

# **Verfahren zum emulationsgestützten MES-Engineering für die Photovoltaikindustrie**

**Von der Fakultät Konstruktions-, Produktions- und Fahrzeugtechnik  
der Universität Stuttgart zur Erlangung der Würde eines  
Doktor-Ingenieurs (Dr.-Ing.) genehmigte Abhandlung**

Vorgelegt von  
Dipl.-Ing. Matthias Meier, M.Sc.  
aus Sindelfingen

Hauptberichter: Prof. Dr.-Ing. Alexander Verl  
Mitberichter: Prof. Dr.-Ing. Michael Weyrich

Tag der mündlichen Prüfung: 03.03.2011

Institut für Steuerungstechnik der Werkzeugmaschinen und  
Fertigungseinrichtungen der Universität Stuttgart

2011

# IPA-IAO Forschung und Praxis

Berichte aus dem  
Fraunhofer-Institut für Produktionstechnik und  
Automatisierung (IPA), Stuttgart,  
Fraunhofer-Institut für Arbeitswirtschaft und  
Organisation (IAO), Stuttgart,  
Institut für Industrielle Fertigung und  
Fabrikbetrieb (IFF), Universität Stuttgart  
und Institut für Arbeitswissenschaft und  
Technologiemanagement (IAT), Universität Stuttgart und  
Institut für Steuerungstechnik der Werkzeugmaschinen  
und Fertigungseinrichtungen (ISW), Universität Stuttgart

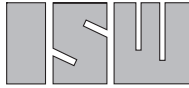
Herausgeber:

Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper  
und

Univ.-Prof. Dr.-Ing. habil. Prof. E.h. mult. Dr. h.c. mult. Hans-Jörg Bullinger  
und

Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath  
und

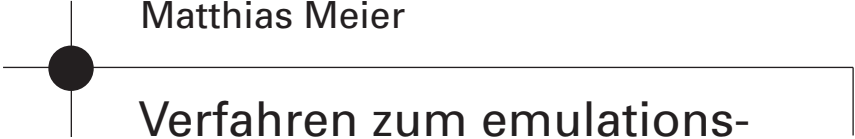
Univ.-Prof. Dr.-Ing. Dr. h.c. Alexander Verl



Institut für Steuerungstechnik  
der Werkzeugmaschinen und Fertigungseinrichtungen



Matthias Meier

A decorative graphic element consisting of a vertical line on the left, a horizontal line extending from it, and a solid black circle at the intersection.

## Verfahren zum emulations- gestützten MES-Engineering für die Photovoltaikindustrie

Nr. 508

Dr.-Ing. Matthias Meier M. Sc.

Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

Univ.-Prof. Dr.-Ing. Prof. e.h. Dr.-Ing. e.h. Dr. h.c. mult. Engelbert Westkämper

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

Univ.-Prof. Dr.-Ing. habil. Prof. E.h. mult. Dr. h.c. mult. Hans-Jörg Bullinger

ord. Professor an der Universität Stuttgart

Präsident der Fraunhofer-Gesellschaft, München

Univ.-Prof. Dr.-Ing. Dr.-Ing. E.h. Dieter Spath

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO), Stuttgart

Univ.-Prof. Dr.-Ing. Dr. h.c. Alexander Verl

ord. Professor an der Universität Stuttgart

Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA), Stuttgart

D 93

ISBN 978-3-939890-76-8

Jost Jetter Verlag, Heimsheim

Dieses Werk ist urheberrechtlich geschützt. Die dadurch begründeten Rechte, insbesondere die der Übersetzung, des Nachdrucks, des Vortrags, der Entnahme von Abbildungen und Tabellen, der Funksendung, der Mikroverfilmung oder der Vervielfältigung auf anderen Wegen und der Speicherung in Datenverarbeitungsanlagen, bleiben, auch bei nur auszugsweiser Verwertung, vorbehalten. Eine Vervielfältigung dieses Werkes oder von Teilen dieses Werkes ist auch im Einzelfall nur in den Grenzen der gesetzlichen Bestimmungen des Urheberrechtsgesetzes der Bundesrepublik Deutschland vom 9. September 1965 in der jeweils gültigen Fassung zulässig. Sie ist grundsätzlich vergütungspflichtig. Zuwiderhandlungen unterliegen den Strafbestimmungen des Urheberrechtsgesetzes.

© Jost-Jetter Verlag, Heimsheim 2011.

Printed in Germany.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Sollte in diesem Werk direkt oder indirekt auf Gesetze, Vorschriften oder Richtlinien (z. B. DIN, VDI, VDE) Bezug genommen oder aus ihnen zitiert worden sein, so kann der Verlag keine Gewähr für die Richtigkeit, Vollständigkeit oder Aktualität übernehmen. Es empfiehlt sich, gegebenenfalls für die eigenen Arbeiten die vollständigen Vorschriften oder Richtlinien in der jeweils gültigen Fassung hinzuzuziehen.

Druck: printsystem GmbH, Heimsheim

## Geleitwort der Herausgeber

Über den Erfolg und das Bestehen von Unternehmen in einer marktwirtschaftlichen Ordnung entscheidet letztendlich der Absatzmarkt. Das bedeutet, möglichst frühzeitig absatzmarktorientierte Anforderungen sowie deren Veränderungen zu erkennen und darauf zu reagieren.

Neue Technologien und Werkstoffe ermöglichen neue Produkte und eröffnen neue Märkte. Die neuen Produktions- und Informationstechnologien verwandeln signifikant und nachhaltig unsere industrielle Arbeitswelt. Politische und gesellschaftliche Veränderungen signalisieren und begleiten dabei einen Wertewandel, der auch in unseren Industriebetrieben deutlichen Niederschlag findet.

Die Aufgaben des Produktionsmanagements sind vielfältiger und anspruchsvoller geworden. Die Integration des europäischen Marktes, die Globalisierung vieler Industrien, die zunehmende Innovationsgeschwindigkeit, die Entwicklung zur Freizeitgesellschaft und die übergreifenden ökologischen und sozialen Probleme, zu deren Lösung die Wirtschaft ihren Beitrag leisten muss, erfordern von den Führungskräften erweiterte Perspektiven und Antworten, die über den Fokus traditionellen Produktionsmanagements deutlich hinausgehen.

Neue Formen der Arbeitsorganisation im indirekten und direkten Bereich sind heute schon feste Bestandteile innovativer Unternehmen. Die Entkopplung der Arbeitszeit von der Betriebszeit, integrierte Planungsansätze sowie der Aufbau dezentraler Strukturen sind nur einige der Konzepte, welche die aktuellen Entwicklungsrichtungen kennzeichnen. Erfreulich ist der Trend, immer mehr den Menschen in den Mittelpunkt der Arbeitsgestaltung zu stellen - die traditionell eher technokratisch akzentuierten Ansätze weichen einer stärkeren Human- und Organisationsorientierung. Qualifizierungsprogramme, Training und andere Formen der Mitarbeiterentwicklung gewinnen als Differenzierungsmerkmal und als Zukunftsinvestition in *Human Resources* an strategischer Bedeutung.

Von wissenschaftlicher Seite muss dieses Bemühen durch die Entwicklung von Methoden und Vorgehensweisen zur systematischen Analyse und Verbesserung des Systems Produktionsbetrieb einschließlich der erforderlichen Dienstleistungsfunktionen unterstützt werden. Die Ingenieure sind hier gefordert, in enger Zusammenarbeit mit anderen Disziplinen, z. B. der Informatik, der Wirtschaftswissenschaften und der Arbeitswissenschaft, Lösungen zu erarbeiten, die den veränderten Randbedingungen Rechnung tragen.

Die von den Herausgebern langjährig geleiteten Institute, das

- Fraunhofer-Institut für Produktionstechnik und Automatisierung (IPA),
- Fraunhofer-Institut für Arbeitswirtschaft und Organisation (IAO),
- Institut für Industrielle Fertigung und Fabrikbetrieb (IFF),
- Institut für Arbeitswissenschaft und Technologiemanagement (IAT),
- Institut für Steuerungstechnik der Werkzeugmaschinen und Fertigungseinrichtungen (ISW),

arbeiten in grundlegender und angewandter Forschung intensiv an den oben aufgezeigten Entwicklungen mit. Die Ausstattung der Labors und die Qualifikation der Mitarbeiter haben bereits in der Vergangenheit zu Forschungsergebnissen geführt, die für die Praxis von großem Wert waren. Zur Umsetzung gewonnener Erkenntnisse wird die Schriftenreihe „IPA-IAO - Forschung und Praxis“ herausgegeben. Der vorliegende Band setzt diese Reihe fort. Eine Übersicht über bisher erschienene Titel wird am Schluss dieses Buches gegeben.

Dem Verfasser sei für die geleistete Arbeit gedankt, dem Jost Jetter Verlag für die Aufnahme dieser Schriftenreihe in seine Angebotspalette und der Druckerei für saubere und zügige Ausführung. Möge das Buch von der Fachwelt gut aufgenommen werden.

Engelbert Westkämper    Hans-Jörg Bullinger    Dieter Spath    Alexander Verl

# Vorwort

Die vorliegende Arbeit entstand während meiner Tätigkeit als wissenschaftlicher Mitarbeiter und Projektleiter am Fraunhofer IPA in Stuttgart.

Herrn Prof. Dr.-Ing. Alexander Verl danke ich für die wissenschaftliche Betreuung und Förderung dieser Arbeit und die Übernahme des Hauptberichts. Mein Dank gilt ebenfalls Herrn Prof. Dr.-Ing. Michael Weyrich für die Durchsicht der Arbeit und die Übernahme des Mitberichts. Frau Heide Kreuzburg danke ich für die Organisation des Promotionsprozesses.

Ich danke den Kolleginnen und Kollegen der Abteilung Reinst- und Mikroproduktion, die die Entstehung dieser Arbeit durch ihre Diskussionsbereitschaft und konstruktive Kritik wesentlich unterstützt haben. Mein besonderer Dank gilt Herrn Dr.-Ing. Dipl.-Inform. Ralf Muckenhirn für die hilfreichen Anregungen, sein kontinuierliches Nachfragen nach dem Projektfortschritt und die Durchsicht des ersten Manuskripts. Herrn Dr.-Ing. Dipl.-Phys. Udo Gommel und Herrn Dipl. Ing.(FH) MEng. Christian Fischmann danke ich für die Unterstützung bei der Strukturierung und die Durchsicht der Arbeit. Danken möchte ich auch den Studierenden, die in Form von Studien- und Diplomarbeiten oder Praktika Beiträge zu dieser Arbeit geleistet haben.

Obwohl die Erstellung dieser Arbeit ein wichtiges Etappenziel meiner Tätigkeit am Fraunhofer IPA darstellte, war sie insbesondere von herausfordernden und abwechslungsreichen Projekten, bereichernden Begegnungen und der Zusammenarbeit mit interessanten Menschen aus der ganzen Welt und tollen Kollegen geprägt. Daher möchte ich an dieser Stelle den Projektpartnern aus Industrie- und Forschungsorganisationen danken, mit denen ich in den zurückliegenden Jahren zusammenarbeiten konnte. Ein besonderes Privileg der Jahre am Fraunhofer IPA war die hervorragende Zusammenarbeit mit den Kollegen aus D2.41, zunächst Herrn Dr.-Ing. Dipl.-Inform. Ralf Muckenhirn und Herrn Dr.-Ing. Dipl.-Inform. Philipp Dreiss und später Herrn Dipl.-Ing. Konstantin Konrad und Frau Dipl.-Ing. (FH) Ursula Rauschecker.

Schließlich möchte ich meinen Eltern und Geschwistern danken, die mich auf dem langen Weg meiner Ausbildung gefördert und unterstützt haben. Mein liebevoller Dank gilt meiner Frau Karin, die mir für die Erstellung dieser Arbeit immer wieder den notwendigen Rückhalt gegeben hat. Ihr und unserer Tochter Lisa widme ich dieses Buch.

Stuttgart, im Mai 2011

Matthias Meier





# Inhaltsverzeichnis

<b>Abkürzungen und Formelzeichen</b>	<b>13</b>
<b>1 Einleitung</b>	<b>17</b>
1.1 Problemstellung	18
1.2 Zielsetzung und Vorgehensweise	19
<b>2 Ausgangssituation</b>	<b>21</b>
2.1 Begriffe und Definitionen	21
2.1.1 Produkt-, Prozess und Fabriklebenszyklus	21
2.1.2 Anlage	22
2.1.3 Produktionsanlauf und Ramp-Up	22
2.1.4 Emulation und Simulation	23
2.1.5 Modell und Metamodell	24
2.2 Die PV-Industrie	25
2.2.1 Einige Grundlagen	25
2.2.2 Die Fertigungsorganisation	26
2.2.3 Mengengerüste	29
2.2.4 Produkt-, Prozess- und Fabriklebenszyklus in der PV-Industrie	29
2.3 Die Produktions-IT-Landschaft	30
2.3.1 Das Ebenenmodell der Produktions-IT-Landschaft	30
2.3.2 Manufacturing Execution Systems	31
2.3.3 Die Ebene der Fertigungsprozesse	35
2.4 Standardisierung in der Produktions-IT	35
2.4.1 Standards zu Produktions-IT-Landschaft und MES-Ebene	36
2.4.2 Standards zur Anlagenschnittstelle	37
2.5 Fazit Industriebedarf	39
<b>3 Analyse der Problemstellung und Konzeption des Verfahrens</b>	<b>41</b>
3.1 Die Produktions-IT in der PV-Industrie	41
3.1.1 Die MES-Anforderungen der PV-Industrie	41
3.1.2 MES – Make or Buy?	43
3.2 Wandlungsfähige MES Systemmodelle	43
3.2.1 Mehrschichtige Metamodelle	44
3.2.2 Mechanismen zur Anpassung von MES Lösungen	45
3.3 Analyse des MES-Engineering-Prozesses	47
3.3.1 Kritische Erfolgsfaktoren	48
3.3.2 Der MES-Engineering-Bedarf im Fabriklebenszyklus	48
3.3.3 Fazit aus der Analyse des MES-Engineering	51

3.4	Softwareengineering . . . . .	52
3.4.1	Prinzipien, Methoden und Werkzeuge . . . . .	52
3.4.2	Qualität von Softwaresystemen . . . . .	53
3.4.3	Softwaresysteme testen . . . . .	54
3.4.4	Fazit Software Engineering . . . . .	54
3.5	Emulationsgestütztes MES-Engineering . . . . .	54
3.5.1	Das System 'Anlage' . . . . .	55
3.5.2	Das System 'Fabrik' . . . . .	57
3.6	Anforderungen an Emulationsplattform und Verfahren . . . . .	59
<b>4</b>	<b>Stand der Technik</b>	<b>61</b>
4.1	Einordnung der Arbeit in den Kontext von Wissenschaft und Technik . . . . .	61
4.2	Engineering auf Anlagenebene . . . . .	63
4.2.1	Engineering mechatronischer Systeme . . . . .	63
4.2.2	Engineering der Fertigungsleittechnik . . . . .	64
4.3	Engineering auf Fabrikebene . . . . .	65
4.4	Integrative Ansätze . . . . .	66
4.5	Optimierungsansätze für das Manufacturing Execution System (MES)- Engineering im Umfeld der SEMI-Standards . . . . .	66
4.5.1	Die emulationsgestützte Anlagenintegration . . . . .	66
4.5.2	Automation Capability Management . . . . .	67
4.5.3	Modellbasierte Anlagenintegration . . . . .	68
4.5.4	Das MES in der Interaktion mit Anlagen testen . . . . .	69
4.6	Weitere Methoden und Werkzeuge . . . . .	69
4.6.1	Ereignisdiskrete Simulation und ereignisbasierte Datenextraktion . . . . .	69
4.6.2	Modellgetriebene Softwareentwicklung . . . . .	70
4.7	Fazit . . . . .	71
<b>5</b>	<b>Metamodell und Implementierung einer Emulationsplattform</b>	<b>73</b>
5.1	Entwurf eines Architekturkonzepts . . . . .	73
5.1.1	Modellierung versus Ausführung . . . . .	74
5.1.2	Die Schicht der Ausführungsumgebung . . . . .	75
5.1.3	Die Schicht der Plattformdienste . . . . .	77
5.1.4	Die Schicht der Benutzerinteraktion . . . . .	79
5.1.5	Zusammenfassung: Architekturkonzept . . . . .	79
5.2	Das statische Modell . . . . .	80
5.2.1	Die virtuelle Anlage . . . . .	81
5.2.2	Die virtuelle Anlagenumgebung . . . . .	89
5.2.3	Die virtuelle Fabrik . . . . .	90
5.3	Das dynamische Modell . . . . .	91
5.3.1	Anlagenverhalten modellieren . . . . .	91
5.3.2	Das Zeitmanagement der Emulationsplattform . . . . .	94
5.3.3	Formale Lebenszyklus-Modelle . . . . .	97
5.3.4	Szenarien ausführen . . . . .	100
5.4	Realisierung der Emulationsplattform . . . . .	102
5.4.1	Die Anwendungsfälle im MES-Engineering-Projekt . . . . .	102
5.4.2	Die Realisierung im Überblick . . . . .	104

5.4.3	Die Realisierung der Ausführungsumgebung . . . . .	107
5.4.4	Die Realisierung der Plattformdienste . . . . .	109
5.4.5	Die Realisierung der Benutzerinteraktionsschicht . . . . .	110
5.5	Fazit: Anforderungen an die Emulationsplattform . . . . .	114
<b>6</b>	<b>Entwicklung des Verfahrens des emulationsgestützten MES-Engineering</b>	<b>115</b>
6.1	Der emulationsgestützte MES-Engineering-Prozess . . . . .	115
6.1.1	Ein iterativer, inkrementeller MES-Engineering-Prozess . . . . .	115
6.1.2	Virtuelle Fabriken effektiv und effizient nutzen . . . . .	118
6.2	Vorgehensweise zur Erzeugung virtueller Fabriken . . . . .	119
6.2.1	Zieldefinition . . . . .	119
6.2.2	Systemanalyse . . . . .	119
6.2.3	Konzeption der Modelle der virtuellen Fabrik . . . . .	121
6.2.4	Erstellung der virtuellen Anlagen . . . . .	122
6.2.5	Modellierung/Konfiguration der virtuellen Fabrik . . . . .	122
6.2.6	MES Integration . . . . .	123
6.3	Fazit: Anforderungen an das Verfahren . . . . .	123
<b>7</b>	<b>Erprobung und Bewertung</b>	<b>125</b>
7.1	Fallbeispiel I (Back-end Konzentratormodule) . . . . .	125
7.1.1	Erzeugung der virtuellen Fabrik . . . . .	126
7.1.2	Einsatz der virtuellen Fabrik . . . . .	129
7.1.3	Fazit . . . . .	130
7.2	Fallbeispiel II (Dünnschicht Front-end / Back-end) . . . . .	130
7.2.1	Erzeugung der virtuellen Fabrik . . . . .	130
7.2.2	Einsatz der virtuellen Fabrik . . . . .	133
7.2.3	Fazit . . . . .	133
7.3	Bewertung . . . . .	134
<b>8</b>	<b>Zusammenfassung und Ausblick</b>	<b>135</b>
	<b>Summary and Outlook</b>	<b>139</b>



# Abkürzungen und Formelzeichen

## Abkürzungen

<b>ACM</b>	Automation Capability Management
<b>API</b>	Application Programming Interface
<b>APS</b>	Advanced Planning and Scheduling
<b>CAD</b>	Computer Aided Design
<b>CAEX</b>	Computer Aided Engineering Exchange
<b>CIM</b>	Computer Integrated Manufacturing
<b>COM</b>	Component Object Model
<b>CPU</b>	Central Processing Unit
<b>EA</b>	Equipment Automation
<b>EDA</b>	Equipment Data Acquisition
<b>EI</b>	Equipment Integration
<b>EMF</b>	Eclipse Modeling Framework
<b>ERP</b>	Enterprise Resource Planning
<b>FIFO</b>	First In First Out
<b>FMEA</b>	Fehlermöglichkeits- und Einflussanalyse (engl. Failure Mode and Effects Analysis)
<b>GBit</b>	Giga Bit
<b>GByte</b>	Giga Byte
<b>GEF</b>	Graphical Editing Framework
<b>GEM</b>	Generic Model for Communication and Control of Manufacturing Equipment
<b>GMA</b>	VDI/VDE-Gesellschaft Mess- und Automatisierungstechnik
<b>GMF</b>	Graphical Modeling Framework
<b>GUI</b>	Graphical User Interface
<b>HiLS</b>	Hardware-in-the-Loop Simulation
<b>HMI</b>	Human Machine Interface
<b>UID</b>	Unique Identifier
<b>IP</b>	Internet Protocol
<b>IPA</b>	Institut für Produktionstechnik und Automatisierung
<b>IPC</b>	Industrie Personalcomputer
<b>ISOC</b>	Internet Society
<b>IT</b>	Informationstechnik
<b>Java EE</b>	Java Enterprise Edition
<b>JMS</b>	Java Message Service
<b>JMX</b>	Java Management Extensions
<b>kWh</b>	Kilowattstunde
<b>LSE</b>	Large Scale Emulation

<b>MDA</b>	Model Driven Architecture
<b>MES</b>	Manufacturing Execution System
<b>MESA</b>	Manufacturing Enterprise Solutions Association
<b>MMS</b>	Mensch Maschine Schnittstelle
<b>MOF</b>	Meta Object Facility
<b>MoM</b>	Message oriented Middleware
<b>NC</b>	Numerical Control
<b>NP</b>	Netzparität
<b>NTP</b>	Network Time Protocol
<b>OLE</b>	Object Linking and Embedding
<b>OMG</b>	Object Management Group
<b>OPC</b>	OLE for Process Control
<b>OPC-UA</b>	OPC Unified Architecture
<b>OSI</b>	Open Systems Interconnection
<b>PC</b>	Personalcomputer
<b>PIM</b>	Platform Independent Model
<b>PLM</b>	Product Lifecycle Management
<b>Produktions-IT</b>	Produktionsnahe Informationstechnik
<b>PSM</b>	Platform Specific Model
<b>PV</b>	Photovoltaik (engl. Photovoltaic)
<b>PVECI</b>	Guide for PV Equipment Communication Interfaces
<b>PV-EIS</b>	Photovoltaik Equipment Interface Specification
<b>PV-Industrie</b>	Photovoltaikindustrie
<b>RCP</b>	Rich Client Platform
<b>RMI</b>	Remote Method Invocation
<b>RPC</b>	Remote Procedure Call
<b>RUP</b>	Rational Unified Process
<b>SCM</b>	Supply Chain Management
<b>SDK</b>	Software Development Kit
<b>SECS</b>	SEMI Equipment Communications
<b>SEM</b>	Specific Equipment Models
<b>SEMI</b>	Semiconductor Equipment and Materials International
<b>SiLS</b>	Software-in-the-Loop Simulation
<b>SOA</b>	Service Orientierte Architektur
<b>SPC</b>	Statistical Process Control
<b>SPS</b>	Speicherprogrammierbare Steuerung
<b>TCP</b>	Transmission Control Protocol
<b>TPTP</b>	Test & Performance Tools Platform
<b>UML</b>	Unified Modeling Language
<b>VDE</b>	Verband der Elektrotechnik Elektronik Informationstechnik e. V.
<b>VDI</b>	Verein Deutscher Ingenieure
<b>Wp</b>	Watt Peak
<b>XMI</b>	XML Metadata Interchange

## Formelzeichen

<b>A</b>	Adjazenzmatrix-Darstellung des virtuellen Fabrik-Logistiksystems
$C_{T_i}$	Echtzeit-Zeitbasis des Prozesses $T_i$
<b>E</b>	Menge der Emulatoren in einer virtuellen Fabrik
$e$	Ereignis $e$
$I$	Eine Iteration im Entwicklungsprozess
$i, j, k$	Indizes; $i, j, k \in \mathbb{N}$ ; $i, j, k \geq 0$
$\vec{L}(P, V)$	Graphen-Darstellung des virtuellen Fabrik-Logistiksystems
<b>N</b>	Menge der Nachrichten einer Host-Schnittstelle
$n_i$	Nachricht $n_i \in \mathbf{N}$
<b>P</b>	Menge der Ports im virtuellen Logistiksystem
$p_i$	Port $p_i \in P$
$\vec{P}_{M_i}$	Position einer Materialinstanz im Materialträger
$t$	Zeit in [s]
$t_{e_i}$	Zeitstempel zu Ereignis $e_i$ in [s]
$t_s$	Zeitpunkt, zu dem ein Datensatz gespeichert wird in [s]
$t_w$	Zeitpunkt, zu dem ein Datensatz wiederhergestellt wird in [s]
$T$	Ein Prozess
<b>V</b>	Menge der Verbindungen im virtuellen Fabrik-Logistiksystem
$\delta$	Maximale Abweichung der Zeit unterschiedlicher Zeitbasen in [s]
$\Delta t$	Zeitlicher Abstand in [s]
$\epsilon$	Auflösung einer Zeitbasis in [s]
$\tau$	Beschleunigungsfaktor





# 1 Einleitung

Die Nutzung der im Gegensatz zu konventionellen Energieträgern nahezu unbegrenzt verfügbaren Sonnenenergie gilt längst als eine wichtige Säule zukünftiger Energiekonzepte [z. B. BMU, 2010]. Einen Ansatz dazu stellt die Photovoltaik dar, also die direkte Umwandlung von Sonnenenergie in elektrische Energie (Abbildung 1.1 links). Die Photovoltaikindustrie (PV-Industrie), die den Anwendungskontext der vorliegenden Arbeit bildet, bietet entsprechende Produkte und Dienstleistungen an. Obgleich die der Photovoltaik zugrunde liegenden physikalischen Effekte bereits seit der zweiten Hälfte des 19. Jahrhunderts bekannt sind und seit dem Ende der 1950er Jahre für Spezialanwendungen in der Raumfahrt eingesetzt werden [Perlin, 2002], erlebte die Branche erst in den letzten Jahren einen gewaltigen Anstieg in der Nachfrage ihrer Produkte [EPIA, 2008]. Weltweit wurden sowohl seitens der Hersteller als auch seitens der Zulieferer und Ausrüster hohe Fertigungskapazitäten aufgebaut, um die starke Nachfrage des Marktes zu bedienen [Ruchat, 2009; EPIA, 2009]. [BSW-Solar, 2009c] gibt an, dass allein in Deutschland in einem Zeitraum von 10 Jahren die Kapazitäten zur Produktion von Solarzellen von 2 MWp<sup>1</sup> im Jahr 1998 auf geschätzte 1450 MWp im Jahre 2008 ausgebaut wurden. Gleichzeitig wuchs der Umsatz der PV-Industrie in Deutschland von 201 Millionen Euro im Jahr 2000 auf geschätzte 7 Milliarden Euro im Jahr 2008.

Vor diesem Hintergrund wird der PV-Industrie in Deutschland grundsätzlich ein hohes Entwicklungspotential zugebilligt [EPIA, 2009]. Allerdings steht sie gleichzeitig unter starkem Druck. Abgesehen von den längerfristigen Auswirkungen der aktuellen Wirtschaftskrise, die auch für die PV-Industrie heute noch nicht absehbar sind, hat sie einige inhärente Herausforderungen des Marktes zu meistern. An prominenter Stelle wird in der Literatur, u. a. bei [EPIA, 2008; SEMI PV Group, 2008], immer wieder die Notwendigkeit genannt, die Erzeugungskosten pro Kilowattstunde Solarstrom zu reduzieren. Zunächst soll das Preisniveau erreicht werden, zu dem eine Kilowattstunde elektrischer Energie für den Verbraucher aus dem Stromnetz verfügbar ist. An diesen Punkten ist jeweils die *Netzparität* (engl. *grid parity*) erreicht (siehe Abbildung 1.1 rechts: Netzparität (NP); vereinfachend wird nur eine Preisklasse für das Netz dargestellt). Schließlich soll dieser Preis unterschritten werden. Das Erreichen bzw. Unterschreiten der Netzparität wird als die notwendige Voraussetzung angesehen, die die Nutzung von Solarstrom wirtschaftlich interessant macht, die PV-Industrie von Subventionen unabhängig werden lässt, der Industrie nachhaltiges Wachstum ermöglicht und schließlich zu einem messbaren Beitrag dieser Industrie zur Lösung des weltweiten Energieproblems führt [EPIA, 2008].

Nachdem die Marktsituation der vergangenen Jahre der eines Verkäufermarkts, geprägt durch schnelles Wachstum und eine hohe Ausbringungsmenge, entsprach, hat diese sich

---

<sup>1</sup>Die Einheit Watt Peak (Wp) gibt die maximal abgegebene elektrische Leistung unter standardisierten Testbedingungen an [Photon-PV-Lexikon, 2007].

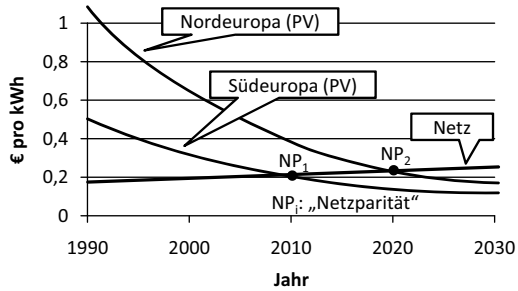
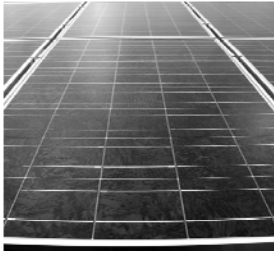


Abbildung 1.1: Links: Beispiel einer Photovoltaikanlage, Quelle: BSW-Solar/Upmann [BSW-Solar, 2009b]. Rechts: Das Konzept der Netzparität, basierend auf [EPIA, 2008]

über die letzten Monate zu der eines Käufermarkts entwickelt. Damit stehen andere Fragestellungen, wie der erreichbare Herstellungspreis, die erzielbare Qualität und die Eröffnung neuer Märkte und Geschäftsmodelle, im Fokus [Berwind, 2009]. Diesen Herausforderungen versucht die PV-Industrie auf unterschiedlichsten Ebenen zu begegnen: Neben Ansätzen zur Optimierung des Wirkungsgrads von Solarzellen, Modulen und Systemen und der Entwicklung gänzlich neuer Zell- und Modulkonzepte liegt ein Schwerpunkt der Bemühungen in der Optimierung der Produktion entlang der Wertschöpfungskette der PV-Industrie. Ansatzpunkte dafür liegen sowohl in der Entwicklung neuer Fertigungsprozesse und der Nutzung anderer Materialien, als auch in der Optimierung der genutzten Produktionssysteme [z. B. Schwirtlich, 2008; Reddig, 2009]. In diesem Zusammenhang spielt die Informationstechnik (IT) eine zunehmend wichtigere Rolle.

## 1.1 Problemstellung

Die Bedeutung der IT für die erfolgreiche Massenfertigung, die in vergleichbaren Branchen seit vielen Jahren allgemein akzeptiert wird, wird erst seit etwa zweieinhalb Jahren auch von der PV-Industrie in größerer Breite wahrgenommen [Baylies u. a., 2008]. Zwischenzeitlich herrscht weitgehend Einigkeit darüber, dass eine leistungsfähige IT-Unterstützung in der Produktion einen wichtigen Beitrag zur Lösung der oben geschilderten Herausforderungen liefern kann und muss [Lillian, 2009]. [Meier, 2007] und [Bergholz, 2008] nennen als Ziele für eine solche die Beschleunigung des Anlaufs, die bessere Integration der Produktion in die übergeordneten Geschäftsprozesse, die Optimierung von Ausbeute und Qualität, die Verbesserung der Ressourcenausnutzung und des Durchsatzes, bessere Möglichkeiten zur Fertigungssteuerung und Prozessregelung und ein erhöhtes Maß an Flexibilität zur Reaktion auf geänderte Anforderungen. Ein Beitrag zur Beschleunigung des Anlaufs wird nach [Bergholz, 2008] dann erzielt, wenn IT-Systeme, die umfassende Analysemöglichkeiten für die Diagnose von Fehlern im Gesamtsystem bieten, bereits zu einem frühen Zeitpunkt zur Verfügung stehen.

[Rein und Preu, 2007; Pfitzner und Schellenberger, 2009] sehen ebenfalls ein großes Potential in der Integration von Qualitätskontrollsystemen in die Produktion und dem Einsatz von Werkzeugen zur Datenanalyse, Prozessregelung und Steuerung. [Mareck, 2009] bestätigt diese Einschätzung für den SPC-Bereich und beschreibt erste Erfolge mit der Einführung solcher Systeme. [Rauter, 2009] nennt für die Arbeitsgruppe der Zellfertiger, in der sich neun große Hersteller organisiert haben, den Bereich der IT-Systeme als ein relevantes Evaluationskriterium für Zulieferer und Ausrüster. Um das Potential der produktionsnahen IT-Systeme in der Praxis nutzen zu können, werden Softwarelösungen benötigt, die die skizzierten Ziele durch entsprechende Funktionalität unterstützen und sich zu einem wettbewerbsfähigen Preis in Produktionssysteme integrieren und betreiben lassen.

Neben den Chancen, die solche IT-Systeme bieten, ist ihr Einsatz allerdings, auch unabhängig vom Umfeld der PV-Industrie, mit einer Reihe von Risiken behaftet. Stellvertretend sei für Projekte zum Aufbau oder Umbau von Produktionsstätten die Verzögerung des Anlaufs aufgrund von Softwarefehlern oder Zeitverzug in Software-Teilprojekten genannt [Zäh und Wunsch, 2005; Ammermann und Lohse, 2007]. Insgesamt werden für die verschiedenen Phasen des Fabriklebenszyklus vielfältige Risiken im Zusammenhang mit dem Einsatz produktionsnaher IT-Systeme beschrieben [z. B. Jindal und Khandelwal, 2005; Dreiss, 2007; Sauer und Ebel, 2007; Buchner, 2008]. Dem Autor ist zum Zeitpunkt der Abfassung dieser Arbeit keine umfassende quantitative Untersuchung solcher Risiken für die PV-Industrie bekannt. Allerdings lassen verschiedene Anforderungen der PV-Industrie an die IT-Schnittstelle von Anlagen, die deren Anbindung an IT-Systeme wie die oben skizzierten ermöglicht, auf ein deutliches Risikobewusstsein schließen: So war eine zentrale Anforderung im Standardisierungsprozess dieser Schnittstelle die Möglichkeit zum "Offline"-Betrieb der Anlagen, um die Abhängigkeit zu den Softwaresystemen auf Fabrikebene zu verringern [Meier, 2008a].

## 1.2 Zielsetzung und Vorgehensweise

Bereits ab einer frühen Phase des Anlaufs bietet der zielgerichtete Einsatz produktionsnaher IT-Lösungen Ansatzpunkte zur Unterstützung der Optimierungsbemühungen der PV-Industrie. Manufacturing Execution Systems (MESs) stellen eine wichtige Klasse produktionsnaher IT-Systeme dar, die das skizzierte Optimierungspotential nutzbar machen. Dabei müssen die Risiken, die mit der Einführung und dem Einsatz solcher IT-Lösungen einhergehen, beherrscht werden.

Ziel dieser Arbeit ist die Entwicklung eines Verfahrens, das im Kontext der PV-Industrie einen Beitrag dazu leistet, angemessene MES-Lösungen frühzeitiger im Fabriklebenszyklus verfügbar und die mit dem MES-Engineering verbundenen Risiken besser beherrschbar zu machen. Es soll damit der PV-Industrie den Zugang zu und die Akzeptanz von MES-Lösungen erleichtern, die die Verfolgung der Optimierungsziele der Branche unterstützen.

Dieses Ziel wird mit dem in Abbildung 1.2 dargestellten Forschungsansatz verfolgt:

**Kapitel 2** legt die Grundlage für die folgenden Schritte. Im Anschluss an die Definition der zentralen Begrifflichkeiten wird zunächst das Umfeld der PV-Industrie, das Konzept des Fabriklebenszyklus im Kontext dieser Industrie und deren Verständnis von produktionsnaher IT erläutert. Aufgrund seiner Bedeutung für diese Arbeit wird das Rahmenwerk verfügbarer Normen für die Domäne der Produktions-IT in besonderer Weise beleuchtet.

**Kapitel 3** analysiert die Herausforderungen des MES-Engineering in der PV-Industrie und leitet daraus das Potential des emulationsgestützten Verfahrens des MES-Engineering ab. Auf dieser Basis werden die Anforderungen an eine geeignete Werkzeugumgebung einerseits und das Verfahren andererseits ermittelt.

**Kapitel 4** spiegelt die Ergebnisse der Analyse am aktuellen Stand der Wissenschaft und Technik und grenzt das im Rahmen dieser Arbeit zu entwickelnde Verfahren von anderen Arbeiten ab.

**Kapitel 5** beschreibt das Architekturkonzept, das Metamodell und die prototypische Implementierung der Emulationsplattform unter Berücksichtigung der in Kapitel 3 ermittelten Anforderungen.

**Kapitel 6** ist der Entwicklung des Verfahrens des emulationsgestützten MES-Engineering gewidmet, das die effiziente und effektive Nutzung der Emulationsplattform in der Industriepraxis ermöglicht.

**Kapitel 7** umfasst schließlich die Validierung der Emulationsplattform und des Verfahrens anhand von Fallstudien aus der industriellen Praxis und gleicht die in Kapitel 3 formulierten Anforderungen mit den erzielten Ergebnissen ab.

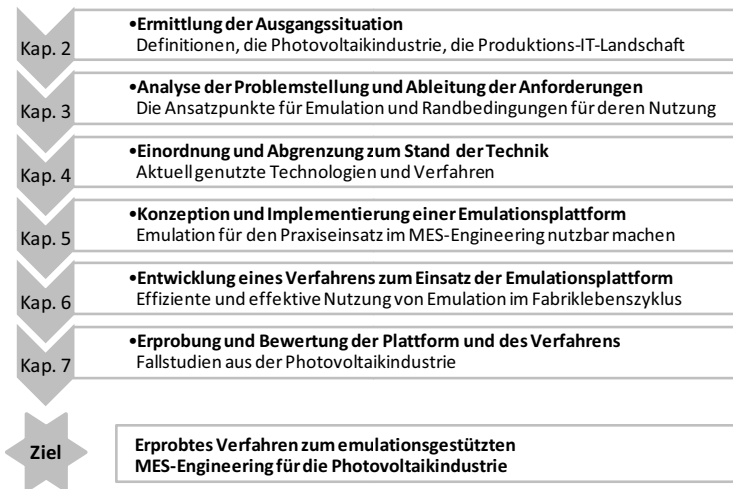


Abbildung 1.2: Der Forschungsansatz im Überblick

## 2 Ausgangssituation

Die folgenden Abschnitte beschreiben den Kontext und wichtige Randbedingungen für das Engineering und den Betrieb von MES-Lösungen. Dabei stehen die Besonderheiten der PV-Industrie im Fokus.

### 2.1 Begriffe und Definitionen

#### 2.1.1 Produkt-, Prozess und Fabriklebenszyklus

Seit längerer Zeit schon wird in der Produktentwicklung, der Prozessentwicklung und der Fabrikplanung ein ganzheitlicher Ansatz zur Betrachtung des jeweiligen Untersuchungsgegenstands gewählt: Ein Ansatz, der den gesamten *Lebenszyklus* berücksichtigt (z.B. [Westkämper, 2006; Constantinescu u. a., 2006]). Die drei angesprochenen Aufgabengebiete sind in der Praxis allerdings nicht völlig unabhängig voneinander zu bearbeiten, sondern müssen spätestens zu dem Zeitpunkt synchronisiert sein, zu dem die Produktion eines gegebenen Produktes mit den gewählten Prozessen in einer Fabrik aufgenommen werden soll. Aus dem Grund werden diese Aufgabengebiete auch als drei Dimensionen der Produktionsplanung betrachtet. Abbildung 2.1 (links) greift die beiden Dimensionen des *Fabriklebenszyklus* und des *Produktlebenszyklus* heraus und stellt sie mit ihren Phasen über die Zeit dar, auf die im weiteren Verlauf dieser Arbeit verschiedentlich Bezug genommen wird. In ähnlicher Weise beschreibt der *Prozesslebenszyklus* die Phasen der Entwicklung von Fertigungsprozessen über die Zeit. Dem durchgängigen Management und der geeigneten Synchronisation dieser Prozesse wird ein hohes Potential bescheinigt: In der Literatur werden beispielsweise die Verkürzung der Produkteinführungszeit (engl. time-to-market), Kosteneinsparungen und ein reduzierter Nacharbeitsbedarf genannt [Westkämper, 2006].

Verschiedene Systemanbieter offerieren integrierte IT-Werkzeuge zur durchgängigen Abwicklung ausgewählter Phasen der beschriebenen Lebenszyklen und zu deren Synchronisation, um das erwartete Potential in der Praxis nutzbar zu machen. So existieren beispielsweise Werkzeugketten, die die Durchgängigkeit von der Entwicklung des Produkts im Computer Aided Design (CAD) System über die Arbeitsplanung bis zur Erstellung von Steuerungsprogrammen auf Automatisierungscomputern ermöglichen und dabei Produkt- und Fabriklebenszyklus synchronisieren. Andere Anwendungen unterstützen in verschiedenen Phasen des Fabriklebenszyklus, angefangen von der frühen konzeptionellen Phase, über die konkrete Designphase bis hin zur Inbetriebnahme, der Wartung und dem Umbau. z.B. [VDI 4499, 2008; Sauer, 2007a; Constantinescu u. a., 2009]

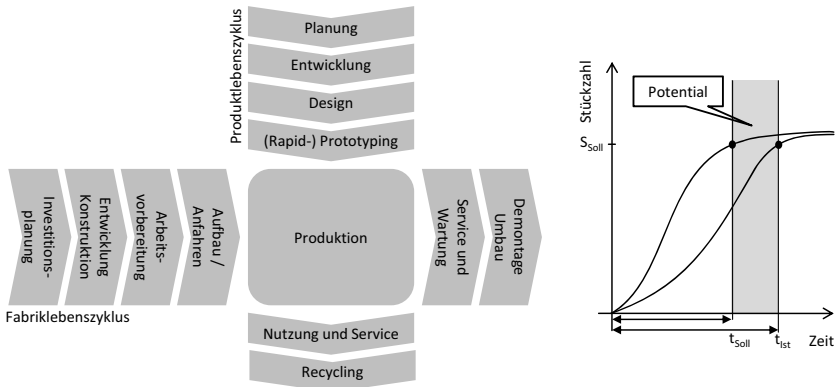


Abbildung 2.1: Links: Fabriklebenszyklus und Produktlebenszyklus, basierend auf [Constantinescu u. a., 2006]. Rechts: Der Produktionsanlauf, basierend auf [Ammermann und Lohse, 2007]

## 2.1.2 Anlage

Eine *Anlage* (engl. equipment) wird als ein eigenständiges, in der Regel ortsfestes, Betriebsmittel definiert, das aus einer Menge von Komponenten (Teilsystemen) besteht. Die Gesamtheit der Komponenten erfüllt durch ihre Interaktion definierte Funktionen, die der Außenwelt über Schnittstellen zur Verfügung gestellt werden. Über weitere Schnittstellen wird die Anlage mit den für den Betrieb benötigten Medien bzw. Hilfsstoffen und Energie versorgt und an externe IT-Systeme angebunden. Wo nicht anders angegeben, werden Anlagen als *Blackboxes* behandelt, also als Systeme, die lediglich durch die Syntax und Semantik ihrer Schnittstellen beschrieben sind.

## 2.1.3 Produktionsanlauf und Ramp-Up

Der deutsche Begriff *Produktionsanlauf* und der englische Begriff *Ramp-up* werden im Folgenden synonym verwendet. Beide beschreiben den Zeitraum, in dem für eine Produktion die Ausbringung der herzustellenden Produkte von der Stückzahl Null bis zur geplanten Stückzahl angehoben wird (siehe Abbildung 2.1, rechts). [Ammermann und Lohse, 2007] nennen die Qualität des Endprodukts, die Reduktion der Kosten für die Anlaufphase und die Reduktion der Dauer der Anlaufphase als die wichtigsten Optimierungsziele der Anlaufphase. Die Notwendigkeit zur Durchführung eines Produktionsanlaufes ergibt sich aus produktbezogenen, parametrischen oder strukturellen Änderungen in der Produktion [Schmahls, 2001]. Abhängig von diesen möglichen Auslösern eines Produktionsanlaufes beschreiben [Ammermann und Lohse, 2007] operative, strukturelle und konzeptionelle Anlaufsznarien, die sich insbesondere im Hinblick auf die auszuführenden Arbeiten und den damit einhergehenden Zeitbedarf unterscheiden. Erstere Anlaufsznarien decken die routinemäßige Umrüstung zum Variantenwechsel ab, die in

der Größenordnung von Stunden abgeschlossen werden. Weitere umfassen komplexere Konfigurationsänderungen bestehender Linien. Die Drittgeantten bilden den Fall umfangreicher Umbaumaßnahmen ab, in denen wieder die ersten Phasen des Fabriklebenszyklus durchlaufen werden und damit ein Neuanlauf erforderlich wird, dessen Dauer sich in der Größenordnung von Wochen oder Monaten bewegt.

Verfahren zum *virtuellen Produktionsanlauf* zielen, neben der Optimierung der betrieblichen Prozesse, auf eine Verkürzung des für den Produktionsanlauf benötigten Zeitraums und eine Reduktion der Kosten für die Anlaufphase ab (siehe “Potential” in Abbildung 2.1 rechts). Dabei kommen Testumgebungen zum Einsatz, die die Inbetriebnahme zu evaluierender Systeme unter möglichst realistischen Bedingungen, unabhängig von der realen Produktion, erlauben. So kann bereits in frühen Phasen des Fabriklebenszyklus, noch bevor die Produktion physisch aufgebaut ist, ein der realen Inbetriebnahme ähnliches Einsatzszenario nachgebildet werden. Verfahren zum virtuellen Produktionsanlauf unterstützen also eine Parallelisierung ursprünglich sequentiell ausgeführter Arbeitsschritte im Fabriklebenszyklus. Eine wichtige Voraussetzung zur Umsetzung des Verfahrens ist die Verfügbarkeit einer geeigneten Testumgebung (z.B. [Klemm u. a., 2006; Ammermann und Lohse, 2007]).

## 2.1.4 Emulation und Simulation

Sowohl im Alltag als auch in der Literatur fällt die bisweilen inkonsistente Nutzung der Begriffe *Emulation* und *Simulation* auf, deren Abgrenzung in der Praxis nicht immer ganz einfach ist.

