oder einzelne Prozessschritte ihrer Hardware, Anlagenteil oder Anlagenmodul zugeordnet werden.

Die Kopplung der Modelle ermöglicht eine Gesamtsicht auf die Produktion und IT, die bisher durch die Modellierung einzelner Sichten nicht erreicht werden konnte.

Verknüpfungen zwischen den Modellen werden als Links den einzelnen Elementen in ihren Attributen hinterlegt und werden somit nicht grafisch in

## 8 MES-Modeling Language – Eine Beschreibungssprache für die interdisziplinäre Anforderungserhebung und Spezifikation von MES

---

den Diagrammen dargestellt. Abb. 8.1 zeigt die Verknüpfungsmöglichkeiten zwischen den einzelnen Modellen.

Tab. 8.1 – Verknüpfungsmöglichkeiten zwischen den Modellen

	<b>Modell des technischen Systems</b>	<b>Produktionsprozessmodell</b>	<b>MES-Modell</b>
<b>Modell des technischen Systems</b>		Datenaustausch	Datenaustausch
<b>Produktionsprozessmodell</b>	Ausführung, Datenaustausch, Entsprechung		Datenaustausch, Entsprechung
<b>MES-Modell</b>	Ausführung, Datenaustausch, Entsprechung	Datenaustausch, Entsprechung	

Mögliche Verknüpfungstypen vom Modell der linken Spalte zu einem Modell der Kopfzeile sind in den entsprechenden Zellen aufgeführt.

### 8.8.2 Datenaustausch

Eine Verknüpfung vom Typ Datenaustausch stellt die Abhängigkeiten und Schnittstellen zwischen der IT und dem Produktionssystem (Automatisierung samt aller Datenerfassungsgeräte in der Produktion und Fertigung) dar. Die Verknüpfungen können jeweils zwischen

- System- und Funktionsbereichen, Aktivitäten, Ereignissen und Datenspeichern vom MES-Modell,
  - Aktivitäten, Ereignisse und Datenspeichern vom Produktionsprozessmodell und
  - Signalen und Prozessobjekten des Technischen System-Modells
- bidirektional definiert werden.

Zu jeder dieser Verknüpfungen kann ein „Connector“ mit verschiedenen Attributen hinterlegt werden, der die Schnittstelle näher beschreibt.

### 8.8.3 Ausführung

Eine Verknüpfung vom Typ Ausführung stellt den Zusammenhang zwischen den Prozess- und Funktionsbeschreibungen des MES- und Produktionsprozessmodells und der Hardwarebeschreibung im Modell des technischen Systems dar. Dabei haben alle Verknüpfungen die Richtung MES- oder Produktionsprozessmodell hin zum Technischen System Modell.

### 8.8.4 Entsprechung - Einbindung von Stellvertretern in das Produktionsprozess- oder MES-Modell

Die Darstellung der Stellvertreter dient der vollständigen Abbildung des Flusses (insbesondere des Nachrichtenflusses) und soll das Verständnis für die Interaktion zwischen den Systemen erleichtern.

Um die Interaktion zwischen der IT-Software und dem Produktionsprozess auch grafisch darstellen zu können, können Stellvertreter von IT-Aktivitäten (egal welchen Subtyps) aus dem MES-Modell in das Produktionsprozess-Modell und Produktionsprozessaktivitäten (ebenfalls egal welchen Subtyps) in das MES-Modell eingebunden werden. Gleiches gilt für Abläufe, bei denen eine direkte Abhängigkeit zwischen der IT und dem Produktionsprozess besteht, beispielsweise bei der Verriegelung von Produktionsprozessschritten durch das MES.

Stellvertreter können mit anderen Elementen jedoch ausschließlich über Nachrichtenflüsse, nicht über Sequenz oder Datenflüsse verbunden werden (siehe Abb. 8.7).