**Emulation** geht auf das lateinische Verb “aemulari” (deutsch: wetteifern) zurück [Duden FWB, 2007] und beschreibt in der Informationstechnik die funktionale “Nachahmung” eines Systems durch ein anderes System. Das nachahmende System wird als *Emulator* bezeichnet. Emulatoren werden in der industriellen Anwendung unter anderem für den Test von Software für spezielle Ziel-Hardwareplattformen genutzt. Abbildung 2.2 zeigt ein Beispiel die Entwicklung von Software für Mobilfunkgeräte. Ein Entwickler, der Software für ein ausgewähltes Mobilgerät an einem Personalcomputer (PC) in einer Standard-Entwicklungsumgebung schreibt, kann den von ihm generierten ausführbaren Code, der nicht direkt auf der PC-Hardware lauffähig ist, innerhalb eines Emulators seines anvisierten Zielsystems ausführen. Der Emulator ahmt die Ziel-Hardwareplattform auf dem Entwickler-PC nach und führt den Programmcode aus. Er ermöglicht damit dem Entwickler, sein Programm auf seinem Entwicklungssystem zu testen, ohne dass er dafür ein reales Exemplar des Mobilgeräts benötigt. Ein Emulator unterstützt dabei nicht notwendiger Weise die vollständige Funktionalität des nachgebildeten Systems. Seine Funktionalität kann sich auf ausgewählte Bereiche beschränken. [Papandreou u. a., 2003; Wellington und Kubischta, 2003; Nikkarinen und Shemyak, 2005; Maeda u. a., 2007] beschreiben weitere Beispiele zum Einsatz von Emulation in der Entwicklung in anderen Industrien.

**Simulation** geht auf das lateinische Substantiv “simulatio” (deutsch: Vorspiegelung) zurück [Duden UWB, 2007] und ist nach [VDI 3633, 1996] “ein Verfahren zur

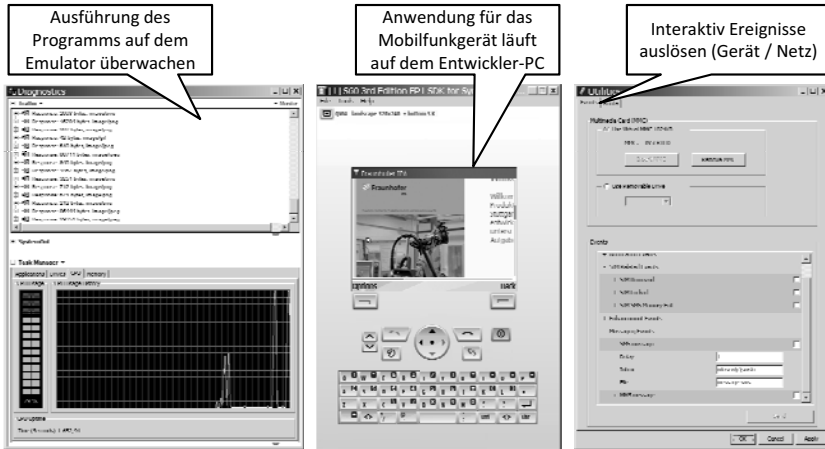


Abbildung 2.2: Emulation in der Praxis – Einsatz eines Emulators für die Entwicklung von Software für Mobilfunkgeräte am Beispiel des S60 3rd Edition SDK for Symbian OS [Nokia, 2009]

Nachbildung eines Systems mit seinen dynamischen Prozessen in einem experimentierbaren [Simulations-]Modell, um zu Erkenntnissen zu gelangen, die auf die Wirklichkeit übertragbar sind.“ Dieses Modell bildet das betrachtete System hinsichtlich der untersuchten Eigenschaften innerhalb einer dem Versuchsziel angepassten Toleranz ab und wird von einem *Simulator* implementiert. An der Implementierung des Modells werden *Experimente* zur Untersuchung der zugrunde liegenden Fragestellung durchgeführt.

Ein Emulator im oben beschriebenen Sinne kann das Verfahren der Simulation für die Nachbildung des Verhaltens des zu emulierenden Systems nutzen. So bietet beispielsweise der in Abbildung 2.2 rechts gezeigte Simulator die Möglichkeit, Ereignisse des Mobilfunknetzes zu generieren, die der Emulator des Mobilfunkgeräts in gleicher Weise wie das reale Telefon an die laufende Anwendung weitergibt.

### 2.1.5 Modell und Metamodell

Die Definition nachfolgender Begriffe basiert auf entsprechenden Definitionen aus [OMG, 2003] und [OMG, 2005]: Ein

**Modell** beschreibt oder spezifiziert einen Betrachtungsgegenstand, in diesem Zusammenhang ein technisches System, für einen definierten Zweck. Die Beschreibung kann aus einer ausgewählten *Perspektive*, das heißt in abstrahierter Form, erfolgen, wobei ausgewählte Details unterdrückt werden. Modelle werden in der Regel durch Kombinationen aus graphischen und textuellen Notationen dargestellt, wobei



letztere in natürlicher Sprache oder in speziellen Modellierungssprachen abgefasst sein können. Die wichtigsten Bestandteile eines solchen Modells sind *Elemente* und *Verbindungen* zwischen Elementen.

**Metamodell** ist selbst ein Modell im oben beschriebenen Sinne und spezifiziert die *abstrakte Syntax*, also Typen, Strukturen und Beziehungen, einer Modellierungssprache. Formale, durch ein Regelsystem beschriebene, Modellierungssprachen sind die Voraussetzung für die Möglichkeit zur automatischen Verarbeitung der Modelle.

## 2.2 Die PV-Industrie

Zur direkten Umwandlung von Sonnenenergie in elektrische Energie stellt die PV-Industrie unterschiedlichste PV-Systeme für eine große Bandbreite von Anwendungsszenarien her. Typische Anwendungsszenarien reichen von der Energieversorgung autonomer Systeme bis hin zu Solarkraftwerken, die ähnlich einem konventionellen Kraftwerk größere Anschlussleistungen in ein Stromnetz einspeisen. Die Energieumwandlung erfolgt durch *Solarzellen*, die zu *Modulen* gruppiert und dann verbaut werden. Abhängig von der vorgesehenen Anwendung unterscheiden sich die Module im Hinblick auf die elektrische Anschlussleistung, die Größe, das Gewicht, die Robustheit gegenüber Umwelteinflüssen, die Geometrie und andere Kenngrößen.

### 2.2.1 Einige Grundlagen

Bei der Herstellung der Solarmodule kommen verschiedene Basistechnologien zum Einsatz. Die aktuell verbreitet eingesetzten sind in Abbildung 2.3 aufgeführt. Zunächst sind die *kristallinen* und die *Dünnschicht*-Technologien zu unterscheiden, die jeweils charakteristische Anforderungen an die Fertigung stellen [Reddig, 2009]: Bei der Nutzung der kristallinen Technologien dient, ähnlich wie bei vielen Produkten der Halbleiterindustrie, eine dünne Siliziumscheibe (engl. Wafer) als Ausgangsprodukt für die Fertigung von *Solarzellen*, wobei der Wafer schließlich einen integralen Bestandteil der Solarzelle darstellt. Bei der Nutzung der Dünnschicht-Technologien werden die Solarzellen in Form einer dünnen Schicht auf ein Trägermaterial (Substrat), typischer Weise Glas, aufgebracht, das nicht Bestandteil der Zelle ist. Zu beiden Basistechnologien gibt es jeweils mehrere konkrete Ausprägungen. Wichtige Unterscheidungsmerkmale stellen insbesondere die verwendeten Ausgangsmaterialien und der technologische Aufbau einer Solarzelle dar.

Kristalline Solarmodule unterscheiden sich in verschiedenen Kenngrößen von Dünnschicht-Solarmodulen. Stellvertretend seien die optische Erscheinung, der Verbrauch an Halbleitermaterialien und der erreichbare Wirkungsgrad genannt. Auf Basis von Dünnschicht-Technologien gefertigte Solarmodule wirken optisch ruhiger. Während die Herstellung von Solarmodulen auf Basis von Dünnschicht-Technologien gegenüber den kristallinen Technologien ein deutlich geringerer Verbrauch von Halbleitermaterialien auszeichnet, lassen sich auf Basis der letzteren Technologien Module mit einem höheren Wirkungsgrad herstellen [EPIA, 2008].

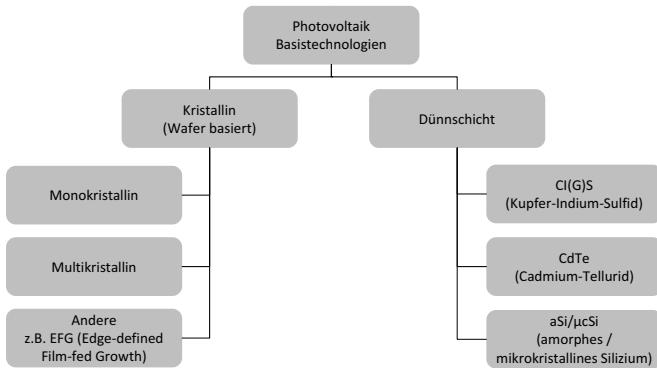


Abbildung 2.3: Basistechnologien der Photovoltaikindustrie basierend auf [Meier, 2008b]

Abbildung 2.4 zeigt einen schematischen Vergleich der beiden Basistechnologien im Hinblick auf die Lieferkette (oben) und die konzeptionellen Prozessketten (unten). Die Liefer- und Prozessketten vor der Systemstufe, in der Komplettsysteme zur Stromerzeugung mit Wechselrichtern und weiterem Zubehör zusammengestellt und vertrieben werden, unterscheiden sich deutlich. Dennoch existieren im Hinblick auf die Produktion eine Reihe von Gemeinsamkeiten, die eine gemeinsame Behandlung beider Welten im Hinblick auf die produktionsnahe IT sinnvoll erscheinen lassen (siehe auch [SEMI PV2, 2009]). Die primären Anwendungsgebiete der nachfolgenden Betrachtungen dieser Arbeit sind die Zellen- und Modulfertigung in kristalliner Basistechnologie bzw. die Front-end und Back-end Fertigung im Dünnschicht-Bereich.

## 2.2.2 Die Fertigungsorganisation

Zum Verständnis des notwendigen und hilfreichen Maßes an IT-Unterstützung für die Produktion ist eine Betrachtung der Leistungsprozesse und der Fertigungsorganisation der PV-Industrie unabdingbar. Um die eingangs geschilderte hohe Nachfrage nach PV-Modulen zu bedienen bzw. wirtschaftlich auszuschöpfen, hat die PV-Industrie sehr konsequent auf das Konzept der *Massenfertigung* gesetzt. Es beschreibt die Produktion einer großen Zahl gleichartiger Produkteinheiten über einen längeren Zeitraum hinweg [Berning, 2001]. Zu diesem Zweck kommen hochspezialisierte Anlagen zum Einsatz, die die oben beschriebenen Kernprozesse weitestgehend automatisch bei geringem Personalbedarf zur Verfügung stellen. Errichtung und Betrieb derartiger Produktionslinien ist mit einem hohen Kapitaleinsatz verbunden. So liegen die Investitionskosten für eine kristalline Zellenfertigung mit einer Jahreskapazität von 361 MWp (bestehend aus 7 Turnkey-Linien > 50 MWp) in der Größenordnung von 130 Millionen Euro [Fath, 2009].

Die Notwendigkeit für Variationen der hergestellten Produkteinheiten sind bisher nur sehr begrenzt gegeben. Vor diesem Hintergrund spielen die Problemstellungen der Vari-

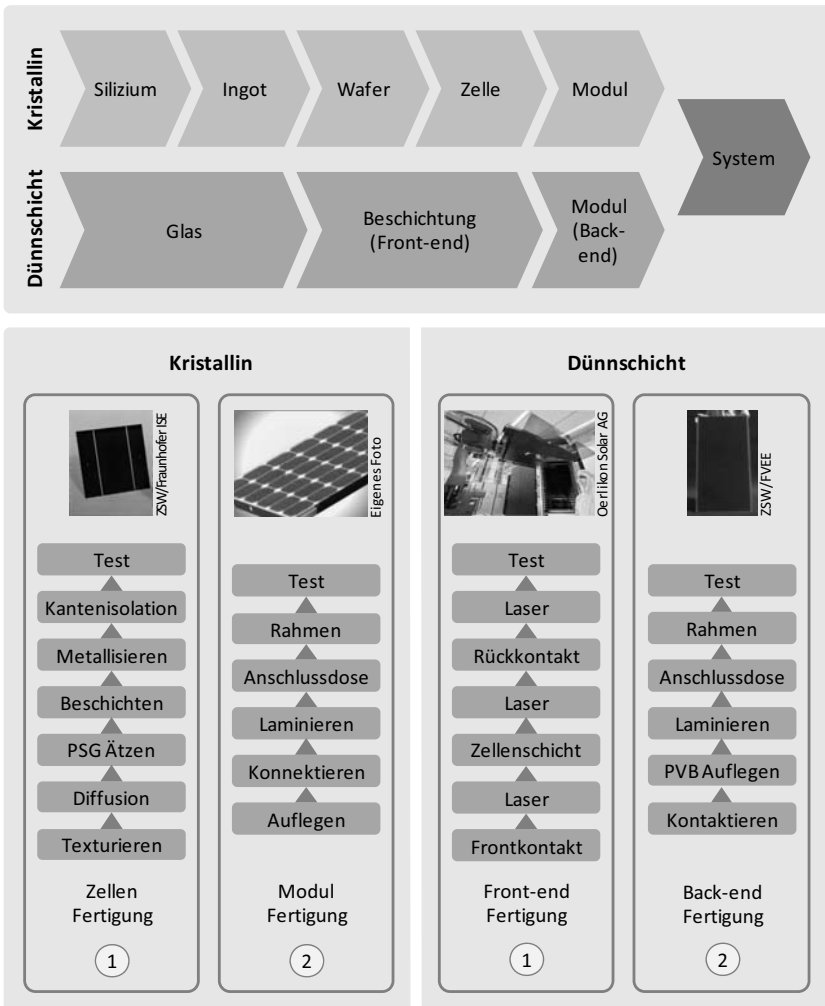


Abbildung 2.4: Kristalline Fertigung versus Dünnschicht Fertigung: Liefer- und Prozessketten im Vergleich, basierend auf [Aulich, 2009] (oben) und [Meier, 2008b] (unten)

antenfertigung im Moment nur eine untergeordnete Rolle. Erste Ansätze in dieser Richtung sind jedoch bereits erkennbar. So bietet der Ausrüster Applied Materials bereits Linien an, die Dünnschicht-Module in voller Größe (Länge: 2,60m, Breite: 2,20m) oder in Vierteln dieser Größe fertigen können [Beitel, 2008]. Weitere Anreize zu Entwicklungen

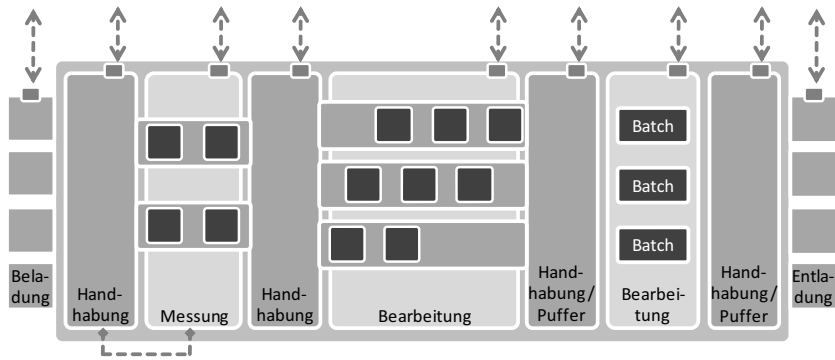


Abbildung 2.5: Schematische Darstellung eines Inline-Fertigungsabschnitts (Beispiel-szenario)

in dieser Richtung könnte in Zukunft der Einsatzbereich der gebäudeintegrierten PV-Systeme schaffen. Gefertigt wird üblicher Weise *kundenanonym*, das heißt auf Basis von Absatzprognosen der Fertiger, wobei die Fertigungen auf den 24/7 Betrieb, also einen möglichst unterbrechungsfreien Betrieb rund um die Uhr, ausgelegt werden [Reddig, 2009].

Im Hinblick auf die Fertigungsorganisation kommen in der PV-Industrie aktuell noch sehr unterschiedliche Konzepte zum Einsatz, was sicherlich mit durch das noch vergleichsweise junge Alter dieser Branche bedingt ist. Prägend für die Fertigungsorganisation ist insbesondere der für Kernprozesse genutzte Typ von Prozessanlagen. In diesem Zusammenhang wird zwischen *Batch*-Prozessen (siehe Abbildung 2.7) und *Inline*-Prozessen unterschieden (siehe Abbildung 2.5). Während im ersten Fall zahlreiche Substrate gemeinsam transportiert und einem Prozess ausgesetzt werden, werden im zweiten Fall Substrate einzeln transportiert und prozessiert.

Daraus ergeben sich unterschiedliche Fertigungsorganisationen. Das Spektrum im Fall von Batch-Prozessen reicht von einer *technologieorientierten Strukturierung*, einer *Werkstattfertigung*, bis hin zu einer *Reihenfertigung* mit *Außenverkettung*. Der erstgenannte Organisationstyp kommt insbesondere in Laborlinien zum Einsatz. Der letztgenannte wird für die Massenfertigung genutzt. Substrate werden für den Batch-Betrieb in *Materialträger* (engl. *Carrier*) eingehodet und in diesen transportiert bzw. in Anlagen eingefahren. Im Fall von Inline-Prozessen wird in der Regel auf Basis von *Transferstraßen getaktet* gefertigt. Das heißt, einzelne Anlagen sind in Innenverkettung miteinander verbunden und werden von den Substraten in fester Abfolge in einer Richtung durchlaufen. Die Materialzuführung und -abführung zu und von Transferstraßen kann ebenfalls über Materialträger gelöst werden. Für einzelne Prozesse können innerhalb der Anlagen Batches gebildet werden. [Berning, 2001; Meier, 2008b; Reddig, 2009]

### 2.2.3 Mengengerüste

Einige Zahlenbeispiele sollen die Randbedingungen für die später beschriebene Implementierung von IT-Systemen in der Produktion verdeutlichen. Für aktuelle Linien zur kristallinen Zellfertigung liegen die Anforderung an den Durchsatz von Anlagen bei etwa 2400-3600 Wafern pro Stunde [Rein u. a., 2008]. Daraus ergeben sich Taktzeiten von bis zu einer Sekunde, in denen einzelne Wafer zu bearbeiten sind. Geht man von einer Fabrik mit der Jahresleistung von 300 MWp, einer durchschnittlichen Leistung einer einzelnen Zelle von ca. 3,5 Wp pro Zelle und dem oben beschriebenen schematischen Prozess aus, so ergibt sich für eine solche Fabrik eine Anlagenkonfiguration bestehend aus mehreren hundert Anlagen.

Betrachtet man zum Vergleich eine Dünnschichtfertigung (Front-end / Back-end), die über eine Jahresproduktionskapazität von 50 MWp verfügt, so ergibt sich bei einer durchschnittlichen Modulleistung von 170 Wp (z.B. [Centrotherm, 2009], ohne Berücksichtigung der real erreichbaren Ausbeute, ein theoretischer Wert von ca. 294 000 zu fertigenden Modulen pro Jahr und damit ein Ausstoß von ca. 34 Modulen pro Stunde, wobei, ohne Berücksichtigung von Logistik- und Messsystemen, eine Anlagenkonfiguration in der Größenordnung von 30 Anlagen im Einsatz ist.

### 2.2.4 Produkt-, Prozess- und Fabriklebenszyklus in der PV-Industrie

Die Entwicklung und Optimierung von Fertigungsprozessen auf der Grundlage der oben beschriebenen Basistechnologien und die Suche nach neuen Basistechnologien bildet einen Schwerpunkt der Forschungstätigkeit der PV-Industrie. Produkte, wie kristalline Zellen oder Module, werden entsprechend der technologischen Randbedingungen entworfen und über die verfügbaren Prozessketten gefertigt. Eine Fabrik in der Produktionsphase ihres Lebenszyklus' stellt ausgewählte Technologien und Prozessketten zur Verfügung, die zur Serienfertigung von Produkten genutzt werden können. z.B. [Q-Cells-GB, 2009]

Besteht der Bedarf eines Kapazitätsaufbaus oder einer Technologieerweiterung, werden Fabriken entweder durch die Hersteller bzw. beauftragte Generalunternehmer in Eigenregie geplant und gebaut bzw. umgebaut oder schlüsselfertig (engl. *turnkey*) als Produkt erworben. Beim Bau in Eigenregie wird die für die Fabrik benötigte Ausstattung am Markt ausgewählt und erworben. Anbieter schlüsselfertiger Fabriken verkaufen vollständig ausgestattete Fabriken mit einer Zertifizierung für ausgewählte Produkte und garantieren für den Betrieb die Einhaltung spezifischer Leistungskennzahlen (z.B. [AppliedMaterials, 2009; Centrotherm, 2009; Oerlikon, 2009]). Der Zeitaufwand für den Bau einer Fabrik bis zur Fertigung des ersten Substrats liegt in der Größenordnung von etwa einem Jahr (Beispiel Signet Solar: 10 Monate [Beitel, 2008]). Eine wichtige Anforderung an das Design von Fabriken in der PV-Industrie bildet die Fähigkeit zur Kapazitätserweiterung [Konrad u. a., 2009].

## 2.3 Die Produktions-IT-Landschaft

Die Begriffe *produktionsnahe IT (Produktions-IT)* bzw. *Produktions-IT-Landschaft* umfassen alle Hard- und Softwaresysteme, die unmittelbar den Betrieb einer Produktion, also die Aktivität “Produktion” in Abbildung 2.1, unterstützen. Die informationstechnische Integration der Prozesse in der Produktion in übergeordnete Geschäftsprozesse, welche die Produkt-, Fabrik- und Prozesslebenszyklen bzw. die Lieferkette abbilden, wird ebenfalls durch die Produktions-IT ermöglicht. Mit dieser Integration werden insbesondere zwei Ziele verfolgt [Meier, 2007]:

- Die Versorgung der Geschäftsprozesse mit korrekten Daten aus der Produktion.
- Die Steuerung der Produktion aus übergeordneten Geschäftsprozessen.

Aus der Produktion werden Daten wie Statusinformationen zu Kundenaufträgen und verfügbaren Kapazitäten oder Prozess- und Qualitätsdaten an übergeordnete Geschäftsprozesse weitergeleitet. An die Produktion wiederum werden Daten wie Prozessparameter übergeben. Auf dieser Basis lassen sich vielfältige Regelschleifen zur Steuerung des Produktionssystems realisieren, die sich über weite Abschnitte der Geschäftsprozessketten erstrecken können. Der abstrakte Begriff der Produktions-IT-Landschaft wird im Folgenden insbesondere im Hinblick auf die Aufgaben der zugehörigen Hard- und Softwaresysteme und Systemarchitekturen näher beleuchtet. Während sich Aufgaben, Architekturen und Begrifflichkeiten im Detail durchaus in Abhängigkeit von der betrachteten Industrie oder Unternehmung unterscheiden, existieren eine Reihe allgemeingültiger Konzepte, die in den folgenden Abschnitten beschrieben und um Spezifika der PV-Industrie ergänzt werden.

### 2.3.1 Das Ebenenmodell der Produktions-IT-Landschaft

Ein weit verbreiteter Ansatz zur Definition des Umfangs und der Aufgabenbereiche der Produktions-IT-Landschaft leitet sich von der Planungs- und Steuerungshierarchie eines Unternehmens ab, die auch in der Norm [ANSI/ISA-S95 Part I, 2000] beschrieben wird. Die Planungs- und Steuerungshierarchie geht von der hierarchischen Zerlegung eines Unternehmens von der Ebene des Gesamtunternehmens, über Standorte, Bereiche, Linien und Zellen bis zum Fertigungsprozess aus. Abbildung 2.6 spiegelt dieses Modell wieder (siehe Ebenen E4-E0). Jede Hierarchieebene ist für ein definiertes Aufgabenspektrum und einen der Ebene angemessenen Planungshorizont verantwortlich und wird auf entsprechende Schichten in der IT-Architektur abgebildet. [ANSI/ISA-S95 Part I, 2000; VDI 5600, 2007]

Die Ebene 4 in der Abbildung bildet die Ebene des *Unternehmensmanagements* ab, die das gesamte Unternehmen berücksichtigt. Das heißt Planungs- und Steuerungsaktivitäten in diesem Umfeld betreffen alle Standorte des Unternehmens, erstrecken sich über einen Zeitraum von mehreren Wochen, haben Antwortzeiten in der Größenordnung von Tagen und berücksichtigen die gesamte Liste der verfügbaren Bestellungen (strategische Ebene). Klassen von Softwaresystemen, die auf dieser Ebene eingesetzt werden, sind beispielsweise Advanced Planning and Scheduling (APS), Enterprise Resource Planning

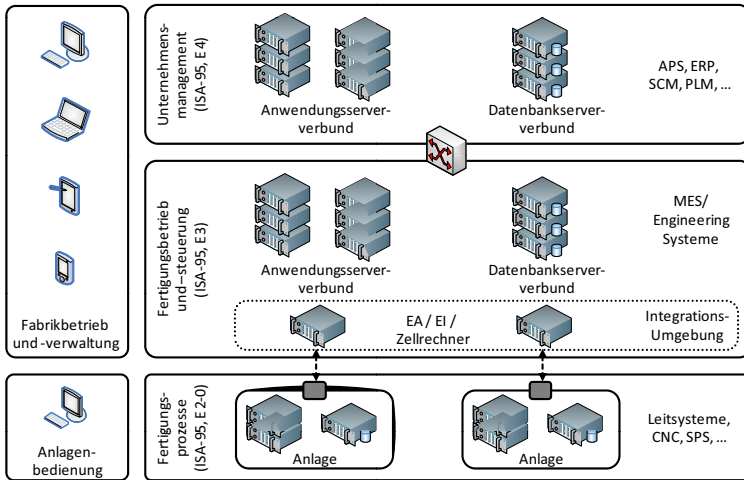


Abbildung 2.6: Das Ebenenmodell der Produktions-IT

(ERP), Product Lifecycle Management (PLM) und Supply Chain Management (SCM) Systeme.

Die Ebene 3 ist mit Aufgabenstellungen des *Fertigungsbetriebs und der Fertigungssteuerung* befasst. Instanzen dieser Schicht sind für einzelne Standorte zuständig. Planungs- und Steuerungsaktivitäten beschränken sich auf diesen Standort, erstrecken sich über den Zeitraum einer oder mehrerer Schichten, haben Antwortzeiten in der Größenordnung von einigen Sekunden bis zu einer Schicht und berücksichtigen den Auftragsvorrat des Standorts (taktische Ebene). Der folgende Abschnitt 2.3.2 beschreibt die Schicht im Detail. Aufgrund ihrer Bedeutung für diese Arbeit ist die Integrationsumgebung zwischen Ebene 2 und 3 gesondert ausgewiesen. Sie bildet das Rückgrat für die Integration der Systeme der Produktions-IT und schafft die notwendige informationstechnische Durchlässigkeit zu den Produktionsprozessen.

Die Ebenen 2-0 bilden insbesondere Prozessüberwachungs und -regelungsaufgaben ab. Hier laufen die *Fertigungsprozesse* bzw. werden einzelne Prozessschritte ausgeführt. Planungs- und Steuerungsaktivitäten auf dieser Ebene betreffen einzelne Anlagen, erstrecken sich über einen Zeitraum von Sekunden bis hin zu einigen Stunden, haben Antwortzeiten in der Größenordnung von Millisekunden bis zu wenigen Sekunden und berücksichtigen Aufträge einer Anlage (operative Ebene).

### 2.3.2 Manufacturing Execution Systems

Auf der Ebene 3 des vorgestellten Modells spielt die Klasse der Fertigungsmanagementsysteme, englisch: Manufacturing Execution System (MES), eine besondere Rolle. Auf-

grund seiner Verbreitung in der Industrie wird in dieser Arbeit ausschließlich der englische Begriff verwendet wird. Der Begriff des MES wurde 1990 geprägt [MESA-WP1, 1997] und war ursprünglich aus der Computer Integrated Manufacturing (CIM) Strömung (u. a. [Williams, 1989]) hervorgegangen. Eine kurze Beschreibung der Begriffsentwicklung findet sich beispielsweise in [VDI 5600, 2007]. Die Manufacturing Enterprise Solutions Association (MESA) [MESA, 2009] versuchte bereits Anfang der 1990er Jahre (überarbeitet 1997) auf Basis eines funktionalen MES-Modells den Umfang dieser Systemklasse genauer zu definieren [MESA-WP2, 1997]. Dennoch blieb die Bedeutung dieses Begriffs in der Praxis sehr unscharf. Seine sprunghafte Weiterverbreitung in den ersten Jahren des 21. Jahrhunderts, in denen Produkte und Lösungen mit einer großen Bandbreite unterschiedlicher Funktionalitäten unter der MES-Flagge vertrieben wurden, trug zu dieser Entwicklung bei. Vor diesem Hintergrund wurde im September 2003 innerhalb des Verein Deutscher Ingenieure (VDI) im Kompetenzfeld Informationstechnik der Fachausschuss 2.5.1 “Manufacturing Execution Systems (MES)” [VDI, 2009] gegründet, der sich zunächst einer aufgabenorientierten MES Beschreibung ausgehend vom MESA-Modell widmete.

Basierend auf diesen Arbeiten und den Erfahrungen der Abteilung Reinst- und Mikroproduktion des Fraunhofer Institut für Produktionstechnik und Automatisierung (IPA) aus der industriellen Praxis in der PV-Industrie und der Halbleiterindustrie wird die Klasse der MES wie folgt definiert: MES stellen das Bindeglied zwischen Fertigungsprozessen mit ihren Steuerungs- und Regelungssystemen und den Unternehmensmanagementsystemen dar. Sie dienen der Steuerung, Regelung und Überwachung der Abläufe in der Produktion in Echtzeit und helfen bei der Identifikation und Nutzung von Optimierungspotentialen in der Produktion. Sie schaffen das notwendige Maß an Transparenz zwischen den Ebenen des Unternehmensmanagements und der Produktion, das für die Implementierung der beschriebenen Ansätze zur Prozessintegration notwendig ist.

Die VDI-Richtlinie 5600 Blatt 1 [VDI 5600, 2007] nennt die folgenden generischen MES Aufgabenbereiche:

- Feinplanung und Feinsteuerung
- Betriebsmittelmanagement
- Materialmanagement
- Personalmanagement
- Datenerfassung
- Leistungsanalyse
- Qualitätsmanagement
- Informationsmanagement

Im Einklang mit o. g. Definition werden für den weiteren Verlauf der Arbeit ausdrücklich Systeme mit berücksichtigt, die einen ausgeprägten aktiven Charakter haben, also direkt steuernd und regelnd in das Geschehen in der Produktion eingreifen.



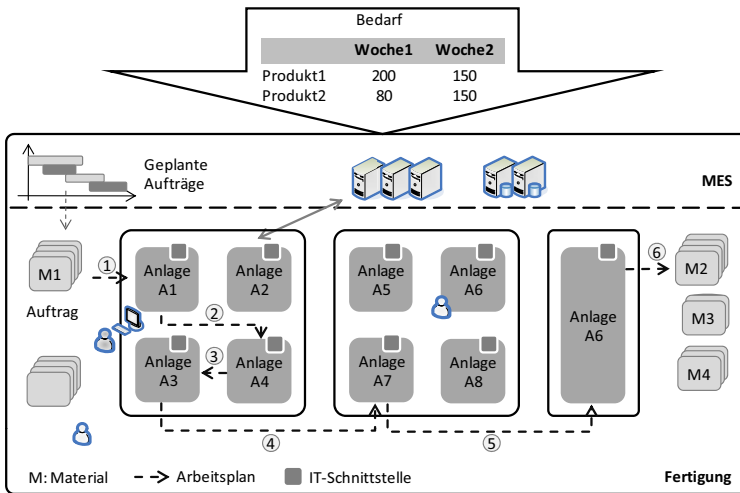


Abbildung 2.7: Beispielszenario einer MES-gesteuerten Fertigung

### Ein Beispiel-Szenario

Definition und Aufgabenstellung des MES sollen anhand des in Abbildung 2.7 visualisierten vereinfachten Betriebsszenarios nochmals verdeutlicht werden:

Ausgehend von Bedarfsdaten von der Unternehmensebene werden (Fertigungs-) *Aufträge* erzeugt und dem MES zur Feinplanung übergeben. Den Aufträgen liegt ein *Arbeitsplan* zugrunde, der die auszuführenden Arbeitsschritte auf Basis verfügbarer Prozesse beschreibt. Zur Feinplanung kommen die Ansätze der Planung (engl. *Scheduling*) oder der Disposition (engl. *Dispatching*) zum Einsatz. Beim Scheduling-Ansatz wird ein Plan für die optimierte Zuordnung einer Menge von Aufgaben auf eine Menge von Ressourcen (z. B. Anlagen) über einen gewählten Planungshorizont (engl. *scheduling horizon*) berechnet. Der Dispatching-Ansatz zielt auf die Optimierung ausgewählter Ressourcen in Echtzeit ab, indem Regelsätze auf eine Menge von Kontextinformationen angewendet werden. Beide arbeiten auf Basis von Zustandsinformationen, die in Echtzeit über die *IT-Schnittstellen* aus der Produktion bezogen werden. Entsprechend der Planungsergebnisse werden die Fertigungsaufträge vom MES durch die Produktion getrieben. Das MES stellt sicher, dass in jedem Prozessschritt das richtige *Material* dem richtigen Prozess zugeführt wird. Es konfiguriert die Anlagen für den anstehenden Prozessschritt und gibt Material nach seiner Verifizierung zur Bearbeitung frei. Alle produktbezogenen Abläufe und Daten, die für die Rückverfolgbarkeit relevant sind, werden dokumentiert. Gleichzeitig werden Qualitätskennzahlen überwacht. Darüber hinaus dokumentiert das MES den Einsatz der Ressourcen, überwacht die Prozessfähigkeit und den Wartungsbedarf und ermöglicht den Zugriff auf relevante Prozessdaten.

### Die Systemtechnik

Während die Großväter heutiger MES aus der CIM-Zeit eher auf datenbankzentrierte Großrechnerarchitekturen (engl. mainframe) setzten, kommen zwischenzeitlich die weit verbreiteten Client-Server Architekturen und in der neueren Zeit Service Orientierte Architektur (SOA) basierte Ansätze in der Produktions-IT zum Einsatz. Die Mensch Maschine Schnittstelle wird beispielsweise über Web- oder Rich-Clients realisiert. Die Geschäftslogik, die die oben beschriebenen Aufgaben implementiert, wird im Hintergrund von Serverdiensten zur Verfügung gestellt, die in Produktionsrechenzentren auf Servern oder Serververbänden (engl. cluster) betrieben werden und für die Datenhaltung auf Datenbanksysteme und Dateiserver zugreifen. Die Kommunikation zwischen den Systemen erfolgt, ähnlich wie auf der Unternehmensebene, über etablierte Standardprotokolle. [z. B. Meier, 2006; Sauer, 2007a]

Innerhalb der Integrationsumgebung auf Ebene 3 in Abbildung 2.6 spielen die Aufgaben der Anlagenintegration (engl. Equipment Integration (EI)) bzw. die Anlagenautomatisierung (engl. Equipment Automation (EA)) eine wichtige Rolle. In erster Linie stellt die Anlagenintegration eine einheitliche Sicht auf alle zu überwachenden und zu steuernden Anlagen her und bindet diese in das Fabrikkommunikationssystem, heute häufig Ethernet-basierte, lokale Netzwerke, ein. Darüber hinaus kann die Anlagenintegration auch Aufgaben wie die Umrechnung von Einheiten oder die Verifikation von Daten übernehmen. Die Anlagenautomatisierung befasst sich mit der Implementierung von anlagen(typ)spezifischen Betriebsszenarien. Sie bietet also die notwendige IT-Unterstützung, um Prozessschritte an einer gegebenen Anlage in definierter Art und Weise auszuführen. Zum Betriebsszenario gehören beispielsweise die instanz- oder typspezifische Materiallogistik, Materialverifikation und Fernsteuerungskommandos. [z. B. Dreiss, 2007; SEMI DRAFT 4557, 2008]

Terminals mit grafischen Benutzerschnittstellen auf der Ebene der Anlagenautomatisierung werden einzelnen Anlagen oder Gruppen von Anlagen zugewiesen, sofern Betriebsszenarien die Interaktion mit dem Benutzer oder eine Überwachung durch den Benutzer erfordern. Die EA/EI-Komponenten für einzelne Anlagen oder Gruppen von Anlagen können auf dedizierten PCs oder IPCs, auch Zellrechner genannt, oder auf Server Systemen im Rechenzentrum der Fertigung betrieben werden. Für weiter verbreitete Anlagenschnittstellen stehen kommerzielle Softwarepakete zur Realisierung der Anlagenintegration zur Verfügung. Die notwendigen Anpassungen für den konkreten Fall oder die Entwicklung von Speziallösungen zur Anbindung werden durch spezialisierte Unternehmen (*Integratoren*) oder durch die Betreiber der Fertigung umgesetzt. [Chung und Jeng, 2004; Dreiss, 2007]

Für die konkrete Gestaltung der Produktions-IT, einschließlich des MES, gibt es vielfältige Ansätze. Üblicher Weise werden die oben beschriebenen Aufgaben nicht durch ein einzelnes System implementiert, sondern durch eine Menge kooperierender Systeme [VDI 5600, 2007]. Vor diesem Hintergrund wird im Folgenden die Schicht 3 auch als *MES-Ebene* bezeichnet. Eine Vielzahl von Herstellern bietet, häufig branchenspezifische, Produkte im MES-Umfeld an, die eine Untermenge der MES-Aufgaben abdecken. Zur Implementierung einer MES-Lösung für eine ausgewählte Produktion müssen also zunächst die

spezifischen Anforderungen ermittelt werden. Auf dieser Basis sind in der Folge geeignete Produkte und Rahmenwerke (engl. *frameworks*) auszuwählen, die anschließend an die spezifischen Anforderungen angepasst und mit den benachbarten IT-Systemen integriert werden müssen. Dieser Prozess wird auch *Projektierung* (engl. *engineering*) genannt. Die Projektierung kann durch spezialisierte Systemhäuser oder den Betreiber der Fertigung erfolgen. Abhängig von ihrem Funktionsumfang und der Zahl der integrierten Systeme variiert die Komplexität der Produktions-IT-Landschaft beträchtlich. [Chung und Jeng, 2004; Meier, 2006; Kletti, 2007; Sauer und Ebel, 2007; Meier, 2010]

### 2.3.3 Die Ebene der Fertigungsprozesse

Auf der Ebene der Fertigungsprozesse sind Anlagen angesiedelt, die jeweils einen oder mehrere Produktionsprozesse ausführen können. Sie werden von internen Steuerungs- und Regelungssystemen überwacht und gesteuert, die auf unterschiedlichen Systemkonzepten basieren können. Die internen IT-Systeme dienen der autonomen Steuerung und Regelung der Prozesse innerhalb der Anlage. Die Mensch Maschine Schnittstelle (MMS) (engl. Human Machine Interface (HMI)) der Anlage bietet Möglichkeiten zur Interaktion mit dem Anwender in der Produktion und zur Visualisierung der Prozesse innerhalb der Anlage. Über IT-Anlagenschnittstellen stellt die Anlage externen IT-Systemen ausgewählte Dienste zur Verfügung.

Als Systemkonzepte kommen in der PV-Industrie *SPS-basierte* Systemkonzepte und *PC-basierte Fertigungsleitsysteme* zum Einsatz. Erstere werden für weniger komplexe Automatisierungsprobleme genutzt. Insbesondere die große Erfahrung von Automatisierern mit der Entwicklung SPS-basierter Lösungen, die Robustheit, die solche Systeme erfahrungsgemäß auszeichnen, ihre harte Echtzeitfähigkeit, die auch die Regelung sehr schneller Prozesse erlaubt und die vergleichsweise günstigen Kosten sind wesentliche Gründe, die zum Einsatz SPS-basierter Automatisierungslösungen führen [PV-EIS, 2007; Reddig, 2009]. Letzere werden in der PV-Industrie für komplexe Automatisierungsaufgaben bzw. Automatisierungsaufgaben, die ein hohes Maß an Flexibilität erfordern, eingesetzt. Die Steuerung läuft in diesem Fall auf einer PC-basierten Plattform. Komponenten des Automatisierungssystems, die wiederum eigene Steuerungs- und Regelungssysteme enthalten, wie beispielsweise Messsysteme, Handhabungssysteme oder Prozessmodule, sind über ein anlageninternes Kommunikationsnetz und entsprechende Schnittstellen mit dem Leitsystem verbunden, das die Ablaufsteuerung übernimmt, die Interaktion mit dem Benutzer und externen IT-Systemen abwickelt, das Gesamtsystem überwacht und auftretende Ausnahmesituationen in geeigneter Art und Weise behandelt [Muckenhirn, 2005; PV-EIS, 2007; Brecher u. a., 2008; Reddig, 2009].

## 2.4 Standardisierung in der Produktions-IT

Die vorausgehenden Abschnitte machen deutlich, dass eine durchgängige Produktions-IT-Landschaft ein hohes Maß an Integration zwischen zahlreichen IT-Systemen erfordert, von der Unternehmensebene angefangen bis hin zur Ebene der Fertigungsprozesse. Auf

breiter Basis akzeptierte und umgesetzte Industriestandards haben sich daher als wichtige Voraussetzung für die Implementierung der Produktions-IT-Landschaft erwiesen. Standards dienen dabei den folgenden Zielen:

- Sie beinhalten abgestimmte Begriffs- und Konzeptdefinitionen und erleichtern damit das gemeinsame Verständnis zwischen Kunden und Lieferanten, was insbesondere für die Projektierungsphase von großer Bedeutung ist.
- Sie definieren Umfang und Verantwortlichkeiten von Systemen.
- Sie schaffen vereinheitlichte Schnittstellendefinitionen und ermöglichen bzw. vereinfachen dadurch die Integration unterschiedlicher Systeme – auch für den Fall, dass sie von unterschiedlichen Herstellern stammen.

Aktivitäten zur Standardisierung finden auf allen Ebenen der Produktions-IT, zumeist am aktuellen Bedarf orientiert, statt. Diese Aktivitäten werden auf unterschiedlichen Plattformen gebündelt:

- Plattformen für spezifische Industrien, wie Semiconductor Equipment and Materials International (SEMI) [SEMI, 2009], die die Infrastruktur für die Entwicklung von Standards für die Produktion in der PV-Industrie, Flachbildschirmfertigung und der Halbleiterindustrie bietet
- Plattformen, die sich industrieübergreifend Produktions-IT Themen widmen, wie MESA [MESA, 2009] oder VDI [VDI, 2009] mit ihren MES-Modellen oder die OPC Foundation [OPC, 2009], die Standards für die vertikale Integration in der Produktion erarbeitet
- Plattformen, die Standards für generische Basistechnologien erarbeiten, wie die Internet Society (ISOC) [ISOC, 2009], die Standards für Internet-Technologien erstellt

### 2.4.1 Standards zu Produktions-IT-Landschaft und MES-Ebene

In den letzten Jahren wurden von verschiedenen Organisationen Anläufe unternommen, um umfassende Standardrahmenwerke oder Referenzmodelle zu erstellen, die eine durchgängige, vollständig integrierte Produktions-IT-Landschaft zum Ziel hatten. Ein Anlauf, der die Diskussionen über mehrere Jahre, insbesondere in den frühen 1990ern, geprägt hat, war das “Reference Model For Computer Integrated Manufacturing (CIM)” [Williams, 1989]. Das Modell beschreibt eine 6-Schichten-Architektur, die auf der oben erwähnten Planungs- und Steuerungshierarchie basiert und als Leitfaden für die Implementierung einer vertikal vollständig integrierten Produktions-IT-Landschaft konzipiert war. Obwohl dieses Konzept nach Kenntnis des Autors in der Praxis nie in signifikanter Verbreitung vollständig umgesetzt wurde, bildete es die Grundlage für zahlreiche Nachfolgearbeiten. Im Jahr 1997 präsentierte die MESA eine aktualisierte Version ihrer MES-Definition, inklusive der Beschreibung eines maximalen Funktionsumfangs bestehend aus elf funktionalen Gruppen, die nach wie vor Zustimmung findet [MESA-WP2, 1997]. Darin wird der Funktionsumfang einer spezifischen MES-Instanz als Untermenge

des maximalen Funktionsaufwands beschrieben, wobei die Untermenge basierend auf den Prioritäten und Anforderungen des Anwenders bzw. Betreibers ermittelt wird.

Im Jahr 2000 wurde [ANSI/ISA-S95 Part I, 2000] veröffentlicht. Der Standard basiert auf CIM-Konzepten, insbesondere der Planungs- und Steuerungshierarchie, und spezifiziert in erster Linie die Schnittstelle zwischen den Ebenen des Unternehmensmanagement und der Fertigungssteuerung bzw. des Fertigungsbetriebs. Darüber hinaus greift ISA-95 Teil I die durch die MESA definierten funktionalen Gruppen auf, um die Zuständigkeiten der MES-Schicht zu beschreiben. [ANSI/ISA-S95 Part II, 2001] und [ANSI/ISA-S95 Part III, 2005], die in der Folge publiziert wurden, ergänzen Teil I um die Spezifikation des Datenmodells und des Prozessmodells (Spezifikation der Aktivitäten und Datenflüsse) für diese Schicht.

Parallel zu den Arbeiten an ISA-95 wurde von SEMI eine Initiative zur Standardisierung eines Rahmenwerks für die Produktions-IT in der Halbleiterindustrie (CIM-Framework) gestartet. Der Standard sollte Halbleiterfertiger in die Lage versetzen, Produkte unterschiedlicher Hersteller auf der MES-Ebene per “plug-and-play” zu integrieren. Dieses Ziel sollte durch die Spezifikation einer Domänenarchitektur erreicht werden, basierend auf einer standardisierten funktionalen Beschreibung der Komponenten eines MES, vereinheitlichten Definitionen von Kernbegriffen und deren datentechnischen Repräsentation, wie “Fertigungsauftrag” oder “Material”, und formalen Schnittstellendefinitionen der Komponenten. Das Ergebnis dieser Initiative kam nie über den Status eines so genannten “Provisional Standard” hinaus, eines Standards, der nach SEMI Definition noch Lücken aufweist, und wurde in der Praxis nie in der ursprünglich geplanten Konsequenz angewandt. Für die Definition der Produktions-IT-Landschaft bzw. bei der Analyse und der Spezifikation von MES-Lösungen in der Halbleiterindustrie und in Industrien, die vergleichbare Konzepte nutzen, leistet der Standard [SEMI E81, 2000], der im Rahmen dieser Initiative entstand, jedoch nach wie vor gute Dienste. Er wurde bei seiner turnusgemäßen Überprüfung im Jahr 2006 erneut verlängert [Meier u. a., 2007b].

Die oben genannte Initiative des VDI, die im Jahr 2003 mit dem Ziel der Schärfung des MES-Begriffes ins Leben gerufen worden war, schließt sich an die vorausgegangenen Arbeiten der MESA an und verfolgt ebenfalls die Ziele, MES-Anwender im Verständnis der Konzepte und Potentiale der MES-Schicht zu unterstützen, Angebote unterschiedlicher Hersteller vergleichbarer zu machen und Hilfestellung bei der Analyse und der Formulierung der Anforderungen zu geben [VDI, 2009].