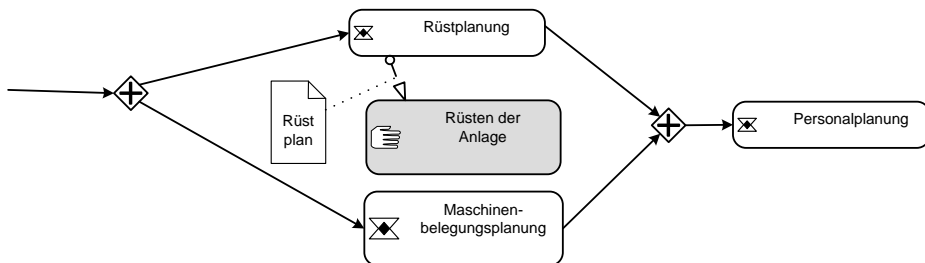


Abb. 8.8 – Einbindung einer Stellvertreter-Produktionsprozessaktivität in das MES Modell

## 8 MES-Modeling Language – Eine Beschreibungssprache für die interdisziplinäre Anforderungserhebung und Spezifikation von MES

---

Ebenso kann in das MES- und Produktionsprozessmodell ein Stellvertreter eines Knotens oder das Signal eines Knoten als Technisches Element eingebunden werden. Dies ist insbesondere dann von Bedeutung, wenn die genaue Ankopplung des MES an die Anlage grafisch dargestellt werden soll.

Stellvertreter müssen in ihrem „Muttermodell“ existieren, um in ein anderes Modell als Stellvertreter eingebunden werden zu können. Beispielsweise kann eine Produktionsprozess-Aktivität „Rüsten der Anlage“ nur dann als Stellvertreter in das MES Modell eingebunden werden, wenn sie im Produktionsprozessmodell zuvor modelliert wurde. Dabei muss sie nicht in einen Sequenzfluss im Produktionsprozessmodell eingebunden sein, sondern kann auch erst mal allein in einem Diagramm auftreten. Einzeln auftretende Aktivitäten können somit Lücken im Verständnis des Gesamtablaufs aufzeigen. Stellvertreter referenzieren mit einer Äquivalenz-Beziehung auf ihre Originale im „Muttermodell“.

Stellvertreter können mit Elementen des Modells in welches sie eingebunden werden nur durch Nachrichtenflüsse verbunden werden. Dabei gelten die normalen Modellierungsregeln zur Verwendung von Nachrichtenflüssen. Das bedeutet, Elemente die im Modell keinen Ein- oder Ausgang von Nachrichtenflüssen haben können, können auch nicht mit den Stellvertretern verbunden werden.

Auf oberster Ebene (Grobprozessbeschreibung) ist die Verwendung von Stellvertretern optional. Die aus dem Produktionsprozessmodell in das MES-Modell (und umgekehrt) projizierten Objekte dienen eher in der Detaillierung der Beschreibung der Interaktion zwischen dem MES und dem Produktionsprozess.

### 8.9 SpeziMES Editor

Zur Unterstützung des Engineeringprozesses von MES und leichterem Anwendung zur Evaluation der MES-ML wurde ein Editor-Prototyp entwickelt. Dieser ermöglicht die Erstellung und Verknüpfung der Modelle der MES-ML. Durch seine einfache Handhabung und die disziplinenübergreifende, intuitive Verständlichkeit der MES-ML können die Beschreibungssprache und der Prototyp ohne vorherige Einarbeitung für die Prozessanalyse und Spezifikation von MES verwendet werden. Insbesondere der Einsatz in interdisziplinären Workshops zur parallelen

Visualisierung und Dokumentation der Diskussionen sowie die eigenständige Weiterbearbeitung der Modelle durch die Teilnehmer sind damit möglich.

## 8.10 Evaluation der Beschreibungssprache

Die MES-ML wurde im Rahmen des Projekts mit Hilfe des Prototypen bei namhaften Unternehmen aus dem Bereich der Fertigungs- (2) und Verfahrenstechnik (2) evaluiert und ihr Nutzen und ihre einfache Anwendbarkeit nachgewiesen. Eine ausführliche Beschreibung eines Evaluationsprojekts und der Ergebnisse findet sich in[5].

Die Evaluation erfolgte in zwei Tages-Workshops bei den Unternehmen vor Ort und anschließender individueller Weiternutzung und Bearbeitung der erstellten Modelle. An den Workshops nahmen bis zu 12 Mitarbeiter aus unterschiedlichen Abteilungen teil. Vertreten waren dabei sowohl Mitarbeiter aus den Bereichen Produktion, IT, Qualitätsmanagement und Automatisierung wie auch aus den Bereichen Finanzen und der Geschäftsführung. Die Beschreibungssprache wurde von allen Bereichen im gleichen Maße sehr positiv bewertet.

Besondere Vorteile durch die Modellierung mit der MES-ML wurden in der intuitiv verständlichen und vollständigen Darstellung und Integration der unterschiedlichen Sichten bei gleichzeitig sehr effizienter Modellierung und Ergebniserzielung gesehen. Obwohl den Evaluationsworkshops reale MES-Projekte zugrunde lagen, konnten keine Sachverhalte identifiziert werden, die nicht vollständig mit der MES-ML abgebildet werden konnten. Durch die direkte Visualisierung der relevanten Prozesse und Anforderungen konnten Inkonsistenzen oder Widersprüche direkt identifiziert und gemeinsam eine von allen beteiligten Abteilungen akzeptierte Lösung entwickelt werden. Diese konnte mit der MES-ML wiederum sofort interpretationsfrei dokumentiert werden.

Nicht nur innerhalb der Workshops, auch für die Weitergabe und Diskussion der Workshopergebnisse mit Dritten stellte sich die MES-ML als sehr hilfreich und effizient heraus. So wurden auch von MES-Anbietern und Automatisierungsdienstleistern, die nicht Teil des Projektkonsortium sind, die Spezifikationsdokumente auf Basis der MES-ML als sehr gute Grundlage für die Erstellung eines Lastenheftes angesehen, auf deren Basis bis zu



85% der Aufwände für die Lastenhefterstellung eingespart werden können bzw. im Rahmen der Evaluation eingespart wurden.

Aufgrund der sehr positiven Erfahrungen in den Workshops und der Weiterverwendung der Modelle wollen alle Evaluationspartner das Beschreibungsmittel auch in zukünftigen Projekten anwenden.

### 8.11 Danksagung

Der Lehrstuhl für Automatisierung und Informationssysteme bedankt sich bei den Projektpartnern des Projekts SpeziMES, der BN Automation AG und der Trebing & Himstedt Prozessautomation GmbH & Co.KG für die gute und konstruktive Zusammenarbeit.

Das Projektkonsortium des Projekts SpeziMES bedankt sich bei den Industriebeiratsmitgliedern für die Möglichkeiten der Evaluation, die interessanten Diskussionen und das hilfreiche Feedback.

Besonderer Dank gilt dem Bundesministerium für Bildung und Forschung (BMBF) für die Förderung des Projekts „Entwicklung eines grafischen Beschreibungsmittels für die Spezifikation von Manufacturing Execution Systems im Spannungsfeld zwischen IT und Produktion –SpeziMES“ (01IS09026 C), wodurch eine Durchführung erst möglich wurde.

### 8.12 Referenzen

- [1] Himstedt, St.; Witsch, M.: MES Projekte effizient spezifizieren - Anforderungen an ein einheitliches Beschreibungsmittel für den Praxisalltag. In: Höhere Verfügbarkeit und transparente Produktion, Tagungsband Automation Symposium 2011, Kassel: Kassel University Press, 2011.
- [2] Ricken, M.; Vogel-Heuser, B.: Modeling of Manufacturing Execution Systems: an Interdisciplinary Challenge. In: Proc. of ETFA 2010 - 15th IEEE International Conference on Emerging Technologies and Factory Automation, Bilbao, 2010.
- [3] [www.bpmn.org](http://www.bpmn.org).