### 2.4.2 Standards zur Anlagenschnittstelle

Während die im vorherigen Abschnitts beschriebenen Standards in der Praxis eher ein einheitlicheres Bild der Inhalte und des Umfangs der Produktions-IT fördern, stellen die Spezifikationen für Anlagenschnittstellen Verträge dar, die diese Schnittstellen im Detail definieren. Die konsequente Umsetzung dieser Verträge durch die zu integrierenden Systeme ist eine wichtige Voraussetzung, um die große Zahl von Anlagen unterschiedlichster Hersteller, wie sie in realen Produktionslinien auftritt, effizient und effektiv in die Produktions-IT einzubinden [z. B. Dreiss, 2007; Meier u. a., 2007b; Baylies u. a., 2008].

Ein Standardrahmenwerk, das mit diesem Ziel in zahlreichen Branchen zum Einsatz kommt, umfasst die OLE for Process Control (OPC) Standards der OPC Foundation, deren erste Spezifikation im September 1996 in einer ersten Version freigegeben wurde. Neben generischen Teilen, die sich branchenübergreifend einsetzen lassen, existieren branchenspezifische Teile, wie OPC Batch, die spezifische Anforderungen einer Industrie aufgreifen. Während die erste Generation auf Microsofts Component Object Model (COM) Technologien basiert, setzt die neue Generation der OPC Standards, genannt OPC Unified Architecture (OPC-UA), auf Webtechnologien. Die ersten Teile der OPC-UA Spezifikation wurden im Jahr 2006 veröffentlicht. [OPC, 2009]

### Die standardisierte Anlagenschnittstelle der PV-Industrie

Die SEMI-Standards für die IT-Anlagenschnittstelle, die insbesondere in den Bereichen der Halbleiterindustrie, Flachbildschirmindustrie und neuerdings der PV-Industrie eingesetzt werden, erstrecken sich über alle Schichten des Open Systems Interconnection (OSI) Referenzmodells [ISO/IEC 7498-1, 1994] und spezifizieren neben den Basismechanismen zur Kommunikation und der Syntax der Anlagenschnittstelle sehr umfassend deren Semantik [Meier u. a., 2007b]. Im September 2007 wurde innerhalb von SEMI die Photovoltaik Equipment Interface Specification (PV-EIS) Taskforce gegründet – eine Arbeitsgruppe, die sich mit der Entwicklung eines geeigneten Standardframeworks für die IT-Anlagenschnittstellen in der PV-Industrie befassen sollte. Nach einer umfangreichen Analyse- und Diskussionsphase wurde die Entscheidung getroffen, zunächst eine Untermenge der Standards aus der Halbleiterindustrie zu übernehmen und an einigen Stellen für die PV-Industrie anzupassen [SEMI DRAFT 4557, 2008]. Das Ergebnis dieser Arbeit wurde im Juli 2009 als Standard [SEMI PV2, 2009] mit dem Titel Guide for PV Equipment Communication Interfaces (PVECI) publiziert.

Die IT-Anlagenschnittstelle ist durch diesen Standard wie folgt definiert, wobei an den referenzierten SEMI-Standards einige Anpassungen und Vereinfachungen für die PV-Industrie vorgenommen wurden (siehe Abbildung 2.8):

- Ein Ethernet-Port realisiert die Schicht 1/2 Anbindung.
- SEMI-E37 [SEMI E37, 2008] spezifiziert, wie der Transmission Control Protocol (TCP)/Internet Protocol (IP) Protokollstapel zu konfigurieren ist. Darüber hinaus definiert SEMI-E37 einen Dienst für den Transport standardisierter SEMI Equipment Communications (SECS) Nachrichten über diesen Protokollstapel.
- SEMI-E5 [SEMI E5, 2007] definiert Form und Inhalt der Nachrichten, die über die Verbindung mit der Anlage ausgetauscht werden können.
- SEMI-E30 [SEMI E30, 2007] beschreibt mit Generic Model for Communication and Control of Manufacturing Equipment (GEM) schließlich die Semantik der so geschaffenen Schnittstelle, also das Verhalten der Anlage aus der Sicht der Schnittstelle für wichtige Automatisierungsszenarien.
- Network Time Protocol (NTP) wird zur Zeit-Synchronisation der Anlage mit der Fabrik vorgesehen.

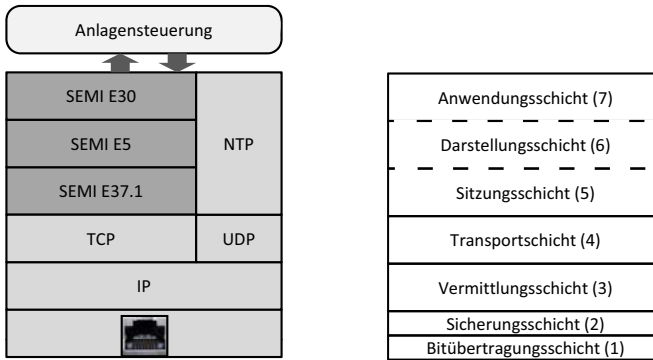


Abbildung 2.8: Der PVECI-Protokollstapel im Vergleich zum ISO/OSI Referenzmodell

In der Halbleiterindustrie wurden diese fundamentalen Standards durch eine Reihe weiterer Standards ergänzt. Einige werden im Folgenden stellvertretend genannt. Neben Standards für die Verwaltung von Prozessprogrammen bzw. Rezepten wurden anlagentypspezifische Standards (engl. Specific Equipment Models (SEM)) erstellt, die auf SEMI E30 aufsetzen und spezielle Anforderungen an die Schnittstelle für eine Klasse von Anlagen abbilden. Beispielsweise wurde ein SEM für automatische Transportsysteme erstellt. Die Familie der GEM 300 Standards, die auf SEMI-E5 aufsetzt, wurde ergänzt, um dem steigenden Automatisierungsgrad beim Übergang der Fertigung von Scheiben mit einem Durchmesser von 200mm auf 300mm Rechnung zu tragen. Sie fügen der Anlagenschnittstelle umfangreiche Funktionalitäten hinzu, beispielsweise im Hinblick auf die Verwaltung von Materialträgern oder die Einzelscheibenverfolgung. Die gestiegenen Anforderungen an die Datenerfassung von Anlagen (engl. Equipment Data Acquisition (EDA)) haben darüber hinaus zur Entwicklung der neuen Familie der EDA Standards geführt, die, ähnlich wie OPC-UA, auf aktuellen Web-Technologien basieren und parallel zum GEM-Stack betrieben werden. [Meier u. a., 2007b]

Ähnliche Erweiterungen von GEM, allerdings in geringerem Umfang, sind in den kommenden Jahren auch für die PV-Industrie zu erwarten. Aktuell wird von der PV-EIS Arbeitsgruppe die Erweiterung des ersten Standards um Funktionalitäten für die Verfolgung einzelner Substrate bearbeitet. Teil dieser Arbeit ist die Definition eines zusätzlichen Protokollstapels zur "horizontalen Kommunikation", der den physikalischen Austausch von Substraten zwischen Anlagen informationstechnisch abbildet [PV-EIS, 2009b]. Die EDA Standards spielen hingegen in der PV-Industrie noch keine Rolle.

## 2.5 Fazit Industriebedarf

Die PV-Industrie steht, wie in Kapitel 1 diskutiert und begründet, vor der Herausforderung, die Produktion entlang der gesamten Wertschöpfungskette optimieren zu müssen,

um mit den Herausforderungen des Marktes Schritt halten zu können. Einen Ansatzpunkt stellt aus Sicht der Industrie die verstärkte Unterstützung der Produktionssysteme durch MES-Lösungen dar.

Erfahrungen aus anderen Branchen zeigen, dass solche Erwartungen durchaus berechtigt sind, dass der verstärkte Einsatz von Software und IT-Systemen allerdings auch seinen Preis hat. [Zäh und Wünsch, 2005] fassen die Ergebnisse einiger Studien aus dem Umfeld von Werkzeugmaschinen zusammen: Danach sei die Steuerungstechnik zwischenzeitlich für mehr als 50% der Funktionalität von Produktionsmaschinen und -anlagen verantwortlich. Trotz ihres hohen Stellenwerts präge nach wie vor die Mechanikentwicklung die Produktgestaltung. Von der Inbetriebnahmephase, die einen Anteil von ca. 15% bis 25% an der Gesamtlauzeit eines Anlagenprojekts habe, würden 90% von Elektronik und Steuerungstechnik beansprucht. 70% dieses Zeitbedarfs seien auf Fehler in der Steuerungstechnik zurückzuführen. Ausgehend von diesen Zahlen betrage der Aufwand für die Behebung von Fehlern in der Steuerungssoftware 60% der Inbetriebnahmephase bzw. 16% der Gesamtprojektdauer.

Die geschilderte Problematik wird durch andere Untersuchungen gestützt. [Ammermann und Lohse, 2007] diskutieren die Ergebnisse einer Studie, wonach für den Fall flexibel verbundener Fertigungszentren ein signifikanter Anteil der in der Anlaufphase auftretenden Fehler der Steuerung bzw. der Software zuzurechnen seien. Dieser Fehlerkategorie sei eine besondere Bedeutung zuzumessen, weil ein vergleichsweise hoher Anteil dieser Fehler zu besonders langwierigen Verzögerungen (in der diskutierten Untersuchung von bis zu 3 Tagen) führe.

Vergleichbare repräsentative Untersuchungen aus der PV-Industrie, speziell für die Ebene der Fabrikautomatisierung, stehen nach dem Kenntnisstand des Autors zum heutigen Zeitpunkt nicht zur Verfügung. Allerdings existieren verschiedene Indikatoren, die zumindest qualitativ eine Transferierbarkeit der geschilderten Beobachtungen in die PV-Industrie nahelegen: Das Interesse der PV-Industrie an Automatisierungsstandards konnte erst spät geweckt werden. Erst durch einen steigenden Leidensdruck aus dem Projektalltag heraus wurde deutlich, dass steuerungs- und softwaretechnische Fragestellungen einen höheren Stellenwert haben als vorher angenommen [Meier, 2009]. Bei der folgenden Ermittlung der Anforderungen der PV-Industrie an die standardisierte Anlagenschnittstelle wurde die Möglichkeit zum Offline-Betrieb der Anlagen, also zum Betrieb der Anlagen während die MES-Ebene nicht verfügbar ist, sehr hoch priorisiert. Diese Anforderungen lassen auf verschiedene Probleme mit der MES-Ebene schließen die zu dem Lösungsversuch führten, die Anforderungen an die Verfügbarkeit dieser Systeme möglichst gering zu halten [Meier, 2008a]. [Jindal und Khandelwal, 2005] bestätigen die zentrale Bedeutung der nicht-funktionalen Systemeigenschaften der Zuverlässigkeit, Verfügbarkeit und Skalierbarkeit für MES-Lösungen in der Halbleiterindustrie, in der die MES-Schicht im Produktionssystem eine unverzichtbare Rolle spielt.

Damit zukünftig MES-Lösungen in der PV-Industrie tatsächlich einen positiven Beitrag zur Optimierung der Wertschöpfungskette leisten können, sind also nicht nur Softwarehersteller gefragt, deren Produkte im Hinblick auf funktionale und nicht-funktionale Anforderungen das Vertrauen der PV-Industrie rechtfertigen, sondern auch ein integraler Engineeringprozess, der einen optimalen Einsatz dieser Systeme über den Fabriklebenszyklus ermöglicht.



# 3 Analyse der Problemstellung und Konzeption des Verfahrens

Die folgenden Abschnitte widmen sich der detaillierten Analyse der Problemstellung des MES-Engineering in dem in Kapitel 2 beschriebenen Kontext und führen zum Vorschlag des Verfahrens zum emulationsgestützten MES-Engineering.

## 3.1 Die Produktions-IT in der PV-Industrie

Aus der in Kapitel 1.1 beschriebenen Problem- bzw. Zielstellung und unter Berücksichtigung der in Kapitel 2 diskutierten Randbedingungen ließen sich vielfältige generische Anforderungen an die Produktionsnahe Informationstechnik (Produktions-IT) der PV-Industrie ableiten. Eine detaillierte, repräsentative Studie über die tatsächlichen Anforderungen der Fertiger an die MES-Ebene, die von ihnen genutzten Lösungen und Architekturen und die Funktionalitäten am Markt erhältlicher Systeme lässt sich allerdings kaum durchführen, da dieses Wissen sowohl seitens der Fertiger als auch seitens der Softwareanbieter als schützenswertes Know-how betrachtet und nicht öffentlich verfügbar gemacht wird. Typischer Weise werden aus diesem Grund Produktions-IT Konzepte von Fertigern und Lösungsangebote von Anbietern nur im Rahmen von Geheimhaltungsvereinbarungen diskutiert. Einen gewissen Einblick bieten dennoch die öffentlichen Aktivitäten zur Standardisierung (siehe Kapitel 2.4.2). Ergebnisse und Teilergebnisse aus diesen Arbeiten werden im Folgenden zur Argumentation genutzt. Zunächst wird aus [SEMI PV2, 2009] deutlich, dass von einem prinzipiellen Aufbau der Produktions-IT-Landschaft wie er in Kapitel 2.3 skizziert wurde auszugehen ist.

### 3.1.1 Die MES-Anforderungen der PV-Industrie

Darüber hinaus geben Ergebnisse der SEMI PV-EIS Taskforce (siehe Kapitel 2.4) einen Einblick in reale MES-Anforderungen der PV-Industrie. Eine Sammlung von MES-Funktionalitäten, die unmittelbar durch die Anlagenschnittstelle unterstützt werden müssen, wurde im Zuge der Anforderungsanalyse durch die Taskforce erhoben. Abbildung 3.1 fasst diese MES-Anforderungen zusammen und ergänzt sie um die in den Abschnitten 1.1 bzw. 2.5 genannten zukünftigen Anforderungen, die durch den Zusatz "zukünftig" gekennzeichnet sind. Für die Darstellung wurden die Anforderungen entsprechend der in [VDI 5600, 2007] definierten MES-Aufgaben gruppiert.

MES-Aufgabe (VDI 5600)	Anforderungen der PV-Industrie
Feinplanung und -steuerung	<ul style="list-style-type: none"> <li>• Unterstützung von Dispatching (zukünftig), z. B.                             <ul style="list-style-type: none"> <li>– Konfigurierbare Regeln</li> </ul> </li> </ul>
Betriebsmittelmanagement	<ul style="list-style-type: none"> <li>• Anlagenmodell in ausreichender Detaillierung</li> <li>• Überwachung von Anlagen                             <ul style="list-style-type: none"> <li>– Zustandsverfolgung und Historisierung (z.B. Betriebszustand, Materialzuweisung, Rüstzustand)</li> <li>– Fehler, Alarmer, Warnungen, Nachrichten</li> <li>– Verbrauch von Hilfsstoffen / Energie</li> <li>– Unterstützung OEE Ermittlung (SEMI E10)</li> </ul> </li> <li>• Wartungs- und Instandhaltungsmanagement (Zeitbasiert / Zählerbasiert)</li> <li>• Rezeptverwaltung (Auswahl, Upload, Download, Parameter)</li> <li>• Carrier-Identifikation</li> </ul>
Materialmanagement	<ul style="list-style-type: none"> <li>• Substratverfolgung und Identifikation (mit realen und virtuellen IDs)</li> <li>• Verfolgung des Materialzustands, insbesondere detaillierte Verfolgung von Verlusten</li> </ul>
Datenerfassung	<ul style="list-style-type: none"> <li>• Konfigurierbare automatische Datenerfassung (von Anlagen), Korrelierbarkeit sicherstellen (über Material-ID und/oder Zeitstempel)                             <ul style="list-style-type: none"> <li>– Qualitätsdaten / Messdaten</li> <li>– Prozessdaten (z.B. Prozesskenngrößen, verwendete Rezepte / Parameter)</li> <li>– Zustandsänderungen / Ereignisse</li> </ul> </li> </ul>
Leistungsanalyse	<ul style="list-style-type: none"> <li>• Analysen auf Basis der ermittelten Daten, z. B.                             <ul style="list-style-type: none"> <li>– OEE Auswertung</li> </ul> </li> </ul>
Qualitätsmanagement	<ul style="list-style-type: none"> <li>• Unterstützung von SPC (und zukünftig FDC)</li> </ul>
Informationsmanagement	<ul style="list-style-type: none"> <li>• Aktive Steuerung von Anlagen, z. B.                             <ul style="list-style-type: none"> <li>– Anpassung von Anlagen-Konstanten / Rezepten / Rezept-Parametern</li> <li>– Freigabe von Material zur Prozessierung</li> <li>– Eingriffsmöglichkeiten für APC (zukünftig)</li> </ul> </li> </ul>

Abbildung 3.1: Grundlegende funktionale Anforderungen aus der PV-Industrie an MES-Systeme auf Basis der Analyse der PV-EIS Taskforce [PV-EIS, 2007], Gruppirt nach den MES-Aufgaben aus [VDI 5600, 2007]

Die Bandbreite der Ziele, die MES-Anwender an die im konkreten Fall von ihnen eingesetzten Systeme stellen, ist allerdings, bedingt durch die unterschiedlichen Produktionssysteme, sehr groß. Sie reicht von Systemen zur Ermittlung einiger wichtiger Kennzahlen, über Systeme, die eine vollständige Nachverfolgbarkeit der hergestellten Produkte durch die Fertigung sicherstellen oder detaillierte Analysen der Prozessqualität und möglicher Fehlerursachen erlauben, bis hin zu Systemen, die sehr umfassend steuernd und regelnd in die Produktion eingreifen (z.B. durch die automatische Zuweisung von Fertigungsaufträgen auf Anlagen und eine entsprechende Steuerung automatischer Transportsysteme oder Eingriffe aufgrund vorliegender Mess- und Analyseergebnisse) und jeglichen Kombinationen solcher Systeme.

#### 3.1.2 MES – Make or Buy?

Ähnlich wie in anderen Branchen stellt sich für einen Anwender im Hinblick auf die Umsetzung der beschriebenen Anforderungen die Frage, ob eine oder eine Kombination mehrerer Softwarelösungen am Markt gekauft und eingeführt oder ob die Anforderungen in einer Eigenentwicklung umgesetzt werden sollen. Anwender, die lediglich eine oder mehrere Fertigungsstätten betreiben, greifen aus vielen Gründen eher zum Einsatz von Standardsoftware. Für einen Turnkey-Anbieter liegt der Gedanke an eine für sein Produkt optimierte, lizenzfreie Produktions-IT Lösung etwas näher. In jedem Fall existiert am Markt eine Vielzahl von Anbietern und Produkten. Diese decken, wie in Abschnitt 2.3 beschrieben, zumeist jeweils eine Teilmenge der denkbaren Funktionalitäten ab und eignen sich für ein konkretes Produktionssystem mehr oder weniger gut. [Wiendahl u. a., 2007] bieten einen branchenunabhängigen Überblick über das Angebot im deutschsprachigen Raum. Abbildung 3.2 zeigt eine Liste von MES-Anbietern, die von der PV-EIS Taskforce aufgrund bereits vorhandener Projekte oder bestehender Verbindungen zur PV-Industrie im Hinblick auf ihre Möglichkeiten zur Unterstützung von Single-Substrate-Tracking befragt wurden [PV-EIS, 2009a].

Vor diesem Hintergrund wird im Folgenden davon ausgegangen, dass in der PV-Industrie in der Mehrzahl der Fälle die MES-Lösung für eine ausgewählte Fertigung auf Basis eines oder mehrerer kommerzieller Produkte entsteht.

## 3.2 Wandlungsfähige MES Systemmodelle

Aus dem Produktionssystem, das den Kontext für den Einsatz einer MES-Lösung darstellt, ergeben sich die konkreten Anforderungen an ein solches Softwaresystem bzw. deren Gewichtung (siehe Kapitel 2.3). Aufgrund der Vielfältigkeit der Produktionssysteme, die schon innerhalb der PV-Industrie zu beobachten ist (siehe Kapitel 2.2), spielt der Aspekt der Anpassungsfähigkeit für MES Software eine wichtige Rolle. MES Systemkonzepte, die ein hohes Maß an Flexibilität bieten, werden am Beispiel des Synapse MES erläutert [u. a. Meier, 2006; Muckenhirn und Meier, 2008; Meier, 2010]. Dieses MES wurde am Fraunhofer IPA in mehreren Forschungsvorhaben entwickelt und zwischenzeitlich an einen Industriepartner transferiert, der die Produktentwicklung übernommen hat

Integrator / Anbieter	Produktbezeichnung
acp-IT	InFrame Syn@pse MES
AIS Automation	VPC – VAC Production Control
Applied Materials	FAB300, FACTORYworks
Camline	LineWorks
CamStar	Camstar Manufacturing
HCL Technologies / Eyelit	Eyelit Manufacturing
InSystems/ Prediktor	APIS Click&Trace
Mpdv Mikrolab	HYDRA
Systema/ Eyelit	Eyelit Manufacturing

Abbildung 3.2: Anbieter von MES-Systemen für die PV-Industrie (Auszug) basierend auf einer Anbieterbefragung der PV-EIS [PV-EIS, 2009a]

[Wissen.Schafft.Arbeit, 2008]. Das System ist am Markt verfügbar und erfolgreich in mehreren Fabriken, u. a. in der PV-Industrie, eingeführt.

### 3.2.1 Mehrschichtige Metamodelle

Abbildung 3.3 links zeigt den Ausschnitt des Architekturkonzepts, der weitgehende Anpassungen ermöglicht. Es besteht zumindest aus den Schichten *Systemkern*, *Domäne* und *Instanz*. Zwischen den beiden letztgenannten Schichten können bei Bedarf weitere eingeführt werden. Der *Systemkern* stellt die Basisinfrastruktur des MES zur Verfügung, die die Grundlage für die Implementierung fachlicher Funktionalität bildet. Dazu gehört insbesondere die Unterstützung von Architekturaspekten, wie:

- Das Rahmenwerk zur Erstellung und Verwaltung von Systemkomponenten
- Persistenzmechanismen, inklusive der für das MES-Umfeld wichtigen Mechanismen für die Versionierung und die Historisierung von Datensätzen
- Mechanismen zur Unterstützung kontrollierter Änderungsprozesse im verteilten System
- die Kommunikationsinfrastruktur für synchronen und asynchronen Nachrichtenaustausch
- Sicherheitsmechanismen, inclusive Authentisierung und Autorisierung
- Grundlegende Konfigurations- und Erweiterungsmechanismen, die direkt im Kern eingreifen

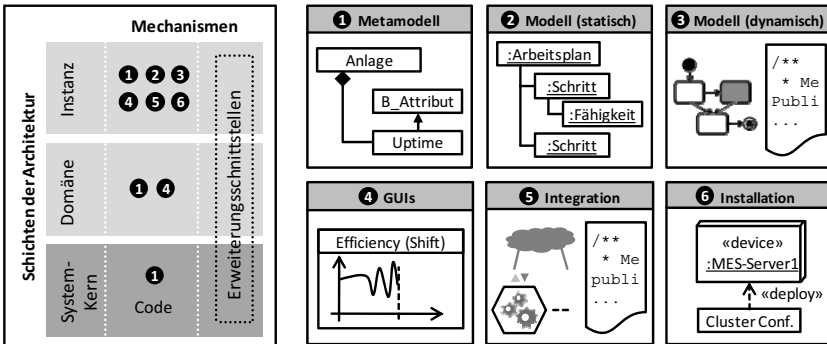


Abbildung 3.3: Mehrschichtiges Architekturkonzept (links) und native Mechanismen zur Anpassung des MES an die Anforderungen einer ausgewählten Produktion (rechts)

Darüber hinaus definiert der Kern *Primitive* (Grundbausteine), die zur Modellierung domänenspezifischer Problemstellungen in oberhalb liegenden Schichten genutzt werden. Dazu gehören Primitive zur Abbildung von Ressourcen, Material, Aufträgen, Arbeitsplänen, etc. (siehe Kapitel 2.3.2). Des Weiteren stellt der Kern generische Funktionalitäten zur Verfügung, die über die Grenzen einer Domäne hinaus relevant sind, wie die Funktionalität zur Modellierung historisierter endlicher Automaten, die über eingehende Signale und dadurch ausgelöste Zustandsänderungen Buch führen.

Die Schicht der *Domäne* stellt auf Basis des Kerns ein spezifischeres Metamodell für eine ausgewählte Domäne zur Verfügung. Dazu gehören die Definition zusätzlicher Modellelemente, vorgefertigte Konfigurationsdaten und Erweiterungsmodule. Eine Domäne entspricht in diesem Zusammenhang einer Branche, z. B. der PV-Industrie, oder einem Branchenweig, z. B. ein Dünnfilm Front-end oder Back-end. Das Domänenmodell vereinfacht die Modellierung von Problemstellungen aus dieser Domäne wesentlich und ermöglicht damit eine schnellere Einführung des MES in einer Fertigung innerhalb dieser Domäne. Ein Domänenmodell für die PV-Industrie könnte unter anderem angepasste Substratelemente oder Anlagenmodelle beinhalten, die häufig wiederkehrende Anforderungen aus dieser Industrie erfüllen.

Die oberste Schicht der *Instanz* steht für das Modell einer MES-Installation, das für den Betrieb einer spezifischen Fabrik genutzt wird. Dieses Modell bildet das Produktionssystem in ausreichendem Detaillierungsgrad ab, um die ausgewählten Geschäftsprozesse mit der entsprechenden Systemfunktionalität unterstützen zu können.

### 3.2.2 Mechanismen zur Anpassung von MES Lösungen

Abbildung 3.3 rechts zeigt die wichtigsten Mechanismen, die zur Anpassung einer MES-Instanz an die konkreten Anforderungen eines Produktionssystems zur Verfügung stehen. Zunächst besteht die Möglichkeit von Anpassungen im Systemkern, die durch den

Hersteller im Rahmen der Systementwicklungszyklen direkt im Quellcode vorgenommen werden und damit einem breiten Kundenkreis zur Verfügung stehen. Darüber hinaus existieren verschiedene Erweiterungsschnittstellen (engl. Application Programming Interfaces (APIs)), die dem Integrator oder Fabrikbetreiber die Integration selbst entwickelter Komponenten oder die Integration von Systemen anderer Hersteller ermöglichen. Diese Erweiterungsmechanismen greifen auf unterschiedlichen Schichten ins System ein. Für die Implementierung solcher Erweiterungen kommen die klassischen Werkzeuge der Softwareentwicklung und Bibliotheken und Rahmenwerke des MES zum Einsatz. Zur Anpassung oder Erweiterung des Metamodells und zur Modellierung einer gegebenen Produktion bringt das MES eigene Werkzeuge und Mechanismen mit, die mit einem höheren Abstraktionsgrad arbeiten. Die wichtigsten Aufgabenfelder zur Anpassung des Systems unter Nutzung dieser Werkzeuge sind:

Das *Metamodell* stellt die Sprache zur Verfügung, in der die konkrete Produktion modelliert wird. Es wird zunächst durch Systemkern, Domänenmodell und etwaige weitere Schichten definiert, die schließlich an die Anforderungen des gegebenen Produktionssystems angepasst werden. Solche Anpassungen umfassen beispielsweise die Ergänzung spezifischer Attribute an ausgewählten Metamodellelementen, die Definition projektspezifischer Metamodellelemente auf Basis verfügbarer Primitive und die Definition von Assoziationen zwischen den Elementen.

Das *statische Modell* bildet die statischen Zusammenhänge der realen Produktionsumgebung in einem informationstechnischen Modell auf Basis des Metamodells ab. Diese Daten werden auch *Masterdaten* genannt und umfassen Modellelemente, die längerfristig im System angelegt und genutzt werden (im Gegensatz zu Bewegungsdaten, die für einen begrenzten Zeitraum zur Verfügung stehen). MES-Komponenten arbeiten mit der virtuellen Repräsentation und wirken über das Modell auf die reale Produktion ein. Im statischen Modell sind beispielsweise die Ressourcen der realen Produktion abzubilden. Dazu gehören u. a. die Anlagen, die durch die Anlagen angebotenen Prozesse und Materialträger.

Das *dynamische Modell* ist eng mit dem statischen Modell integriert und ergänzt es um die Geschäftslogik, die Aspekte des Systemverhaltens zur Laufzeit beschreibt. Über das dynamische Modell wird das MES in Echtzeit mit der Produktionsumgebung synchronisiert. Im Zusammenspiel mit dem statischen Modell werden beispielsweise Dispatchregeln, Prozesse zur Fehlerbehandlung und die Verifikationsstrategie von Material an einer Anlage modelliert.

*Graphical User Interfaces (GUIs)* als Schnittstellen des MES zum menschlichen Anwender erfordern erfahrungsgemäß besondere Aufmerksamkeit im Zuge der Projektierung. Die Domänen-Ebene stellt domänenspezifische, generische GUI zur Verfügung. Um Geschäftsprozesse, die die Interaktion mit Anwendern benötigen, bestmöglich zu unterstützen, müssen dem Anwender zu jedem Zeitpunkt die von ihm benötigten Informationen und die im jeweiligen Kontext relevanten Funktionen in geeigneter Art und Weise aufbereitet zur Verfügung gestellt werden. Ebenso müssen sich Abläufe, die sich aus der Definition der Geschäftsprozesse ergeben, über die Benutzeroberfläche abbilden lassen. Folglich sind grafische Benutzeroberflächen mit hoher Wahrscheinlichkeit im Engineering-Prozess anzupassen.

Die *Integration*. Aufgrund der in Abschnitt 2.3 beschriebenen Rolle des MES besteht ein hoher Integrationsbedarf bei der Einführung solcher Systeme. Für die vertikale Integration der Produktions-IT sind insbesondere die Anlagen der Produktion in der beschriebenen Form zu integrieren. Dazu kommen die Integrationswerkzeuge des MES oder eigenständige Zusatzprodukte zum Einsatz (siehe die Integrationsebene in Abbildung 2.6). Diese stellen die benötigten Treiber für die eingesetzten Kommunikationsprotokolle, inklusive der für die Integration benötigten Konfiguratoren und Frameworks zur Implementierung komplexer Logik zur Verfügung. Bei Verwendung der Integrationswerkzeuge des MES wird die Logik in Form konfigurierbarer endlicher Automaten, interpretierter Skripte und kompilierter Workflows spezifiziert. Nach oben hin ist das MES mit Systemen der strategischen Ebene zu integrieren. In diesem Zusammenhang ist insbesondere der Bedarf zur Transformation von Daten zwischen den in beiden Welten verwendeten Datenmodellen zu erwarten der konfiguriert bzw. implementiert werden muss. Schließlich gibt es den Bedarf zur Integration anderer IT-Systemen innerhalb der MES-Ebene, wobei Integrationsaufwand und verwendete Werkzeuge unter anderem davon abhängen, ob standardisierte Schnittstellen zum Einsatz kommen oder ob die Integration auf Basis der von den Systemen zur Verfügung gestellten APIs neu implementiert werden muss.

Bei der *Installation* wird das MES in der Zielumgebung ausgerollt. Bei der Auswahl der Ziel-Hardware, des Softwarestapels unterhalb des MES und der Konfiguration der Laufzeitumgebung des MES sind zahlreiche nicht-funktionale Randbedingungen zu berücksichtigen. Dazu gehören insbesondere die geforderten Verfügbarkeit des Systems, das erwartete Lastprofil und vorgegebener Netzwerk- und Hardware-Infrastrukturen.

## 3.3 Analyse des MES-Engineering-Prozesses

Der Begriff des *MES-Engineering* fasst Prinzipien, Methoden und Werkzeuge zusammen, die zur Auswahl, Anpassung und kontinuierlichen Wartung der MES-Ebene für Produktionsumgebungen benötigt werden. Er beinhaltet damit eine größere Bandbreite an Aufgabenstellungen als die reine Lösungsentwicklung, die vielfach mit dem deutschen Begriff der *Projektierung* verbunden wird. Ziel des MES-Engineering ist, Fabriken mit einer MES-Ebene auszustatten, die die initialen Anforderungen der Betreiber erfüllt und mit den sich über den Lebenszyklus der Fabrik ändernden Anforderungen Schritt hält. Dabei wird sichergestellt, dass die Software ihren Beitrag zur Lösung der eingangs geschilderten Problemstellung liefert (siehe Kapitel 1.1 und 2.5) und so tatsächlich den erwarteten Nutzen im Sinne einer Kapitalrendite (engl. return on investment) bringt.

Abhängig von der Phase bzw. der Aufgabenstellung wird der Engineering-Prozess von unterschiedlichen Akteuren ausgeführt. Falls beim Fabrikbetreiber die benötigten Kompetenzen vorhanden sind, kann dieser das Engineering fast vollständig selbst übernehmen. In vielen Fällen werden zumindest die Anpassungsarbeiten, die auf Mechanismen wie die oben beschriebenen zurückgreifen, von einem Dienstleister durchgeführt. Dieser Dienstleister kann entweder der MES-Hersteller selbst oder ein für die gewählten Produkte zertifiziertes Systemhaus oder ein zertifizierter Integrator sein.

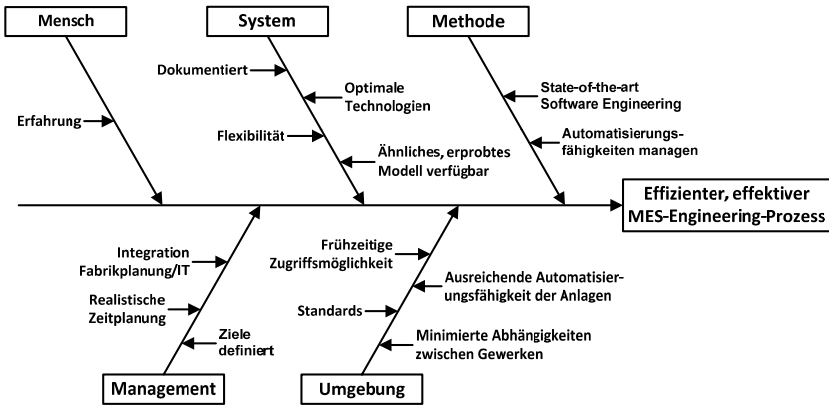


Abbildung 3.4: Kritische Erfolgsfaktoren des MES-Engineering-Prozesses

Neben grundsätzlichen kritischen Erfolgsfaktoren werden insbesondere die Aufgabenstellungen des MES-Engineering in den unterschiedlichen Phasen des Fabriklebenszyklus betrachtet.

### 3.3.1 Kritische Erfolgsfaktoren

Das Ursache-Wirkungsdiagramm (siehe z. B. [Balzert, 2008]) in Abbildung 3.4 fasst einige kritische Erfolgsfaktoren zusammen, die für die Definition des Engineering-Prozesses in einem konkreten Projekt zu berücksichtigen sind. Neben Erfahrungswerten aus Industrieprojekten sind Faktoren in die Darstellung eingeflossen, die in der Literatur als kritische Erfolgsfaktoren identifiziert wurden [Erdmann und Dreiss, 2006; Dreiss, 2007; Kletti, 2007; Meier u. a., 2007a, 2008; Meier und Konrad, 2009]. Während den Haupteinflussgrößen 'Mensch', 'System' und 'Management' in der Praxis zweifellos eine hohe Bedeutung zuzumessen ist, sollen mit dieser Arbeit schwerpunktmäßig die Haupteinflussgrößen 'Methode' und 'Umgebung' Berücksichtigung finden. Ein gute Strategie zum Umgang mit den beiden letztgenannten Haupteinflussgrößen liefert, unabhängig von der gewählten Softwarelösung und der spezifischen Projektorganisation, einen wichtigen Beitrag zum Projekterfolg.

### 3.3.2 Der MES-Engineering-Bedarf im Fabriklebenszyklus

Den folgenden Betrachtungen liegt ein Fabriklebenszyklus entsprechend der Definitionen aus den Kapiteln 2.1.1 und 2.2.4 zugrunde. Neben dem offensichtlichen Entwicklungs- bzw. Anpassungsbedarf bis zur Inbetriebnahme einer MES-Lösung besteht auch in den späteren Phasen des Lebenszyklus des MES immer wieder die Notwendigkeit für Anpassungen. Der Lebenszyklus des MES muss dabei nicht zwangsläufig von Anfang an mit



dem Fabriklebenszyklus synchronisiert sein. In Fällen, in denen das MES nicht zwingend für den Betrieb einer Fabrik erforderlich ist, kann dieses zu einem späteren Zeitpunkt ausgerollt werden. Allerdings ist auch in diesem Fall eine große Ähnlichkeit des Lebenszyklus des IT-Systems und des Fabriklebenszyklus zu erwarten. Aufgrund dessen und der beschriebenen wachsenden Bedeutung des MES in der PV-Industrie (siehe Kapitel 1.1) gehen die folgenden Abschnitte von einem Vorgehen aus, bei dem das MES-Engineering von Anfang an synchron zum Fabriklebenszyklus erfolgt.

#### **(Investitions-)Planung**

Die Auswahl geeigneter Lieferanten und Produkte für die MES-Schicht in den frühen Phasen des Fabriklebenszyklus hat wichtige Auswirkungen auf die späteren Phasen. Dies soll an zwei Aspekten verdeutlicht werden [Wiendahl u. a., 2007; Kletti, 2007]: Der Überdeckungsgrad der Anforderungen aus dem geplanten Produktionssystem mit den durch die evaluierten Systeme nativ zur Verfügung gestellten Funktionalitäten und den von ihnen berücksichtigten Randbedingungen lässt auf den zu erwartenden Engineering-Aufwand und das damit verbundene Risiko für das Gesamtprojekt schließen. Die Flexibilität der ausgewählten IT-Systeme geht direkt in die Gesamtflexibilität des Produktionssystems für zukünftige Modifikationen oder Erweiterungen ein.

Eine besondere Herausforderung bei der Auswahl von Lieferanten und Produkten für die PV-Industrie liegt in den begrenzten Erfahrungswerten aus dem praktischen Einsatz der Produkte. Während vergleichbare Referenzprojekte ansonsten eine wichtige Hilfe im Auswahlprozess darstellen, stehen solche aufgrund des jungen Alters der Branche und ihres vergleichsweise geringen Reifegrads noch nicht in einem wünschenswerten Umfang zur Verfügung.

**Fazit:** Wege zur Verbesserung der Präzision des Auswahlprozesses lassen einen positiven Beitrag zu Effizienz und Effektivität des Produktionssystems erwarten.

#### **Entwicklung**

Ein guter Teil des Aufwands für das MES-Engineering ist in der Phase 'Entwicklung' des Fabriklebenszyklus zu investieren. Wichtige Aktivitäten sind die detaillierte Anforderungsanalyse und die Spezifikation der Kerngeschäftsprozesse, die an die initiale Definition des Produktionssystems gekoppelt ist und die Basis für die Entwicklung des MES-Modells bildet. Falls keine standardisierte IT-Schnittstelle zur Anlagenintegration genutzt wird, ist eine proprietäre Schnittstelle zu definieren. Ebenso sind Erweiterungen und Festlegungen auf Basis der Freiheitsgrade für den Fall der Verwendung von Standard-Schnittstellen zu spezifizieren. Schließlich ist unter Nutzung der beschriebenen Anpassungsmechanismen die Instanz-Schicht aufzubauen. Architektur und Designentscheidungen, die in dieser Phase getroffen werden, bestimmen wichtige Eigenschaften der Systeme, die zu einem späteren Zeitpunkt nur schwer zu ändern sind. Dies gilt insbesondere, wenn man von dem in der PV-Industrie häufig geforderten 24/7 Betrieb ausgeht. Zu diesen Eigenschaften gehören die Skalierbarkeit des Systems und erreichbare Transaktionsraten bzw. -zeiten. Die physikalische Fabrik ist in dieser Phase noch

nicht vorhanden. Im Hinblick auf das geplante Anlagenset ist erfahrungsgemäß noch mit Änderungen zu rechnen. Aufgrund der begrenzten Erfahrungen der PV-Industrie mit den jeweils neuesten Technologien und Produktionssystemen in der Serienproduktion sind ein entsprechender Aufwand und geeignete Mechanismen für Lernprozesse zu einzuplanen. [Jindal und Khandelwal, 2005; Kletti, 2007; Meier u. a., 2007a, 2008]

**Fazit:** Aus Sicht des MES-Engineering entsprechen Aufgaben und Vorgehen in den Phasen 'Entwicklung' und 'Aufbau' denen eines Softwareprojekts mit hohem Integrationsaufwand und können folglich entsprechend der Maßgaben bekannter Strategien zur Softwareentwicklung ausgeführt werden. Ein besonderes Charakteristikum stellen die Abhängigkeiten zu den physikalisch noch nicht vorhandenen Anlagen der zu errichtenden Fabrik dar.

#### **Aufbau und Produktionsanlauf**

Auf Basis der Ergebnisse der vorausgehenden Phase wird das MES in der Zielumgebung installiert. Mit der physikalischen Verfügbarkeit der Anlagen wird deren IT-Integration vorgenommen, wobei der Gesamtaufwand für diese Aktivität in erster Linie von der Zahl der Anlagentypen, deren Komplexität und der Zahl der Instanzen pro Typ abhängt [Erdmann und Dreiss, 2006; Dreiss, 2007]. Erst nach der erfolgreichen Integration der Anlagen kann das MES in der Produktionsumgebung in vollem Umfang in Betrieb genommen werden. Tests in der realen Umgebung fordern für viele Anwendungsfälle die reale Prozessierung von Material. Das reale Material muss so vorprozessiert sein, dass es für die zu untersuchenden Anwendungsfälle genutzt werden kann. Im Anschluss an die Tests ist das Material nicht zwingend weiter verwendbar. Während der Tests sind die Anlagen für andere Arbeiten nicht verfügbar. Das zeigt, dass bereits die Tests von Normalabläufen mit einem nicht zu vernachlässigenden Zeit- und Kostenaufwand verbunden sind. Das Verhalten des Systems in Ausnahmesituationen lässt sich dabei aufgrund einer möglichen Gefährdung von Anlagen oder Material nur schwer oder gar nicht in der Produktionsumgebung validieren. Andererseits legen die im Kapitel 2.5 diskutierten Herausforderungen im Blick auf die Auswirkungen von Fehlern in der Produktions-IT den Bedarf umfangreicher Testläufe nahe. Während diese Faktoren auf einen größeren Zeitbedarf für die Inbetriebnahme der Produktions-IT auf MES-Ebene hindeuten, wäre gleichzeitig eine frühestmögliche produktive Verfügbarkeit des MES in der zu errichtenden Fabrik wünschenswert: Viele Werkzeuge, die später einen stabilen Produktionsbetrieb gewährleisten, könnten bereits in einer frühen Phase des Anlaufs zum besseren Verständnis komplexer Zusammenhänge beitragen und damit einen Beitrag zur Beschleunigung des Anlaufs liefern (siehe Kapitel 1.1 und 2.5).

**Fazit:** Die Integration zahlreicher Teilsysteme im Zielproduktionssystem und deren Inbetriebnahme stellt eine kritische Hürde aus der Perspektive des MES-Engineering dar. Die Möglichkeit zur virtuellen Inbetriebnahme der MES-Ebene in einer mit der Produktionsumgebung vergleichbaren Umgebung, die Konfigurationsmöglichkeiten für vielfältige Anwendungsszenarien bietet, könnte die Aktivitäten in der Anlaufphase entzerren, typische, durch IT-Systeme verursachte, Probleme vermeiden helfen und zu einem optimierten Anlaufprozess führen.

## Der operative Produktionsbetrieb

Auch im Rahmen des operativen Produktionsbetriebs fallen Engineering-Aufgaben im MES-Umfeld an. Solche Aufgaben können u. a. durch Änderungen an Geschäftsprozessen oder die Einführung neuer Produkte im laufenden Betrieb begründet sein, die Anpassungen auf der MES-Ebene nach sich ziehen. Darüber hinaus ist mit Änderungen zu rechnen, die ihre Ursache im IT-Umfeld haben, wie das Einspielen von Fehlerkorrekturen oder neuer Softwareversionen auf MES-Ebene oder auf Anlagenebene. Im Einzelfall sind derartige Änderungen mit hohen Risiken verbunden, die vor der Umsetzung der Änderungen im Produktivsystem in geeigneter Weise evaluiert werden müssen. Ein Teilaspekt der Evaluierung kann der Test kritischer Szenarien in einer Testumgebung sein. Soweit Funktionalitäten betroffen sind, die die Interaktion mit Anlagen benötigen, muss die Testumgebung die Kommunikation mit Anlagen ermöglichen.

**Fazit:** MES-Engineering-Aufgaben im operativen Produktionsbetrieb erfordern fallweise ein Bündel von Maßnahmen, die den störungsfreien Weiterbetrieb sicherstellen. Eine Testumgebung, die die vorhandenen Produktionsanlagen berücksichtigt, bietet zusätzliche Möglichkeiten zur Minimierung des Risikos von Störungen.

## Umbau und Erweiterungen

Die Art der Engineering-Aufgaben im Zusammenhang mit Umbau- oder Erweiterungsmaßnahmen hängt wesentlich vom Typ des Umbaus ab. So ist der Fall umfassender struktureller Umbauten, die bei abgeschalteter Produktion stattfinden und anschließend eine umfassende Anlaufphase notwendig machen, aus der Perspektive des MES-Engineering sehr gut mit den ursprünglichen Phasen von Entwicklung, Aufbau und Produktionsanlauf zu vergleichen. Besondere Herausforderungen stellen hingegen Umbauprojekte dar, die bei minimalen Abschalt- und Anlaufzeiten gleichzeitig einen hohen Integrationsaufwand aufweisen, also eine passgenaue Vorbereitung des MES bei minimalem Risiko, Verzögerungen zu verursachen, erfordern. Gleiches gilt für Umbaumaßnahmen, die bei laufender Produktion ausgeführt werden sollen und dabei den Betrieb nicht stören dürfen. Letzterer Fall hat für die PV-Industrie eine besondere Relevanz vor dem Hintergrund des Einsatzes skalierbarer Produktionslinien, die im Rahmen ihres Lebenszyklus signifikante Kapazitätserweiterungen um ein vielfaches der Ursprungskapazität vorsehen.

**Fazit:** Im Hinblick auf das Engineering des MES fallen vergleichbare Aufgaben wie in den Phasen Entwicklung, Aufbau und Produktionsanlauf an, allerdings unter den Randbedingungen der störungsfreien Integrierbarkeit in die bestehende Systemlandschaft.