- [4] Allweyer, Th.: BPMN 2.0 – Business Process Model and Notation – Einführung in den Standard für die Geschäftsprozessmodellierung, 2. Aufl., Norderstedt: Books on Demand, 2009, S. 26.
- [5] Dendorfer, J; Witsch, M: Spezifikation eines generischen MES-Clients mit der MES-ML, in: Höhere Verfügbarkeit und transparente Produktion, Tagungsband Automation Symposium 2011, Kassel: Kassel University Press, 2011.

## 9 Spezifikation eines generischen MES-Clients mit der MES-ML

Jürgen Dendorfer  
*Continental Automotive GmbH*

Maria Witsch  
*Lehrstuhl für Automatisierung und Informationssysteme(AIS), TU München*

**Zusammenfassung:** Die im Projekt SpeziMES entwickelte Beschreibungssprache MES-ML wurde im Rahmen eines realen Projektes bei Continental Automotive evaluiert. In diesem Beitrag werden das Vorgehen und die Ergebnisse der Evaluation dargestellt.

### 9.1 Einleitung

Die im Rahmen des Projekts SpeziMES entwickelte und in Kapitel 8 vorgestellte Beschreibungssprache MES-Modeling Language wurde im Rahmen eines realen MES Projektes bei der Continental Automotive GmbH evaluiert. Dabei wurden insbesondere das MES Modell und die Aufgaben und Interaktionen des MES innerhalb des Produktionsprozesses betrachtet.

Die Evaluation wurde in zwei Schritten durchgeführt. Zuerst wurde mit der für die Spezifikation des MES verantwortlichen IT-Abteilung von Continental ein zweitägiger Workshop durchgeführt, bei dem insbesondere die Softwarefunktionalitäten des MES durch Modellierung mit der MES-ML generisch definiert wurden. Anschließend wurden die definierten Soll-Funktionen des MES und die Art der Darstellung bei einer Werksbesichtigung, Analyse des Produktionsprozesses einer Beispielanlage, Diskussion mit den Programmierern des MES und Diskussion des Soll-Produktionsprozesses auf ihre Eignung und Anwendbarkeit für das konkrete Werk untersucht. Die Evaluation wurde parallel zum realen Projekt durchgeführt, sodass die Basis der Evaluation die zu dem Zeitpunkt vorhandenen Spezifikationsdokumente und bekannten Anforderungen waren.

### 9.2 Vorstellung des Evaluationsprojekts

Für die Evaluation der Beschreibungssprache wurde ein aktuelles Projekt ausgewählt, welches in Zusammenarbeit zwischen der zentralen IT-Abteilung von Continental, einem MES-Anbieter und einem Werk in Ungarn durchgeführt wird. Ziel des Projekts ist die Spezifikation und Neuentwicklung einer generischen MES-Software zur Materialrückverfolgung sowie seine Anwendung auf eine Kabelfertigung in einem Werk im europäischen Ausland. Dabei soll die bestehende MES-Lösung abgelöst werden.

Durch diese Konstellation ergeben sich besondere Herausforderungen:

- Anforderungserhebung vor Inbetriebnahme der neuen Fertigungslinie anhand einer bestehenden Beispiellinie und zwischenzeitlicher Wechsel der Beispiellinie
- Räumliche Distanz zwischen IT-Abteilung in Regensburg und Fertigungsstandort im Ausland
- unterschiedliche Sprachkenntnisse: unterschiedliche Muttersprachen, unterschiedlich gute Englischkenntnisse
- Gleichzeitige Spezifikation des generischen Clients und des Clients für das Pilotprojekt
- Planung der Aufwände für die Cliententwicklung und Implementierung ohne eine Analyse der Produktionsprozesse nur sehr grob möglich.

Die Kabelfertigung ist ein größtenteils manueller Fertigungsprozess an Maschinen. In 50er Batches werden pro Tag 11.000 Kabel als Serienfertigung im 3-Schichtbetrieb sechs Tage die Woche gefertigt. Somit wird eine Jahresstückzahl von drei Millionen erreicht.

Der grobe Produktionsprozess besteht aus den in Abb. 9.1 dargestellten Schritten:

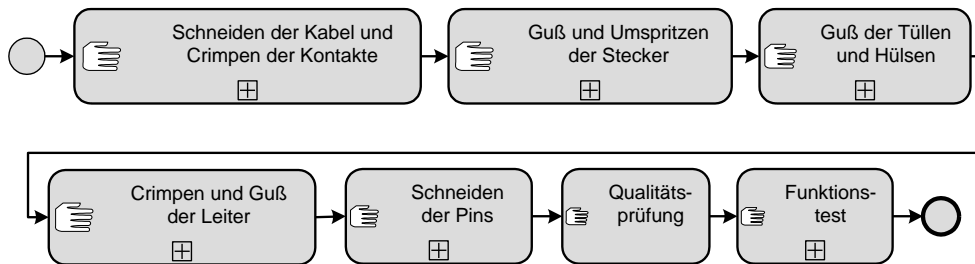


Abb. 9.1 – Produktionsprozessmodell grob

### 9.3 Modellierung des generischen Clients und Abgleich des Produktionsprozesses

#### 9.3.1 Modellierungsworkshop mit der MES-ML

Die Moderation des Workshops und die direkte Modellierung der Anforderungen und Prozesse wurde von zwei Mitarbeitern der Projektpartner des Projekts SpeziMES durchgeführt, sodass sich die Workshopteilnehmer vollständig auf die inhaltliche Diskussion konzentrieren konnten. Dabei wurde erstmalig der im Projekt für die Evaluation entwickelte Prototyp als Modellierungswerkzeug eingesetzt.

Die erste Modellierung wurde auf Grundlage eines Lastenhefts durchgeführt, welches von den Workshopteilnehmern auf Basis eines Besuchs der Beispielproduktion erstellt und zu Beginn des Workshops kurz vorgestellt wurde. Abgesehen von einer 15minütigen Einführung in die MES-ML, war die Beschreibungssprache für die Workshopteilnehmer neu. Einzelne besaßen jedoch Vorkenntnisse in der grafischen Modellierung mit Aktivitätsdiagrammen der Unified Modeling Language.

Bei der Kabelfertigung handelt es sich fast ausschließlich um manuelle Fertigungsprozesse. In Folge dessen verfügt das MES nur über eine sehr begrenzte Anbindung an die Automatisierung. Das MES muss seine Informationen größtenteils durch direkte Benutzereingaben an Terminals erhalten. Daher wurden bei der Modellierung nur das MES- und Produktionsprozessmodell, sowie die Verlinkungen zwischen beiden Modellen angewendet.

Durch die Trennung der MES- und Produktionsprozesssicht und Dekomposition der relevanten Funktionen wurden beide Prozesse nur soweit nötig

beschrieben und somit der Aufwand für die detaillierte Beschreibung von Schritten bzw. Funktionen, die im späteren Verlauf als nicht wichtig eingestuft wurden, vermieden. Der Gesamtablauf im Produktionsprozessmodell und die Funktionsübersicht im MES-Modell wurden dennoch vollständig abgebildet. Zudem bietet die Trennung der beiden Sichten die Möglichkeit, sich je nach Interessenslage eine Sicht bis ins Detail anzuschauen und sich über die andere nur einen Überblick zu verschaffen. So kann der Programmierer die MES-Funktionalitäten bis zur Business-Logik untersuchen und nur einen groben Überblick über den gesamten Produktionsprozess gewinnen. Je nach seinem Erfahrungsschatz und Anwendungsfall kann so die passende Detailebene für die MES Implementierung ausgewählt werden. Genauso könnte sich der Produktionsleiter die Modelle seiner Produktionsprozesse bis ins Detail anschauen und sich einen schnellen Überblick über die Funktionalitäten des MES auf oberster Ebene des MES-Modells verschaffen. Auf tieferer Detailebene wurde genau geplant, an welchen Stellen des Produktionsprozesses welche MES-Clients mit welchen Funktionalitäten eingesetzt werden. Abb. 9.2 zeigt einen Ausschnitt aus dem MES-Modell und der Dekomposition der MES Funktion *Production Tracking*. Die Abbildung verdeutlicht auch den Übergang zwischen der strukturellen Darstellung der Funktionalitäten (Module des Clients) und der Ablaufbeschreibung der Business-Logik des Clients auf den tieferen Detailebenen. So sind auf der ersten und zweiten Ebene keine Sequenzflüsse zwischen den Aktivitäten des MES abgebildet und der Aufruf der einzelnen Aktivitäten erfolgt, wie in den Subprozessdiagrammen dargestellt, durch ein Link-Ereignis.