### 3.3.3 Fazit aus der Analyse des MES-Engineering

Der Erfolg von Projekten zum Aufbau und der kontinuierlichen Weiterentwicklung von MES-Lösungen hängt von zahlreichen kritischen Erfolgsfaktoren ab. Ein Lösungsvorschlag zur Optimierung des MES-Engineering, der unabhängig von einzelnen Systemanbietern mit ihren vielfältigen Technologien und von ausgewählten Projektorganisationen

sein soll, muss sich in erster Linie auf die Haupteinflussgrößen 'Methode' und 'Umgebung' beziehen. Geht man von den vielfältigen Möglichkeiten für produktionssystembezogene Anpassungen aus, die beispielhaft für ein MES-Produkt betrachtet wurden, und berücksichtigt dazu MES-Anforderungen der PV-Industrie und die Ergebnisse der Analyse der typischer Aufgabenstellungen für das MES-Engineering, so wird deutlich:

1. Ziel des MES-Engineering ist eine lauffähige MES-Ebene, die in der Fabrik zu jedem Zeitpunkt die der aktuellen Phase des Lebenszyklus angemessene Funktionalität zur Verfügung stellt. Dazu gehört, dass die MES-Ebene den Anlauf zum frühestmöglichen Zeitpunkt unterstützt und keine zusätzlichen Verzögerungen auslöst.
2. Die Integration der MES-Ebene mit den Anlagen einer Produktion liefert einen wesentlichen Beitrag zur Funktionalität dieser Systeme.
3. Die fehlende Zugriffsmöglichkeit auf die reale Produktion über weite Teile des Engineering-Prozesses und Einschränkungen im Hinblick auf die an realen Anlagen untersuchbaren Szenarien stellen ein Risiko für einen effektiven Engineering-Prozess dar. Desweiteren verhindert die späte Interaktion mit der realen Produktion kurze Lernzyklen.

Diese Schlussfolgerungen legen die Entwicklung einer virtuellen Fabrikumgebung nahe, die sich nahtlos in den Prozess des MES-Engineering integrieren lässt und das MES-Engineering in den unterschiedlichen Phasen des Fabriklebenszyklus unterstützt.

## 3.4 Softwareengineering

Nachdem wesentliche Teilbereiche des MES-Engineering als klassische softwaretechnische Problemstellungen identifiziert wurden, die unter den Randbedingungen des Fabriklebenszyklus und der Fabrikumgebung zu lösen sind, sollen die Aufgabenstellungen aus der Perspektive der Softwaretechnik beleuchtet werden. Die Disziplin des Softwareengineering (deutsch: Softwaretechnik) beschäftigt sich mit der ingenieurmäßigen Entwicklung und Anwendung von Softwaresystemen [Balzert, 2001]. Ziel des Softwareengineering ist letztlich die Beherrschbarkeit der hohen Komplexität dieser Systeme.

### 3.4.1 Prinzipien, Methoden und Werkzeuge

Das Ziel des Softwareengineering wird nach [Balzert, 2001] durch die Berücksichtigung von *Prinzipien* und die Bereitstellung und Verwendung geeigneter *Methoden* und *Werkzeuge* verfolgt. Die Zahl der Freiheitsgrade, die das Softwareengineering theoretisch im Hinblick auf die Auswahl der beiden letztgenannten bietet, wird im MES-Engineering durch die Auswahl der verwendeten Produkte und Plattformen eingeschränkt. So liegt aufgrund der Auswahl fest, ob im Rahmen der Modellierung und Pflege von Datenmodell und Geschäftslogik Möglichkeiten, die durch das Paradigma der Objektorientierung

nutzbar werden, zum Tragen kommen oder in wie weit das Paradigma der Serviceorientierten Architektur unterstützt wird und mit seinen Vor- und Nachteilen Entwicklung, Betrieb und Wartung der MES-Schicht beeinflusst.

### Vorgehensmodelle

Die Disziplin des Softwareengineering beschäftigt sich eingehend mit dem Lebenszyklus von Softwaresystemen. Ein besonderes Augenmerk liegt dabei auf dem Entwicklungsprozess komplexer Softwaresysteme und in der Fragestellung, wie dieser Prozess so gestaltet werden kann, dass Softwareprojekte mit hoher Wahrscheinlichkeit zum gewünschten Erfolg führen. Einen Lösungsansatz stellen *Vorgehensmodelle* dar, die eine stark strukturierte Vorgehensweise für Softwareentwicklungsprojekte vorgeben, indem sie vorschreiben, wer im Projekt wann, was, wie zu tun hat. [V-Modell-XT, 2009] und [RUP, 2009] seien beispielhaft als Vertreter für solche Vorgehensmodelle genannt. Zur praktischen Anwendung müssen beide auf das konkrete Projekt angepasst werden (engl. tailoring).

### Agile Softwareentwicklung

Ogleich in vielen Organisationen verbindlich vorgegeben, ist der Beitrag formaler Vorgehensmodelle zum Erfolg von Softwareentwicklungsprojekten nicht unumstritten. Die *Agile* Strömung, deren Prinzipien das *Agile Manifest* [Beck, 2001] beschreibt, stellt daher die individuellen Projektbeteiligten und deren Kommunikation, lauffähige Software, die Zusammenarbeit mit dem Kunden und die Reaktionsfähigkeit auf Änderungen vor formalen Prozessen und Werkzeugen in den Vordergrund. [Wolf und Roock, 2008] kommen in einer Studie zum Einsatz agiler Konzepte im deutschsprachigen Raum zum Ergebnis, dass diese zwischenzeitlich, über die akademische Diskussion hinaus, erfolgreich in der Industrie eingesetzt werden. Eine Reihe von Gründen spricht für die Berücksichtigung agiler Konzepte beim MES-Engineering in der PV-Industrie: Aktuell ist, bedingt durch das junge Alter der Industrie, von einem hohen Lernbedarf in Bezug auf die genutzten Produktionssysteme auszugehen [Reddig, 2009]. Dieser Lernbedarf erschwert die vollständige a-priori Spezifikation von MES-Lösungen und erfordert ein höheres Maß an Flexibilität bezüglich Anforderungsänderungen.

### 3.4.2 Qualität von Softwaresystemen

Die Qualität eines Softwaresystems ist nach [ISO/IEC 9126, 2001] nicht als eine absolute Größe zu sehen, sondern als "Gesamtheit der Merkmale und Merkmalswerte eines Produktes oder einer Dienstleistung, die sich auf deren Eignung beziehen, festgelegte oder vorausgesetzte Erfordernisse zu erfüllen" [ISO/IEC 9126, 2001]. Die Definition der Qualität unterscheidet einerseits die Produktqualität (hier: Qualität der MES-Lösung) und andererseits die Prozessqualität (hier: Qualität der Organisation und der Abläufe, die zum lauffähigen MES führen). Zwischen beiden Aspekten der Qualität besteht ein Zusammenhang (siehe auch die beiden vorhergehenden Abschnitte). [ISO/IEC 9126, 2001] definiert ein formales Modell zur Beschreibung der Produktqualität, das die Grundlage zur Messung der Qualität bietet.

### 3.4.3 Softwaresysteme testen

Testverfahren stellen eine wichtige Möglichkeit dar, um die Qualität von Softwaresystemen im Hinblick auf Kriterien zu überprüfen, die durch das Qualitätsmodell beschrieben werden. Das Testen komplexer Softwaresystem kann als eigene Teildisziplin des Software Engineering betrachtet werden. Einen Überblick bieten beispielsweise [Versteegen u. a., 2006; SIGS DATACOM GmbH, 2008]. Da Testverfahren in der vorhergehenden Analyse des MES-Engineering als ein wichtiger Baustein zur Lösung der beschriebenen Defizite identifiziert werden, sei an dieser Stelle auf die Bedeutung einer geeigneten *Testumgebung* hingewiesen, die nach [RTI, 2002; Versteegen u. a., 2006] eine entscheidende Voraussetzung für die effiziente Testdurchführung darstellt. Hervorgehoben wird insbesondere die Bedeutung der Anforderung, dass der Tester in der Testumgebung die Kontrolle über die Vorbedingungen für die Durchführung eines Tests hat. Zur Sicherstellung der Nachvollziehbarkeit der Testergebnisse muss sich der Zustand der Testumgebung eindeutig beschreiben lassen [Versteegen u. a., 2006].

### 3.4.4 Fazit Software Engineering

Es ist zu erwarten, dass Auswahl und Art der Umsetzung von Prinzipien, Methoden und Werkzeugen des Softwareengineering im Rahmen des MES-Engineering signifikante Auswirkungen auf dessen Effizienz und Effektivität haben. Unabhängig davon, ob dessen Grundausrichtung auf einem formalen Vorgehensmodell beruht oder verstärkt auf agile Konzepte baut, spielt die Verfügbarkeit einer geeigneten Integrations- und Testumgebung eine wichtige Rolle im Risikomanagement- und Qualitätssicherungskonzept. Der Aufwand zum Aufbau und zur Pflege der Testumgebung muss allerdings einen echten wirtschaftlichen Vorteil bieten und darf nicht vom eigentlichen Ziel des MES-Engineering, einer MES-Lösung, die die Anforderungen des Fabrikbetreibers in der realen Fertigung erfüllt, ablenken.

## 3.5 Emulationsgestütztes MES-Engineering

Sowohl das Ergebnis der Analyse des MES-Engineering als auch die Betrachtung aus der Sicht des Softwareengineering weist auf das Potential einer virtuellen Fabrikumgebung hin, die für das MES-Engineering als Explorations- und Testumgebung zur Verfügung steht. Dieses Potential soll mit dem folgenden Teil der Arbeit nutzbar gemacht werden. Dabei wird stets die Übertragung des Leitgedankens der Agile-Strömung berücksichtigt, dass das eigentliche Ziel des MES-Engineering die lauffähige MES-Ebene ist, die die Anforderungen des Fabrikbetreibers erfüllt. Etwaiger Zusatzaufwand, der durch den Aufbau und die Pflege der virtuellen Fabrikumgebung entsteht, darf den angestrebten wirtschaftlichen Nutzen der MES-Lösung nicht gefährden. Daher kann die Entwicklung, Wartung und Nutzung einer solchen virtuellen Fabrikumgebung nur integriert im Prozess des MES-Engineering stattfinden. Dieses Vorgehen wird als *emulationsgestütztes MES-Engineering* bezeichnet.

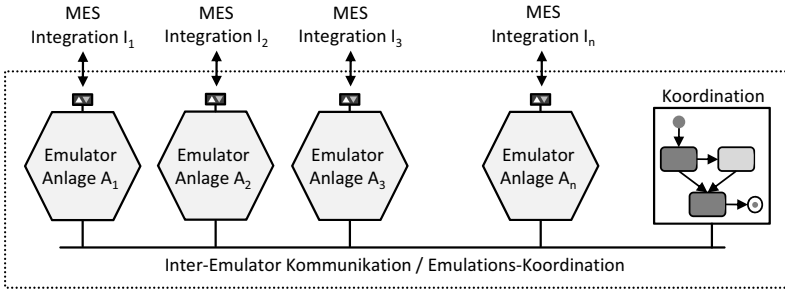


Abbildung 3.5: Das Konzept einer Emulationsumgebung

Im Zentrum dieses Konzepts steht eine *virtuelle Fabrik* auf Basis einer *Emulationsplattform*, die ein Modell der realen Fabrik aus Sicht des MES nutzt und damit das Verhalten dieser Fabrik an ihren realen IT-Schnittstellen emuliert. Abbildung 3.5 zeigt schematisch den Aufbau der virtuellen Fabrik. Sie besteht aus autonomen Anlagen in Form von *Emulatoren*, welche mit dem realen MES interagieren und sich dabei aus Sicht des MES nicht von realen Anlagen unterscheiden. Diese Anlagen sind in eine *virtuelle Fabrikumgebung* eingebunden, die alle aus MES-Sicht relevanten Aspekte der realen Fabrikumgebung abbildet. Ein ergänzendes Steuerungssystem, das in der realen Fabrik keine direkte Entsprechung hat, ermöglicht die Koordination der verteilten Emulatoren zum Anfahren ausgewählter Szenarien und unterstützt damit die Umsetzung der Anforderungen an ein Testsystem aus Kapitel 3.4. Zunächst werden schrittweise einige zentrale Aspekte der realen Fabrik analysiert, die für ihre virtuelle Abbildung wichtig sind.

### 3.5.1 Das System 'Anlage'

Den wichtigsten Basisbaustein der virtuellen Fabrik stellt die 'Anlage' entsprechend der Definition in Kapitel 2.1 dar. Zum aktuellen Zeitpunkt sind die Grenzen des Systems einer Anlage in der PV-Industrie noch nicht allgemeingültig definiert [SEMI PV2, 2009]. Es existieren zahlreiche Varianten von Integrations-, Automatisierungs- und Steuerungskonzepten. Dennoch bildet eine Anlage in allen betrachteten Fällen aus Sicht der IT-Schnittstellen ein zustandsbehaftetes System, das auf hierarchische endliche Automaten abgebildet werden kann. Neben der IT-Anbindung wird dieses System über verschiedene Schnittstellen in seine Umgebung integriert. Dabei verbirgt die durch die IT-Schnittstellen markierte Systemgrenze ein Automatisierungssystem, dessen Komplexität im Vergleich unterschiedlicher Anlagen sehr stark schwankt. Abhängig von der zu lösenden Automatisierungsaufgabe werden innerhalb des Systems Anlage die in Kapitel 2.3.3 dargestellten Systemkonzepte eingesetzt.

#### Die Schnittstellen

Anlagen werden über zahlreiche Schnittstellen in den Kontext einer Produktion integriert, über die sie mit benachbarten und umschließenden Systemen interagieren. Über

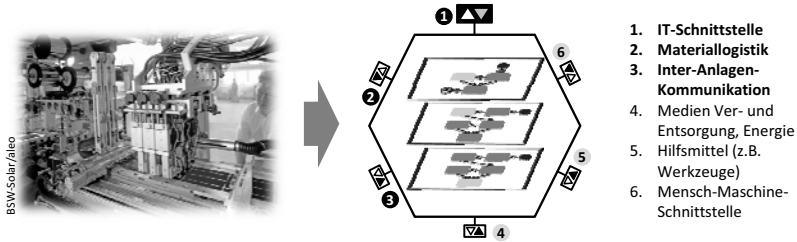


Abbildung 3.6: Die Systemgrenze der Anlage und relevante Schnittstellen

diese Schnittstellen beeinflusst einerseits die Umgebung den Zustand des Systems Anlage; andererseits resultieren Änderungen des Zustands der Anlage in der Interaktion der Anlage mit ihrer Umgebung. Abbildung 3.6 zeigt die wichtigsten Klassen dieser Schnittstellen. Die Klassen (1), (2) und (3) haben im Hinblick auf die Zielstellung der Entwicklung einer virtuellen Fabrik eine besondere Bedeutung, da (1) die virtuelle Umgebung mit dem realen Steuerungssystem verbindet und (2) und (3) die Anlage in logistische Zusammenhänge einbettet, die vielfältige Synchronisationspunkte mit dem MES-Modell haben.

Die Schnittstellen der Klasse (1) entsprechen der in Kapitel 2.3.3 konzeptionell beschriebenen IT-Anlagenschnittstelle und werden zukünftig in der PV-Industrie mehr und mehr entsprechend des in Kapitel 2.4.2 vorgestellten PVECI-Standards ausgeführt sein, sofern sich der Standard durchsetzt. Zum heutigen Zeitpunkt kommen, neben PVECI, zahlreiche proprietäre Integrationsschnittstellen zum Einsatz und müssen folglich durch die virtuelle Umgebung in entsprechender Weise unterstützt werden. Ihr Funktionsumfang kann als Untermenge des PVECI-Standards betrachtet werden. Während mit PVECI langfristig das Ziel verfolgt wird, dass eine Anlage eine IT-Schnittstelle zur Anbindung an übergeordnete Systeme aufweist, besitzen heute bisweilen mehrere Komponenten einer Anlage eigene Schnittstellen der Klasse (1).

Die Schnittstellen der Klasse (2) binden die Anlage in die Materiallogistik der Fabrik ein. Die Be- und Entladung der Anlage mit zu prozessierendem bzw. prozessiertem Material erfolgt über eine oder mehrere dieser Schnittstellen. Für ihre Implementierung kommen sehr unterschiedliche Technologien zum Einsatz: Von Schnittstellen zur Aufnahme von Automatisierungscarriern, über solche zur Aufnahme spezieller Prozesscarrier bis hin zu vielfältigen Lösungen zur Zuführung einzelner Substrate.

Für die Schnittstellen der Klasse (3) zur Inter-Anlagen-Kommunikation existieren bisher nur proprietäre Lösungen. Sie spielen insbesondere für die Verkettung von Anlagen eine Rolle, um die Verfolgung einzelner Substrate ohne den Zugriff auf physikalische Identifikationsmerkmale des Substrats zu ermöglichen. Die PV-EIS Arbeitsgruppe arbeitet aktuell an der Normierung dieser Schnittstelle (siehe Kapitel 2.4.2). Falls der Funktionsumfang der Schnittstelle bei den Normierungsarbeiten ausgedehnt wird, ist diese Schnittstelle zukünftig zu berücksichtigen.



**Fazit:** Die Klassen (1) und (2) der Anlagenschnittstellen sind im Anlagenmodell unmittelbar zu berücksichtigen. (3) ist für zukünftige Erweiterungen zu vorzusehen. Darüber hinaus ist ein Mechanismus einzurichten, der bei Bedarf die Nachbildung der Schnittstellen (4), (5) und (6) erlaubt.

### Automatisierungsszenarien aus Sicht der IT-Schnittstelle

Für diese Arbeit wird vorausgesetzt, dass sich die Interaktion des MES mit einer ausgewählten Anlage im Sinne von [SEMI PV2, 2009] bzw. [SEMI E30, 2007] in einer endlichen Menge spezifizierbarer *Automatisierungsszenarien* ausdrücken lässt. Automatisierungsszenarien beschreiben die Konversation zwischen MES-Ebene und Anlage beim Ablauf der korrespondierenden Anwendungsfälle des Systems Anlage. Dabei wird die Interaktion der Anlage über die als relevant identifizierten Schnittstellen berücksichtigt, z. B. das Ereignis “Carrier detektiert” beim Aufsetzen eines Materialträgers auf einen Port. Anwendungsfälle des Systems Anlage können vielfältig parametrierbar sein, z. B. die zu verwendenden Prozessprogramme oder Parameter solcher Programme. Diese können aber in gleicher Weise durch parametrisierte Automatisierungsszenarien abgebildet werden.

## 3.5.2 Das System 'Fabrik'

Im System 'Fabrik' werden Anlagen zu einem Gesamtsystem integriert, das die Fertigung spezifischer Produkte ermöglicht. Aus Sicht des MES sind für die Integration die folgenden Eigenschaften des Gesamtsystems von besonderem Interesse.

### Parallelität und Synchronisation in der realen Fabrik

Aus der Perspektive der MES-Ebene stellt eine Fabrik, deren Produktions-IT Architektur entsprechend Kapitel 2.3 aufgebaut ist, ein hochgradig parallelisiertes System dar. Dieses System ist in Abbildung 3.7 links abstrahiert dargestellt. Jede Anlage ist als eigenständiger Prozess zu betrachten, der über die IT-Schnittstelle bidirektional Nachrichten mit der MES-Ebene austauscht. Ausgehend vom Systemmodell der Anlage findet im Normalbetrieb unmittelbar zwischen den Anlagen allenfalls eine Interaktion über deren Logistikschnittstellen und deren Anlage-zu-Anlage Kommunikationsschnittstellen statt (letztere ist in Abbildung 3.7 nicht dargestellt). Im Folgenden wird davon ausgegangen, dass nur ein kleiner Anteil der Nachrichten bzw. Ereignisse, die von einer Anlage über die IT-Schnittstelle kommuniziert werden, unmittelbar von der Interaktion zwischen Anlagen ausgelöst wird. Gleichmaßen parallelisiert ist die Anbindung der Anlagen an das MES entsprechend der in Kapitel 2.3 beschriebenen Architektur. Abhängig von der genutzten Hardwareumgebung laufen die Prozesse, die die Anbindung realisieren, echt parallel auf unabhängigen Rechnerressourcen oder quasi-parallel in Multi-Tasking Umgebungen, die die Rechnerressourcen im Zeitmultiplexing-Verfahren an die auszuführenden Prozesse verteilen. Entsprechendes gilt für die MES-Dienste selbst. Die beschriebene Konstellation kann zu verschiedenen Ausnahmesituationen führen, die am Beispiel von Abbildung 3.7 erläutert werden sollen.

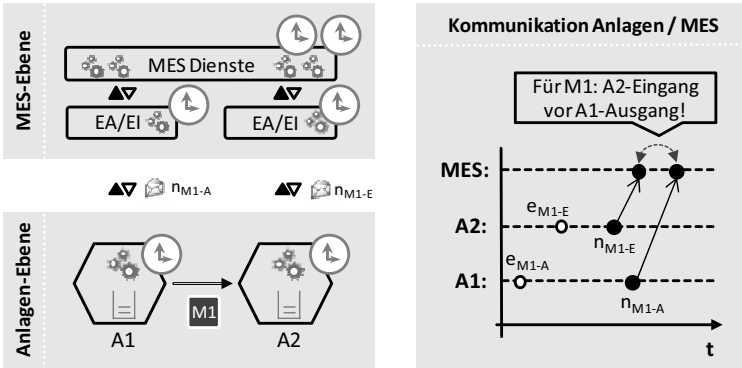


Abbildung 3.7: Beispiel: Wichtige Problemstellungen durch Parallelität und Synchronisation in der realen Fabrik

- Aufgrund der limitierten Synchronisation der Anlagenprozesse und der losen Kopplung durch das Kommunikationssystem besteht eine hohe Wahrscheinlichkeit, dass die Signalisierung von Ereignissen nicht in der Reihenfolge beim MES eingeht, in der sie physisch im Produktionssystem aufgetreten sind. Die Anlagen  $A_1$  und  $A_2$  sind verkettet und signalisieren die Ereignisse des physischen Materialeingangs ( $e_{M_1-E}$ ) und -ausgangs ( $e_{M_1-A}$ ) des Substrats  $M_1$  in Form der Nachrichten  $n_{M_1-E}$  und  $n_{M_1-A}$ . Die Verkettung stellt sicher, dass  $e_{M_1-A}$  bei  $A_1$  für alle Materialbewegungen  $M_i$  vor  $e_{M_1-E}$  bei  $A_2$  auftritt. Dennoch kann die Signalisierung dieser Ereignisse beim MES in umgekehrter Reihenfolge eingeht. Falls die Reihenfolge des Eintreffens der Ereignisse am MES zu unterschiedlichen Ergebnissen führt, spricht man von einer Wettlaufsituation (engl. *race-condition*).
- Nachrichten  $e$  gehen verloren. Solche Verluste führen dazu, dass der Zustand der realen Fabrik und das Zustandsabbild im MES auseinanderlaufen. Falls Prozesse auf MES-Ebene auf bestimmte Nachrichten warten, könnten sie aufgrund verlorener Nachrichten blockiert werden.
- Nachrichten  $e$  werden dupliziert. Ein physikalisches Ereignis wird dadurch mehrfach kommuniziert.
- Der Zugriff der nebenläufigen Prozesse (z. B. bei EA/EI) auf gemeinsam genutzte Ressourcen führt zu Verklemmungen durch wechselseitiges Warten (engl. *deadlocks*), sofern alle notwendigen Bedingungen erfüllt sind.

Auf der MES-Ebene sind Vorkehrungen zu treffen, um Fehler, die durch solche Ausnahmen entstehen können, zu vermeiden. Ausgewählte Problemstellungen könnten bei einem gegebenen System und bei Kenntnis aller relevanten Parameter mit formalen Ansätzen untersucht werden. Da solche Parameter aber proprietäres Wissen eines MES-Anbieters betreffen und die formale Untersuchung darüber hinaus zumeist mit hohem Aufwand verbunden ist, bleibt in vielen Fällen nur die Untersuchung in einem geeigneten Testsystem,

das von der MES-Implementierung unzureichend berücksichtigte Konstellationen mit einer Wahrscheinlichkeit aufdecken kann. Die Testumgebung muss zu diesem Zweck ein hochgradig nebenläufiges Verhalten aufweisen. Gleichzeitig erfordern einige der in Kapitel 3.1 beschriebenen Anforderungen an das MES die Möglichkeit zur Sequenzierung von Ereignissen über das gesamte verteilte System einer Fabrik. Diese Randbedingung wird immer dann wichtig, wenn Kausalketten von Ereignissen identifiziert werden sollen. Andere Anforderungen machen die Zuordnung von Ereignissen zu bestimmten Zeitpunkten der Realzeit erforderlich. Beide Anforderungen lassen sich erfüllen, indem alle Prozesse, die Zeitstempel generieren, synchronisierte Uhren verwenden [SEMI PV2, 2009]. Diese Anforderungen müssen auch in der verteilten Testumgebung abgebildet werden.

#### **Lastverhalten der realen Fabrik**

Die virtuelle Fabrik muss in der Lage sein, das Lastverhalten der realen Fabrik in gleicher Weise abzubilden. Legt man Performance-Anforderungen zugrunde, wie sie in Kapitel 2.2 beschrieben sind, muss pro Anlage im Maximalfall (Zelltester) mit 500-1000 Werten pro Sekunde gerechnet werden, die an die MES-Ebene zu übermitteln sind.

## **3.6 Anforderungen an Emulationsplattform und Verfahren**

Die Entwicklung des Verfahrens zum emulationsgestützten MES-Engineering zerfällt in zwei größere Teilprobleme. Das erste Teilproblem umfasst die Entwicklung der skizzierten Emulationsplattform und deren Implementierung. Abbildung 3.9 fasst die Anforderungen an die Emulationsplattform zusammen. Das zweite Teilproblem umfasst die Integration der virtuellen Fabrikumgebung in das MES-Engineering, also die Verfahrensentwicklung. Abbildung 3.8 fasst die Vorgehensweise der Verfahrensentwicklung zusammen und beschreibt die Zielsetzung und die wichtigsten Eigenschaften des Verfahrens.

<b>Anforderungen an das Verfahren</b>
<b>1. Zielstellung des Verfahrens</b>
a) Kompensation der späten Verfügbarkeit der realen Fabrik
b) Erweiterung der begrenzten Validierungsmöglichkeiten in der realen Fabrik
c) Absicherung kritischer Entscheidungen in einer möglichst frühen Phase des Fabriklebenszyklus
d) Reduktion der Risiken bei Inbetriebnahme und Anlauf
<b>2. Eigenschaften des Verfahrens</b>
a) Der für die Modellbildung zu investierende Aufwand muss in einem wirtschaftlich sinnvollen Verhältnis zum erzielbaren Nutzen stehen
b) Das Verfahren muss kompatibel zu den in der Industrie eingesetzten Vorgehensmodellen sein
c) Verfahren muss in der Projektorganisation implementierbar sein

Abbildung 3.8: Zusammenfassung der Anforderungen an das Verfahren

<b>Anforderungen an die Emulationsplattform</b>	
<b>1. Modellierung</b>	<ul style="list-style-type: none"> <li>a) Abbildung realer Fabriken durch virtuelle Fabriken in Form gekoppelter Anlagenemulatoren, die das Verhalten jeweils einer Anlage aus Sicht ihrer IT-Schnittstelle beschreiben</li> <li>b) Unterstützung eines variablen, das heißt den Anforderungen des Anwendungsfalls angemessenen Detaillierungsgrads des Verhaltens der Anlagen</li> <li>c) Modellelemente zur Abbildung des Logistik-Subsystems in einem Detaillierungsgrad der eine automatische Ablaufsteuerung erlaubt               <ul style="list-style-type: none"> <li>• Modellierung der Verkettung von Anlagen (Push, Pull)</li> <li>• Modellierung von Material und Materialträgern (z.B. Substrat, Carrier)</li> <li>• Fähigkeit zur Integration von Transportsystemmodellen</li> </ul> </li> <li>d) Unterstützung der Wiederverwendung zentraler Modellelemente               <ul style="list-style-type: none"> <li>• Virtueller Fabriken</li> <li>• Emulatoren</li> <li>• Konfigurationen</li> <li>• Zustandsdaten</li> </ul> </li> </ul>
<b>2. Integration</b>	<ul style="list-style-type: none"> <li>a) Fähigkeit zur (minimal-invasiven) Integration unterschiedlicher Emulator-Technologien               <ul style="list-style-type: none"> <li>• Durch den Anlagenhersteller zur Verfügung gestellte Emulatoren</li> <li>• Auf Basis von Integrationsframeworks erstellte Emulatoren</li> <li>• Integration verfügbarer Simulationsmodelle für die Nachbildung des Verhaltens einer Anlage</li> <li>• Entwicklungsumgebung für die Entwicklung eigener Emulatoren</li> </ul> </li> <li>b) Plattformunabhängigkeit im Umfeld der Emulatoren</li> <li>c) Integrierbarkeit der Emulationsplattform in die MES-Entwicklungs- und Testumgebung</li> </ul>
<b>3. Ausführung</b>	<ul style="list-style-type: none"> <li>a) Fähigkeit zur Nachbildung des Verhaltens der Anlagen in Echtzeit</li> <li>b) Anpassbarkeit der Ablaufgeschwindigkeit</li> <li>c) Fähigkeit zur Koordination von Szenarien, die beliebig viele Anlagen im Umfang des spezifizierten Skalierungsbereiches enthalten</li> <li>d) Analyse- und Interaktionsmöglichkeit für den Anwender bis auf die Ebene einzelner Emulatoren („Drill-down“)</li> <li>e) Fähigkeit zur Auswertung der Konversation zwischen Anlage und anbindendem IT-System</li> <li>f) Zustandsverwaltung               <ul style="list-style-type: none"> <li>• Fähigkeit zur Erzeugung konsistenter Zustandsabbilder aus einzelnen Anlagen und der gesamten virtuellen Fabrik</li> <li>• Fähigkeit zur Initialisierung virtueller Fabriken auf Basis verfügbarer Zustandsabbilder</li> </ul> </li> </ul>
<b>4. Randbedingungen</b>	<ul style="list-style-type: none"> <li>a) Skalierbarkeit               <ul style="list-style-type: none"> <li>• Virtuelle Fabriken im Umfang von 1 bis zu 1000 Anlagen</li> <li>• Paralleler Betrieb der virtuellen Anlagen entsprechend deren Einsatz in der realen Produktion bei industrieüblichem Kommunikationsaufkommen</li> </ul> </li> </ul>

Abbildung 3.9: Zusammenfassung der Anforderungen an die Emulationsplattform

# 4 Stand der Technik

Vor dem Hintergrund der in den Kapiteln 1 und 2.5 geschilderten Herausforderungen widmen sich seit Jahren immer wieder Arbeitsgruppen aus Forschungseinrichtungen und Industrie der Suche nach geeigneten Lösungen des beschriebenen Dilemmas, den Entwicklungs- und Konfigurationsprozess für Softwaresysteme im Produktionsumfeld zu optimieren und die Nachteile, die der verstärkte Softwareeinsatz mit sich bringt, zu minimieren. In diesen Kontext von Wissenschaft und Technik wird das Verfahren des emulationsgestützten MES-Engineering eingeordnet.

## 4.1 Einordnung der Arbeit in den Kontext von Wissenschaft und Technik

Der Leitgedanke, der dem Gros der Arbeiten zugrunde liegt, ist der bereits in Kapitel 2.1.1 vorgestellte Ansatz der lebenszyklusorientierten Betrachtung bzw. der integrierten Betrachtung der mit dem Lebenszyklus des Untersuchungsgegenstands zu synchronisierenden Prozesse. Diesen Gedanken manifestiert insbesondere das Konzept der *Digitalen Fabrik*. Es verfolgt nach [VDI 4499, 2008] die Ziele der ganzheitlichen Planung, Evaluierung und kontinuierlichen Verbesserung der relevanten Strukturen, Prozesse und Ressourcen der realen Fabrik mit Bezug zum herzustellenden Produkt. Der Fokus des

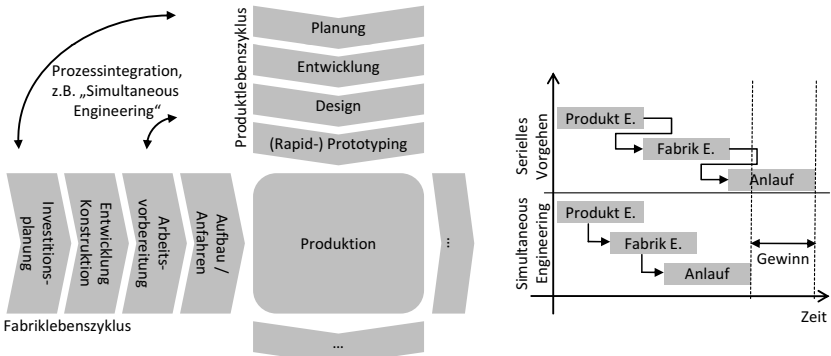


Abbildung 4.1: Integrierte Betrachtung von Lebenszyklen mit dem Ziel des Zeitgewinns (basierend auf Abbildung 2.1)

Konzepts liegt auf der Produktionsplanung [VDI 4499, 2008]. Während sich verschiedene Ansätze des Softwareengineering im Automatisierungsumfeld explizit aus dem Konzept der Digitalen Fabrik ableiten (hierarchische Planung, englisch: top-down), verfolgen andere, wie der später vorgestellte ACM-Ansatz, eher ein Vorgehen von unten nach oben (engl. bottom-up). Dennoch lassen sich die Ziele beider Vorgehensweisen im Lebenszyklusmodell darstellen. Letztlich verfolgen diese Ansätze die generischen Ziele der schnelleren Platzierung von Produkten am Markt (engl. time-to-market) und der Senkung der Kosten. Dazu wird der Mechanismus der Parallelisierung der Aufgabenbearbeitung (engl. *simultaneous engineering*, siehe Abbildung 4.1 genutzt, der die Entwicklungszeit verkürzen soll. Weitere Mechanismen sollen das Risiko weitreichender Fehlentscheidungen in frühen Phasen des Entwicklungsprozesses vermeiden, den Anlaufprozess verkürzen und die Identifikation optimaler Lösungen und Betriebspunkte unter gegebenen Randbedingungen unterstützen.

Ausgehend von dem in Kapitel 2.3 beschriebenen hierarchischen Charakter der Produktions-IT Landschaft einer Fabrik lässt sich der Lebenszyklus der gesamten Fabrik in die Lebenszyklen der Teilsysteme auf den tieferen Ebene der Produktions-IT zerlegen. Die im Anschluss diskutierten Lösungsansätze werden demnach durch die betrachtete Ebene der Produktions-IT, die eingesetzten Methoden und die berücksichtigten Lebenszyklen bzw. Lebenszyklusphasen charakterisiert. Abbildung 4.2 zeigt eine Klassifizierung nach den beiden erstgenannten Charakteristika. Die Beschreibung des Standes der Wissenschaft und Technik erfolgt ausgehend von der Anlagenebene hin zur Fabrikebene. Anschließend werden integrierte Lösungsansätze diskutiert und Lösungsansätze aus dem Kreis der Nutzer von SEMI-Standards beleuchtet.

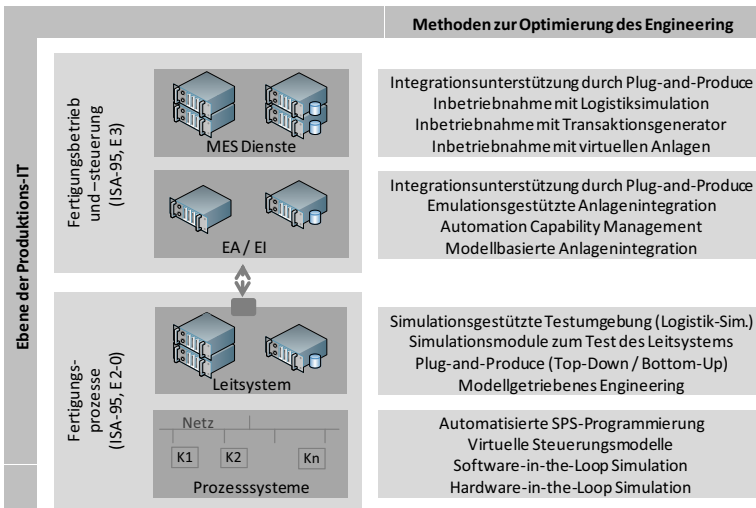


Abbildung 4.2: Klassifizierung der betrachteten Ansätze zur Optimierung der Engineering-Prozesse im Bereich der Produktions-IT

## 4.2 Engineering auf Anlagenebene

### 4.2.1 Engineering mechatronischer Systeme

Eine besondere Herausforderung des Software-Engineering auf der Anlagenebene ist nach [Klemm u. a., 2006] in dessen enger Integration mit dem Engineering mechanischer und elektrischer Komponenten einer Anlage begründet. Die Aufgabenstellung in jeder dieser Disziplinen wird üblicherweise durch Fachexperten der jeweiligen Domäne mit eigenen Werkzeugen und Methoden bearbeitet, die aber letztlich zu einem integrierten Ergebnis führen müssen. Klemm u. a. beschreiben einen integrierten, durchgängigen, baukastenbasierten Engineering-Ansatz für mechatronischen Systeme, der auf Werkzeuge der digitalen Fabrik abbildbar ist. Dieser nutzt Quality-Gates zur Synchronisation der Domänen und setzt für die Kommunikation bzw. das Wissensmanagement zwischen den Domänen und deren Synchronisation umfassend auf Simulationsmethoden. [Klemm u. a., 2006] und [Lohse u. a., 2007] nennen darüber hinaus weitere Zielstellungen für den Einsatz von Simulation im Softwareentwicklungsprozess: Die Beschleunigung der Auftragsabwicklung, die Reduktion der Entwicklungskosten und die virtuelle Inbetriebnahme. Letztere dient als Basis für die Durchführung systematischer Tests, die teure Änderungen zu einem späten Zeitpunkt im Entwicklungsprozess vermeiden sollen. Weitere Zielstellungen sehen sie in der Überprüfung und Optimierung von Steuerungsprogrammen, der Schulung an der entstehenden Software und der Beobachtung bzw. Diagnose der Anlage im Betrieb. [Ammermann, 2007] diskutiert ebenfalls den Einsatz virtueller Steuerungsmodelle in Rahmen von Bedienschulungen.

Wichtige Werkzeuge für diesen und vergleichbare Engineering-Ansätze stellen Umgebungen für die Hardware-in-the-Loop Simulation (HiLS) bzw. Software-in-the-Loop Simulation (SiLS) dar. HiLS-Umgebungen, wie die von [Röck, 2007] beschriebene, lassen sich über den Feld- bzw. Antriebsbus an reale Steuerungshardware anbinden. Dabei interagiert eine virtuelle Maschine oder Anlage, die in der Simulationsumgebung ausgeführt wird, mit der realen Steuerungssoftware, die auf der realen Hardware läuft; die virtuelle Maschine emuliert also das Verhalten der realen Maschine an der realen Schnittstelle im Sinne von Kapitel 2.1. [Amans, 2008] verwendet in diesem Zusammenhang tatsächlich den Begriff der Emulation. Im Fall von SiLS Umgebungen [z. B. Lohse u. a., 2007] wird auf den Einsatz der Hardware der Automatisierungsgeräte, wie Speicherprogrammierbare Steuerung (SPS) oder NC-Steuerungen, verzichtet. Stattdessen wird die reale Steuerungssoftware auf einer virtuellen Steuerung ausgeführt, die mit der Simulationsumgebung gekoppelt ist. Virtuelle Werkzeugmaschinen, die reale Werkzeugmaschinen (serien) abbilden, werden bereits von verschiedenen Werkzeugmaschinenherstellern in Produktform angeboten, [z. B. Gundelach, 2008; Großmann, 2008].

Als Grenzen der beschriebenen Methoden und Werkzeuge nennen [Lohse u. a., 2007] konzeptionelle und technische Grenzen. Eine konzeptionelle Grenze liegt beispielsweise in der Laufzeitbegrenzung für das Simulationsmodell aufgrund der notwendigen Synchronisation mit der Steuerung im Fall der HiLS (siehe dazu auch [Röck, 2007]). Die mangelnden Kopplungsmöglichkeiten der Werkzeuge stellen eine technische Grenze dar. Als signifikante Herausforderung wird der Aspekt der Wirtschaftlichkeit identifiziert: Modellierungsaufwand versus wirtschaftlicher Nutzen.

## 4.2.2 Engineering der Fertigungsleittechnik

Die für das Engineering mechatronischer Systeme beschriebene Grundproblematik setzt sich prinzipiell auf der Ebene der Fertigungsleittechnik fort. Eine ausführliche Diskussion der historischen Entwicklungsstufen des Engineerings von Leitsystemen für *flexible Fertigungssysteme* findet sich z. B. in [Kapitza, 2007].

[Strassacker, 1997] betrachtet die Problemstellung, die sich aus der gleichzeitigen Weiterentwicklung der Leitsystemebene und der Verwendung der Systeme dieser Ebene in Engineering-Aktivitäten ergibt unter Berücksichtigung der Leitsystemebene und der Anlagenebene. Er analysiert die dadurch entstehenden Abhängigkeiten und ihre Folgen, insbesondere die Problematik der Testbarkeit von Softwareänderungen an produktiv genutzten Systemen. Strassacker fokussiert sich auf die Testproblematik im Rahmen des Engineering-Prozesses und schlägt zur Lösung eine Modul- und eine Systemtestumgebung vor, wobei die erstere den unabhängigen Test von Leitsystemmodulen, die zweitere den Test des Leitsystems ohne den Zugriff auf das flexible Fertigungssystem ermöglichen soll. Für die Realisierung der Systemtestumgebung greift Strassacker auf die Kopplung des Leitsystems mit einem zeitdiskreten Simulationsmodell über ein proprietäres Kommunikationsprotokoll zurück.

[Wiechers u. a., 2004] beschreiben am Beispiel des Leitsystems einer Lithographieanlage den Nutzen einer detaillierten Ablaufsimulation der Anlage für den Softwareentwicklungsprozess und heben dabei das Potential der virtuellen Anlage für die Analyse- und Testphase hervor. Das Logistiksystem wird über eine proprietäre Integrationsschnittstelle an das Leitsystem gekoppelt, um die Ablaufsteuerung zu testen.

[Kapitza, 2007] zielt ebenfalls auf die Unterstützung der Testaktivitäten und die virtuellen Inbetriebnahme ab und beschreibt dazu eine Testumgebung, die auf dem Konzept von *Simulationsmodulen* basiert, die das Maschinenverhalten, im Gegensatz zu [Strassacker, 1997], direkt an der Maschinenschnittstelle nachbilden. Dazu kommt die kommerzielle MCIS RPC Rechnerkopplung von Siemens zum Einsatz. Die von Kapitza präsentierte Testumgebung unterstützt die vollautomatische und manuelle Interaktion mit dem Leiterrechner unter Berücksichtigung der folgenden Funktionalitäten: Des Programmtransfers von/zur Maschine, des Werkzeugdatentransfers von der Maschine, des Palettenwechsels (intern oder über ein zentrales Transportsystem), des Austauschs von Palettendaten zwischen Transportsimulation und Maschinensimulation und des NC-Programmablaufs (Start, Ende, Abbruch und Zustandsänderungen von Palette und Maschine).

[Ebel u. a., 2007; Schleipen, 2008] führen die Lebensdauer der Anlagen, die die der produzierten Produkte bei Weitem übersteigt, als Begründung für den Bedarf eines hohen Maßes an Flexibilität von Leitsystemen und deren Anbindung an Anlagenkomponenten an. Um diese Flexibilität zu erhöhen, stellen die Autoren ein integriertes Änderungsmanagement beim Engineering von Leitsystemen vor: Änderungen im Umfeld einer Anlage werden direkt aus dem Werkzeug zur Anlagensimulation an das integrierende Leitsystem kommuniziert. Dabei werden Nachrichten auf Basis des Computer Aided Engineering Exchange (CAEX) Standards eingesetzt. Die Änderungsnachrichten werden für die teilautomatisierte Anbindung der Anlagen an das Leitsystem und die Generierung von Prozessführungsbildern genutzt. Für die Anlagenintegration kommt OPC-UA zum Einsatz. Die Autoren sehen ihre Arbeit als einen Schritt in Richtung der Umsetzung der



Vision der sofortigen Betriebsbereitschaft von Automatisierungssystemen (engl. *plug-and-play*, Technik allgemein; engl. *plug-and-produce*, produktionspezifisch). [Brecher, 2008] beschreiben einen etwas anderen Ansatz dieses Aspekts von *plug-and-produce*, bei dem von Änderungen betroffene Anlagenmodule entsprechende Beschreibungen direkt an die Anlagenvisualisierung liefern.

[Buchner, 2008] analysiert den Engineering-Prozess am Beispiel eines verteilten, agentenbasierten Fertigungsleitsystems für den Einsatz in flexiblen Fertigungssystemen. Er entwickelt auf dieser Basis ein Vorgehensmodell zum virtuellen Engineering von Fertigungsleitsystemen und Werkzeuge zur modellgetriebenen Projektierung und der simulationsgestützten Validierung dieses Systems. Buchner legt einen besonderen Schwerpunkt auf die Ableitung von Testfällen im Zuge des Entwicklungsprozesses, um damit eine systematische Validierung des dezentralen Steuerungssystems, dessen Verhalten nicht notwendiger Weise deterministisch ist, zu unterstützen. Verfahren und Werkzeugkette orientieren sich sehr stark am betrachteten Fertigungsleitsystem.

[Schertl u. a., 2008] liefern zu den Ergebnissen des GMA-Fachausschusses 6.12 “Durchgängiges Engineering von Leitsystemen” einen Bericht über die “Systematische Beurteilung und Verbesserung des Engineerings von automatisierten Anlagen”. Daraus lässt sich zusammenfassend konstatieren, dass der Steigerung der Effizienz des Engineering nach wie vor eine entscheidende Bedeutung zuzumessen ist, dass alle Optimierungsansätze aber den Ist-Zustand und die Randbedingungen, die sich aus dem Kontext der Engineering-Organisation ergeben, berücksichtigen müssen.

## 4.3 Engineering auf Fabrikebene

Die Klassifizierung der Engineering-Ansätze auf Basis der Automatisierungsebene, auf die sie abzielen, ist aufgrund der unterschiedlichen Definition der Produktions-IT, die sie zugrunde legen, nicht immer eindeutig durchführbar. Einige der Arbeiten zum Engineering auf Anlagenebene sind daher, zumindest konzeptionell, auf die Fabrik- und damit auf die MES-Ebene im Sinne von Kapitel 2.3, übertragbar. [Sauer und Ebel, 2007] beschreiben explizit das Potential des oben beschriebenen *plug-and-produce* Konzepts für die MES-Ebene. U. a. nennen sie den Transfer statischer Anlageneigenschaften aus Werkzeugen der Digitalen Fabrik in das Ressourcenmodell der MES-Umgebung bzw. zur Erstellung der Visualisierung auf Fabrikebene.

Neben modellgetriebenen Ansätzen zur Optimierung des MES-Engineering ausgehend vom Datenbestand der digitalen Fabrik wird der Kopplung realer Steuerungssoftware an Simulationsmodelle auch auf der Fabrikebene ein hohes Potential zugeschrieben [z. B. [Schlögl, 2007]]. Einen wichtigen Einsatzbereich für dieses Vorgehen stellt die Untersuchung der Leistungsfähigkeit des Feinplanungs- und Steuerungssubsystems auf der Fabrikebene (siehe Kapitel 2.3) dar. Stellvertretend seien [Zimmermann und Mönch, 2004; Mönch u. a., 2003] genannt, die Anwendungsfälle und Lösungsansätze im Detail diskutieren. [Meier u. a., 2007a, 2008] stellen über die Beschreibung der Integrationsproblematik und des Nutzens für den Test und die virtuelle Inbetriebnahme der Steuerungskomponente eines realen MES hinaus vor, wie der Prozess der Modellentwicklung und dessen

Integration für den Wissenstransfer zwischen Fabrikplanern und MES-Entwicklern genutzt werden kann.