Bei der Erstellung der Modelle konnten viele Konkretisierungen getroffen und einige noch offene Fragestellungen identifiziert werden. Letzteres diente als Basis für die weitere Bearbeitung der Modelle.

## 9 Spezifikation eines generischen MES-Clients mit der MES-ML

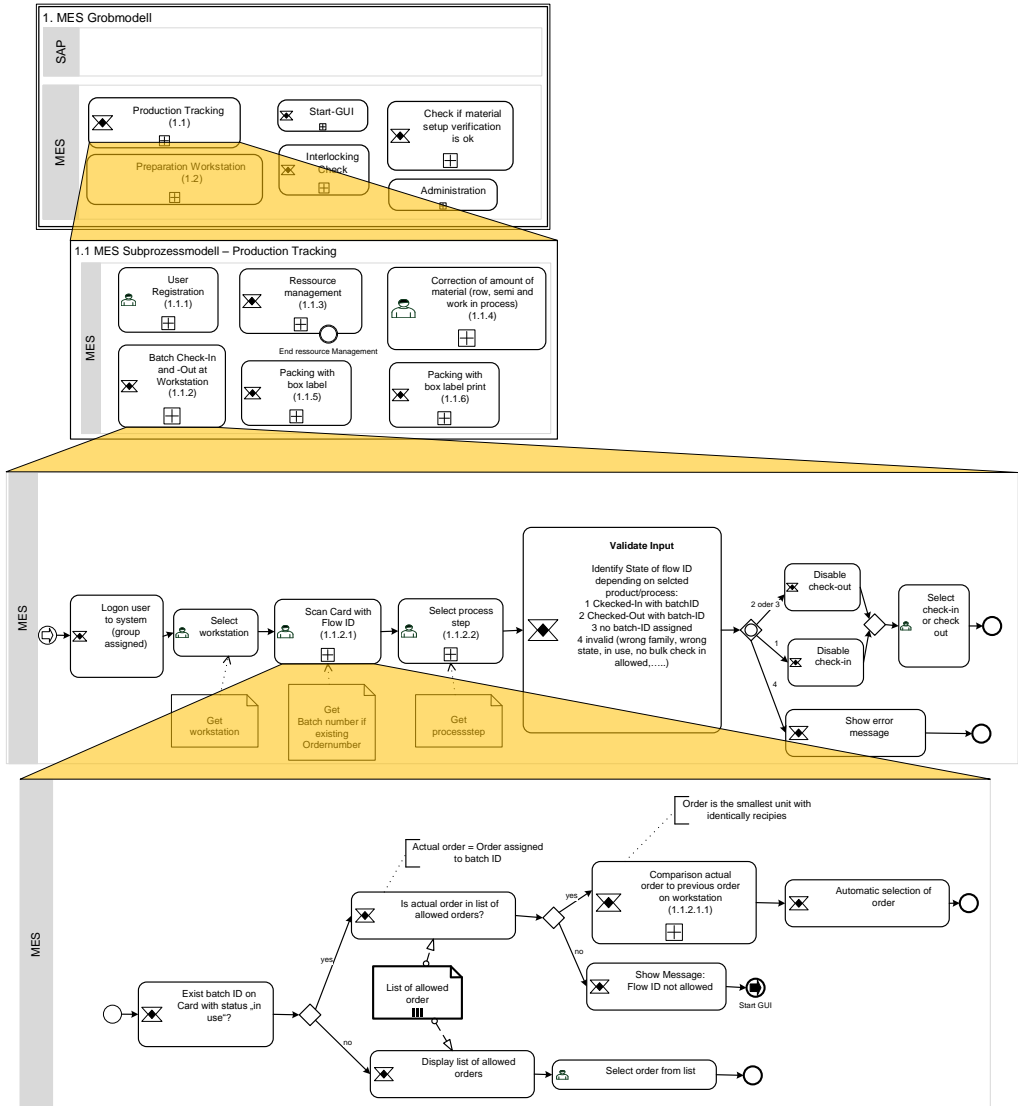


Abb. 9.2 – Ausschnitt aus dem MES Modell

### 9.3.2 Eigenständige Überarbeitung der Modelle

Die im ersten Workshop erstellten Modelle wurden durch die Workshop-teilnehmer in den zwei Wochen nach dem Workshop ohne weitere Anleitung oder Schulung in der MES-ML weiter verfeinert und die im

Workshop vorläufig entwickelte Lösungsvorschläge überdacht und gegebenenfalls korrigiert.

Zudem wurden die Modelle verwendet, um mit anderen Softwareentwicklern, die nicht am Workshop teilgenommen haben, die Anforderungen zu diskutieren und zu verifizieren.

Außerdem wurden aus den Funktionen des generischen Clients, die Funktionen für den Client des Pilotprojekts abgeleitet bzw. selektiert. Hierfür wurden die Modelle mit geringem Aufwand für die konkrete Umsetzung angepasst.

### **9.3.3 Analyse der ausgewählten Funktionalitäten anhand der Pilotlinie**

Unter Berücksichtigung der in den Workshops und während der anschließenden Weiterentwicklung identifizierten Unklarheiten, wurde die Pilotlinie besichtigt und die Funktionalitäten des Clients des Pilotprojekts mit den Produktionsverantwortlichen vor Ort diskutiert. Aufgrund der detaillierten Planung des Clients vorab konnten in der begrenzten Zeit im Werk viele wichtige Fragen effizient geklärt werden.

## **9.4 Ergebnisse der Evaluation**

In den Workshops können durch die gemeinsame Betrachtung der visualisierten Prozesse häufig unterschiedliche, teils konträre Sichten auf die Prozesse und vorhandenen IT-Systeme identifiziert werden. Durch die gemeinsame Diskussion und direkte Dokumentation in den Workshops, sowie die Unterscheidung in Ist- und Soll-Beschreibung konnten die Unterschiede in der Pilotlinie zur geplanten Linie sowie zwischen dem aktuellen MES, welches abgelöst werden soll und den Anforderungen an das neue MES einfach dokumentiert werden.

Durch die direkte Modellierung der diskutierten Abläufe des MES sowie des Produktionsprozesses waren die Workshopteilnehmer gezwungen, sich konkret auszudrücken. Unklarheiten und Inkonsistenzen in den Abläufen und Daten wurden sofort sichtbar und mussten gelöst werden. Hierdurch wurde eine sehr schnelle Konkretisierung der Soll-Prozess- und Funktionsbeschreibung erreicht. Aufgrund der transparenten Darstellung der Ist-Prozesse konnten Optimierungspotentiale leicht identifiziert werden.



Durch die parallele Modellierung im Workshop zur Diskussion und damit die direkte Visualisierung ergaben sich einige weitere Vorteile:

- Die direkte Visualisierung von Lösungsvorschlägen erleichtert die Diskussion und Optimierung dieser.
- Alle diskutierten Aspekte wurden direkt dokumentiert.
- Somit werden gemeinsame Ergebnisse direkt transparent und für alle Teilnehmer sichtbar festgehalten. Die Workshopteilnehmer identifizieren sich mit den erstellten Modellen und können sie ohne schulungsaufwand selbstständig weiterbearbeiten. Auch im Workshop selbst ging keine Zeit für die Schulung im Beschreibungsmittel verloren sondern die immer sehr "teure" Zeit in Workshops, wenn mehrere Mitarbeiter von ihrem Tagesgeschäft fernbleiben, konnte sehr effizient genutzt werden.
- Ergebnisse der Diskussionen gehen nicht verloren oder können durch den Protokollanten verfälscht werden. Es ist keine Nachbereitung des Workshops notwendig, sondern die Modelle können direkt ohne Zeitverzug zur weiteren Diskussion mit Dritten und Klärung von offenen Fragen verwendet werden.
- Die Art der Dokumentation ist auch für Dritte ohne weitere Vorkenntnisse verständlich.