[Jindal und Khandelwal, 2005] stellen einen Ansatz zur Validierung von MES-Lösungen bzgl. der Aspekte Zuverlässigkeit, Verfügbarkeit und Skalierbarkeit im Umfeld hochautomatisierter 300mm Halbleiterfabriken vor. Dazu koppeln sie die realen Softwaresysteme an ein abstraktes Simulationsmodell. Anstelle der Anlagen- und Logistikinfrastuktur bildet das Simulationsmodell das Verhalten der realen Fabrik hinsichtlich der Systemtransaktionen nach, die durch die Fabrik in der MES-Ebene ausgelöst werden. Das Transaktionsverhalten wird aus der Analyse der Historie vergleichbarer Fabriken abgeleitet.

### 4.4 Integrative Ansätze

Viele der zitierten Engineering-Ansätze, die auf den Einsatz von Emulations- und Simulationskonzepten zurückgreifen, zielen auf ausgewählte Ebenen der Produktions-IT ab. Dennoch werden in der Literatur auch integrative Ansätze diskutiert [z. B. Schlögl, 2007]. Eine vielbetrachtete Fragestellung beim Einsatz virtueller Umgebungen ist die Abschätzung zwischen Modellierungsaufwand und erzielter Präzision der Abbildung einerseits und dem wirtschaftlichen Nutzen andererseits [z. B. VDI 3633, 2007]. [Zäh und Wunsch, 2005] beschreiben ein Vorgehensmodell zur Abschätzung von Aufwand und Nutzen für die virtuelle Inbetriebnahme flexibler Fertigungssysteme. Darüber hinaus schlagen sie ein Konzept zur Kopplung eines Materialflussmodells, das sehr abstrakte Anlagenmodelle nutzt, mit detaillierten HiLSs vor, um damit an ausgewählten Punkten des Materialflussmodells den Detaillierungsgrad der Simulation zu erhöhen, wo er für die virtuelle Inbetriebnahme gewinnbringend ist.

### 4.5 Optimierungsansätze für das MES-Engineering im Umfeld der SEMI-Standards

Insbesondere die Phasen des Aufbaus und des Anlaufs im Fabriklebenszyklus geraten im SEMI-Umfeld im Zusammenhang mit dem MES-Engineering seit Jahren immer wieder in den Fokus der Aufmerksamkeit. Der in Kapitel 3.3.2 beschriebene signifikante Aufwand für die Integration von Anlagen führte, neben den in Kapitel 2.4 diskutierten Standardisierungsbemühungen, zur Entwicklung weiterer Optimierungsansätze.

#### 4.5.1 Die emulationsgestützte Anlagenintegration

Der Einsatz von Emulatoren im Zuge der Anlagenintegration entspricht seit vielen Jahren dem Stand der Technik. Entgegen der Definition aus Kapitel 2.1 wird im SEMI-Umfeld häufig der Begriff des “Simulators” für diese Anwendungen verwendet. Softwareentwickler, die mit der Aufgabe der Implementierung einer Integrationslösung für einen Anlagentyp betraut sind, nutzen solche Emulatoren, um ein Verständnis für das Verhalten der Anlagen an der IT-Schnittstelle zu entwickeln und schließlich ihre Integrationslösung

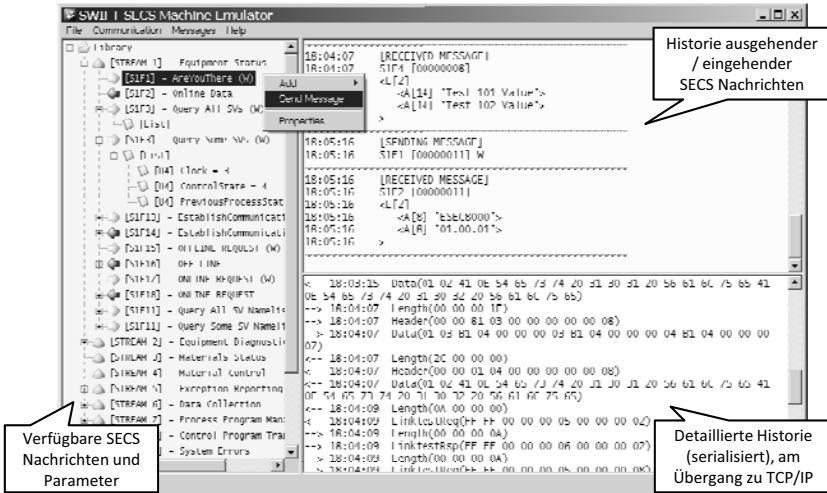


Abbildung 4.3: Ein einfacher Anlagen- bzw. Host-Emulator aus dem SEMI-Umfeld am Beispiel der Freeware SWIFT SECS Machine Emulator [INSPHERE, 2009]

Schritt für Schritt zu testen, ohne ständig den Zugriff auf eine reale Anlage zu benötigen. Auf Seiten des Emulators führt der Anwender das gewünschte Szenario interaktiv mittels einer GUI, wie in Abbildung 4.3 dargestellt, oder automatisch unter Verwendung entsprechender Skripte aus. Zur Nachbildung komplexer Abläufe werden bisweilen im Vorfeld aufgezeichnete Kommunikationshistorien realer Anlage herangezogen, die der Emulator Schritt für Schritt im Zuge der Interaktion mit dem Host-System wiedergibt. Emulatoren entstehen auf sehr unterschiedlichen Wegen: Sie werden von Integratoren auf Basis ihrer Bibliotheken für einen bestimmten Anlagentyp selbst entwickelt, sofern das Verhalten der realen Anlage bekannt ist, werden in Form generischer Emulatoren auf dem Markt angeboten oder direkt vom Anlagenhersteller zur Verfügung gestellt. [z. B. Cimatrix, 2009; INSPHERE, 2009]

## 4.5.2 Automation Capability Management

Anlagensoftware bzw. Implementierungen der IT-Schnittstelle, die nicht den Anforderungen des Kunden entsprechen, tragen in den Phasen von Aufbau und Anlauf aufgrund des hohen Integrationsbedarfs erheblich zum Risiko von Verzögerungen bei [Erdmann und Dreiss, 2006; Dreiss, 2007]. Dieser Tatsache wird im SEMI-Umfeld verbreitet durch die Anwendung des Automation Capability Management (ACM) Rechnung getragen. ACM steht für ein Vorgehensmodell, das die Zusammenarbeit zwischen Anlagenherstellern und Anwendern bzw. Integratoren im Hinblick auf die Automatisierungsfähigkeit der Anlagen vom Zeitpunkt der Anforderungsanalyse bis zur Endabnahme der Anlage

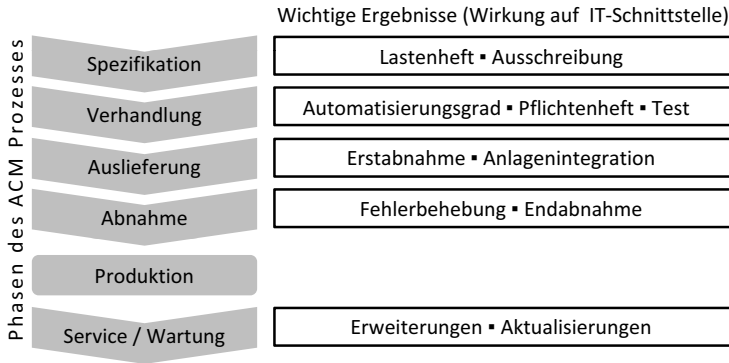


Abbildung 4.4: Phasen und wichtige Ergebnisse des ACM-Vorgehens im Überblick

in der Produktion regelt. ACM kann darüber hinaus auf den gesamten Lebenszyklus der Anlage ausgedehnt werden (siehe Abbildung 4.4). Besondere Schwerpunkte des ACM liegen auf der Erstellung des Lastenhefts (möglichst weitgehend basierend auf den vorhandenen Standards), einer soliden Verankerung des Pflichtenhefts in den abzuschließenden Kaufverträgen und mehrstufigen Test- und Abnahmeprozessen, die nicht erfüllte Anforderungen zum frühestmöglichen Zeitpunkt aufdecken und einer entsprechenden Fehlerkorrektur zuführen sollen [Dreiss, 2007; Meier u. a., 2007b].

Geeignete Testwerkzeuge, die die Überprüfung der Anlagensoftware an ihren IT-Schnittstellen auf Standardkonformität oder auf den Grad der Unterstützung der definierten Betriebsszenarien erlauben, haben unabhängig von der Einbettung in ein formales Vorgehensmodell eine hohe Bedeutung für die erfolgreiche Integration der Anlagen. Für die Durchführung der Konformitätstests von IT-Schnittstellen auf Basis verschiedener SEMI-Standards stehen kommerzielle Sammlungen von Testfällen zur Verfügung. [z. B. Cimatrix, 2009; PEERGROUP, 2009b,a]

Der Vorgang der Analyse des Verhaltens der Anlage aus Sicht ihrer IT-Schnittstellen wird im SEMI-Umfeld auch als *Charakterisierung* bezeichnet. Er liefert wichtige Informationen zu der tatsächlich an der Anlagenschnittstelle verfügbaren Funktionalität und erlaubt eine Einschätzung des zu erwartenden Integrationsaufwands und des mit der Integration verbundenen Risikos.

### 4.5.3 Modellbasierte Anlagenintegration

[Dreiss, 2007] erweitert das Basiskonzept des ACM um eine durchgängige Werkzeugkette unter Berücksichtigung der großen Anzahl von Anlagen (Größenordnung 600 Anlagen) und Anlagentypen in realen Produktionen. Er berücksichtigt insbesondere die Verwaltung der Lastenhefte auf Basis verfügbarer Standards, die Erstellung der Pflichtenhefte in Zusammenarbeit mit den Anlagenherstellern und die Überwachung des mehrstufigen Test und Abnahmeprozesses, inklusive der Behandlung identifizierter Abweichungen von

der Spezifikation. Darüber hinaus schlägt Dreiss ein modellbasiertes Verfahren für die Anlagenintegration vor. Dieses macht sich Datensätze aus dem ACM zunutze, die für jede Anlage beschreiben, in wie weit sie mit ihrer Implementierung der IT-Schnittstelle den Anforderungskatalog des Pflichtenhefts erfüllt. Der Anforderungskatalog wird dazu in Modellen formalisiert, aus denen in einem modellgetriebenen Ansatz die für die EDA-Integration der Anlagen benötigten Treiber generiert werden.

[Erdmann und Dreiss, 2006] berichten von einer erheblichen Optimierung des Integrationsprozesses durch einen werkzeuggestützten ACM-Prozess am Beispiel des Aufbaus einer 300mm Halbleiterfertigung.

### 4.5.4 Das MES in der Interaktion mit Anlagen testen

[Ow u. a., 2006] entwickeln für den Kontext einer ausgewählten Halbleiterproduktion (Scheibenfertigung) eine Testumgebung, die signifikante Änderungen am produktiven MES oder die Migration der Produktionsumgebung von einer MES Lösung zu einer anderen absichern soll. In der Testumgebung sollen insbesondere die Synchronisation paralleler Prozesse, die Robustheit und das Lastverhalten der betrachteten MES-Lösung untersucht werden, ohne die reale Produktion zu stören. Dazu schlagen Ow u. a. die Nutzung einer Simulation (sic.) der Fabrik vor, die das Verhalten der Anlagen an ihrer IT-Schnittstelle mit hoher Detailtreue nachbildet. Die virtuellen Anlagen werden in gleicher Weise wie die realen Pendanten an das MES angebunden. Das Verhalten der virtuellen Anlagen in der Interaktion mit dem MES wird durch das Abspielen nachbearbeiteter Kommunikations-Logdateien realer Anlagen erzeugt, wobei Kontext-Information, wie Carrier-IDs, bei der Wiedergabe ersetzt werden können. Die für die Emulation benötigten Szenarien sind zunächst an der realen Anlage aufzuzeichnen und werden im Rahmen der Emulation immer in derselben Sequenz und unter Verwendung desselben Timings zurückgespielt. Logistische Randbedingungen (durch ein automatisches Transportsystem oder eine anderweitige Verkettung von Anlagen) werden nicht näher betrachtet; darüber hinaus beinhalten Carrier in der Testumgebung beispielsweise immer eine einheitliche, konstante Zahl von Wafern.

## 4.6 Weitere Methoden und Werkzeuge

Im Zuge der Entwicklung des Metamodells der Emulationsplattform und dessen prototypischer Implementierung bzw. der im Anschluss beschriebenen Verfahrensentwicklung werden die folgenden Basiskonzepte verwendet. Sie werden daher überblicksartig eingeführt.

### 4.6.1 Ereignisdiskrete Simulation und ereignisbasierte Datenextraktion

[Page u. a., 2005] stellen am Beispiel von DESMO-J, eines Rahmenwerks zur Modellierung und Simulation ereignisdiskreter Systeme in Java, verschiedene Modellierungs-

und Ausführungskonzepte vor. Dazu gehören u. a. prozessorientierte und ereignisorientierte Ansätze. [Sturm, 2006] beschreibt ein Konzept und dessen Implementierung zur Extraktion von Transportketten in Halbleiterfabriken aus Ereignisdaten, das sowohl im Online-Betrieb als auch im Offline-Betrieb einsetzbar ist. Dabei kommen konfigurierbare Mealy-Zustandsautomaten zum Einsatz.

### 4.6.2 Modellgetriebene Softwareentwicklung

Die durch die Object Management Group (OMG) spezifizierte Model Driven Architecture (MDA) bildet die Grundlage zahlreicher, in der Literatur diskutierter Ausprägungen von Vorgehensweisen zur modellgetriebenen Softwareentwicklung. MDA steht gemäß [OMG, 2003] und [OMG, 2005] für einen Ansatz zur Systementwicklung, dessen Fokus auf der Software dieser Systeme liegt. MDA stellt Hilfsmittel zur Verfügung, um Modelle zur Führung der Aktivitäten der Systementwicklung über den gesamten Lebenszyklus des Systems zu nutzen: Von der Analyse, über das Design, die Implementierung, die Installation und den Betrieb bis hin zu Wartung und Anpassung. Wesentliche Kernkonzepte des MDA-Ansatzes bilden das in Kapitel 2.1 diskutierte Modell bzw. Metamodell und die *Transformation*, die der Überführung eines oder mehrerer Quellmodelle in ein oder mehrere Zielmodelle dient. Zwar stellt die MDA, wie [OMG, 2003] konstatiert, damit lediglich einen weiteren Entwicklungsschritt in der Geschichte der Softwareentwicklung dar. In jedem Fall hat die Einführung zusätzlicher Abstraktionsebenen und die automatische Transformation von Modellen zwischen diesen Abstraktionsebenen in der Geschichte der Softwareentwicklung immer wieder zu wesentlichen Fortschritten bei der Beherrschbarkeit der Komplexität und zur verbesserten Wiederverwendbarkeit einmal erstellter Lösungen geführt.

[OMG, 2003] definiert drei Standpunkte (engl. viewpoint) zu einem zu entwerfenden System, die eine besondere Bedeutung für die MDA haben: Den *Computation Independent Viewpoint*, den *Platform Independent Viewpoint* und den *Platform Specific Viewpoint*. Die im folgenden beschriebenen korrespondierenden Modelle bilden von den genannten Standpunkten aus *Sichten* auf das zu entwerfende System:

**Computation Independent Model (CIM)** wird auch als *Domänenmodell* bezeichnet und beschreibt die fachliche Problemstellung in einer Sprache, die ein Domänenexperte verstehen kann. Diese Sprache ist unabhängig von dem später zur Problemlösung genutzten Rechnersystem.

**Platform Independent Model (PIM)** beschreibt das Lösungskonzept in so weit generalisierter Form, dass die Lösung auf unterschiedlichen Plattformen ähnlichen Typs eingesetzt werden kann, z. B. auf unterschiedlichen Betriebssystemen.

**Platform Specific Model (PSM)** ergänzt das PIM um die Informationen, wie es auf einer spezifischen Zielplattform zum Einsatz kommt.

Diese drei Sichten vom Computation Independent Model bis zum Platform Independent Model bilden eine Modellhierarchie mit abnehmendem Abstraktionsgrad. Die Modelle werden durch Transformation ineinander überführt und ermöglichen damit, wie oben beschrieben, die Führung der Entwicklungsaktivitäten.

Für das Verständnis dieser Arbeit spielen die folgenden OMG-Normen aus dem Umfeld der modellgetriebenen Softwareentwicklung eine Rolle. Sie sind als Werkzeuge zu verstehen, die zur Lösung der beschriebenen Problemstellung genutzt werden:

- Mit der *Meta Object Facility (MOF)* Norm hat die OMG ein Framework spezifiziert, das die formale Beschreibung von Metamodellen erlaubt. Neben MOF selbst sind zahlreiche andere OMG-Technologien auf Basis dieses Frameworks beschrieben, u. a. die Unified Modeling Language (UML). Diese PIMs lassen sich, über Abbildungen, die in der Norm beschrieben sind, in verschiedenste PSMs überführen. Eine Möglichkeit stellen XML Metadata Interchange (XMI) Repräsentationen dar, die zum Austausch zwischen unterschiedlichen Modellierungswerkzeugen dienen. [OMG, 2006]
- Die *Unified Modeling Language (UML)* ist eine visuelle Modellierungssprache, die ursprünglich aus mehreren objektorientierten Methoden hervorgegangen ist, sich zwischenzeitlich zur am weitesten verbreiteten Modellierungssprache in der Softwareindustrie entwickelt hat und für vielfältige Anwendungsgebiete einsetzbar ist [OMG-UML-IS, 25.02.2009]. Sowohl PIMs als auch PSMs lassen sich auf Basis der UML beschreiben. Die UML-Version, die auch an verschiedenen Stellen in dieser Arbeit zur Anwendung kommt, ist die UML 2.0, spezifiziert durch die UML Infrastructure Version 2.2 [OMG-UML-IS, 25.02.2009] und die UML Superstructure Version 2.2 [OMG-UML-SS, 09.03.2009]. *Profile* erlauben die Anpassung des Meta-Modells an spezifische Plattformen oder Domänen. Sofern nicht anders angegeben, sind die abgedruckten UML-Diagramme im Sinne dieser Spezifikation zu verstehen. Eine weniger formale Einführung in die UML bietet z. B. [Rupp u. a., 2007]. [Kraemer und Herrmann, 2006] stellen einen Ansatz zur Spezifikation reaktiver Dienste auf Basis von UML 2.0 Kollaborationen und Aktivitätsdiagrammen vor, der eine Transformationskette bis zum ausführbaren Code umfasst.

## 4.7 Fazit

Die Darstellung der Ausgangssituation in Kapitel 2 und die Analyse der Produktions-IT in der PV-Industrie in Kapitel 3.1 verdeutlichen die hohe und weiter wachsende Bedeutung, die IT-Systeme in aktuellen Produktionssystemen im Allgemeinen und in der PV-Industrie im Speziellen in der industriellen Praxis haben. Das Spannungsfeld zwischen den Möglichkeiten, die der verstärkte Einsatz von IT-Systemen verspricht, und dem Zusatzaufwand und den Problemen, die vielfach mit dem verstärkten IT-Einsatz im Produktionsumfeld einhergehen (siehe Kapitel 1.2 und 2.5), hat bis zum heutigen Tag zu vielfältigen Lösungsvorschlägen geführt, die dieses zugunsten der Möglichkeiten des IT-Einsatzes auflösen sollen.

In der Diskussion des Standes der Wissenschaft und Technik wurden, strukturiert durch die Ebenen der Produktions-IT (siehe Kapitel 2.3), Engineering-Methoden und Werkzeuge betrachtet, die durch die Optimierung des Engineering-Prozesses zur Auflösung des beschriebenen Spannungsfeldes beitragen sollen. Die virtuelle Inbetriebnahme wurde als eine der wesentlichen Methoden identifiziert, die auf allen Ebenen genutzt wird.

Diese greift auf Werkzeuge zurück, die eine virtuelle Abbildung relevanter Ausschnitte Produktionssysteme erlauben. Aufgrund der unterschiedlichen Anforderungen und Randbedingungen werden auf den unterschiedlichen Ebenen der Produktions-IT unterschiedliche Werkzeuge eingesetzt. Angefangen von HiLS-Umgebungen, die die reale Steuerungshardware in eine Simulationsumgebung einbinden, über SiLS-Systeme bis hin zu Testumgebungen, die das Verhalten von Anlagen oder Fabrikbereichen auf höherer Ebene nachbilden.

Die virtuelle Inbetriebnahme von MES-Lösungen für die PV-Industrie stellt hohe Anforderungen an Werkzeuge zur Modellierung und zum Betrieb der virtuellen Fabrikumgebung. Es sei auf die Anforderungen an die Skalierbarkeit bis zu einer sehr großen Anzahl von Anlagen (Größenordnung  $10^3$ ) bei ausreichenden Möglichkeiten zur Überwachung und Steuerung, die Anforderungen an Parallelität und Synchronisation, die Vielfalt der genutzten Produktions- und Logistiksysteme, die Unabhängigkeit vom verwendeten MES-Produkt bzw. Architekturkonzept und die unterschiedlichen IT-Schnittstellen verwiesen. Insbesondere muss das Werkzeug im Umfeld der PV-Industrie die wirtschaftliche Modellerstellung und -pflege erlauben. Ein Werkzeug, das die identifizierten Anforderungen zufriedenstellend erfüllt, konnte nicht gefunden werden. Vor diesem Hintergrund wird im folgenden Kapitel die Entwicklung einer Emulationsplattform diskutiert, die diese Anforderungen erfüllt.

Neben der Methodik der virtuellen Inbetriebnahme wird der Optimierung der Vorgehensweise im Engineering-Prozess und der Entwicklung geeigneter Werkzeugketten zur Unterstützung des Engineering-Prozesses hohes Potential zugemessen. Dabei sind insbesondere die Basisansätze top-down, dessen Ziel die Weiterverwendung einmal im Engineeringprozess erhobener Daten in späteren Prozessphasen ist, und bottom-up, dessen Ziel die Optimierung des Engineeringprozesses von den späten Phasen her kommend ist, zu unterscheiden. Vertreter der ersten Gattung sind z.B. die vorgestellten Konzepte zur Kopplung des MES-Engineering an die Digitale Fabrik oder der modellgetriebene Ansatz zur Anlagenintegration. Vertreter der zweiten Gattung sind z.B. das "plug-and-produce" Konzept, das auf der Selbstbeschreibung der Fähigkeiten der Anlage beruht oder die emulationsgestützte Anbindung von Anlagen. In jedem Fall muss der Engineering-Prozess den Ist-Zustand der PV-Industrie und die Randbedingungen der typischen Engineering-Organisation berücksichtigen. Unabhängig vom gewählten Produkt ist die Einführung eines MES in einem Produktionssystem der PV-Industrie aufgrund der vielfältigen Ausprägungen dieser Produktionssysteme mit einem signifikanten Engineering-Anteil verbunden. Eine wirtschaftliche Nutzung der virtuellen Inbetriebnahme in der PV-Industrie ist nur im Rahmen eines integrierten Engineering-Prozesses denkbar, der die Methodik der virtuellen Inbetriebnahme, unabhängig vom genutzten Produkt, nahtlos in den MES-Engineeringprozess integriert und dabei die Randbedingungen der PV-Industrie berücksichtigt. Diese Anforderungen an ein holistisches Vorgehen werden durch die verfügbaren Verfahren nicht zufriedenstellend erfüllt. Daher wird in Kapitel 6 ein Verfahren entwickelt, das die Emulationsplattform in den MES-Engineeringprozess integriert.



# 5 Metamodell und Implementierung einer Emulationsplattform

Die folgenden Abschnitte bieten mehrere Sichten auf die Emulationsplattform. Ausgehend von der Strukturierung des Systems durch das Architekturkonzept wird das Metamodell der Emulationsplattform entwickelt. Zunächst werden wichtige Elemente des statischen Modells im Detail betrachtet. Das dynamische Modell ergänzt die statische Sicht um relevante Laufzeitaspekte. Das Ergebnis wird anhand der Betrachtung einiger Anwendungsfälle überprüft. Die Beschreibung der prototypischen Implementierung geht schließlich auf Fragestellungen ein, die sich im Zuge der prototypischen Umsetzung des Konzepts ergeben.

## 5.1 Entwurf eines Architekturkonzepts

In der Diskussion des Architekturkonzepts wird wiederholt der Begriff des *Dienstes* verwendet. Ein Dienst beschreibt eine definierte Funktionalität, die ein System über eine Schnittstelle nach außen anbietet. Die Implementierung des Dienstes ist, transparent für den Nutzer des Dienstes, austauschbar. Zunächst werden die in Kapitel 3 ermittelten Anforderungen an eine skalierbare Emulationsplattform in ein abstraktes Architekturmodell überführt, das aus der Problemstellung disjunkte Verantwortungsbereiche aufgreift und diese im späteren Verlauf interagierender Teilsystemen zuordnet. Dabei ergeben sich die folgenden Verantwortungsbereiche:

**Ausführungsumgebung.** *Emulatoren*, die in der virtuellen Fabrik das Verhalten realer Anlagen nachbilden und im folgenden auch synonym als *virtuelle Anlagen* bezeichnet werden, werden zur Laufzeit innerhalb der Ausführungsumgebung instanziiert und ausgeführt. Die Ausführungsumgebung übernimmt dabei das Management des Lebenszyklus von Emulatoren aus Sicht der Emulationsplattform und deren Anbindung an die Plattformdienste.

**Plattformdienste.** Verschiedene Anforderungen legen die Umsetzung der in Kapitel 3.5 beschriebenen virtuellen Fabrik auf Basis einer Gruppe von Infrastrukturdiensten nahe. Das sind insbesondere die Anforderung zur Unterstützung von Wiederverwendung und die geforderten Mechanismen zur Zustandsverwaltung und zur Koordination von Szenarien. Die Schicht der Plattformdienste wird sowohl interaktiv von Benutzern als auch programmatisch von der Ausführungsumgebung genutzt, die Dienste zur Koordination, Überwachung und Auswertung des Betriebs benötigt. Alle Dienstanbieter greifen über geeignete Kommunikationsmechanismen auf die Dienstanbieter zu.

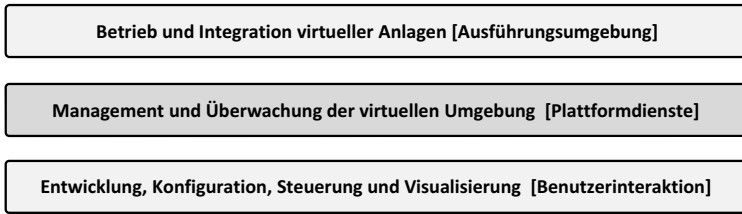


Abbildung 5.1: Black-Box-Darstellung des 3-schichtigen Architekturkonzepts der Emulationsplattform

**Benutzerinteraktion.** Für zahlreiche Anwendungsfälle der Emulationsplattform beim MES-Engineering spielen die Möglichkeiten zur Interaktion mit menschlichen Benutzern eine wichtige Rolle. Besonders hervorzuheben sind die Anwendungsfälle 'Emulatorentwicklung', 'Modellierung', 'Konfiguration' und 'Verwaltung virtueller Fabriken und Szenarien', und die Anwendungsfälle 'Steuerung', 'Überwachung' und 'Auswertung von Experimenten' zur Laufzeit. Ressourcen und Dienste, die von der Emulationsplattform angeboten werden, könnten zeitgleich von mehreren Benutzern in unterschiedlichem Kontext angefordert werden.

Diese Verantwortungsbereiche werden durch das in Abbildung 5.1 dargestellte *3-schichtige* Architekturkonzept (engl. *3-tier*) abgebildet. Jede Schicht bietet Dienste an, die durch andere Dienste oder Anwender innerhalb oder außerhalb des Systems genutzt werden können und damit einen Beitrag zur Erfüllung der Anforderungen leisten. Implementierungen der Schichten könnten gegen andere ausgetauscht werden. Innerhalb jeder Schicht ist eine Zerlegung in weitere Schichten möglich, die auf dieser Abstraktionsebene jedoch nicht sichtbar ist.

### 5.1.1 Modellierung versus Ausführung

Für die Betrachtung der Architektur und der Detailkonzepte sind zwei Hauptkategorien von Anwendungsfällen zu unterscheiden:

**Die Modellierungszeit (engl. build time)** gruppiert die Anwendungsfälle, die im Zusammenhang der Erstellung und Verwaltung virtueller Fabriken zu sehen sind. Wesentliche Architekturtreiber aus diesen Anwendungsfällen sind die Anforderungen an die Wiederverwendbarkeit von Modellelementen und die Berücksichtigung des für die Modellbildung notwendigen Aufwands (siehe Abbildungen 3.8 und 3.9).

**Die Ausführungszeit (engl. runtime)** fasst die Anwendungsfälle zusammen, in denen virtuelle Fabriken in der Interaktion mit Softwaresystemen der MES-Ebene genutzt werden. Für diese Kategorie wird, soweit die Bedeutung an der verwendeten Stelle klar ist, synonym der Begriff der *Laufzeit* verwendet. Wesentliche Architekturtreiber aus diesen Anwendungsfällen sind die geforderten Analyse- und Interaktionsmöglichkeiten des Anwenders und die geforderte hohe Skalierbarkeit (siehe Abbildung 3.9).

## 5.1.2 Die Schicht der Ausführungsumgebung

Als charakteristische Architekturtreiber aus den Anforderungen an die Ausführungsumgebung werden die Anforderungen der Integrierbarkeit virtueller Anlagen, die auf Basis unterschiedlicher Technologien implementiert sind, und das geforderte Skalierungsvermögen der Emulationsplattform im Hinblick auf die Anzahl parallel betreibbarer virtueller Anlagen identifiziert. Beide legen die Umsetzung der Ausführungsplattform als *verteiltes System* nahe.

Für die Organisation verteilter Systeme, die einen Ressourcenpool zur Verfügung stellen, um Dienste hochgradig skalierend anbieten zu können, sind aus der Informatik zahlreiche mögliche Formen bekannt. Dazu gehören die Organisationen des Clusters, des Grids oder der Cloud<sup>1</sup>. Jedes Organisationskonzept stellt eigene Anforderungen an die IT-Infrastruktur, auf der die Lösung des Problems implementiert wird. Die Entscheidung für das bestgeeignete Organisationskonzept leitet sich aus spezifischen Eigenschaften und Zielstellungen des zu verteilenden Problems ab, in diesem Fall des verteilten Betriebs einer virtuellen Fabrik:

- Eine virtuelle Fabrik ist als *föderiertes System* virtueller Anlagen zu betrachten; sie bildet also einen Zusammenschluss selbständiger virtueller Anlagen, für deren Implementierung unterschiedlichste Konzepte und Basistechnologien genutzt werden können.
- Virtuelle Anlagen sind ausschließlich über die in Kapitel 3.5 identifizierten Schnittstellen gekoppelt.
- Jede virtuelle Anlage benötigt Zugriff auf ausreichende Rechnerressourcen, insbesondere auf Prozessor, Speicher und Netzanbindung. Der tatsächliche Ressourcenbedarf einer konkreten virtuellen Anlage wird bestimmt durch den Ressourcenbedarf, der zur Nachbildung des Verhaltens der Anlage an ihrer IT-Schnittstelle im gewünschten Detaillierungsgrad notwendig ist.
- Die Emulationsplattform wird innerhalb einer definierten Organisation eingesetzt, die über ein Rechnernetz hoher Bandbreite verfügt, d.h. Emulationsplattform und gekoppelte IT-Systeme werden in einem eigenen Netz betrieben. Das Konzept virtueller Organisationen wird nicht berücksichtigt.

Aufgrund dieser Eigenschaften wird für die Emulationsplattform ein Organisationskonzept auf Basis von *Agenten* gewählt (siehe Abbildung 5.2). Ein Agent bezeichnet dabei eine aktive Softwarekomponente, die aus einer beliebigen Anzahl von Bausteinen besteht, über einen eigenen Ausführungsstrang verfügt mit dem sie Interaktionen mit anderen Komponenten initiieren kann und wird auf einem Rechnersystem zur Ausführung gebracht. Das Rechnersystem wird als *Knoten* bezeichnet und stellt dem Agenten die Ressourcen zur Verfügung, die er benötigt.

Entsprechend der Definition der UML (siehe Kapitel 4.6.2) stellen Komponenten eigenständige Einheiten dar, die über wohldefinierte Schnittstellen mit der Außenwelt

<sup>1</sup>Einen guten Überblick über diese Organisationskonzepte bieten in der Reihenfolge ihrer Nennung [Baker, 2001], [Foster und Kesselman, 2007] und [Foster, 2008], wobei letzterer die konzeptionellen Zusammenhänge zwischen Grid und Cloud diskutiert.

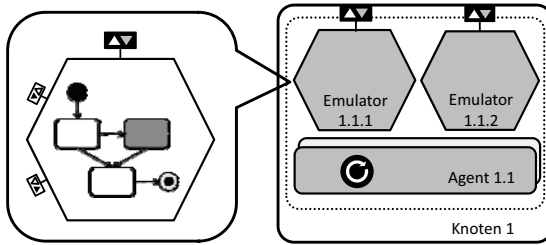


Abbildung 5.2: Konzept: Virtuelle Anlage und Ausführungsumgebung

interagieren und innerhalb ihrer Umgebung durch andere Implementierungen ersetzt werden können, die dieselben Schnittstellen anbieten. Emulatoren werden als Komponenten realisiert und, wie in Abbildung 5.2 dargestellt, mit Hilfe des Agenten in die Infrastruktur der Emulationsplattform eingebunden. Sie stellen daher nicht nur logische Komponenten dar, die zusammengehörige Funktionalitäten in einer Einheit bündeln, sondern auch physikalische Komponenten, die durch unabhängig verteilbare Artefakte (hier: Installationsbündel) manifestiert werden. Der Agent bietet damit die standardisierte Umgebung an, in der Emulatoren installiert und ausgeführt werden können – er übernimmt die Funktion eines so genannten *Containers*. Das System unterstützt sowohl den Betrieb mehrerer Agenten auf einem physikalischen Knoten als auch die Installation mehrerer Emulatoren auf einem Agenten.

Mit der Abbildung von Emulatoren auf Komponenten wird die Wiederverwendung auf der Ebene der Emulatoren unterstützt. Desweiteren sind Emulatoren dadurch flexibel auf der Emulationsplattform verteilbar und unabhängig voneinander betreibbar. Abbildung 5.3 zeigt eine schematische Black-Box-Darstellung der Komponente. Für die Interaktion der Komponente mit der Umgebung werden vier komplexe Ports definiert, die jeweils durch mehrere Schnittstellen spezifiziert werden:

- Der Port 'Host-Schnittstelle' dient der Kommunikation mit dem MES-System. Der Emulator bildet das Verhalten des realen Automatisierungssystems an diesem Port ab.
- Der Port 'Logistik' bindet den Emulator in das Logistiksystem der virtuellen Fabrik ein. Über diesen Port werden dem Emulator virtuelle Repräsentationen von Logistikobjekten zu- bzw. vom Emulator abgeführt, die für den Betrieb der virtuellen Anlage notwendig sind.
- Der Port 'Horizontale Kommunikation' ermöglicht die Kommunikation zwischen Emulatoren analog zu dieser Schnittstelle bei realen Anlagen.
- Der Port 'Management' wird durch den Container genutzt, um den Lebenszyklus der Komponente zu steuern, gewährt gleichzeitig dem Emulator Zugang zu den Diensten der Plattform und der Ausführungsumgebung und ermöglicht der Plattform die Fern-Koordination der Emulation.

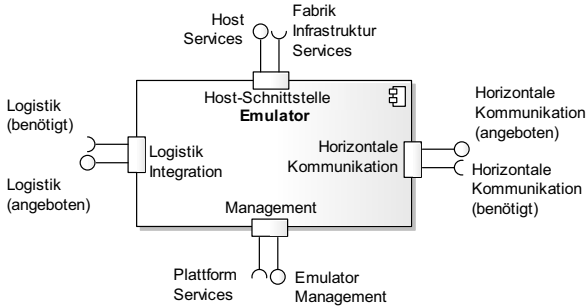


Abbildung 5.3: Der Emulator als Komponente – Black-Box-Darstellung

Diesen standardisierten Ports können bei Bedarf weitere hinzugefügt werden, um zusätzliche domänenspezifische Schnittstellen abzubilden.

### 5.1.3 Die Schicht der Plattformdienste

Die Schicht der Plattformdienste stellt, komplementär zur Ausführungsumgebung, die Infrastruktur zur Verfügung, die eine Menge unabhängiger, parallel betreibbarer Emulatoren zu virtuellen Fabriken integriert und die Modellierung und Nutzung dieser virtuellen Fabriken für konkrete Anwendungsfälle ermöglicht. Die folgenden Dienste spielen dabei eine Schlüsselrolle.

#### Das Repository

Der Dienst 'Repository' verwaltet die Modellelemente aus denen virtuelle Fabriken aufgebaut werden. Die Aufgabenstellung der Verwaltung umfasst insbesondere Funktionalitäten zur dauerhaften Speicherung von Modellelementen, zur Suche nach verfügbaren Modellelementen anhand ihrer Metadaten und zur Bearbeitung bzw. zum lesenden Zugriff auf Modellelemente. Er bietet damit einen Teil der Basisfunktionalität zur Modellierung virtueller Fabriken und zu deren Überführung in die Ausführungsumgebung. Genutzt wird der Dienst von Agenten der Ausführungsumgebung und Anwendern, die über die Benutzerinteraktionsschicht mit dem Repository kommunizieren. Das Design des Repository-Dienstes ermöglicht die parallele Bearbeitung von Anfragen und skaliert im geforderten Umfang, der in erster Linie von der maximalen Anlagenzahl der auszuführenden virtuellen Fabriken bestimmt wird.

#### Mechanismen zur Kommunikation

Für die Interaktion der Teilsysteme der Plattform stellt die Infrastruktur zwei Basismechanismen zur Verfügung, die für den Austausch von Nachrichten zwischen kommu-

nizierenden Systemen genutzt werden: *Synchrone* und *asynchrone* (vgl. Kapitel 4.6.2, UML-Sequenzdiagramme). Während die synchronen Mechanismen den Initiator der Interaktion so lange blockieren, bis eine Antwort auf die Anfrage zur Verfügung steht, erfolgt im Fall der Nutzung der asynchronen Mechanismen die Bearbeitung der Anfrage in entkoppelter Form: Der Sender setzt unmittelbar nach dem Versenden einer Nachricht seinen Bearbeitungsstrang fort. Falls die Anfrage eine Antwort erfordert, erhält der Initiator diese zu einem späteren Zeitpunkt und muss diese zur ursprünglichen Anfrage korrelieren. Während die synchronen Mechanismen für *Punkt-zu-Punkt*-Kommunikationsszenarien relevant sind, also für den Fall, dass die betrachtete Interaktion aus zwei Partnern besteht, werden die asynchronen Mechanismen auch in *Punkt-zu-Mehrpunkt*-Kommunikationsszenarien eingesetzt. Der Zugriff eines Subsystems der Benutzerinteraktionsschicht auf den Repository-Dienst wird beispielsweise über den synchronen Mechanismus abgewickelt.

Punkt-zu-Mehrpunkt-Szenarien spielen für die Koordination virtueller Fabriken eine wichtige Rolle, um Nachrichten, die von einem Koordinationsdienst generiert werden und von allen Emulatoren der virtuellen Fabrik verarbeitet werden müssen, diesen Empfängern zuzustellen. Das Kommunikationssystem<sup>2</sup>, das durch die Infrastrukturschicht zur Verfügung gestellt wird, muss den Skalierungsanforderungen genügen, die für die Emulationsplattform definiert wurden. Das bedeutet, es muss eine ausreichende Zahl von Nachrichten einer gegebenen durchschnittlichen Größe innerhalb eines definierten Zeitraums an eine Zahl von Emulatoren verteilen können, die sich innerhalb des durch die Anforderungen spezifizierten Bereichs bewegt. Gleichzeitig muss die *Dienstgüte*, die das Kommunikationssystem für den Versand der Nachrichten zur Verfügung stellt, der Problemstellung angemessen sein. Die Dienstgüte charakterisiert insbesondere die Zuverlässigkeit, mit der Nachrichten, die von einem Sender asynchron an das Kommunikationssystem übergeben wurden, bei allen Empfängern ausgeliefert werden. Für den betrachteten Anwendungsfall wird ein Maß an Zuverlässigkeit gefordert, das die Zustellung von Nachrichten, die dem Kommunikationssystem zur Verteilung übergeben wurden, an alle Empfänger garantiert. Darüber hinaus sind die Nachrichten allen Empfängern in derselben Reihenfolge zuzustellen, d. h. die Auslieferungsreihenfolge der Nachrichten eines ausgewählten Senders an seine Empfänger muss der entsprechen, mit der sie versendet wurden. Die verfügbare Dienstgüte hat einen wesentlichen Einfluss bei der Koordination der Abläufe zur Laufzeit, die in der Folge noch diskutiert wird.

Für die Integration von Sender (z. B. dem Koordinationsdienst) und Empfängern (z. B. den Emulatoren einer virtuellen Fabrik) wird eine *lose Kopplung* angestrebt. Das bedeutet, dass der Sender weder eine bestimmte Anzahl von Empfängern noch eine besondere Ausprägung der Empfänger voraussetzt. Aus diesem Grund wird das Architekturmuster *publish/subscribe* für die Anbindung der Empfänger an die Sender gewählt. Dieses Muster beschreibt ein Konzept, demzufolge der Sender Nachrichten auf einem spezifischen Kanal veröffentlicht (engl. *publish*). Empfänger, die sich für Nachrichten des Senders interessieren, registrieren (engl. *subscribe*) sich als Empfänger für diesen Kanal und erhalten von diesem Zeitpunkt an die über diesen Kanal versendeten Nachrichten. *Filter* ermöglichen dem Empfänger, anhand von Regeln eine Untermenge von Nachrichten zu spezifizieren, die ihm für die weitere Verarbeitung zugeführt werden sollen.

---

<sup>2</sup>Einen tieferen Einblick in die verwendeten Basiskonzepte bietet z. B. [Sun, 2002].

## Management und Überwachung zur Laufzeit

Zur Laufzeit zeichnet die Schicht der Plattformdienste insbesondere für die Verwaltung des Lebenszyklus virtueller Fabriken und damit die Koordination und Integration des föderierten Systems von Emulatoren zum Gesamtsystem der virtuellen Fabrik verantwortlich. Der Lebenszyklus einer virtuellen Fabrik aggregiert den Lebenszyklus aller eingebundenen Modellelemente von der Installation der benötigten Komponenten in der Ausführungsumgebung bis zu ihrer Deinstallation. Gültige Zustände und Zustandsübergänge werden als dynamische Aspekte der verwendeten Modellelemente definiert. Ereignisse im verteilten System, die Zustandswechsel auslösen und über Zustandswechsel informieren, werden asynchron über das Kommunikationssystem mit den oben beschriebenen Eigenschaften verteilt. Die Koordinationsmechanismen der Plattform erlauben definierte Eingriffe in die Abläufe virtueller Fabriken, um alle beteiligten virtuellen Anlagen und die virtuelle Fabrikumgebung konsistent zu initialisieren, zu starten, anzuhalten, Abbilder des Gesamtzustandes zu speichern, die zu einem späteren Zeitpunkt zur Initialisierung genutzt werden können, und den Betrieb abzubrechen. Dabei sorgen die Koordinationsmechanismen für das erforderliche Maß an Synchronisation aller parallelen Prozesse.

Zur Überwachung der im verteilten System ablaufenden Prozesse stellt die Plattform Log-Mechanismen zur Verfügung, die die a-posteriori-Analyse der im System aufgetretenen Ereignisse und die Visualisierung der Abläufe unterstützen. Zusätzlich vermittelt ein Plattformdienst Punkt-zu-Punkt-Verbindungen zu ausgewählten Agenten, über die die Zustände einzelner Komponenten, die auf dem Agenten ausgeführt werden, beobachtet und interaktiv verändert werden können.

### 5.1.4 Die Schicht der Benutzerinteraktion

Bei der Interaktion von Benutzern mit der Emulationsplattform tritt die in Kapitel 5.1.1 beschriebene Differenzierung zwischen Modellierungszeit und Ausführungszeit deutlich zutage. Die Benutzerinteraktionsschicht stellt Werkzeuge zur Verfügung, die in der Interaktion mit den Plattformdiensten bzw. der Ausführungsumgebung beide Kategorien von Anwendungsfällen unterstützen. Neben plattformeigenen grafischen Benutzeroberflächen bietet die Schicht Schnittstellen, sogenannte APIs an, über die Werkzeuge anderer Anbieter, wie Entwicklungsumgebungen oder Testmanagement-Werkzeuge, mit der Plattform integriert werden können.

### 5.1.5 Zusammenfassung: Architekturkonzept

Abbildung 5.4 fasst den Entwurf des Architekturkonzepts in einer vereinfachten White-Box-Darstellung zusammen, die die wichtigsten Aspekte im Hinblick auf die Strukturierung des Systems beinhaltet. Die folgenden Abschnitte beschreiben die Konzepte, die in den einzelnen Schichten und Diensten zur Anwendung kommen, im Detail.

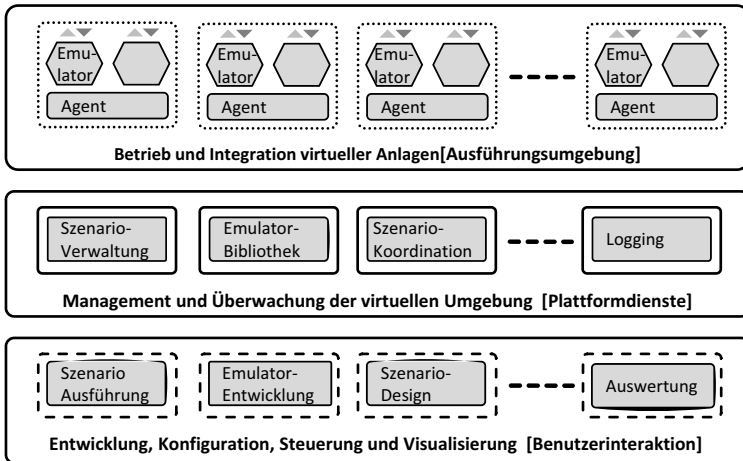


Abbildung 5.4: Vereinfachte White-Box-Darstellung des 3-schichtigen Architekturkonzepts

## 5.2 Das statische Modell

Zur Beschreibung der statischen Zusammenhänge der realen Fabrik in der virtuelle Welt der Emulationsplattform stellt deren Metamodell einige fundamentale Primitive (auch als Bausteine oder Modellelemente bezeichnet) zur Verfügung, die in den folgenden Abschnitten spezifiziert werden. Die Entwicklung der Primitive geht von einer virtuellen Anlage aus, die schrittweise unter Verwendung der virtuellen Anlagenumgebung hin zur virtuellen Fabrik ausgebaut wird. Zur formalisierten Darstellung der statischen Zusammenhänge werden UML (siehe Kapitel 4.6.2) Klassendiagramme benutzt. Abbildung 5.5 verdeutlicht den Zusammenhang zwischen den Ebenen des Metamodells, des Modells und der Instanz einer virtuellen Fabrik: Das Metamodell stellt das Rahmenwerk der Primitive zur Verfügung, das zur Modellierung virtueller Fabriken genutzt werden kann. Der Anwender bildet reale Fabriken in einem Modell ab, das diese unter Verwendung der angebotenen Primitive im gewünschten Detaillierungsgrad beschreibt. Teile des Modells werden durch Werkzeuge der Emulationsplattform in Modelle transformiert, die in der Ausführungsumgebung instanziiert und ausgeführt werden können. Die übrigen Teile des Modells werden von der Ausführungsumgebung zur Laufzeit interpretiert.

Zur besseren Verständlichkeit wird das Modell in vier Teilmodelle aufgeteilt, die in ihrer Gesamtheit die virtuelle Fabrik beschreiben:

**Das Emulatortypmodell (ETM)** beschreibt eine Menge von Emulatoren in einem Detaillierungsgrad, der die Abbildung des Verhaltens vergleichbarer Systeme bzw. Teilsysteme im Sinne von Kapitel 2.1.2 an ihren IT-Schnittstellen erlaubt.



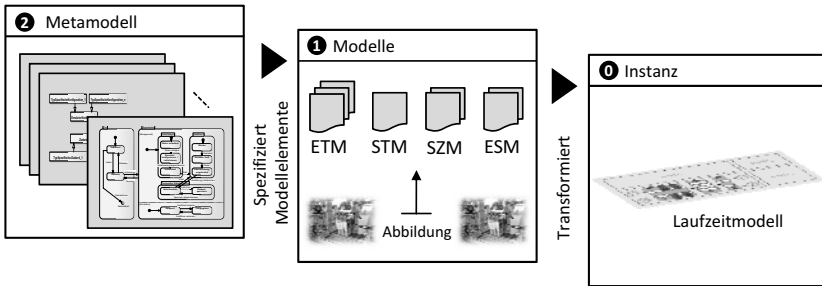


Abbildung 5.5: Zusammenhang Metamodell – Modelle – Instanz

**Das Strukturmodell (STM)** integriert virtuelle Anlagen zur virtuellen Fabrik. Dabei kommen Emulatortypmodelle aus dem Repository zum Einsatz, die durch das Strukturmodell für die Modellinstanz parametrierbar werden.

**Das Einsatzmodell (ESM)** bildet die im Strukturmodell spezifizierten Emulatoren auf die in der Laufzeitumgebung verfügbaren Agenten ab und berücksichtigt dabei den Ressourcenbedarf des Emulators und die Randbedingungen der Anbindung der MES-Ebene.

**Das Szenarienmodell (SZM)** beschreibt alle für eine virtuelle Fabrik definierten, benannten Konfigurationssätze, die im Betrieb der virtuellen Fabrik aufgerufen werden können.

Bei der Diskussion der Primitive wird jeweils auf diese Teilmodelle verwiesen. Zu den Teilmodellen korrespondierende Laufzeitaspekte werden im folgenden Kapitel zum dynamischen Modell ergänzt.

### 5.2.1 Die virtuelle Anlage

Der *Emulator* stellt das zentrale Modellelement zur virtuellen Abbildung von Anlagen dar. Er repräsentiert eine Menge realer Automatisierungssysteme bzw. -teilsysteme im Sinne von Kapitel 2.1.2, die aus Sicht ihrer IT-Schnittstellen *vergleichbar* sind. Vergleichbar bedeutet hier, dass sich die Automatisierungsszenarien (siehe Kapitel 3.5.1) dieser Systeme lediglich in ihrer Parametrierung unterscheiden. Ein Emulator kann dadurch zur Beschreibung beliebig vieler Instanzen vergleichbarer Automatisierungssysteme herangezogen werden. Instanzen eines Automatisierungssystems werden in der virtuellen Fabrik durch jeweils einen *Emulator* repräsentiert, der seine für die Abbildung relevanten instanzspezifischen Eigenschaften umfasst und so einen Emulator parametrierbar. Auf diese Weise wird, zusammen mit der Abbildung von Emulatoren auf Komponenten, die Anforderung der Wiederverwendbarkeit von Modellen auf der Anlagenebene umgesetzt, wo sie für die praktische Nutzung besonders wichtig ist. Unabhängig vom Detaillierungsgrad, der für die Lösung einer konkreten Aufgabenstellung mit Hilfe der

virtuellen Fabrik benötigt wird, wird die Strategie verfolgt, die abzubildenden Automatisierungssysteme auf eine möglichst kleine Menge hinreichend generischer Emulatorarten abzubilden, um den Aufwand der Modellbildung zu reduzieren. Darüber hinaus erlaubt die konzeptionelle Trennung von Emulator und Emulatorart den einfachen Austausch der Implementierung der virtuellen Anlage, um bei Bedarf den Detaillierungsgrad für einen ausgewählten Emulator zu erhöhen. Abbildung 5.6 zeigt diese grundlegenden Zusammenhänge auf.

Neben der instanzspezifischen Parametrierung werden Emulatorarten durch *Emulator-Konfigurationen* konfiguriert. Allein die IT-Schnittstellen, die in den Kapiteln 2.4.2 und 3.5.1 vorgestellt wurden, erfordern für jedes reale Automatisierungssystem zahlreiche Einstellungen, die sich in entsprechender Weise in der virtuellen Repräsentation wiederfinden müssen. Die Menge verfügbarer Konfigurationsoptionen und die zur Konfiguration verwendeten Technologien werden durch den Emulatorart vorgegeben. Konfigurationen sind aufgrund dieser Definition nicht notwendigerweise durch die Emulationsplattform interpretierbar, können also proprietärer Natur sein. Über die Definition einer generischen Konfiguration seitens der Plattform, die durch die Implementierung von Emulatorarten genutzt werden kann, können generische Werkzeuge zur Anpassung der Konfigurationen durch die Plattform zur Verfügung gestellt werden.

In ähnlicher Weise wird die Repräsentation des *Zustands* durch den Emulatorart definiert. Die Zustandsrepräsentation eines Emulatorart umfasst die Zustände aller seiner Teilsysteme zu einem gegebenen Zeitpunkt  $t_s$  dergestalt, dass ein Emulator dieses Typs zu einem Zeitpunkt  $t_w$ , wobei  $t_w > t_s$ , auf Basis der Daten vom Zeitpunkt  $t_s$  wiederhergestellt werden und die Ausführung anschließend konsistent fortsetzen kann. Zustandsdatensätze unterschiedlicher Emulatorarten sind also nicht zwingend kompatibel. Neben diesen typspezifischen Anteilen umfasst der Zustand Metadaten, die durch die Emulationsplattform definiert werden und der Verwaltung der Datensätze innerhalb der Plattform dienen.

Abbildung 5.7 rechts zeigt, wie Emulatorarten einer virtuellen Fabrik in der Ausführungsumgebung der Emulationsplattform durch Emulatorartefakte manifestiert werden. Diese Artefakte stellen die physikalische Repräsentationen der im Modell definierten Emulatoren dar. Sie beinhalten oder referenzieren jeweils das Modell des Emulatorart

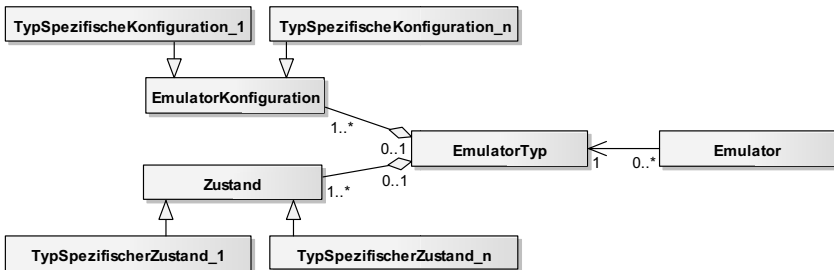


Abbildung 5.6: Statischer Zusammenhang: Emulatorart und Emulator

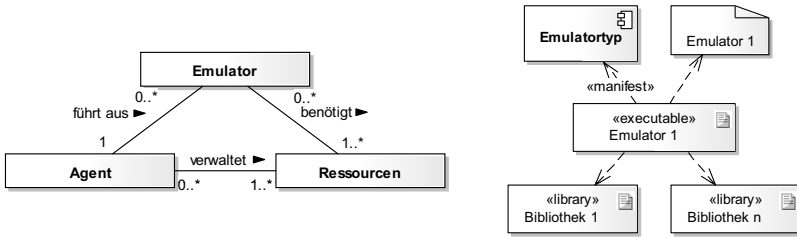


Abbildung 5.7: Links: Statischer Zusammenhang: Agent – Emulator. Rechts: Das Artefakt eines Emulators als konkrete Ausprägung eines Emulatortyps

und Metadaten, die aus dem Datensatz der Modellinstanz des Emulators generierbar sind. Emulatorartefakte stellen die kleinste Einheit dar, die in der Ausführungsumgebung verteilt und ausgeführt werden kann. Wie aus Abbildung 5.7 links ersichtlich, wird ein Emulator für dessen Ausführung einem geeigneten Agenten zugewiesen. Geeignet bedeutet in diesem Zusammenhang, dass der Agent dem Emulator neben Diensten der Emulationsplattform weitere benötigte Ressourcen zur Verfügung stellen können muss. Solche Ressourcen können z.B. Bibliotheken sein, von denen das Emulatorartefakt abhängt. Neben dem Ressourcenbedarf sind Randbedingungen bezüglich der Konfiguration der durch den Emulator zur Verfügung gestellten IT-Anlagenschnittstelle zu beachten.

Abbildung 5.8 zeigt in einer White-Box-Darstellung die vier wesentlichen Teilmodelle des Emulatortypmodells, die die in Abschnitt 5.2 definierten Ports versorgen: 'Host-Schnittstelle', 'Logistikintegration', 'Horizontale Kommunikation' und 'Anlagenmodell'.

Das Teilmodell der *Host-Schnittstelle* erzeugt eine Sicht auf die virtuelle Anlage, die der Spezifikation des eingesetzten Schnittstellentyps genügt (siehe Kapitel 2.4.2 und Kapitel 3.5.1). Wie im Fall der realen Anlage verbirgt diese Sicht den Implementierungsansatz der virtuellen Anlage und bildet diesen auf die durch die Schnittstellenspezifikation definierten Primitive ab. Für die generische Schnittstellenspezifikation PVECI (siehe Kapitel 2.4.2) umfasst das statische Modell der Host-Schnittstelle eine Reihe von Definitionen, die charakteristisch für eine konkrete Anlage oder einen Anlagentyp sind. Der lesende und schreibende Zugriff auf die virtuelle Anlage erfolgt über Dienstschnittstellen, die in der Norm in Form von 'Capabilities' definiert sind. Der Baustein der Host-Schnittstelle wird in der virtuellen Anlage über die Kopplung an das Verhaltensmodell mit Daten versorgt und wirkt, sofern durch den implementierten Schnittstellentyp unterstützt, steuernd auf das Anlagenmodell ein.

Das Teilmodell der *Logistikintegration* bindet das Anlagenmodell in das virtuelle Logistiksystem der Emulationsplattform ein und repräsentiert dabei die realen Logistikschnittstellen des Automatisierungssystems. Diese Anbindung wird über das Konzept des Ports realisiert. Ports sind typisierbar und ermöglichen damit die Abbildung spezifischer Eigenschaften der realen Logistikschnittstelle. Über Ports werden Logistikobjekte zwischen Emulator und Umgebung ausgetauscht, die ebenfalls typisierbar sind. Dadurch lässt sich im Teilmodell der Logistikintegration sicherstellen, dass im Anlagenmodell nur

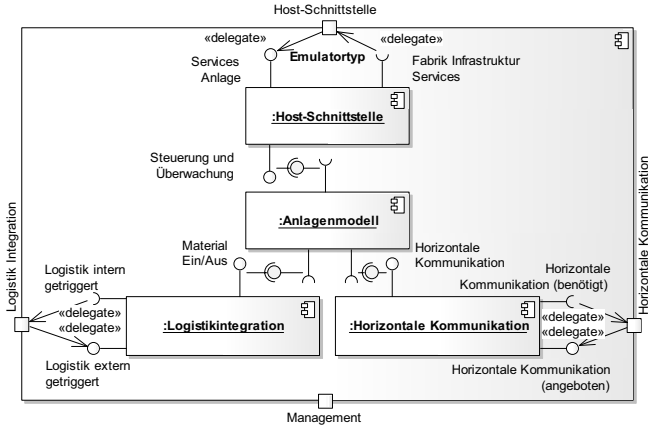


Abbildung 5.8: Das Emulator Typmodell – White-Box-Darstellung

kompatible Logistikobjekte verarbeitet werden. Bei Bedarf übernimmt die Logistikintegration die Transformation der Datentypen von Logistikobjekten zwischen den von der Emulationsplattform definierten und den vom Verhaltensmodell erwarteten Datentypen. Eine detaillierte Diskussion des virtuellen Logistiksystems folgt in Abschnitt 5.2.2.

Das Teilmodell *Horizontale Kommunikation* unterstützt die in Kapitel 3.5.1 beschriebene Verkettung von Anlagen, die einen Kommunikationskanal zwischen diesen Anlagen, unabhängig von den IT-Systemen der MES-Ebene, erforderlich macht. Solange lediglich die Übergabe von Substraten von einer Anlage zur nächsten durch die synchronisierte Übertragung eines die Substrate beschreibenden Datensatzes benötigt wird, kann diese Funktionalität durch die Logistikintegration zur Verfügung gestellt werden. Das Teilmodell *Horizontale Kommunikation* wird in diesem Fall nicht benötigt. Es wird aber für die zukünftige Verwendung vorgesehen (siehe Kapitel 3.5.1).

Das *Anlagenmodell* bildet schließlich die Struktur und das Verhalten des realen Automatisierungssystems mit einem dem Einsatzszenario angepassten Detaillierungsgrad ab. Prinzipiell überlässt das Metamodell dem Implementierer des Anlagenmodells die Entscheidung, wie das Verhalten einer konkreten Anlage auf ein Anlagenmodell abgebildet wird. Die Definition des Teilmodells erlaubt sowohl die Verwendung nativer Anlagenmodelle, die unter Verwendung von Modellelementen der Emulationsplattform erstellt werden, als auch die Integration externer Anlagenmodelle, die durch die übrigen Teilmodelle des Emulator Typmodells in den Kontext der Emulationsplattform integriert werden. Letzere ermöglicht die Einbindung vorhandener Emulatoren, die auf anderen Technologien basieren. Falls verfügbar, werden zusammen mit dem Anlagenmodell passende Implementierungen der Host-Schnittstelle und des Teilmodells *Horizontale Kommunikation* eingebunden. Damit wird im Idealfall der Aufwand zur Implementierung eines Emulator Typmodells auf die Einbindung in das virtuelle Logistiksystem und die Plattforminfrastruktur reduziert. Für den erstgenannten Fall steht die Strategie zur spezifika-

tionsgetriebenen Implementierung von Teilmodellen, insbesondere des Anlagenmodells und der Host-Schnittstelle zur Verfügung.

### Das statische Emulatortypmodell im Detail

Dem Primitiv des Emulatortyps bzw. dem Emulatortypmodell fällt im Metamodell der Emulationsplattform eine zentrale Rolle zu, da ein Emulatortyp für korrespondierende reale Anlagen alle Aspekte abbilden muss, die über deren IT-Schnittstelle für externe Systeme sichtbar sind. Abhängig von der Komplexität des abzubildenden Anlagentyps lässt diese Aufgabenstellung den größten Modellierungsaufwand beim Aufbau virtueller Fabriken erwarten. Damit virtuelle Fabriken in realen Projekten tatsächlich den erwarteten positiven Beitrag zum Projekterfolg liefern können, muss der für die Modellierung benötigte Aufwand in einem wirtschaftlich vertretbaren Verhältnis zum erwarteten Nutzen stehen. Gleichzeitig muss die Schnittstelle der realen Anlage durch den Emulator in der Präzision abgebildet werden, die einen Erkenntnisgewinn aus der Nutzung der virtuellen Fabrik erlaubt. Eine Schnittstelle mit umfangreicher Funktionalität vorausgesetzt, könnte die Komplexität der virtuellen Abbildung einer Anlage potentiell ein vergleichbares Maß wie die der realen Anlagensteuerung erreichen. Der Versuch einer exakten Nachimplementierung der Anlagensteuerung scheidet damit schon aus wirtschaftlichen Gründen aus. Geht man darüber hinaus von einer ähnlichen relativen Häufigkeit von Implementierungsfehlern in Programmcode vergleichbarer Komplexität aus, so wären bei diesem Vorgehen Abweichungen im Verhalten zu erwarten, die an der Host-Schnittstelle bzw. den Schnittstellen zur Horizontalen Kommunikation in Erscheinung treten. Diese Abweichungen würden den Nutzen von Experimenten mit der realen Fabriksteuerungssoftware in der Interaktion mit der virtuellen Fabrik in Frage stellen. Der in den folgenden Abschnitten beschriebene Lösungsansatz soll diesen widersprüchlichen Anforderungen gerecht werden.

### Die spezifikationsgetriebene Modellierung

Der Ansatz der spezifikationsgetriebenen Modellierung zielt auf einen praxisnahen Kompromiss zwischen Modellierungsaufwand und Funktionalität für die Erstellung einzelner Emulatortypen ab. Dieses Konzept zum Aufbau des Emulatortypmodells basiert auf der in Kapitel 3.5.1 formulierten Prämisse, dass für das virtuelle MES-Engineering lediglich eine endliche Menge von Automatisierungsszenarien zu berücksichtigen sind. Entsprechend dieser Strategie implementierte Emulatortypmodelle unterstützen an der Hostschnitt-Stelle eine endliche Menge von Automatisierungsszenarien, die für Experimente im Zusammenspiel mit der MES-Ebene zur Verfügung stehen und dabei den höchsten Nutzwert erwarten lassen. Die zu berücksichtigenden Automatisierungsszenarien, die im Beispiel in Abbildung 5.9 als UML-Kollaborationen (Basis Dekomposition nach Kraemer und Herrmann, siehe Kapitel 4.6.2) im UML-Kollaborationstyp 'Automatisierungsszenarien' dargestellt sind, sind für den konkreten Fall im Rahmen einer Analyse zu identifizieren und lassen sich durch die folgenden Eigenschaften charakterisieren:

- Sie beschreiben jeweils Interaktionen zwischen dem Host als Stellvertreter des MES und dem Emulator bzw. der Anlage, die ein spezifisches, aus einem Geschäftsprozess

abgeleitetes, Ziel verfolgen und durch Rollen realisiert werden, welche vom Host bzw. vom Emulator eingenommen werden.

- Sie sind im allgemeinen Fall zustandsbehaftet.
- Alle Automatisierungsszenarien eines Emulators greifen zur Interaktion mit dem MES auf eine Menge  $N$ , wobei  $0 < |N| < \infty$ , definierter Nachrichten zurück; innerhalb eines Automatisierungsszenarios (AS) wird die Menge  $N_{AS} \subseteq N$  von Nachrichten genutzt.
- Sie definieren für die Nutzung von  $N_{AS}$  gültige Pfade, d.h. Sequenzen im Nachrichtenaustausch zwischen MES und Emulator.

*Kollaborationen* spezifizieren strukturelle Aspekte der Anwendung von Verhaltensmustern in einem bestimmten Kontext. Für die Automatisierungsszenarien ist insbesondere die Interaktion zwischen den beiden Classifiern 'Host' und 'Anlage/Emulator' relevant,

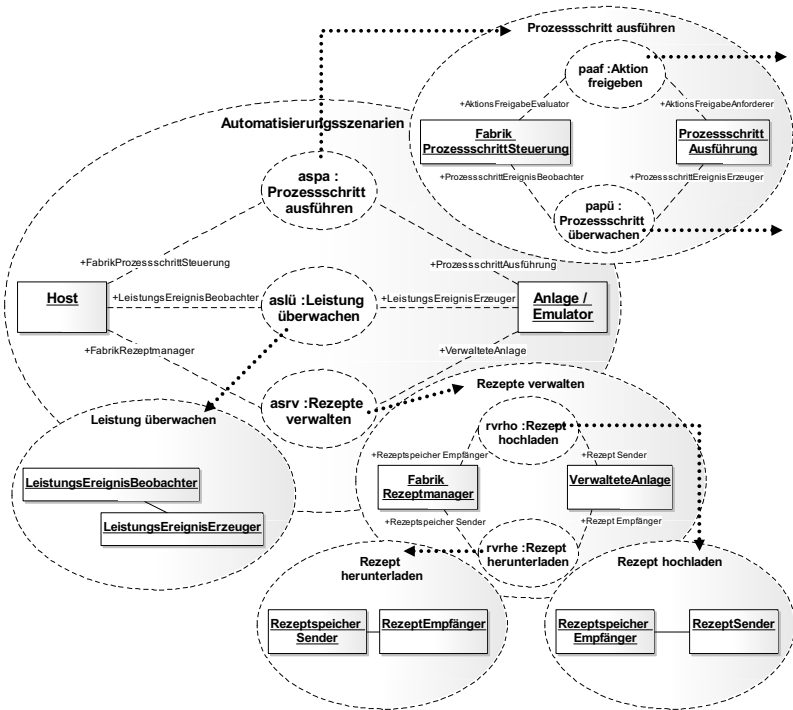


Abbildung 5.9: Beispiel: Die strukturelle Dekomposition dreier einfacher Automatisierungsszenarien mit Hilfe von UML Kollaborationstypen und Kollaborationen

die im Rahmen der Kollaborationen in unterschiedlichen Rollen interagieren. Die referenzierten Verhaltensmuster werden ihrerseits in *Kollaborationstypen* definiert, welche die interagierenden Rollen identifizieren, die für das jeweilige Verhaltensmuster relevant sind. Jene statische Beschreibung des Kollaborationstyps kann wiederum um eine geeignete dynamische Verhaltensbeschreibung ergänzt werden, die die Interaktion der Rollen im Detail festlegt. Ein Kollaborationstyp beschreibt eine Rolle nicht notwendigerweise vollständig. Er greift lediglich die Aspekte auf, die für das betrachtete Verhaltensmuster relevant sind. Eine Rolle kann also in verschiedenen Kollaborationstypen auftreten.

Damit eignen sich Kollaborationen und Kollaborationstypen sehr gut, um Automatisierungsszenarien top-down auf die Interaktion von Rollen über Konnektoren herunterzubrechen, die im weiteren Verlauf um die Spezifikation der dynamischen Aspekte erweitert werden (siehe auch Kapitel 5.3). Diese Strukturierung stellt damit den ersten Schritt der spezifikationsgetriebenen Modellierung dar. Abbildung 5.9 zeigt den Strukturierungsschritt für die drei Automatisierungsszenarien 'Prozessschritt ausführen', 'Leistung überwachen' und 'Rezepte verwalten', die schrittweise, wie durch die fett gedruckten, punktierten Pfeile angedeutet, verfeinert werden. Aus Platzgründen wird auf die Darstellung der in 'Prozessschritt ausführen' referenzierten Kollaborationstypen verzichtet.

### Definition der Menge gültiger Nachrichten

Für jeden Emulatorotyp wird auf Basis der Kollaborationstypen die Menge der gültigen Nachrichten  $\mathbf{N}$  ermittelt, die für die Kommunikation über die Host-Schnittstelle genutzt wird. Diese Definition beschreibt für alle Nachrichten  $n \in \mathbf{N}$  deren Struktur und Inhalte auf dem Detaillierungsgrad, den der für die Host-Schnittstelle eingesetzte Protokollstapel erforderlich macht. Kommt einer der in Kapitel 3.1 beschriebenen standardisierten Protokollstapel zum Einsatz, ist zumindest eine Teilmenge dieser Nachrichten bereits durch den Standard definiert. Sofern Inhalte, die durch den Emulator generiert werden, später weiterverarbeitet werden sollen, werden zusätzlich zu Informationen über den Datentyp gültige Wertebereiche spezifiziert. Die Nachrichtendefinitionen werden im Nachrichten-katalog der Host-Schnittstelle gespeichert.

### Anbindung eines virtuellen Logistiksystems

Für alle nativen Teilmodelle sieht das Metamodell einen objektorientierten Modellierungsansatz vor. Dabei sind insbesondere *permanente* und *transiente* Modellelemente zu unterscheiden. Permanente Modellelemente stehen über die gesamte Laufzeit eines Szenarios zur Verfügung. Transiente Modellelemente werden bei der Ausführung eines Szenarios erzeugt und zu einem späteren Zeitpunkt wieder zerstört. Für beide Klassen von Modellelementen stellt das Metamodell Basistypen zur Verfügung, die kompatibel zur Infrastruktur der Emulationsplattform sind und vom Modellierer bei Bedarf erweitert werden können. Die *Ports* der in einem Modell verwendeten Emulatorotypen stellen permanente Modellelemente dar und bilden die Ausgangsbasis zum Aufbau des virtuellen Logistiksystems. Ports definieren eine eindeutige Semantik zur Übergabe von Logistikobjekten und stellen adressierbare Endpunkte für das virtuelle Logistiksystem dar.

Neben den permanenten Ports sieht das virtuelle Logistiksystem transiente Logistikobjekte vor, die über die Grenzen einzelner Emulatoren durch die Ports übergeben werden können. Hinsichtlich der Übergabe von Logistikobjekten unterscheidet das Metamodell zwei grundlegende Porttypen: Ports, die die Verantwortung zur Verwaltung von Instanzen von Logistikobjekten übergeben bzw. übernehmen (*aktive Ports*) und Ports, die auf Instanzen von Logistikobjekten auf anderen Ports zugreifen, ohne deren Verwaltung zu übernehmen (*passive Ports*). Erstere bedienen sich eines definierten Puffers und tauschen Instanzen von Logistikobjekten zwischen Systemgrenzen aus. Mit der Übergabe einer Instanz geht die Verantwortung zur Verwaltung des Logistikobjekts vom Quellsystem zum Zielsystem über. Letztere ermöglichen den Zugriff auf Logistikobjekte, ohne dass die Verantwortung zu deren Verwaltung übertragen wird. Übernimmt ein Teilsystem die Verwaltung einer Instanz eines Logistikobjekts, obliegen ihm von diesem Zeitpunkt an die Verwaltung der Instanz im Speicher und einer Referenz auf die Instanz, die den lesenden und schreibenden Zugriff auf Zustandsinformation, einschließlich der Steuerung des Lebenszyklus, erlaubt. Ohne die Verantwortung zur Verwaltung für die Instanz eines Logistikobjekts ist kein schreibender Zugriff auf das Objekt möglich.

Als Voraussetzung zur Integrierbarkeit eines Emulatortyps in das virtuelle Logistiksystem ist auf Basis der durch das Metamodell zur Verfügung gestellten Port-Typen die Logistikanbindung der realen Anlage abzubilden. Falls die virtuelle Anlagenumgebung nicht bereits die benötigten Logistikobjekte zur Verfügung stellt, sind diese auf Basis der verfügbaren Logistikobjekte zu modellieren (siehe Kapitel 5.2.2) .

### Das Anlagenmodell

Das Anlagenmodell bildet eine logische Sicht der realen Anlage in Form interagierender Objekte ab. Es orientiert sich an den Teilsystemen der realen Anlage, die für die Umsetzung der Automatisierungsszenarien relevant sind. Sofern durch den Emulator Automatisierungsszenarien unterstützt werden sollen, in denen Material in der Anlage prozessiert wird, stellt die interne Logistik der realen Anlage einen geeigneten Ausgangspunkt zum Aufbau des Modells dar. Unter Voraussetzung der Verwendung der in Kapitel 3.5.1 betrachteten Host-Schnittstellen kann von einer stark abstrahierten Sicht des MES auf die interne Anlagenlogistik ausgegangen werden. Diese Sicht beschränkt sich auf eine Menge statisch definierter *logischer Positionen* innerhalb der Anlage, die Logistikobjekte bei der Ausführung von Automatisierungsszenarien erreichen und die bei der Datenerfassung bzw. Steuerung durch das MES relevant sind. Logische Positionen werden im Anlagenmodell durch beschränkte Puffer abgebildet, die einem Modellelement zugeordnet sind; ihre Belegung wird durch das dynamische Modell gesteuert und ist Teil des Gesamtzustands eines Emulators. Neben der Belegung der logischen Positionen kann jedes Anlagenmodell eine beliebige Anzahl weiterer Zustandsvariablen verwalten.

### Fazit Virtuelle Anlage

Unter Berücksichtigung der vorgestellten Modellelemente entsteht die Abbildung einer realen Anlage in der virtuellen Fabrik im Zusammenspiel von einem oder mehreren Emulatoren mit entsprechenden Konfigurationen. Das Metamodell bietet dem Anwender um-



fangreiche Freiheitsgrade hinsichtlich der Entscheidung, welcher Anteil eines Automatisierungssystems durch den Emulatorotyp vorgegeben bzw. durch dessen Konfiguration anpassbar ist. Bei der Modellierung ist immer die Randbedingung zu beachten, dass ein Emulatorotyp, unabhängig von seiner aktuellen Konfiguration, kompatible Zustandsrepräsentationen nutzen und Zustandsdatensätze, die für den Emulatorotyp gespeichert wurden, lesen und verarbeiten können muss.

### 5.2.2 Die virtuelle Anlagenumgebung

Zum Betrieb virtueller Anlagen in einer virtuellen Fabrik werden eine Reihe weiterer Modellelemente benötigt. An erster Stelle sind die bereits eingeführten Logistikobjekte zu nennen. Diese umfassen zunächst diskretes virtuelles *Material*, das durch virtuelle Anlagen bearbeitet wird. Material wird zur Laufzeit durch Instanzen von Material-Modellelementen repräsentiert. In einer virtuellen Fabrik werden die benötigten Material-Modellelemente durch die eingesetzten Emulatortypmodelle und deren Detaillierungsgrad bestimmt.

Neben produktivem Material, das unmittelbar in die hergestellten Produkte eingeht, bietet das Metamodell die Möglichkeit zur Beschreibung von *Hilfsstoffen*, die Anlagen zum Betrieb benötigen und dabei verbrauchen (engl. *consumables*), und dauerhaften virtuellen *Betriebsmitteln* (engl. *durables*), die mehrfach wiederverwendbar sind. Für die Abbildung des Logistikkonzepts spielen insbesondere Betriebsmittel eine Rolle, die für den Transport des Materials durch die Fabrik genutzt werden. In der realen Fabrik kommen entlang der Prozesskette eine Vielzahl von *Materialträgern* (engl. *carrier*) zum Einsatz, die sich in der virtuellen Fabrik wiederfinden. Die physikalischen Materialträger unterscheiden sich im Zugriffskonzept auf das enthaltene Material (wahlfreier Zugriff versus sequenzieller Zugriff), ihrer Kapazität, den physikalischen Abmessungen und den korrespondierenden Ports, über die Materialträger einer Anlage zur Be- und Entladung oder zur Speicherung des Materialträgers zugeführt werden. Im Metamodell werden Materialträger für diskretes Material als mehrdimensionale Systeme betrachtet, in denen sich die Position einer enthaltenen Materialinstanz  $M_i$  durch einen Positionsvektor  $\vec{P}_{M_i}$  im Koordinatensystem des Materialträgers eindeutig beschreiben lässt.

Das Metamodell unterstützt sowohl eine typisierte Modellierung von Logistikobjekten und Ports, die bereits zur Modellierungszeit die Überprüfung der Kompatibilität im Modell erlaubt, als auch eine einfache untypisierte Modellierung mittels generischer Logistikobjekte. Im letzteren Fall führen Inkompatibilitäten zu Ausnahmen bei der Ausführung des Modells zur Laufzeit. Logistikobjekte sind in jedem Fall über eine Kennzeichnung (engl. *Unique Identifier (UID)*) eindeutig identifizierbar, die der Kennzeichnung des korrespondierenden realen Objekts entspricht. Sie haben einen Zustand und eine *Historie*, in der zur Laufzeit Ereignisse protokolliert werden, die diese Instanz betreffen. Die Historie wird für die Auswertung von Experimenten und die Validierung des Modells herangezogen.

Die logistischen Teilsysteme einzelner Emulatoren werden über Modellelemente der Anlagenumgebung zum Logistiksystem der virtuellen Fabrik gekoppelt. Dieses System wird für eine virtuelle Fabrik in Form eines endlichen, gerichteten Graphen  $\vec{L}(P, V)$  abgebildet,

wobei  $P$  die endliche Menge aller Ports der Emulatoren einer virtuellen Fabrik (Knoten) und  $V$ , mit  $V \subseteq P^2$  die ebenfalls endliche Menge gerichteter Verbindungen (Kanten) beschreibt. In einer Verbindung  $v = (p_q, p_z) \in V$  entspricht  $p_q$  dem Quellport und  $p_z$  dem Zielport. Das logistische Teilsystem eines Emulators entspricht dem induzierten Untergraph  $\tilde{L}_{E_i}(P', V')$  mit  $P' \subseteq P$  und  $V' \subseteq V$ , wobei  $P'$  alle zu einem Emulator  $E_i$  gehörigen Ports und  $V'$  die Verbindungen innerhalb des Emulators  $E_i$  beschreiben. Im Gegensatz zu Ports können Verbindungen keine Instanzen von Logistikobjekten speichern. Komplexere Logistiksysteme in der Anlagenumgebung, die Puffer enthalten, werden stattdessen als Emulatoren abgebildet. Diese Emulatoren basieren auf Emulatortypen, die lediglich die für die Logistikfunktion relevanten Teilmodelle implementieren. Das Logistiksystem der virtuellen Fabrik lässt sich damit als Adjazenzmatrix  $\mathbf{A}_{(n,n)}$  darstellen, wobei  $n = |P|$  und

$$\mathbf{A} = (a_{ij}), a_{ij} = \begin{cases} 1 & \text{für } \{p_i, p_j\} \in V \\ 0 & \text{sonst} \end{cases}$$

### 5.2.3 Die virtuelle Fabrik

Die *virtuelle Fabrik* bildet schließlich den Kontext, in dem eine Menge von Emulatoren  $E$ , basierend auf einer Menge von Emulatortypen unter Verwendung des virtuellen Logistiksystems integriert werden. Sie repräsentiert damit die statische Struktur einer realen Fabrik, also die Menge der Anlagen und Automatisierungssysteme und deren Integration, soweit diese für deren Nachbildung an den Host-Schnittstellen der Anlagen erforderlich ist. Abbildung 5.10 stellt die wesentlichen Zusammenhänge im Klassendiagramm dar und bildet die Grundlage für die folgenden Beschreibungen zum dynamischen Verhalten.

Darüber hinaus spielt im Kontext der virtuellen Fabrik, insbesondere für das dynamische Modell, das Konzept des *Szenarios* eine wesentliche Rolle. Ein Szenario beschreibt orthogonal zur Struktur der virtuellen Fabrik die vollständige, konsistente Konfiguration einer virtuellen Fabrik zu einem Zeitpunkt  $t_s$ . Die Konfiguration besteht aus einer Menge von *Konfigurationssätzen*, jeweils einem pro Emulator. Ein Konfigurationssatz referenziert aus der Menge der verfügbaren Zustände und Konfigurationen des korrespondierenden Emulatortyps den Initialzustand des Emulators und dessen Konfiguration für dieses Szenario. Szenarios, deren Spezifika wiederum durch Metadaten beschrieben werden, bilden damit die Grundlage für vielfältige Experimente, um die Interaktion der MES-Ebene mit der virtuellen Fabrik zu untersuchen. So bieten Szenarien die Möglichkeit, Vorbedingungen, die die Anlagenseite beziehungsweise die Fabrik betreffen, in einfacher Art und Weise zu modellieren und vor ihrer Ausführung reproduzierbar wiederherzustellen. Der statische Modellanteil für die Erstellung eines Szenarienmodells umfasst folglich die Aufzeichnung der benötigten Zustandsabbilder für die genutzten Emulatortypen, die Erstellung der Emulatorkonfigurationen für diese Emulatortypen, deren Gruppierung zu Konfigurationssätzen und schließlich die Zuordnung zu Emulatoren. Die tatsächliche Zuweisung von Emulatoren zu Agenten auf Basis des Einsatzmodells erfolgt ebenfalls für den Kontext eines Szenarios (in Abbildung 5.10 nicht dargestellt). In Abhängigkeit vom gewählten Szenario lässt sich so die virtuelle Fabrik in unterschiedlicher Art und Weise

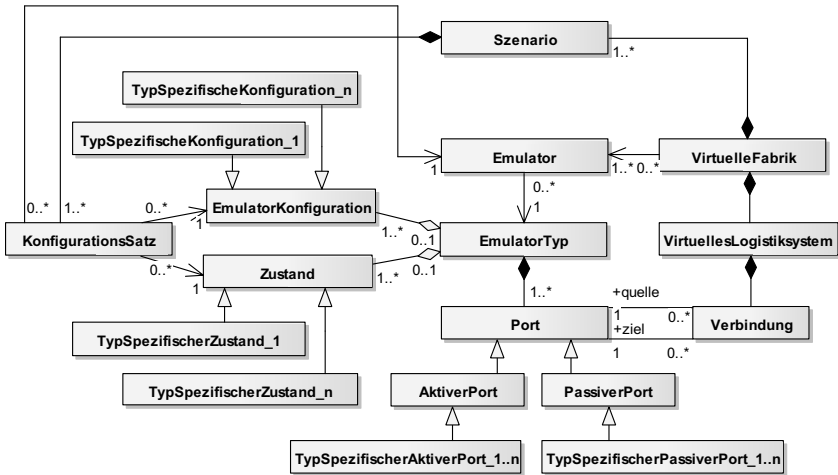


Abbildung 5.10: Statischer Zusammenhang: Die virtuelle Fabrik

auf die verfügbare Agenten-Infrastruktur verteilen, um den spezifischen Anforderungen des Szenarios gerecht zu werden. Die Szenarien werden auf der Infrastruktur der Emulationsumgebung installiert und zur Ausführung gebracht. Über die Aufgabe der Konfigurationsbeschreibung der virtuellen Fabrik hinaus bildet das Szenario den Kontext für die ereignisbasierte Steuerung von Experimenten, die im folgenden Kapitel detailliert beschrieben werden.

## 5.3 Das dynamische Modell

Das dynamische Modell bezieht den Zeitfaktor in die Diskussion des Metamodells der Emulationsplattform ein. Es beschreibt das Verhalten von Modellelementen und ihre Interaktionen. Daraus ergibt sich die Dynamik des Gesamtsystems der virtuellen Fabrik. Zunächst werden die dynamischen Aspekte des Emulatortypmodells und des virtuellen Logistiksystems beleuchtet. In der Folge werden die Lebenszyklusmodelle von Emulatoren und Szenarien eingeführt, die schließlich zur Diskussion des Ausführungsmodells der Emulationsplattform führen.

### 5.3.1 Anlagenverhalten modellieren

Im Kontext der spezifikationsgetriebenen Modellierung wird die komponentenbasierte Modellierung des Anlagenverhaltens, abgeleitet aus der bereits diskutierten statischen Zerlegung der Automatisierungsszenarien in Kollaborationstypen und Kollaborationen,

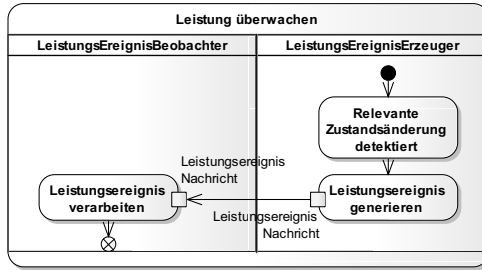


Abbildung 5.11: Beispiel: Interaktion der im Automatisierungsszenario 'Leistung überwachen' definierten Rollen

nativ durch das Metamodell unterstützt. Für die in den Kollaborationstypen interagierenden Rollen wird zunächst das Interaktionsverhalten auf Basis der Menge der definierten Nachrichten  $N$  im Detail spezifiziert. Abbildung 5.11 zeigt ein einfaches Beispiel einer solchen Spezifikation basierend auf UML Aktivitäten für die Rollen 'LeistungsEreignisErzeuger' und 'LeistungsEreignisBeobachter'. Die Rollen sind schließlich auf Modellkomponenten abzubilden, welche die spezifizierten Rollen zur Laufzeit einnehmen.

Sowohl hinsichtlich des Zeitverhaltens als auch hinsichtlich der Zustandsrepräsentation werden Anlagen in ihrer Modellrepräsentation als *diskrete* Systeme beschrieben (unter Verwendung der in Kapitel 4.6.1 referenzierten Basiskonzepte). Kontinuierliche Vorgänge werden im Modell abstrahiert. Der Zustand einer Anlage ist über eine endliche Zahl von *Zustandsvariablen* definiert. Änderungen an Zustandsvariablen werden immer durch *diskrete Ereignisse* ausgelöst. Solange kein Ereignis eintritt, das die Änderung von Zustandsvariablen bewirkt, bleibt der Zustand des Modells konstant. Es wird also vorausgesetzt, dass alle relevanten Vorgänge auf entsprechende Ereignisse abgebildet werden. Der dynamische Aspekt des Modells spezifiziert folglich, wie sich sein Zustand aufgrund bestimmter Ereignisse ändert. Solche Ereignisse treten zur Laufzeit zu definierten Zeitpunkten auf. Während aus konzeptioneller Sicht die Verarbeitung von Ereignissen keine Zeit "verbraucht", verstreicht während des Wartens auf Ereignisse Zeit. Generiert werden solche Ereignisse sowohl innerhalb des Anlagenmodells, als auch außerhalb des Anlagenmodells. Externe Ereignisse werden in den Systemkomponenten der 'Host-Schnittstelle', der 'Logistikintegration' und der 'Horizontalen Kommunikation' generiert (siehe Abbildung 5.8). Interne Ereignisse werden in den Komponenten des Anlagenmodells selbst erzeugt.

Zur Verarbeitung anstehende Ereignisse werden zur Laufzeit durch eine Planungskomponente (engl. scheduler) verwaltet (vgl. Kapitel 4.6.1). Die Planungskomponente stellt sicher, dass diese Ereignisse zum richtigen Zeitpunkt verarbeitet werden. Ereigniserzeuger können Ereignisse generieren, die zum aktuellen Zeitpunkt, also sofort, oder zu einem späteren Zeitpunkt in der Zukunft zu verarbeiten sind. Dabei kann der Fall eintreten, dass zwei oder mehrere Ereignisse für den gleichen Zeitpunkt eingeplant werden. Im Kontext der realen Anlage entsprechen diesem Fall zeitlich parallel ablaufende Vorgänge. Der Scheduler arbeitet die durch Ereignisquellen geplanten Ereignisse entsprechend der

zeitlichen Vorgaben des Plans ab. Parallele Ereignisse werden durch den Scheduler serialisiert und sequentiell, aufgrund der oben beschriebenen Randbedingung quasiparallel, ausgeführt. Zur Serialisierung der Ereignisse sind aus der Literatur vielfältige Strategien bekannt. Falls nicht anders angegeben, wird im Folgenden davon ausgegangen, dass zum gleichen Zeitpunkt eingeplante Ereignisse in der Reihenfolge, in der sie eingeplant wurden, bearbeitet werden. Jedem Emulator ist ein Scheduler zugeordnet, der in dieser Weise die Ausführung der im Anlagenmodell zu berücksichtigenden Ereignisse steuert.

Umfang und Detaillierungsgrad des Verhaltensmodells leiten sich konsequent aus den Automatisierungsszenarien und damit aus den verfeinerten Kollaborationstypen ab. Zur Beschreibung der Dynamik kommt die Kombination einer *prozessorientierten* und einer *ereignisorientierten Modellierungsstrategie* zum Einsatz (vgl. Kapitel 4.6.1). Unter Verwendung der prozessorientierten Strategie wird das Verhalten des betrachteten Systems durch die Interaktion von *Prozessen* abgebildet. Prozesse gruppieren in diesem Zusammenhang Ereignisse  $e_i$ , die einem Modellelement zuzuordnen sind, beschreiben also den Lebenszyklus eines Modellelements. Prozesse interagieren durch das Senden von Nachrichten. Konzeptionell integriert ein Prozess statische Anteile und die Verhaltensbeschreibung, d. h. das statische Modell und das dynamische Modell werden an dieser Stelle eng verwoben. Die Struktur des Modells hat auf diese Weise große Ähnlichkeit mit der des abgebildeten Systems. Die ereignisorientierte Strategie hingegen betrachtet das abzubildende Gesamtsystem, identifiziert relevante querschnittliche Klassen von Ereignissen und beschreibt für jede dieser Klassen in Ereignisbehandlungsroutinen, wie solche Ereignisse im Kontext der Modellelemente zu behandeln sind.

Offensichtliche Kandidaten zur Abbildung durch Prozesse sind die Module einer Anlage, die Material unter Verwendung von Betriebsmitteln prozessieren. Das Auftreten von Alarmen, die wiederum Auswirkungen auf diese Module haben, lässt sich hingegen sehr gut ereignisorientiert abbilden. Um den Implementierungsaufwand zu minimieren gilt die Prämisse, die Problemstellung mit möglichst wenigen, einfachen Prozessen bzw. Ereignisbehandlungsroutinen abzubilden. Die Modellierung einer Anlage über ein Modul bzw. einen Prozess bietet sich insbesondere für den Fall an, in dem das betrachtete Automatisierungsszenario in einem Prozess beschreibbar ist und ein Emulator gleichzeitig nur ein Automatisierungsszenario ausführt. Dieser Fall ist z. B. gegeben, wenn eine Anlage Material aus einem Materialträger prozessiert und anschließend wieder in diesen Materialträger ausgibt. In Fällen, in denen mehrere Automatisierungsszenarien gleichzeitig auf der Anlage ausgeführt werden, bietet sich die Einführung des transienten Modellelements eines *Auftrags* (engl. job) an. Aufträge werden als Prozesse mit einem spezifischen Lebenszyklus modelliert, die Automatisierungsszenarien in der Interaktion mit weiteren Prozessen des Anlagenmodells realisieren.

Zur Implementierung der beschriebenen Modellierungsstrategien werden lediglich einige wenige Basiskonzepte benötigt (vgl. Kapitel 4.6.1). Bei der prozessorientierten Modellierung wird ein Modellelement programmatisch durch einen Codeabschnitt “angetrieben”, der über dessen gesamte Lebensdauer ausgeführt wird, also das aktive Verhalten des Prozesses definiert. Die Ausführung dieses Codes verbraucht aus konzeptioneller Sicht keine Zeit. Bei der Ausführung auf realen Rechnersystemen ist diese Forderung natürlich nicht direkt realisierbar. Diese Problemstellung wird später gesondert betrachtet. Innerhalb der Lebensdauer des Modellelements kann die Ausführung zeitlich befristet oder unbefristet

unterbrochen werden. Im unterbrochenen Zustand des Prozesses wird Zeit verbraucht. Im Falle der zeitlich befristeten Unterbrechung wird die Ausführung des Prozesses nach Ablauf der gewählten Zeitspanne mit der nächsten Aktion nach der Unterbrechung wieder aufgenommen. Im Falle der zeitlich unbefristeten Unterbrechung wartet der Prozess auf das Eintreten eines bestimmten Ereignisses und wird beim Auftreten des Ereignisses ebenfalls mit der nächsten Aktion nach der Unterbrechung fortgesetzt. Der interne Zustand des Prozesses zum Zeitpunkt der Unterbrechung steht in beiden Fällen bei der Wiederaufnahme der Verarbeitung wieder zur Verfügung. Bei der ereignisbasierten Modellierung werden die Auswirkungen des Auftretens eines Ereignisses  $e_i$  einer gegebenen Ereignisklasse zu einem Zeitpunkt  $t_{e_i}$  auf das gesamte Modell betrachtet. Für jede Klasse von Ereignissen sind entsprechende Behandlungsroutinen zu definieren, die spezifizieren, wie die Ereignisse im Modell verarbeitet werden.