Die Modelle sind auch nach mehreren Monaten noch nachvollziehbar und eignen sich daher auch für Projekte, die über einen sehr langen Zeitraum laufen sowie für die Dokumentation der Implementierung und späteren Weiterverwendung für eine Weiterentwicklung des MES.

Durch die generische Prozessbeschreibung und einfache Verständlichkeit der Modelle kann Prozesswissen einfach festgehalten und weitergegeben werden. Hierdurch wird ein gutes Knowledge Management für bestehende Prozesse unterstützt. Ebenso können mit der Beschreibungssprache Prozesse leicht standardisiert werden. Doch nicht nur die Prozesse, auch MES-Funktionalitäten können beispielsweise durch Aufbau eines Standard-MES-Funktionalitäts-Katalogs die Standardisierung eines MES unterstützen. Durch die so entstehende Bibliothek mit Standard-MES-Funktionalitäten können die Modelle für die Spezifikation weiterer MES in anderen Betrieben wiederverwendet und der Spezifikationsaufwand in den Folgeprojekten stark reduziert werden.

Aufgrund der Einfachheit der Modelle können Anpassungen und Ergänzungen der Modelle um neue Erkenntnisse bspw. aus der Analyse des Produktionsprozesses vor Ort deutlich einfacher durchgeführt werden, als es zuvor mit Umformulierung der Lastenhefte auf textueller Basis möglich gewesen wäre.

Die Moderation eines Workshops zur Anforderungserhebung durch Dritte hat den Vorteil, dass sich die Teilnehmer vollständig auf die inhaltlichen Themen konzentrieren können. Die personelle Trennung zwischen Moderator und Modellierer führt zudem zu einer schnelleren Entwicklung der Modelle und der Möglichkeit bereits während der Erstellung Rückfragen zum Modell bzw. dem dargestellten Ablauf diskutieren zu können.

Die Modellierung im Workshop mit dem prototypischen Editor konnte gut parallel zu den Diskussionen durchgeführt werden, da die Handhabung einfach und auf die MES-ML zugeschnitten ist. Unzulänglichkeiten eines Prototyps mussten dabei in Kauf genommen werden. Insbesondere für die Gegenüberstellung der MES und Produktionsprozesssicht ist die Unterstützung durch das prototypische Werkzeug sehr hilfreich. Die Interaktionen zwischen den Prozessen und Systemen können durch gleichzeitige Betrachtung verschiedener Sichten leicht nachvollzogen werden.

Inzwischen lässt sich nachweisen, dass nicht nur die Spezifikation, sondern auch die Testphasen der auf dieser Basis erstellten Software erheblich von der Klarheit der Darstellungen und von der Verknüpfung von MES-Model, Prozess-Model und Technischem-System-Model profitieren.

Die für den Modul und Integrationstest aufgestellten Testmuster wurden mit Hilfe der in MES-ML erstellten Diagramme verifiziert und ergänzt. Hierdurch konnte die Testabdeckung erhöht werden und die Beschreibung von „Bugs“ in der Testphase konnte sehr präzise mit Verweis auf die Strukturelemente der Dokumentation erfolgen.

Da die Vorteile der MES-ML auch im Praxiseinsatz bestätigt wurden, wird dieses Beschreibungsmittel bei Continental Automotive als Standard für weitere MES Projekte eingesetzt. Dies umfasst sowohl Anforderungserhebung, Spezifikation, Prozessoptimierung, Dokumentation als auch die Unterstützung von Modul und Integrationstests von auf dieser Grundlage erstellter MES-Software.

ISBN 978-3-86219-178-9