Der Einsatz der beiden Unterbrechungsmechanismen soll für den Fall der prozessorientierten Modellierung an dem einfachen Beispiel aus Abbildung 5.12 verdeutlicht werden. Die reale Anlage, in der Abbildung links schematisch dargestellt, entnimmt Wafer aus einem Eingangsmaterialträger aus, führt sie einer Inspektionsstation zu und legt die Wafer anschließend wieder in einen Ausgangsmaterialträger ein. Diese Anlage wird im Anlagenmodell des Emulatortyps durch einen Modulprozess abgebildet. Abbildung 5.12 rechts stellt den Ausführungszustand dieses Prozesses für einen möglichen Ausführungspfad über einen Zeitabschnitt dar. Wird das Anlagenmodell leer in Betrieb genommen, wartet der betrachtete Modulprozess zunächst im Zustand 'blockiert' (zeitlich unbefristete Unterbrechung) auf die Ankunft der beiden Materialträger. Zunächst trifft der Eingangsmaterialträger (1) über das virtuelle Logistiksystem am Emulator ein und löst ein Ereignis der Klasse 'MAE' aus. Dieses Ereignis bewirkt die Fortsetzung der Ausführung des unterbrochenen Prozesses, dargestellt durch die Transition in den Zustand 'ausführen'. Der Prozess liest den Identifikationscode des Materialträgers aus und generiert seinerseits ein Ereignis, das von der 'Host-Schnittstelle' in eine Nachricht auf Basis des Nachrichten-katalogs überführt und an das MES weitergeleitet wird. Anschließend wartet der Prozess wieder im Zustand 'blockiert' auf die Freigabe des Materialträgers durch das MES. In analoger Weise wird der Ausgangsmaterialträger (2) angemeldet. Der Transportschritt (3), in dem jeweils ein einzelner Wafer einer Inspektionsstation zugeführt wird und die Inspektion (4) selbst werden über den Zustand 'warten' (temporäre Unterbrechungen) abgebildet. Die Dauer der temporären Unterbrechungen wird aus statistischen Verteilungen ermittelt, die die Verteilung der realen Zeiten widerspiegelt.

### 5.3.2 Das Zeitmanagement der Emulationsplattform

Bei der Beschreibung des Ansatzes zur Verhaltensmodellierung wurde mehrfach der Begriff der Zeit verwendet, der nun im Kontext der virtuellen Fabrik genauer betrachtet werden soll. Dazu ist zunächst der im Vorfeld eingeführte Begriff des Prozesses zu erweitern. Prozesse  $T_i$  bezeichnen im Folgenden in Anlehnung an [Lampport, 1978] Einheiten innerhalb eines verteilten Systems, die über den Austausch von Nachrichten kommunizieren und aus einer Folge von Ereignissen  $e_{i_1}$  bis  $e_{i_n}$  bestehen. Als Ereignisse werden das Senden und Empfangen von Nachrichten oder die Verarbeitung von Anweisungen auf Recheneinheiten bezeichnet. Von besonderer Relevanz für das Zeitmanagement der

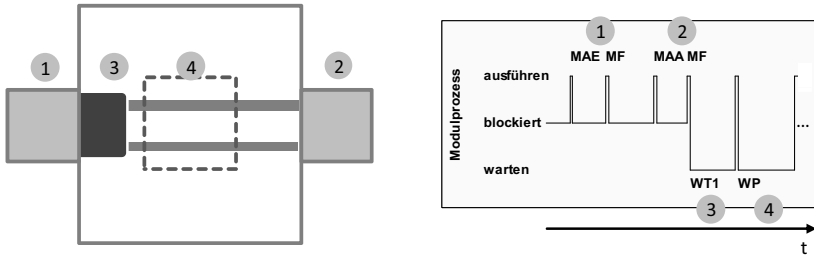


Abbildung 5.12: Beispiel: Ausschnitt eines einfachen Automatisierungsszenarios und korrespondierende Prozesszustände

Emulationsplattform sind die Prozesse der Emulatoren, die entsprechend des Einsatzmodells verteilt auf den Agenten der Plattforminfrastruktur laufen, die Prozesse des Szenariomanagers und die des MES, insbesondere auf der Ebene der Anlagenanbindung. Der Grad der Abhängigkeit von den MES-Prozessen hängt von dem zur Anbindung genutzten Protokollstapel ab.

### Zeitmanagement in der virtuellen Fabrik

Die Analyse in Kapitel 3.5.2 gibt Hinweise auf eine Reihe von Randbedingungen hinsichtlich des Zeitmanagements in der virtuellen Fabrik, die für den effektiven Betrieb der Emulationsplattform in der Interaktion mit realen Systemen der Produktions-IT entscheidend sind. Soweit kausale Zusammenhänge nicht anderweitig eindeutig gekennzeichnet sind, werden Zeitstempel zu ihrer Analyse herangezogen, die den Zeitpunkt des Auftretens von Ereignissen dokumentieren. Unter Berücksichtigung typischer Architekturen (siehe Kapitel 2.3.3) wird im Folgenden vorausgesetzt, dass Ereignisse und Zeitstempel auf allen Ebenen des verteilten Systems der Produktions-IT generiert werden. Daher müssen alle Prozesse kompatible Zeit-Repräsentationen nutzen, die über das System der gesamten Fabrik hinweg möglichst präzise Aussagen hinsichtlich der Reihenfolge des Auftretens von Ereignissen erlauben. Da die Emulationsplattform bei der virtuellen Inbetriebnahme mit kommerziellen Softwarelösungen unterschiedlicher Hersteller auf der MES-Ebene interagieren können soll, ist das Zeitmanagement der Systeme der Produktions-IT außerhalb der Emulationsplattform als gegeben hinzunehmen. Die Mechanismen zur Generierung von Zeitstempeln, zur Verarbeitung von Zeitstempeln und zur zeitgesteuerten Verarbeitung von Ereignissen sind in diesen Systemen als nicht austauschbar anzusehen. In Summe machen diese Bedingungen für die Emulationsplattform die Nutzung von Echtzeit-Zeitbasen erforderlich, wie sie auch von den Systemen der Produktions-IT genutzt werden. Folglich steht jedem Prozess der Emulationsplattform und der anzubindenden Systeme der Produktions-IT eine Echtzeit-Zeitbasis  $C_{T_i}$  zur Verfügung, die mit einer gemeinsamen Zeitbasis synchronisiert wird. Wenn  $C_{T_i}(t)$  diese Zeitbasis beschreibt, so stellt der Synchronisationsmechanismus sicher, dass zu jedem Zeitpunkt  $t$  gilt:  $\forall j, k : |C_{T_j}(t) - C_{T_k}(t)| < \delta$ , wobei  $\delta$  eine Konstante ist, die die maximale Abweichung der Zeit der unterschiedlichen Zeitbasen angibt. Diese Rahmenbedingung

ist durch den Einsatz eines geeigneten Synchronisationsmechanismus sicher zu stellen und lässt sich auch für Systeme der Produktions-IT gewährleisten (vgl. Kapitel 2.4.2: Nutzung von NTP).

### Zeit-Management in der virtuellen Anlage

Bei der Nachbildung des Verhaltens einer Anlage durch einen Emulator spielen zeitabhängige Vorgänge eine besondere Rolle. Solche Vorgänge sind durch eine Sequenz von Ereignissen  $e_{i_1}$  bis  $e_{i_n}$  innerhalb eines Prozesses  $T_i$  gekennzeichnet, die in definierten zeitlichen Abständen auftreten (vgl. Kapitel 5.3.1: zeitlich beschränkte Unterbrechung bzw. geplante Ereignisse). Aufgrund der Randbedingung, dass das Zeitmanagement der IT-Systeme der MES-Ebene nicht angepasst werden kann, ist der Zeitabstand  $\Delta t_{ij} > 0$ , der eine Sequenz von zwei Ereignissen  $e_i, e_j$  trennt, in der Ausführungsumgebung der Emulationsplattform in Realzeit zu berücksichtigen. So entspricht die Dauer  $\Delta t = 1s$  bei der Ausführung eines Szenarios auf der Emulationsplattform der durch die SI-Einheit 1s definierten Dauer. Für Experimente, bei denen diese strikte zeitliche Kopplung zwischen Ausführungsumgebung und Produktions-IT nicht erforderlich ist, wird ein Beschleunigungsfaktor  $\tau \in \mathbb{R}^+$  eingeführt, der für  $0 < \tau < 1$  zu einer Streckung bzw. für  $\tau > 1$  zu einer Stauchung zeitabhängiger Vorgänge in der Ausführungsumgebung führt. Um die Synchronität zwischen Ausführungsumgebung und den Systemen der Produktions-IT nicht zu gefährden, werden die Zeitbasen der Ausführungsumgebung (AU)  $C_{TAU}(t)$ , ohne Berücksichtigung etwaiger Drift, weiterhin mit derselben Frequenz betrieben wie die der Produktions-IT (P-IT)  $C_{TP-IT}(t)$ , also gilt für  $t \geq 0$ :  $\frac{d}{dt}C_{TP-IT}(t) = \frac{d}{dt}C_{TAU}(t)$ . Der Faktor  $\tau$  wird lediglich bei der lokalen Berechnung der Dauer zeitabhängiger Vorgänge innerhalb der Prozesse  $T_{AU}$  mit einbezogen. Für die Berechnung des Eintrittszeitpunkts  $t_1$  eines Ereignisses  $e_{i_1}$ , das mit dem Abstand  $\Delta t_{01}$  auf das Ereignis  $e_0$  zum Zeitpunkt  $t_0$  folgt, gilt:  $t_1 = t_0 + \frac{\Delta t_{01}}{\tau}$ . Damit können Zeitstempel zwischen beiden Welten problemlos ausgetauscht werden. Gleichzeitig lässt sich die Dauer von Abläufen für ausgewählte Experimente anpassen.

Hinsichtlich der Präzision des Schedulers, der geplante zukünftige Ereignisse zur Ausführung bringt bzw. Prozesse zu definierten Zeitpunkten aktiviert, sind einige Randbedingungen zu beachten. Zunächst ist die Begrenzung der Auflösung  $\epsilon$  der in realen Computersystemen verfügbaren Echtzeit-Zeitbasen zu beachten, die den kleinsten unterscheidbaren Abstand zweier Zeitstempel begrenzt. In dieser Definition ist der Aufwand des Zugriffs auf die aktuelle Zeit aus einem Programm heraus berücksichtigt. Zwei Ereignisse  $e_{i_0}$  zum Zeitpunkt  $t_0$  und  $e_{i_1}$  zum Zeitpunkt  $t_1$ , wobei  $t_0 < t_1$ , die in kleinerem zeitlichen Abstand  $t_1 - t_0 < \epsilon$  auftreten, können nicht sicher durch ihre Zeitstempel unterschieden werden. Sofern Emulatoren nicht auf einer Echtzeit-Plattform ausgeführt werden, ist darüber hinaus zu beachten, dass der Start der Verarbeitung eines für den Zeitpunkt  $t_p$  geplanten Ereignisses  $e_p$  nicht in festen Grenzen garantiert werden kann. Ebenso wenig kann eine bestimmte Laufzeit für den Programmcode zur Ereignisverarbeitung oder zur Bearbeitung von Aktivitäten innerhalb von Prozessen garantiert werden. Da sich der Scheduler-Prozess des Emulators die Ressourcen des Agentensystems mit anderen Prozessen teilt, werden beide Fälle stark durch den Scheduler des Betriebssystems und folglich durch andere laufende Prozesse beeinflusst. Damit können sich weitere



Verschiebungen im Vergleich zum ursprünglichen Plan des Schedulers ergeben.

Im Einklang mit den in der Analyse ermittelten Anforderungen wird ein weicher Planungsansatz verfolgt: Für Start-Zeitpunkte  $t_{p_i}$  wird lediglich garantiert  $t_{p_i} \geq t_p - \epsilon$ . Folglich sind zeitliche Intervalle  $\Delta t$  lediglich als Mindestabstand zweier Ereignisse zu betrachten. Geht man entsprechend der Analyseergebnisse von schnellsten Vorgängen innerhalb einer Anlage aus, deren Dauer sich in der Größenordnung von 1s bewegt, so stehen auf einem anderweitig wenig ausgelasteten, aktuellen Rechnersystem hinreichende Reserven zur Verfügung, um das Verhalten der realen Anlage mit ausreichender Genauigkeit abbilden zu können und unterscheidbare Zeitstempel zu erhalten. Die Konstante  $\delta$  hat für die Verkettung virtueller Anlagen eine besondere Relevanz. Für den Zeitstempel des letzten Ereignisses der Quellanlage  $t_{e_q}$  im Vergleich zum Zeitstempel des ersten Ereignisses der Zielanlage  $t_{e_z}$  muss unbedingt gelten  $t_{e_q} < t_{e_z}$ . Für ein technisch gut realisierbares  $\delta = 10ms$  und  $\epsilon = 15ms$  könnte, falls die Übergabe der Logistikobjekte keine Zeit in Anspruch nähme, der Fall  $t_{e_z} = t_{e_q} - 2(\delta + \epsilon) = t_{e_q} - 50ms$  auftreten. Das Auftreten dieses Fehlers lässt sich leicht über einen zusätzlichen Zeitstempel, den die Quellanlage dem Logistikobjekt hinzufügt, erkennen. Auf dieser Basis kann eine geeignete Fehlerbehandlungsstrategie umgesetzt werden.

### 5.3.3 Formale Lebenszyklus-Modelle

Emulatoren und Szenarien werden zur Laufzeit durch die Emulationsplattform verwaltet. Ein reproduzierbares Verhalten der virtuellen Fabrik unter dieser Verwaltung ist nur durch eine hinreichend formale Definition der Semantik an den Schnittstellen des Ports 'Management' zu erreichen. Dazu werden im Folgenden die Lebenszyklen von Emulatoren und Szenarien formal definiert. Zur Beschreibung und Visualisierung werden UML Zustandsautomaten verwendet. Diese basieren auf dem Konzept endlicher Automaten, welche durch die Menge gültiger Zustände, die Menge gültiger Operationen, den Startzustand, die Menge der Endzustände und eine Zustandsübergangsfunktion beschrieben sind.

#### Der Lebenszyklus des Emulators

Unabhängig vom verwendeten Emulator typ gehorchen alle Emulatoren dem in Abbildung 5.13 dargestellten Lebenszyklus. Auf der obersten Hierarchieebene des Automaten sind zunächst die beiden zusammengesetzten Zustände 'Frei' und 'Zugewiesen' zu unterscheiden. Der erstere erlaubt die Existenz eines Emulators auf einem Agenten, ohne dass der Emulator einem Szenario oder einer interaktiven Benutzersitzung zugewiesen ist. Damit können Szenarien und manuelle Sitzungen auf Agenten verfügbare Emulatorinstallationen wiederverwenden. Insbesondere bei umfangreichen Experimenten, die eine große Zahl von Emulatoren erfordern, kann so beim Szenarienwechsel die Anzahl der benötigten Neuinstallationen begrenzt werden. Der Ressourcen bzw. Zeitaufwand für das Experiment reduziert sich dadurch. Mit der Initialisierung wird ein Emulator einem koordinierenden Szenario oder einer interaktiven Benutzersitzung exklusiv zugewiesen. Diese Zuweisung bleibt bis zur Signalisierung von 'Beenden' durch das Szenario oder den Benutzer erhalten.

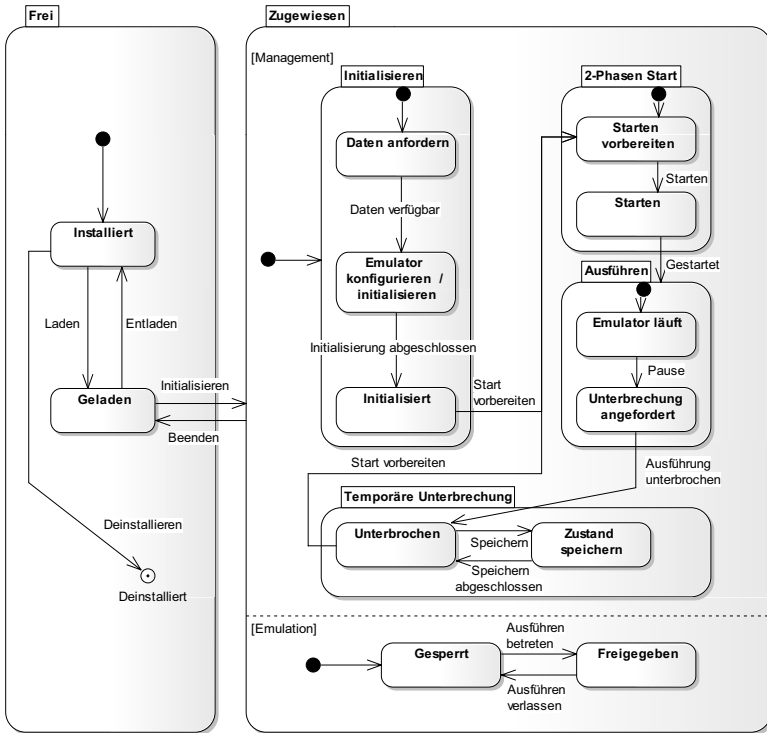


Abbildung 5.13: Der Lebenszyklus eines Emulators

Im Zustand 'Zugewiesen', also während des eigentlichen Betriebs des Emulators, sind insbesondere die Zustände 'Initialisieren', '2-Phasen Start', 'Ausführen' und 'Temporäre Unterbrechung' zu unterscheiden. Im Zuge der Initialisierung wird zunächst der mit dem Szenario assoziierte Konfigurationssatz über einen Plattformdienst angefordert, die Konfiguration des Emulators dementsprechend angepasst und die Zustandsvariablen aller Komponenten des Emulators unter Verwendung des Zustandsdatensatzes initialisiert. Die Menge der verfügbaren Zustandsvariablen, welche die für die Beschreibung des Komponentenzustands relevanten Daten speichern, wird jeweils durch den Emulatortyp bestimmt. Nach dem Empfang des Signals 'Start vorbereiten' wechselt der Emulator in den Zustand '2-Phasen Start', der die Emulatoren eines Szenarios synchronisiert startet (siehe auch Kapitel 5.3.4). Nach einem erfolgreichem Start wechselt der Emulator in den Zustand 'Ausführen'. In diesem Zustand kann der Emulator für Experimente genutzt werden. Dies wird durch den Wechsel in den Zustand 'Freigegeben' in der Region 'Emulation' angezeigt.

Über das Signal 'Pause' wird die Ausführung der Emulation temporär unterbrochen. Abhängig vom Emulatortyp ist eine Unterbrechung allerdings nicht zu jedem Zeitpunkt

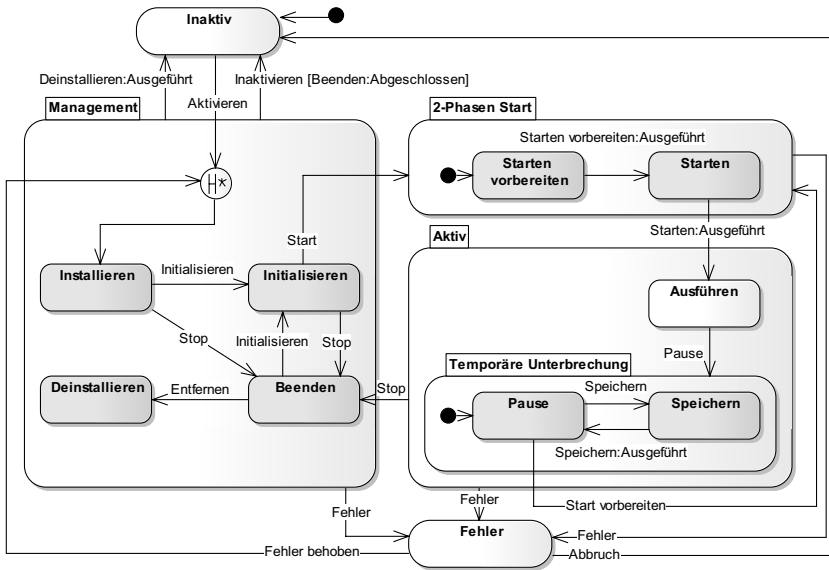


Abbildung 5.14: Der Lebenszyklus eines Szenarios – Ebene 0

in einer Art und Weise möglich, die eine spätere Fortsetzung des Betriebs von diesem Punkt aus erlaubt. Aus diesem Grund führt die Verarbeitung des Ereignisses 'Pause' zunächst zur Generierung der Anforderung einer Unterbrechung. Die angeforderte temporäre Unterbrechung wird vom Emulator zum nächstmöglichen Zeitpunkt ausgeführt und führt zum Wechsel in den Zustand 'Temporäre Unterbrechung'.

Im Zustand 'Unterbrochen' kann ein konsistentes Zustandsabbild des Emulators abgespeichert werden, das in der Folge für die Modellierung neuer Szenarien zur Verfügung steht. Während der Unterbrechung gilt für die emulierte Schnittstelle, dass keine Operationen ausgeführt werden können, die zu einer Änderung des eingefrorenen Zustands des Emulators führen würden. In diesem Fall ist in der Region 'Emulation' der Zustand 'Gespart' aktiv. Der Zustandswechsel von 'Frei' nach 'Gespart' wird beim Verlassen des zusammengesetzten Zustands 'Ausführen' ausgelöst.

Die Signalisierung von 'Beenden' führt aus dem Zustand 'Zugewiesen' schnellstmöglich zurück in den Zustand 'Frei', wobei alle genutzten Ressourcen ordnungsgemäß zurückgegeben und Konfiguration bzw. Zustandsdaten verworfen werden.

### Der Lebenszyklus des Szenarios

Der Lebenszyklus des Szenarios ist eng mit dem Lebenszyklus des Emulators verbunden, da ein Szenario die Lebenszyklen einer Menge von Emulatoren integriert. Die Ver-

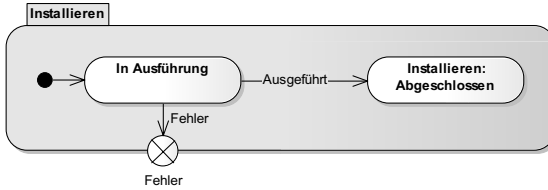


Abbildung 5.15: Der zusammengesetzte Zustand 'Installieren' – Ebene 1

bindung zwischen beiden Lebenszyklusmodellen wird über korrespondierende Zustände und definierte Ereignisse hergestellt, die jeweils von einem Automatentyp generiert und asynchron vom anderen Automatentyp verarbeitet werden. Entsprechend des statischen Modells aus Kapitel 5.2 stellt das Szenario das Modellelement dar, das eine spezifische, verteilbare Konfiguration einer virtuellen Fabrik beschreibt und sich damit auch für deren Koordination zur Laufzeit anbietet. Die Ausführung von Szenarien wird vom Plattformdienst "Szenariomanager" unter Berücksichtigung des Lebenszyklusmodells gesteuert.

Abbildung 5.14 links zeigt die oberste Hierarchieebene (Ebene 0) des Zustandsautomaten eines Szenarios. Dieser Zustandsautomat stellt die Wurzel einer baumförmigen Organisation zusammengesetzter Zustände dar. Jeder der gezeigten Zustände kann also prinzipiell beliebig viele Unterzustände besitzen. Alle grau-schattiert dargestellten Zustände korrespondieren mit jeweils einer Transition im Lebenszyklusmodell des Emulators. Die so gekennzeichneten Zustände sind analog zum Zustand 'installieren' aufgebaut, dessen Ebene 1 in Abbildung 5.15 dargestellt ist: Mit dem Wechsel in den Unterzustand 'Ausführen' sendet das Szenario allen beteiligten Emulatoren das im Zustandsnamen auf Ebene 0 erkennbare Signal. Das Szenario verbleibt so lange im Unterzustand 'Ausführen' auf Ebene 1, bis alle beteiligten Emulatoren erfolgreich in den gewünschten Zielzustand versetzt werden konnten und diese Tatsache durch die ausführenden Agenten quittiert wurde. In diesem Fall wechselt das Szenario in den zugehörigen Unterzustand 'Abgeschlossen'. Tritt in der Interaktion mit den Emulatoren ein Fehler auf, den der Szenariomanager nicht selbständig beheben kann, führt das zum Übergang in den Zustand 'Fehler'. Sofern der Fehler behoben werden kann, kann die Ausführung vom Zustand vor dem Auftreten des Fehlers fortgesetzt werden. Ansonsten muss die Ausführung abgebrochen werden.

Der zusammengesetzte Zustand 'Management' bildet gebündelt die Infrastrukturaufgaben eines Szenarios ab. Die Bedeutung der Zustände '2-Phasen Start' und 'Aktiv' ergibt sich unmittelbar aus dem Emulator-Lebenszyklus. Im letztgenannten Zustand steht die virtuelle Fabrik für Experimente zur Verfügung.

### 5.3.4 Szenarien ausführen

Aus dem verteilten Prozess-Ausführungsmodell (engl. execution model) ergeben sich weitere Fragen bezüglich der Synchronisation eines Emulators mit seiner Umgebung und der systemweiten Synchronisation von Emulatoren auf Basis der beschriebenen Lebenszyklen.

## Lokale Synchronisation

Im Zentrum des Ausführungsmodells stehen die Emulatoren einer virtuellen Fabrik, die jeweils in eigenen Prozessen ausgeführt werden. Diese Prozesse werden durch Bedingungen blockiert oder zur Ausführung freigegeben, die den internen Zustand des Emulators berücksichtigen. Von außen wird dieser Zustand über die Host-Schnittstelle, die Schnittstelle zur horizontalen Kommunikation, die Management-Schnittstelle der Emulationsplattform und den Zustand des virtuellen Logistiksystems, soweit er für den betrachteten Emulator relevant ist, beeinflusst. Der letztgenannte Aspekt spielt für das Verständnis des Gesamtsystems eine wichtige Rolle. Relevant für den Emulator ist (siehe Kapitel 5.2.2) einerseits die Teilmenge  $P_E = \{p \in P | (q, p) \in V\}$ , wobei  $q \in P \setminus \{p\}$  der Eingangsports und andererseits die Teilmenge  $P_Z = \{p \in P | (p, z) \in V\}$ , wobei  $z \in P \setminus \{z\}$  der Zielports. Ein Emulator wird so lange durch das virtuelle Logistiksystem in seiner Ausführung blockiert, solange die auf Eingangsports  $p_E \in P_E$  verfügbare Konfiguration von Logistikobjekten nicht der Vorbedingung entspricht oder der Versuch des Ausschleusens durch die Belegung benötigter Ausgangsports  $p_A \in P_A$  verhindert wird. Sind alle äußeren Bedingungen zur Verarbeitung erfüllt, obliegt die Freigabe des Prozesses allein dem Emulator.

Der Transport von Logistikobjekten zur Laufzeit über Verbindungen des virtuellen Logistiksystems erfolgt *transaktional*. Das bedeutet insbesondere, dass der Transfer in einer *atomaren* Operation erfolgt, die entweder vollständig ausgeführt oder nicht ausgeführt wird. Blockierte Zielports führen in jedem Fall zum Abbruch des Transfervorgangs. Damit wird sichergestellt, dass sich das virtuelle Logistiksystem stets in einem konsistenten Zustand befindet und die Verantwortung zur Verwaltung von Instanzen von Logistikobjekten stets eindeutig geregelt ist.

## Systemweite Synchronisation

Das Ausführungsmodell sieht als systemweite Prozesssynchronisationspunkte bei der Ausführung von Szenarien lediglich die Initialisierung bzw. den Start eines Szenarios, dessen temporäre Unterbrechung und seine Beendigung vor. Diese Synchronisation wird, wie oben beschrieben, mit Hilfe des Szenariomanager-Plattformdienstes auf Basis des Szenario-Lebenszyklusmodells gesteuert. Obwohl sich das Ausführungsmodell aufgrund seiner Parallelität außerhalb der Prozesssynchronisationspunkte, wie die reale Fabrik, nicht deterministisch verhält, wird für den Zeitpunkt des Starts eines Szenarios  $t_{Start}$  ein höchstmögliches Maß an Synchronität angestrebt. Die Möglichkeit, dasselbe Szenario wiederholt unter möglichst ähnlichen Ausgangsbedingungen anlaufen zu lassen, erleichtert das Experimentieren mit der virtuellen Fabrik.

Dieses Ziel wird mit dem in Abbildung 5.16 dargestellten Zwei-Phasen Start von Szenarien verfolgt. In Phase I fordert die Szenario-Koordination alle beteiligten Emulatoren asynchron über ihre Agenten zur Vorbereitung des Start-Vorgangs auf. Die Emulatoren bestätigen die Anforderung asynchron unter Angabe einer Zeitspanne  $t_{minVerzögerung_i}$ , die sie aus technischen Gründen zum Start benötigen werden. In Phase II ermittelt die Szenario-Koordination aus den Rückmeldungen aller Emulatoren ausgehend von der aktuellen Systemzeit  $t_0$  den optimalen Startzeitpunkt  $t_{Start} = t_0 + \max(t_{minVerzögerung_i})$ ,  $0 <$

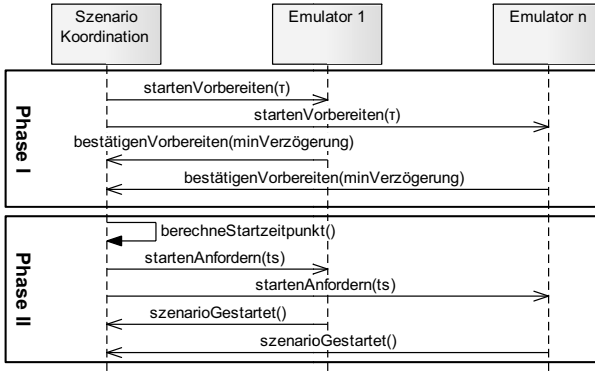


Abbildung 5.16: Der Zwei-Phasen-Start von Szenarien (ohne Berücksichtigung involvierter Agenten)

$i \leq n$  und fordert von allen beteiligten Emulatoren den Start zu diesem Zeitpunkt an. Die Freigabe der Emulation zum Zeitpunkt  $t_{start}$  wird von den einzelnen Emulatoren umgesetzt.

## 5.4 Realisierung der Emulationsplattform

Die in den vorigen Abschnitten beschriebenen Konzepte werden in Form einer prototypischen Implementierung umgesetzt, die die praktische Anwendung und Erprobung der Konzepte ermöglicht. Das Ergebnis der Implementierung firmiert unter dem Acronym *Large Scale Emulation (LSE)*. Den Einstieg in die Darstellung der Realisierung bildet ein Überblick über die Anwendungsfälle aus Sicht eines Benutzers, der LSE in der Praxis beim MES-Engineering einsetzt. Das Prototypsystem wird entlang des zu Beginn dieses Kapitels eingeführten Architekturkonzepts diskutiert und greift verschiedene Fragestellungen der Realisierung für die drei Schichten auf. Ein Aspekt dieser Fragestellungen ist die Auswahl geeigneter Basistechnologien und Werkzeuge, die eine effiziente Implementierung ermöglichen und die Integration in eine durchgängige Werkzeugkette für den Anwender erlauben. Die an der statischen Struktur orientierte Diskussion wird ergänzt um ein beispielhaftes Laufzeitszenario, das die Interaktion der LSE-Teilsysteme verdeutlicht.

### 5.4.1 Die Anwendungsfälle im MES-Engineering-Projekt

Abbildung 5.17 zeigt die wichtigsten Gruppen von Anwendungsfällen der Emulationsplattform im MES-Engineering-Projekt und deren Zusammenhang im Überblick. Dabei

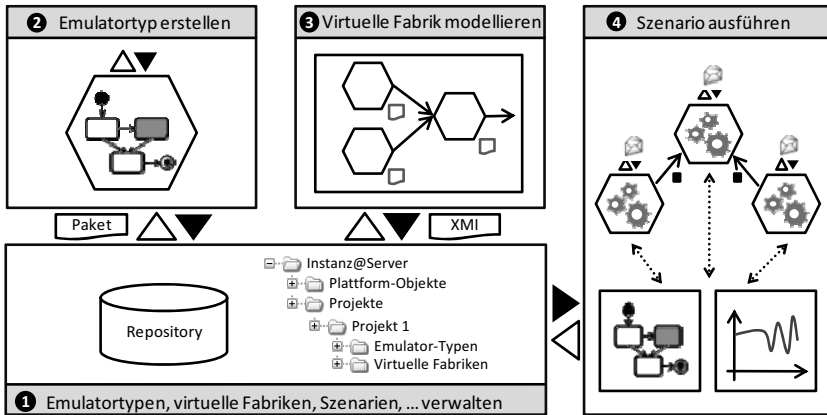


Abbildung 5.17: Die wichtigsten Gruppen von Anwendungsfällen der Emulationsplattform im Überblick

tritt die in Kapitel 5.1.1 beschriebene Differenzierung zwischen build time und runtime deutlich zutage. Die Gruppen 1, 2 und 3 von Anwendungsfällen sind der build time zuzuordnen. Die Gruppe 4 umfasst die Anwendungsfälle der runtime.

## Build time

Die Anwendungsfälle der Gruppe 1 beschreiben Interaktionen des Benutzers mit dem Dienst 'Repository' zur Verwaltung von Modellelementen. Benutzer verbinden sich über eine GUI mit einer Instanz dieses Dienstes, um verfügbare Modellelemente zu suchen, zur Ansicht bzw. zur Bearbeitung zu öffnen oder zu löschen. Darüber hinaus ermöglicht die Verwaltung dem Benutzer, das Anlegen neuer Modellelemente anzustoßen. Die GUI bietet dem Benutzer zur Unterstützung dieser Aufgabenstellungen verschiedene Sichten auf die vom Repository-Dienst verwalteten Datenstrukturen.

Die Gruppen 2 (Emulatortyp erstellen) und 3 (virtuelle Fabrik modellieren) werden getrennt von der Gruppe 1 betrachtet, da jede dieser Gruppen sehr spezifische Anforderungen an die Benutzeroberfläche bzw. die benötigten Werkzeuge stellt. Falls Emulatoren, die nicht auf LSE-Technologien basieren, in virtuelle Fabriken eingebunden werden, werden die Anwendungsfälle der Gruppe 2 außerhalb der Werkzeugumgebung der Emulationsplattform ausgeführt. Sowohl die Integration solcher externer Emulatoren, als auch das Erstellen und Bearbeiten von Emulatoren auf Basis der Technologien, die durch die Emulationsplattform zur Verfügung gestellt werden, wird durch die Plattform unterstützt. Einmal erstellt bzw. eingebunden obliegt die Verwaltung der Emulatoren dem Repository-Dienst. Unabhängig von der gewählten Emulatortechnologie umfassen die Anwendungsfälle der Gruppe 3 die Erstellung, Bearbeitung und Konfiguration virtueller Fabriken unter Verwendung der Infrastruktur und der Modellelemente der Emulati-

onsplattform. Virtuelle Fabriken und ihre Konfigurationen werden ebenfalls durch den Repository-Dienst verwaltet.

### Runtime

Die Anwendungsfälle der Gruppe 4 umfassen die Nutzung der zur build time modellierten virtuellen Fabriken. Auf Anforderung durch den Benutzer wird das Modell einer virtuellen Fabrik in der Ausführungsumgebung installiert, initialisiert und zur Ausführung gebracht. Während die virtuelle Fabrik mit der realen MES-Schicht interagiert, bietet die Emulationsplattform dem Benutzer verschiedene Möglichkeiten zur Überwachung und zum manuellen (steuernden) Eingriff. Insbesondere lässt sich der Zustand einzelner Emulatoren oder der gesamten virtuellen Fabrik vom Benutzer einfrieren, abspeichern und wieder fortsetzen. Zustandsdatensätze werden durch den Dienst 'Repository' verwaltet und können zu einem späteren Zeitpunkt erneut geladen werden. Nach dem Abschluss der Experimente mit einer virtuellen Fabrik steht es dem Benutzer frei, die zugehörigen Emulatoren aus der Ausführungsumgebung zu deinstallieren, oder die Installation für eine spätere Verwendung vorzuhalten.

Weitere Sichten auf die Emulationsplattform, die durch entsprechende GUIs implementiert werden, geben dem Anwender in Echtzeit einen Überblick über den Gesamtzustand der virtuellen Fabrik und bieten darüber hinaus die Möglichkeit zum *Drill-down*. Beim Drill-down wechselt der Benutzer die Betrachtungsebene von der virtuellen Fabrik zu einer ausgewählten virtuellen Anlage und erhält, soweit durch den ausgewählten Emulator unterstützt, lesenden und schreibenden Zugriff auf freigegebene Eigenschaften des Emulators. Dieses Werkzeug bietet die Voraussetzungen für eine detaillierte Online-Analyse des Verhaltens der virtuellen Anlage im Zusammenspiel mit dem MES.

Neben der Online-Analyse findet der Fall der Offline-Analyse Berücksichtigung, bei der ein Benutzer Ereignisse, die während des Betriebs der virtuellen Fabrik aufgezeichnet wurden, a posteriori auswertet. Eine besondere Rolle spielt in diesem Zusammenhang die Kommunikationshistorie, die für jeden Emulator geführt werden kann. Sie spiegelt den Nachrichtenverkehr zwischen Emulator und MES wieder. Werkzeuge zur Analyse der aufgezeichneten Ereignisse helfen dem Benutzer, Zusammenhänge zu verstehen.

## 5.4.2 Die Realisierung im Überblick

### Statische Sicht

Das Verteilungsdiagramm in Abbildung 5.18 stellt für ein Beispielszenario die Infrastruktur und darauf ausgeführte Systemkomponenten dar, die in ihrer Gesamtheit das eingangs diskutierte Architekturkonzept widerspiegeln. Die Ausführungsumgebung besteht hier aus drei physikalischen Rechnerknoten (in Abbildung 5.18 oben als Gerät, engl. device, dargestellt), die über ein lokales Netz gekoppelt sind. Über dieses Netz kann jedes Gerät mit anderen Geräten kommunizieren. Jeder LSE-Knoten stellt eine Ausführungsumgebung (engl. execution environment) in Form eines LSE-Agenten zur Verfügung, in der zum dargestellten Zeitpunkt jeweils ein Emulator installiert ist und



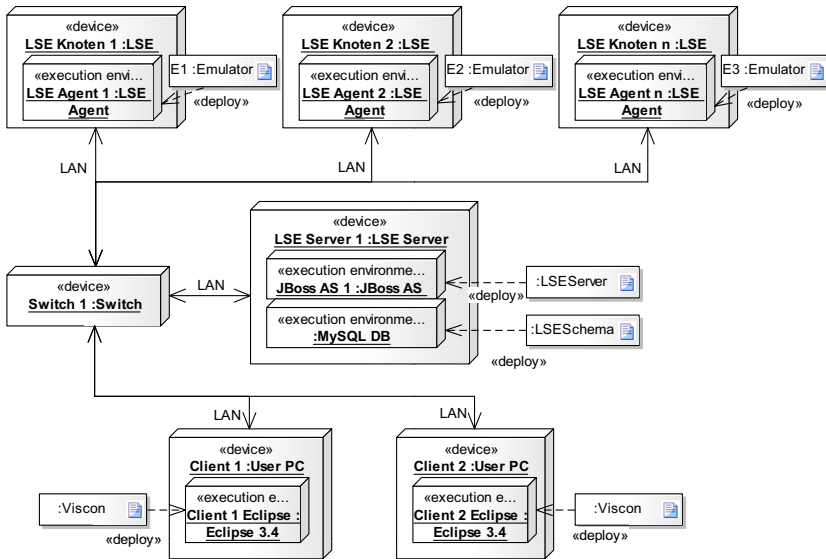


Abbildung 5.18: Verteilungssicht der realisierten Plattform in einem Beispiel-Szenario, das die drei Schichten des Architekturmodells widerspiegelt.

ausgeführt wird. Während LSE-Agenten im Zuge der Installation der Emulationsplattform als Betriebssystemdienste auf physikalischen Knoten installiert und konfiguriert werden und dadurch automatisch nach dem Start der Knoten zur Verfügung stehen, werden Emulatoren unter Berücksichtigung des Einsatzmodells durch die Emulationsplattform zur Laufzeit installiert. Externe Emulatoren werden, soweit sie nicht automatisch verteilbar sind, zusammen mit dem Agenten installiert. Agenten kommunizieren während der Ausführung von Szenarien in erster Linie untereinander und mit dem LSE-Server. Darüber hinaus wertet die Benutzerinteraktionsschicht zur Visualisierung der Emulation eine Untermenge der Multicast-Nachrichten aus, die von Agenten und Emulatoren generiert werden. Zur Realisierung der Drill-down Funktionalität baut die Benutzerinteraktionsschicht zusätzliche dedizierte Verbindungen zu einzelnen Agenten auf.

Die Plattformdienste werden durch den LSE-Server realisiert und im Netz zur Verfügung gestellt, der in der dargestellten Ein-Server-Konfiguration auf einem physikalischen Gerät innerhalb eines Applikationsservers läuft. Dieser greift auf eine relationale Datenbank auf demselben Gerät zurück. Sinnvoller Weise kommt beim physikalischen Gerät ein Mehrprozessor- oder ein Mehrkernsystem zum Einsatz, das den echt-parallelen Betrieb mehrerer Prozesse unterstützt. Neben dem LSE-Server laufen auf dem Applikationsserver die oben diskutierten Kommunikationsdienste, die LSE-Komponenten für ihre Interaktion nutzen.

Die beiden in Abbildung 5.18 unten dargestellten GUIs-Instanzen der Benutzerinterak-

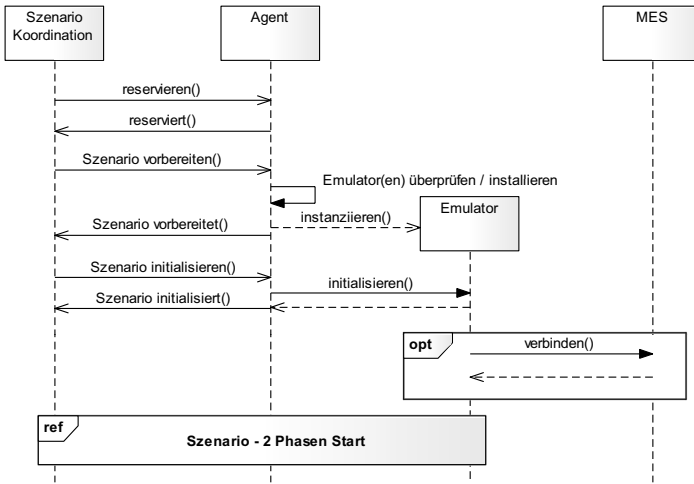


Abbildung 5.19: Beispiel: Ausschnitt aus dem Lebenszyklus eines Emulators – Szenario ausführen

tionsschicht laufen auf zwei weiteren physikalische Geräten, den Rechnern Client1 und Client2, die von zwei Anwendern parallel benutzt werden können. Auf beiden Rechnern ist eine Ausführungsumgebung installiert, in der die GUI-Komponenten der Emulationsplattform ausgeführt werden. Die für den Benutzer verfügbare Funktionalität wird durch die Interaktion der GUI-Instanz auf einem Rechner mit dem LSE-Server bzw. mit ausgewählten Agenten erbracht.

Das vorgestellte Verteilungsszenario wird für reale Installationen an die Anforderungen der Problemstellung im MES-Engineering angepasst. In der Ausführungsumgebung variiert die Anzahl der genutzten Knoten, die Anzahl der Agenten pro Knoten und die Zahl der Emulatoren pro Agent. Die konkrete Konfiguration ergibt sich aus Ressourcenangebot und -bedarf. Auf der Ebene der Plattformdienste variiert die Zahl der genutzten Server-Knoten, die Verteilung der Serveranwendungen auf diese Knoten und die Lastverteilung zur Laufzeit. Die Zahl der Rechnerknoten in der Benutzerinteraktionsschicht ergibt sich aus der Zahl der parallel arbeitenden Benutzer.

### Dynamische Sicht

Abbildung 5.19 zeigt für einen Emulator einen Ausschnitt aus dessen Lebenszyklus und dessen Steuerung durch den Plattformdienst “Szenariomanager”. Die Anforderung des Starts einer virtuellen Fabrik, deren Spezifikation durch den Repository-Dienst zur Verfügung gestellt wird, löst zunächst die Reservierung der benötigten Ressourcen, hier eines Agenten, durch den Szenariomanager aus und fordert diesen Agenten anschließend zur Installation eines oder mehrerer Emulatoren auf. Falls diese noch nicht auf dem

Agent verfügbar sind, werden die Emulatoren durch den Agenten vom Repository bezogen (nicht dargestellt) und installiert. Im nächsten Schritt werden die Emulatoren in den Arbeitsspeicher geladen und an der Laufzeitumgebung registriert. Die Emulatoren warten nun auf weitere Anweisungen durch den Szenariomanager. Der Szenariomanager löst in der Folge die Initialisierung aller Emulatoren der virtuellen Fabrik aus. Diese beinhaltet insbesondere die Wiederherstellung eines ausgewählten Ausgangszustands auf Basis von Zustandsdatensätzen, die über den Agent vom Repository-Dienst bezogen werden (der Zugriff auf die Zustandsdaten ist nicht dargestellt). Abhängig vom Protokollstapel, der für die Kommunikation zwischen MES und Emulator genutzt wird, und dessen Konfiguration kann der Emulator im Anschluss bereits die Verbindung zum MES aufbauen. Schließlich werden die Emulatoren der virtuellen Fabrik synchron gestartet und emulieren, beginnend vom gewählten Ausgangszustand, das Verhalten der realen Fabrik an der IT-Schnittstelle der involvierten Anlagen.

### 5.4.3 Die Realisierung der Ausführungsumgebung

#### Basistechnologien und Werkzeuge

Die beste Übereinstimmung der Basisanforderungen aus der Architekturdefinition zu existierenden Lösungen ergibt sich für Agentenplattformen, die in Testplattformen für die Ausführung von Lasttests oder als Mess- und Monitoring-Agenten genutzt werden. Von der Nutzung einer komplexen Multiagentenplattform ist hingegen kein gesteigerter Nutzen zu erwarten, da die zu lösende Aufgabenstellung diese Konzepte nicht erfordert. Nach einer Vorauswahl wird die im Eclipse Test & Performance Tools Platform (TPTP) Projekt [EclipseTPTP, 2009] enthaltene Agentenplattform im Detail auf ihre Eignung als technologische Basis untersucht. Diese ist grundsätzlich gegeben. So bietet der native C++ Code vielfältige Möglichkeiten zur Integration externer, also nicht nativer LSE-Emulatoren. Allerdings steht dem Einsatz dieser Technologie innerhalb des LSE-Projekts zum Zeitpunkt der Agentenentwicklung eine äußerst knappe Dokumentation entgegen, die die effektive Nutzung stark erschwert, da die Umsetzung der LSE-Anforderungen verschiedene Erweiterungen notwendig machen würde.

Vor diesem Hintergrund fällt die Entscheidung, eine speziell auf die Anforderungen der Ausführungsumgebung zugeschnittene, portable Agentenplattform in Java zu entwickeln, auf der der LSE-Agent aufgebaut werden kann. Um diese Agenten als Betriebssystemdienste betreiben zu können, werden jeweils auf die Zielplattform angepasste native Wrapper [JavaServiceWrapper, 2009] genutzt, die zur Laufzeit von entfernten Systemen aus über Java Management Extensions (JMX), einer Java-Technologie für das Management verteilter Systeme, verwaltet und überwacht werden. Die Kommunikation zwischen Agenten und Plattformdiensten bzw. zwischen Agenten und Benutzerinteraktionsschicht wird über Java Message Service (JMS) (asynchrone Interaktion) und Remote Method Invocation (RMI) (synchrone Interaktion) abgewickelt.

## Der LSE-Agent Implementierungsansatz

Innerhalb der Ausführungsumgebung übernehmen Agenten die Rolle des Containers, in dem Emulatoren zur Laufzeit installiert, ausgeführt und deinstalliert werden. Nicht-native, externe LSE-Emulatoren werden in eigenen Prozessen außerhalb des Agenten ausgeführt. In diesem Fall übernimmt ein vom Agenten ausgeführter nativer Emulator die Rolle eines *Stellvertreters* (engl. proxy) für den externen Emulator. Die Zuständigkeit für die Verwaltung der Lebenszyklen von Emulatoren liegt bei den verwaltenden Agenten. Abhängigkeiten zwischen Emulator und Agent werden in der Form aufgelöst, dass der Agent dem Emulator die Implementierung benötigter Schnittstellen im Rahmen seiner Initialisierung übergibt (engl. *dependency injection*).

Die von einem Agent verwalteten Emulatoren können, eine geeignete Hardwareplattform vorausgesetzt, über mehrere Prozessoren oder Kerne verteilt werden. Um innerhalb eines Agenten die Parallelität von Emulatoren abzubilden, wird jeder Emulator in einem eigenen Thread ausgeführt. Sobald ein Agent auf einem Knoten gestartet wurde, registriert er sich beim Agentenmanagement-Dienst der Plattform und kann von diesem Zeitpunkt an über die Plattform gesteuert und überwacht werden. Eingehende Nachrichten der Plattform an den Agenten bzw. von ihm verwaltete Emulatoren werden in einem First In First Out (FIFO)-Puffer zwischengespeichert und reihenfolgetreu an den jeweiligen Empfänger zur Bearbeitung weitergeleitet.

Die Agenten bilden für die Emulationsplattform einen wichtigen Baustein im Konzept der hochgradig skalierbaren, integrationsfähigen Architektur. Der gewählte Implementierungsansatz erlaubt eine hohe Zahl von Agenten, die gleichzeitig bei der Plattform registriert und parallel an der Ausführung eines Szenarios teilnehmen können. Die Anzahl von einigen hundert Agenten, die für die Ausführung eines realistischen Testszenarios vorstellbar ist, läßt sich auf Basis der beschriebenen Architektur problemlos erreichen. Neben dem Parallelbetrieb von Agenten erlaubt die Implementierung die parallele Ausführung mehrerer Emulatoren innerhalb eines Agenten.

Die maximale Zahl von Emulatoren, die auf einem Agenten parallel betrieben werden kann, hängt vom Ressourcenangebot des Agenten, dem Ressourcenbedarf der Emulatoren und den Anforderungen an das Zeitverhalten der Emulation ab. Einen deutlichen Einfluss auf die Auslastung des Systems hat beispielsweise der Nutzungsgrad der Mechanismen zur Datenerfassung. Da sich die Grenzen für den konkreten Fall a priori lediglich abschätzen lassen, können die Agenten über oben genannte Management-Mechanismen zur Laufzeit überwacht werden, um einen Betrieb im gewünschten Betriebsbereich sicherzustellen. Wichtige Kenngrößen sind in diesem Zusammenhang der verfügbare Heap, die prozentuale Auslastung der Prozessorkerne und jene der Netzwerkanbindung. Für den Fall, dass sich Emulatoren gemeinsame Ressourcen teilen, helfen die beschriebenen Management-Mechanismen bei der Einschätzung, in wie weit virtuelle Anlagen, die aus konzeptioneller Sicht nebenläufig betrieben werden sollten, durch die gemeinsame Ressourcennutzung in unerwünschter Weise gekoppelt werden.

## 5.4.4 Die Realisierung der Plattformdienste

### Basistechnologien und Werkzeuge

Für die Umsetzung der beschriebenen Plattformdienste stehen mit dem aktuellen Stand der Technik einige sehr mächtige Basistechnologien zur Verfügung. In diesem Zusammenhang sind in erster Linie die Java-Technologie von Oracle (ehemals Sun Microsystems) [Java, 2009] und Microsofts .Net-Technologie [Net, 2009] zu nennen. Vor allem die große Zahl qualitativ hochwertiger, frei verfügbarer Bibliotheken, Rahmenwerke und Integrationsplattformen auf Java-Basis haben zu der Entscheidung geführt, für die Implementierung der Schicht der Plattformdienste Java zu nutzen.

Mit dem Java Enterprise Edition Software Development Kit (Java EE SDK) stellt Oracle Werkzeuge zur Softwareentwicklung zur Verfügung, die die Entwicklung komplexer verteilter Unternehmensanwendungen in Umsetzung von SOA Paradigmen erlauben. Neben dieser Basistechnologie wird die Integrationsplattform (engl. *Middleware*) ausgewählt, die die durch Java EE spezifizierten Programmierschnittstellen implementiert. Für die LSE-Implementierung werden frei verfügbare Middleware-Produkte der JBoss Community [JBoss, 2009], der JBoss Application Server und die Infrastruktur zur nachrichtenbasierten Kommunikation (engl. Message oriented Middleware), genutzt. Diese Wahl liegt in der hohen Verbreitung dieser Produkte, deren Stabilität, Dokumentationsstand und der Breite der unterstützten Funktionalität begründet. Eine relationale Datenbank auf Basis von MySQL [MySQL, 10] übernimmt die persistente Speicherung anfallender Daten.

### Der LSE-Server Implementierungsansatz

Die Geschäftslogik der LSE-Plattformdienste wird in Form dedizierter Komponenten, sogenannter Java Enterprise Beans, implementiert, die jeweils eine definierte Dienst-Schnittstelle anbieten, einem von der Java Enterprise Edition (Java EE) Spezifikation vorgegebenen Lebenszyklus gehorchen und zur Laufzeit von einem Java EE Container verwaltet werden. Der Container stellt den Enterprise Beans eine Vielzahl häufig in Geschäftsanwendungen benötigter Dienste zur Verfügung. Dienstanwender, wie LSE-Agenten, externe Prozesse oder graphische Benutzeroberflächen, greifen über verschiedene, vom Applikationsserver unterstützte, Protokolle auf die durch Enterprise Beans implementierten Dienste zu. Für die Implementierung der Emulationsplattform kommen in erster Linie synchrone Aufrufe über Referenzen innerhalb des Containers, synchrone Remote Procedure Calls (RPCs) über Java RMI und asynchrone Aufrufe über JMS zum Einsatz.

Eine zusätzliche Schicht zwischen dem eingesetzten Persistenz-Dienstanbieter des Anwendungsservers und den die Geschäftslogik implementierenden Enterprise Beans sorgt für die Versionierung von Modellelementen, die der Anwender auf Basis des LSE-Metamodells erzeugt. Die Versionierung eröffnet dem Anwender die Möglichkeit, auf Vorversionen seines Modells zuzugreifen.

In der Umgebung des Application Servers stehen zahlreiche Basisdienste zur Verfügung, die die Implementierung der Plattformdienste als ein stabiles Mehrbenutzersystem erleichtern. Stellvertretend seien Mechanismen zur *Authentisierung* und *Autorisierung* genannt. Der erstgenannte Mechanismus sorgt für die zweifelsfreie Identifikation von Anwendern, die mit Plattformdiensten interagieren. Er stellt damit also sicher, dass ein Anwender tatsächlich der ist, der zu sein er vorgibt. Der zweitgenannte Mechanismus stellt sicher, dass ein Benutzer nur solche Aktionen im System ausführen kann, für die er die notwendigen *Berechtigungen* besitzt. Die Implementierung der Plattformdienste greift, neben den Mechanismen zur Authentisierung und Autorisierung, auf weitere Basisdienste zurück.

Für die Schicht der Plattformdienste bedeutet die Fähigkeit zur Skalierung die Möglichkeit, bei einer steigenden Zahl von Agenten, Emulatoren, Endanwendern und gleichzeitig ausgeführten Szenarien und der damit einhergehenden steigenden Belastung der Plattformdienste, möglichst linear, mitwachsen zu können. Diese Anforderung wird durch den ausgewählten Technologiestapel unterstützt. So bietet die Middleware Mechanismen zur Cluster-Konfiguration, also der Gruppierung mehrerer Anwendungsserver zu einem Serververbund, und zur Implementierung geeigneter Strategien der Lastverteilung (engl. *load balancing*) über dieses Cluster. Die Nutzbarkeit dieser Mechanismen wird durch ein geeignetes Design der LSE-Serverkomponenten sichergestellt.

### 5.4.5 Die Realisierung der Benutzerinteraktionsschicht

#### Basistechnologien und Werkzeuge

Während für die Implementierung jedes der in Abbildung 5.17 genannten Anwendungsfälle für sich genommen eine Vielzahl von Basistechnologien zum Einsatz kommen könnten, schränken die zusätzliche Anforderung der Integrierbarkeit der implementierten Lösungen in Entwicklungs- und Testumgebungen und der Offenheit für zukünftige Erweiterungen den möglichen Lösungsraum deutlich ein. Aufgrund der beiden letztgenannten Anforderungen wird die Eclipse Plattform [Eclipse, 2009] als Basistechnologie für alle graphischen Benutzeroberflächen in der Benutzerinteraktionsschicht gewählt. Zunächst als Entwicklungsumgebung für Java-Anwendungen konzipiert, hat Eclipse sich in der Zwischenzeit zu einer mächtigen Plattform für die Implementierung grafischer Anwendungen entwickelt. Eclipse-Anwendungen bestehen aus einer Zusammenstellung von Komponenten, so genannten *Plug-ins*, die definierte Dienste zur Verfügung stellen. Ihr Lebenszyklus wird von der Eclipse-Umgebung verwaltet. Von der Eclipse-Entwicklergemeinde werden eine Vielzahl solcher Plug-ins frei oder unter kommerziellen Lizenzen zur Nutzung angeboten. Die Möglichkeit zur Wiederverwendung und Erweiterung dieser vorhandenen Funktionalität beschleunigt die Entwicklung neuer Anwendungen. In den folgenden Abschnitten werden einige Kerntechnologien ausführlicher diskutiert, die für die Realisierung relevant sind.

Im Eclipse-Umfeld existieren einige mächtige Werkzeuge in Form von Plug-ins, die die Integration von modellgetriebenen Ansätzen für die Softwareentwicklung in eigene Anwendungen wesentlich erleichtern. An erster Stelle ist in diesem Zusammenhang das Eclipse Modeling Framework (EMF) zu nennen. EMF stellt Anwendungen die Infrastruktur für

die Arbeit mit strukturierten Datenmodellen zur Verfügung. Im Zentrum steht das EMF Meta-Metamodell namens Ecore, das ausgehend von der OMG MOF-Spezifikation entstanden ist. Mit Hilfe der Ecore-Primitive kann ein Anwendungsentwickler, der EMF einsetzt, eigene Metamodelle spezifizieren. Diese Modellspezifikationen werden in XMI beschrieben und können mit EMF-Werkzeugen in Java Klassen transformiert werden, deren Instanzen mittels Adapterklassen angezeigt und über ein Framework, das die Bearbeitung des Modells über Kommandos erlaubt, editiert werden können. Damit steht eine wichtige Basisfunktionalität für die Entwicklung von Modellierungswerkzeugen innerhalb einer Eclipse-Anwendung zur Verfügung. Für ein praktisch nutzbares Modellierungswerkzeug fehlt bisher noch die Fähigkeit zur Interaktion mit dem Anwender, also ein geeigneter Modell-Editor, mit dem ein Anwender auf Basis des vom Anwendungsentwickler definierten Meta-Modells eigene Modelle betrachten und erstellen kann. Die Basisfunktionalität für entsprechende grafische Editoren bietet das Eclipse Graphical Editing Framework (GEF), das damit an zweiter Stelle zu nennen ist. Mit dem Graphical Modeling Framework (GMF) steht ein drittes Hilfsmittel zur Verfügung, das, zumindest für weite Teile der Infrastruktur, die Generierung von GEF-Editoren zu EMF-Modellen unterstützt.

Obleich die Eclipse TPTP-Plattform in der Schicht der Ausführungsumgebung nicht zum Zug kommt, bietet sich die Nutzung eines Teils der verfügbaren Funktionalität innerhalb der LSE-GUI an. Insbesondere die Funktionen zur Auswertung von Log-Daten, speziell die zur Korrelation von Ereignissen, ermöglicht ein besseres Verständnis der Vorgänge zur Laufzeit auf der Emulationsplattform und kann damit einen wichtigen Beitrag bei der Analyse von unerwartetem Verhalten liefern.

Für den interaktiven Management-Zugriff auf einzelne Agenten und/oder Emulatoren kommt die JConsole zu Einsatz, die nicht als Eclipse-Plug-in zur Verfügung steht, sondern Bestandteil des Java Software Development Kit (SDK) ist und Javas natives GUI-Framework *Swing* nutzt. Allerdings wird sie aus der Eclipse-GUI heraus aufgerufen und initialisiert.

### Der LSE-GUI Implementierungsansatz

Die Anwendung LSE-Viscon wird mit oben skizzierten Funktionsumfang unter Verwendung der beschriebenen Frameworks in Form mehrerer Eclipse-Plug-ins implementiert. Für den Betrieb werden diese Plugins entweder in einer eigenen Eclipse Rich Client Platform (RCP)-Anwendung gebündelt oder einer bestehenden Eclipse-Installation hinzugefügt. Nach Installation der Plugins bzw. dem Start der RCP-Anwendung und der erfolgreichen Anmeldung des Benutzers stehen verschiedene *Views* und *Editoren* zur Verfügung, die dem Benutzer unterschiedliche Sichten auf die Emulationsplattform bieten. Kombinationen aus Views und Editoren werden zu Eclipse-*Perspektiven* gebündelt, die jeweils eine Gruppe zusammengehöriger Anwendungsfälle unterstützen. Abbildung 5.20 zeigt die Perspektive zur Bearbeitung von Strukturmodellen.

Konzeptionell ist Viscon als ein so genannter *Rich Client* ausgeführt. Die Anwendung greift zur Realisierung der für den Anwender sichtbaren Funktionalität auf Plattformdienste zurück, verantwortet aber einen Teil der Zustandsverwaltung selbst und erweitert darüber hinaus die Geschäftslogik der Plattformdienste an ausgewählten Punkten. So

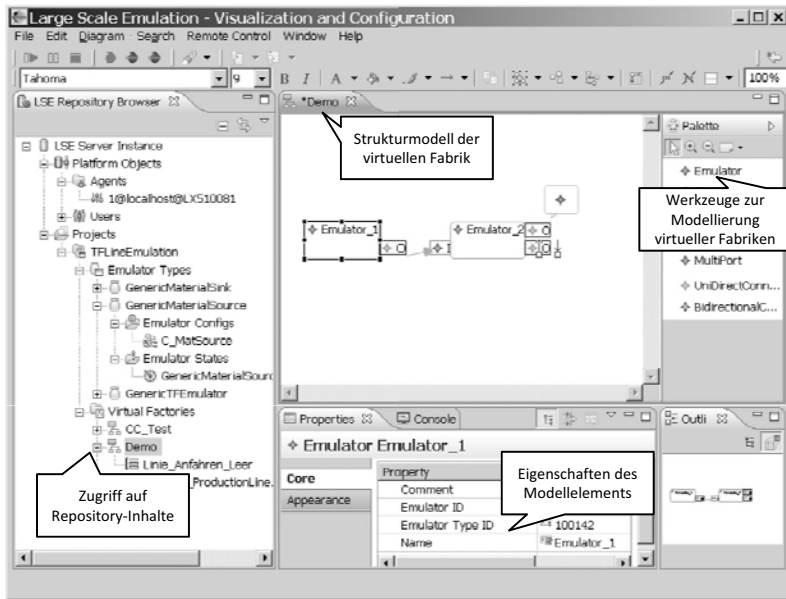


Abbildung 5.20: Viscon-Perspektive für die Bearbeitung von Strukturmodellen

sorgt Viscon beispielsweise lokal für den syntaktisch richtigen Aufbau der Strukturmodelle, während der Nutzer diese editiert. Die folgenden Gruppen von Anwendungsfällen werden bisher von Viscon implementiert:

*Modellelemente verwalten.* Die 'Repository Browser View' stellt den Einstiegspunkt des Anwenders dar und ermöglicht diesem die Navigation durch das Datenmodell der verbundenen Emulationsplattform. Das Datenmodell wird im Browser baumförmig dargestellt, wobei auf oberster Ebene zwischen Plattformressourcen und projektspezifischen Ressourcen unterschieden wird. Plattformressourcen werden über Projektgrenzen hinaus genutzt. Projekte helfen Modelle bedarfsorientiert zu strukturieren. Die meisten 'Verwaltungs'-Anwendungsfälle werden aus der Repository Browser View heraus initiiert.

*Emulatortyp erstellen.* Neben den oben beschriebenen Frameworks zur Unterstützung von Ansätzen der modellgetriebenen Softwareentwicklung steht mit der Eclipse-Plattform eine vollständige Java-Entwicklungsumgebung zur Verfügung, die mit Hilfe der integrierten Erweiterungsmechanismen für die Emulatorentwicklung angepasst wird. Von besonderer Bedeutung ist das LSE-Emulator-SDK. Es beinhaltet Rahmenwerke zur Erstellung nativer Emulatoren, die später in der Ausführungsumgebung betrieben werden können, entsprechend der konzeptionellen Diskussion in Kapitel 5.2 und 5.3 und Werkzeuge für die Entwicklung. Bestandteil des SDK sind auch Rahmenwerke zur Anbindung dreier in der PV-Industrie genutzter IT-Schnittstellentypen an Emulatoren. Neben dem PVECI Standard werden zwei proprietäre Protokolle unterstützt, die zum jetzigen Zeitpunkt



```

<?xml version="1.0" encoding="UTF-8"?>
<xml:XMI xmi:version="2.0" xmlns:xmi="http://www.omg.org/XMI"
xmlns:notation="http://www.eclipse.org/gmf/runtime/1.0.1/notation"
xmlns:viscon="http://ipa.fraunhofer.de/lse/viscon">
<viscon:VirtualFactory xmi:id="__7FFMCiEd-RO935ayGmxQ">
  <emulator xmi:type="viscon:Emulator" xmi:id="AwId4CIsEd-RO935ayGmxQ"
name="Emulator_1" comment="Demo Emulator_1" emulatorID="19200142"
emulatorTypeID="100142">
    <port xmi:type="viscon:OutPort" xmi:id="C5qWMCIsEd-RO935ayGmxQ"
outgoing="_HHhdwCIsEd-RO935ayGmxQ"/>
  </emulator>
  |

```

Abbildung 5.21: Beispiel: Ausschnitt eines serialisierten Strukturmodells einer virtuellen Fabrik

in der PV-Industrie häufig zum Einsatz kommen. Für deren Implementierung kommen kommerzielle InFrame®Synapse [InFrame, 2010] Bibliotheken zum Einsatz.

*Virtuelle Fabrik modellieren.* Für das Betrachten und Editieren von Strukturmodellen stellt Viscon einen dedizierten *Editor* zur Verfügung. Alle Modellelemente des Strukturmodells enthalten im Editor eine grafische Repräsentation und können in dieser Ansicht bearbeitet werden. Die wichtigsten Elemente sind *Emulatoren*, *Ports* und *Konnektoren*. Der Austausch von Modellen zwischen Viscon und den Plattformdiensten erfolgt über XMI Dokumente (siehe auch Abbildung 5.21). Auf Basis der Strukturmodelle werden in der Interaktion mit dem Anwender Einsatz- und Szenarienmodelle erzeugt und im Repository abgelegt. Wiederverwendbare Modellelemente können aus dem Repository abgerufen und in das zu bearbeitete Modell eingefügt werden.

*Szenario ausführen.* Die Ausführung von Szenarien wird durch den Anwender über eine Variante des Stukturmodelleditors gesteuert. Wird ein Szenario zur Ausführung geöffnet, präsentiert der Editor das zugehörige Strukturmodell “nur lesend”. Dem Benutzer stehen über die Oberfläche die Steuerbefehle für Szenarien, Installieren, Initialisieren, Pause, Start und Stop, zur Verfügung. Steuerbefehle, die der Benutzer über die Oberfläche absetzt, werden in Form von Anforderungen asynchron an den Szenario-Koordinationsdienst weitergeleitet, der in der Interaktion mit den Agenten die Anforderungen umsetzt. Über die Steuerbefehle auf Szenarienebene hinaus besteht die Möglichkeit, Agenten und Emulatoren einzeln anzusteuern.

Die Granularität der manuellen Eingriffsmöglichkeiten auf der Ebene des Emulators hängt vom Emulatortyp und vom Umfang ab, in dem er für die Steuerung über JMX instrumentiert wurde. Die Bandbreite der Möglichkeiten reicht vom Zugriff auf wenige ausgewählte Kommandos, wie “Anlage entleeren”, bis hin zu komplexen anlagenspezifischen GUIs. Die Visualisierung des aktuellen Zustands der virtuellen Fabrik für den Anwender erfolgt in erster Linie aggregiert auf das Szenario annähernd in Echtzeit. Dieser wird in der grafischen Repräsentation des Strukturmodells durch die Farbgebung der Modellelemente und/oder die Anzeige von zusätzlichem Text dargestellt. Detaillierten Einblick in den Zustand einzelner Emulatoren erhält der Nutzer über die JMX-Schnittstelle der Emulatoren. Wie für den Fall der Steuerung hängt auch für den Fall der Überwachung der verfügbare Detaillierungsgrad vom Umfang der Instrumentierung des betrachteten Emulatortyps ab. Weitere Einblicke bietet die Analyse der detaillierten Kommunikationslogs der Emulatoren.

<b>Umsetzung der Anforderungen an die Emulationsplattform</b>	
<b>1. Modellierung</b>	<ul style="list-style-type: none"><li>a) Abbildung realer Fabriken durch virtuelle Fabriken in Form gekoppelter Anlagenemulatoren, die das Verhalten jeweils einer Anlage aus Sicht ihrer IT-Schnittstelle beschreiben [5.1.2, 5.2.1, 5.2.3]</li><li>b) Unterstützung eines variablen, das heißt den Anforderungen des Anwendungsfalls angemessenen Detaillierungsgrads des Verhaltens der Anlagen [5.2.1, 5.3.1]</li><li>c) Modellelemente zur Abbildung des Logistik-Subsystems in einem Detaillierungsgrad der eine automatische Ablaufsteuerung erlaubt [5.2.2, 5.2.3, 5.3.4]</li><li>d) Unterstützung der Wiederverwendung zentraler Modellelemente [5.1.1, 5.1.2, 5.2.1, 5.3.1]</li></ul>
<b>2. Integration</b>	<ul style="list-style-type: none"><li>a) Fähigkeit zur (minimal-invasiven) Integration unterschiedlicher Emulator-Technologien [5.1.2, 5.2.1, 5.5.1]</li><li>b) Plattformunabhängigkeit im Umfeld der Emulatoren [5.4.3]</li><li>c) Integrierbarkeit der Emulationsplattform in die MES-Entwicklungs- und Testumgebung [5.1.4, 5.4.5]</li></ul>
<b>3. Ausführung</b>	<ul style="list-style-type: none"><li>a) Fähigkeit zur Nachbildung des Verhaltens der Anlagen in Echtzeit [5.3.1, 5.3.2]</li><li>b) Anpassbarkeit der Ablaufgeschwindigkeit [5.3.2]</li><li>c) Fähigkeit zur Koordination von Szenarien, die beliebig viele Anlagen im Umfang des spezifizierten Skalierungsbereiches enthalten [5.1.2, 5.3.2, 5.3.3, 5.3.4]</li><li>d) Analyse- und Interaktionsmöglichkeit für den Anwender bis auf die Ebene einzelner Emulatoren („Drill-down“) [5.1.3]</li><li>e) Fähigkeit zur Auswertung der Konversation zwischen Anlage und anbindendem IT-System [5.1.3]</li><li>f) Zustandsverwaltung [5.2.1, 5.2.3, 5.3.1, 5.3.3]</li></ul>
<b>4. Randbedingungen</b>	<ul style="list-style-type: none"><li>a) Skalierbarkeit [5.1.1, 5.1.2]</li></ul>

Abbildung 5.22: Abgleich des Metamodells und der Implementierung der Emulationsplattform mit den Anforderungen aus Abbildung 3.9

## 5.5 Fazit: Anforderungen an die Emulationsplattform

Mit der Definition des in diesem Kapitel beschriebenen Metamodells und der prototypischen Implementierung der Emulationsplattform wird schrittweise das Anforderungsprofil an eine Emulationsplattform umgesetzt, das aus der Analyse in Kapitel 3 abgeleitet worden war. Um den Lesefluss zu erleichtern, wurde im Fließtext dieses Kapitels auf die detaillierte Referenzierung einzelner Anforderungen verzichtet. Abbildung 5.22 bietet dem interessierten Leser stattdessen im Überblick den Zusammenhang zwischen den einzelnen Anforderungen aus Kapitel 3 und den Abschnitten dieses Kapitels, die diese Anforderungen jeweils in der Umsetzung aufgreifen.

# 6 Entwicklung des Verfahrens des emulationsgestützten MES-Engineering

Während im Kapitel 5 das Metamodell der Emulationsplattform und die Implementierung der Werkzeuge, die das Metamodell in der Praxis nutzbar machen, im Vordergrund stehen, erweitern die folgenden Abschnitte die “Modell- und Werkzeugwelt” um ein Vorgehen, das einen nutzbringenden Einsatz der Emulationsplattform in der Projektpraxis ermöglicht. Im ersten Teil des Kapitels wird der emulationsgestützte MES-Engineering-Prozess entwickelt, der sich in den Fabrik- bzw. MES-Lebenszyklus der PV-Industrie (siehe Kapitel 3.3.2) einbetten lässt. Der zweite Teil des Kapitels ist der integrierten Entwicklung virtueller Fabriken gewidmet, deren Verfügbarkeit im ersten Teil vorausgesetzt wird.

## 6.1 Der emulationsgestützte MES-Engineering-Prozess

Nach der Analyse in Kapitel 3.3.2 entspricht ein wesentlicher Teil der zu bearbeitenden Problemstellung der von Softwareprojekten unter Berücksichtigung der beschriebenen Rahmenbedingungen. Für deren strukturierte Bearbeitung stehen aus der Disziplin der Softwaretechnik eine Vielzahl von Methoden zur Verfügung (vgl. Kapitel 3.4). Dabei wurde speziell auf das Potential agiler Elemente hingewiesen, die vor dem Hintergrund des aktuellen Entwicklungsstands der PV-Industrie Vorteile erwarten lassen.

### 6.1.1 Ein iterativer, inkrementeller MES-Engineering-Prozess

Als Ausgangspunkt für die Entwicklung des emulationsgestützten MES-Engineering-Prozesses wird für das MES-Engineering ein *iteratives, inkrementelles* Vorgehen auf Basis der Prozessarchitektur des Rational Unified Process (RUP) (vgl. Kapitel 3.4) definiert, in dem *agile Elemente* integriert werden. Der emulationsgestützte MES-Engineering-Prozess lässt sich in analoger Art und Weise auf der Grundlage anderer Vorgehensmodelle aufbauen. Um eine vergleichbare Verzahnung und damit einen vergleichbaren Nutzwert des Emulationsaspekts zu erreichen, müssen diese alternativen Vorgehensmodelle in der Lage sein, die drei hervorgehobenen Eigenschaften des Vorgehens abzubilden.

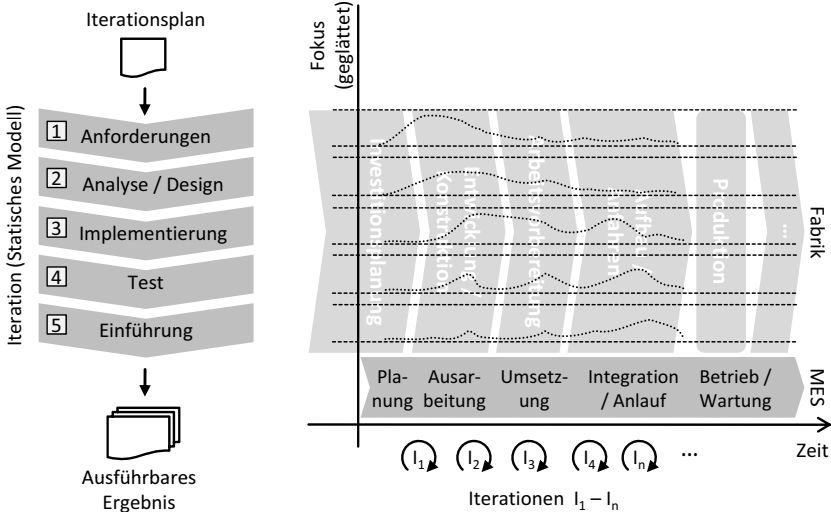


Abbildung 6.1: MES-Engineering auf der Grundlage eines iterativen, inkrementellen Vorgehens (Basis RUP, siehe Kapitel 3.4)

Abbildung 6.1 zeigt die wichtigsten Konzepte des MES-Engineering-Prozesses im Überblick. Im Zentrum von Abbildung 6.1 rechts ist der für das Vorgehen wesentliche Abschnitt des MES-Lebenszyklus dargestellt, der in der Analyse in Kapitel 3.3.2 in Verbindung mit dem Fabriklebenszyklus untersucht wurde. Diesem Abschnitt des Lebenszyklus sind fünf Projektphasen des MES-Engineering-Prozesses zugeordnet, die in Anlehnung an die RUP-Architektur wie folgt bezeichnet werden: *Planung*, *Ausarbeitung*, *Umsetzung*, *Integration/Anlauf* und *Betrieb/Wartung*. Oberhalb des MES-Lebenszyklus ist zur Orientierung der erste Abschnitt des in Kapitel 2.1 eingeführten Fabriklebenszyklus abgebildet. Die zeitliche Synchronisation der beiden Lebenszyklen in Abbildung 6.1 entspricht der für den Fall eines Fabrikneubaus, bei dem parallel zum Neubau das MES-Engineering erfolgt. Für den Fall einer nachträglichen Einführung des MES oder der Migration einer bestehenden MES-Lösung des MES kommt derselbe Engineering-Prozess zum Einsatz. Er verschiebt sich lediglich entlang der Zeitachse gegenüber dem Fabriklebenszyklus. Für eine korrekte Darstellung ist darüber hinaus der Fabriklebenszyklus zeitlich entsprechend zu skalieren.

Die Aufgaben des MES-Engineering, die in Kapitel 3.3.2 im Kontext des Fabriklebenszyklus beschrieben wurden, werden in den MES-Engineering-Prozess überführt. Jeder Phase des MES-Engineering-Prozesses werden damit charakteristische Aufgabenstellungen zugeordnet. Entsprechend der iterativen Eigenschaft des Vorgehens werden die Aufgaben jeder Phase verteilt auf eine oder mehrere *Iterationen* bearbeitet. Eine Iteration kann als ein abgeschlossenes Teilprojekt betrachtet werden. Das Ziel und der Umfang jeder Iteration werden vor ihrer Bearbeitung im *Iterationsplan* klar definiert. Grundsätzlich

ist die Möglichkeit vorgesehen, Iterationen mit definiertem Endtermin oder mit flexiblem Endtermin zu planen. Ein flexibler Endtermin bietet die Möglichkeit, innerhalb der Iteration durch die Investition von Mehraufwand auf ungeplante Ereignisse zu reagieren. Ein weiterer Freiheitsgrad zur Reaktion auf ungeplante Ereignisse besteht in der Anpassung des Umfangs der Iteration. Beide Strategien werden in der Literatur mit ihren Vor- und Nachteilen ausführlich diskutiert (siehe Kapitel 3.4). In der Praxis, insbesondere im Falle eines Fabrikneubaus, stellt das gesamte MES-Engineering-Projekt lediglich ein Teilprojekt in einer komplexeren Projekthierarchie dar. Daraus ergeben sich eine Reihe von Synchronisationspunkten des MES-Projekts mit anderen Teilprojekten zu definierten Meilensteinen. Vor diesem Hintergrund bietet sich eher die Planung von Iterationen mit festem Endermin und die Flexibilisierung des Iterationsumfangs an.

Das Ziel einer für ein spezifisches Produktionssystem angepassten MES-Lösung wird damit in einer Abfolge mehrerer Iterationen  $I_i$  (in der Abbildung  $I_1$  bis  $I_n$ ) erarbeitet. Die inkrementelle Eigenschaft des Vorgehens beschreibt, dass jede dieser  $n$  aufeinanderfolgenden Iterationen über die Zeit einen Beitrag zur finalen MES-Lösung liefert. Beide Eigenschaften entlasten Engineering-Projekte dadurch, dass die Notwendigkeit zur vollständigen Durchdringung aller Anforderungen zum Beginn des Projektes und deren Überführung in die weiteren Entwicklungsphasen in einem einzigen Schritt vermieden wird. In entsprechender Weise wird die technische Problemstellung schrittweise gelöst. Damit werden Lerneffekte nutzbar und Projektrisiken minimiert. Idealer Weise werden dem Kunden frühzeitig im Projekt Teilergebnisse zur Verfügung gestellt, die bereits einen konkreten Nutzen haben. Darüber hinaus erlaubt dieses Vorgehen ein höheres Maß an Flexibilität bei Änderungen der Anforderungen: Während der überschaubare Plan einer Iteration für klare Rahmenbedingungen sorgt, lassen sich Änderungen in Folgeiterationen einplanen.

Jede Iteration  $I_i$  wird wiederum durch das statische Modell beschreiben, das in Abbildung 6.1 links dargestellt ist. In jeder Iteration werden, unter Berücksichtigung des Iterationsplans, die Aktivitäten <sup>1</sup> (1) - (5) durchlaufen, die am Ende ein ausführbares, und damit testbares, Ergebnis liefern sollen. Die genannten Aktivitäten klassifizieren zunächst die innerhalb der Iteration zu bearbeitenden Aufgaben. Darüber hinaus kann für jede Aktivität ein standardisiertes Vorgehen mit den beteiligten Rollen, den zu generierenden Ergebnissen und Abhängigkeiten definiert werden. Jede der Aktivitäten einer Iteration ist in Abhängigkeit zur Phase im MES-Engineering-Prozess, der die Iteration zugeordnet ist, unterschiedlich gewichtet. Die Gewichtung der Aktivitäten über den MES-Engineering-Prozess ist in Abbildung 6.1 rechts jeweils über den Verlauf des Indikators *Fokus* über die Zeit skizziert. Während der Fokus von Iterationen in den frühen Phasen des MES-Engineering-Prozesses eher auf den Aktivitäten 'Anforderungen' und 'Analyse / Design' liegt, liegt er in der Phase 'Integration / Anlauf' des MES-Engineering-Projekts eher auf den Aktivitäten 'Implementierung', 'Test' und 'Einführung'. Die Aufgabenstellung der Anforderungsanalyse spielt in früheren Iterationen des Projekts eine wesentlich größere Rolle (höherer Fokus-Wert) als in späteren Iterationen, da zunächst die grundsätzliche Ausrichtung des Projekts festgelegt wird. Entsprechend Kapitel 3.3.2 ist die Analyse der charakteristischen Eigenschaften des Produktionssystems, die einen hohen Einfluss auf

---

<sup>1</sup>Der Aspekt "Geschäftsprozess", der im RUP durch einen eigenen Workflow beschrieben wird, wird in Aktivität (1) berücksichtigt.

die entstehende MES-Lösung hat, in diesen Iterationen vorzunehmen.

Im skizzierten Vorgehen werden bereits eine Reihe agiler Elemente berücksichtigt (siehe Kapitel 3.4); insbesondere ermöglicht das Vorgehen das Lernen im laufenden Prozess und betont die Bedeutung ausführbarer Ergebnisse. Für die konkrete Ausgestaltung des Vorgehens bedeutet der agile Ansatz darüber hinaus, den Overhead des Vorgehens, also Aufwände, die nicht unmittelbar dem Projektziel dienen, auf das absolut Notwendige zu beschränken. Für die Ausgestaltung der Iterationen soll nochmals auf die Beutung ausführbarer Ergebnisse schon im möglichst frühen Phasen des Engineering-Projekts hingewiesen werden. Darüber hinaus ist der Zusammenarbeit mit dem Kunden innerhalb der Iterationen eine hohe Bedeutung zugemessen. Beide Maßgaben sollen sicherstellen, dass die entstehende MES-Lösung bestmöglich an das Produktionssystem angepasst ist, technische Herausforderungen klar identifiziert und Risiken minimiert werden.

### 6.1.2 Virtuelle Fabriken effektiv und effizient nutzen

Mit dem beschriebenen MES-Engineering-Prozess steht das Rahmenwerk eines Vorgehens zur Verfügung, in dem sich virtuelle Fabriken *effektiv*, also wirksam im Hinblick auf die in den Kapiteln 1.2 und 2.5 formulierten Problemstellungen, einsetzen lassen. Die Wirkung wird über einen schnellen Feedbackmechanismus an den Ansatzpunkten erzeugt, die in der Analyse des MES-Engineering-Prozesses in Kapitel 3.3 identifiziert wurden. Die Gesamtwirkung des Einsatzes virtueller Fabriken im MES-Engineering-Prozess ergibt sich prinzipiell aus deren Einsatz in allen fünf Prozessphasen. Abbildung 6.2 fasst die potentiellen Einsatzszenarien virtueller Fabriken in jeder dieser Phasen zusammen. Welche Einsatzszenarien im konkreten Fall umgesetzt werden, um virtuelle Fabriken effizient zu nutzen, ist auf der Basis von Wirtschaftlichkeitsbetrachtungen unter den gegebenen Randbedingungen zu entscheiden.

Phase MES-Engineering	Nutzung virtueller Fabriken
Planung	<ul style="list-style-type: none"> <li>• Unterstützung der MES-Auswahl (Nachweis wichtiger Systemeigenschaften – z.B. Funktionalität, Lastverhalten, Skalierbarkeit, Gebrauchstauglichkeit)</li> </ul>
Ausarbeitung	<ul style="list-style-type: none"> <li>• Unterstützung der Analyse und des Designs durch Prototyping (frühzeitige Rückmeldung vom Kunden)</li> <li>• Architekturprototypen zur Architekturevaluierung</li> </ul>
Umsetzung	<ul style="list-style-type: none"> <li>• System/Integrationstests (z.B. Funktion, Last, Verfügbarkeit, frühzeitige Rückmeldung vom Kunden)</li> </ul>
Integration / Anlauf	<ul style="list-style-type: none"> <li>• Schulung von Anwendern und Administratoren</li> </ul>
Betrieb / Wartung	<ul style="list-style-type: none"> <li>• Absicherung kritischer Engineering-Aufgaben (z.B. Softwareaktualisierungen)</li> </ul>

Abbildung 6.2: Nutzung von Emulation im MES-Engineering-Prozess

## 6.2 Vorgehensweise zur Erzeugung virtueller Fabriken

Dem Nutzen virtueller Fabriken im MES-Engineering-Prozess steht der für ihre Erzeugung notwendige Zusatzaufwand gegenüber. Dieser Aufwand dient nicht unmittelbar dem Ziel des MES-Engineering-Projekts und ist daher zu minimieren. Das im folgenden Teil beschriebene Vorgehen hat die effiziente und effektive Erzeugung virtueller Fabriken zum Ziel. Analog zu der im vorhergehenden Abschnitt beschriebenen Vorgehensweise beim MES-Engineering wird ein iteratives Vorgehen genutzt. Abbildung 6.3 zeigt die Aktivitäten (1) bis (6) einer Iteration  $I_i$  und die Ergebnisse, die in jeder Aktivität entstehen. Abhängig vom Prozesskontext, in dem  $I_i$  ausgeführt wird, sind die Aktivitäten unterschiedlich gewichtet (vgl. Kapitel 6.1). Grundsätzlich sollte jede Iteration eine lauffähige virtuelle Fabrik liefern, die den Anforderungen der Iteration genügt.

### 6.2.1 Zieldefinition

Die Aktivität der Formulierung von Zweck und Ziel des Emulationseinsatzes hat eine hohe Bedeutung, da sie eine wichtige Grundlage für die Abwägung zwischen dem zu investierenden Modellierungsaufwand einerseits und dem erwarteten Nutzen andererseits bildet. Vor dem Hintergrund der Ergebnisse der Analyse in Kapitel 3.3.2 bieten sich zur Ermittlung dieser Ziele insbesondere Werkzeuge des Risikomanagements, wie die Fehlermöglichkeits- und Einflussanalyse (engl. Failure Mode and Effects Analysis) (FMEA) und Ursache-Wirkungs-Diagramme (siehe Kapitel 3.3.1), an. Sie werden in diesem Zusammenhang zur Identifikation kritischer Erfolgsfaktoren genutzt. Risiken, deren Eintrittswahrscheinlichkeit durch die Emulationsunterstützung des Engineering-Prozesses reduziert werden sollen, sind als **Ziele für den Emulationseinsatz** zu formulieren. Die Ziele gehen im weiteren Verlauf des Verfahrens in die Auswahl geeigneter Einsatzmöglichkeiten virtueller Fabriken im MES-Engineering-Prozess und die Ermittlung des Detaillierungsgrads der Emulation ein.

### 6.2.2 Systemanalyse

Die Systemanalyse dient der Erhebung von Informationen über das reale Produktionssystem, die für die virtuelle Abbildung der realen Fabrik von Belang sind. Drei Aspekten fällt dabei eine besondere Bedeutung zu:

- **Das Betriebskonzept der Fabrik aus MES-Perspektive** beschreibt die Rolle des MES im betrachteten Produktionssystem. Von besonderem Interesse ist die Frage, ob und in welcher Form das MES steuernd in den Produktionsbetrieb eingreift, ob es also Material aktiv durch die Fabrik treibt, dabei die beanspruchten Ressourcen nach vorgegebenen Kriterien optimiert und Anlagen aktiv ansteuert, oder ob das MES passiv Daten aus der Produktion sammelt, auswertet und über externe Regelschleifen, wie durch manuellen Werkereingriff an einer Anlage, in das Produktionssystem zurückkoppelt. Damit hat das Betriebskonzept wichtige Auswirkungen auf den Funktionsumfang der Anlagenschnittstellen, der im konkreten

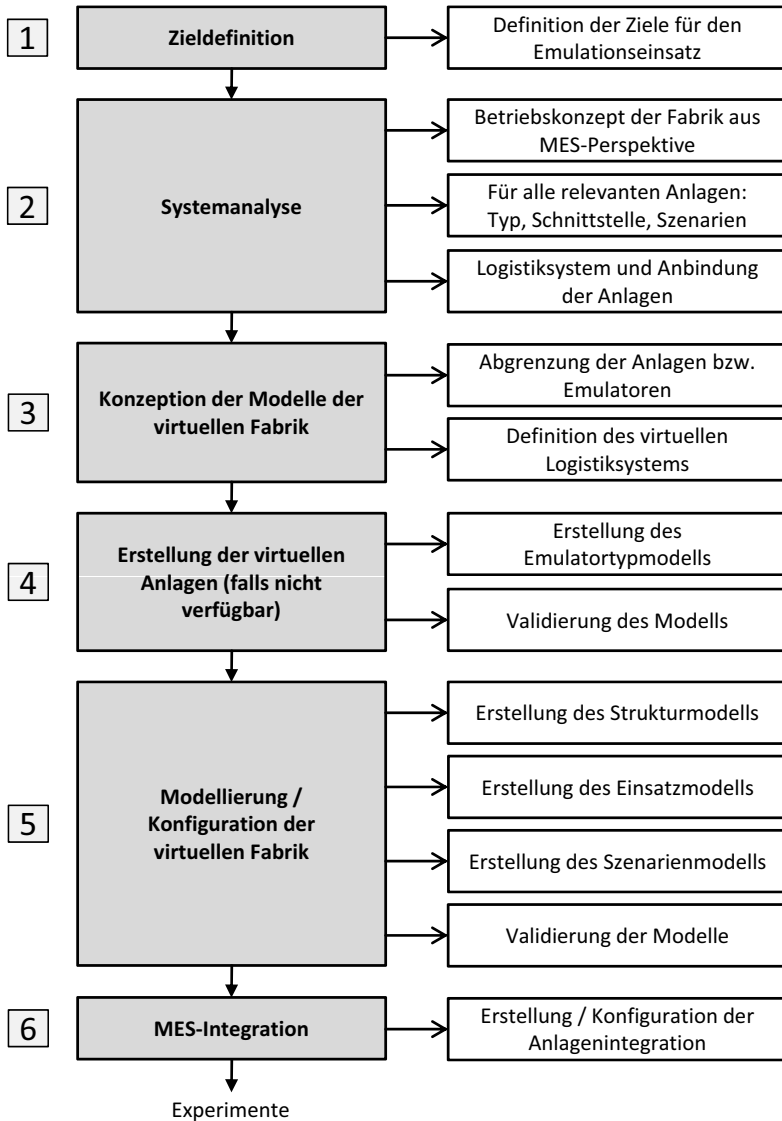


Abbildung 6.3: Aktivitäten zur Erzeugung virtueller Fabriken, die in jeder Iteration zu berücksichtigen sind.



Fall tatsächlich benötigt wird und potentiell durch Emulatoren zur Verfügung zu stellen ist.

- **Alle für das Betriebskonzept relevanten Anlagen** sind im Hinblick auf **ihren Typ, ihre Schnittstelle und die genutzten Betriebsszenarien** zu analysieren. Mit Hilfe der Klassifizierung der Anlagen durch Typen werden Anlagen, die ein gleichartiges Automatisierungskonzept verwenden, und Anlagen, die vergleichbare Prozessfähigkeiten anbieten, ermittelt. Im ersten genannten Fall wird schwerpunktmäßig das Logistik- und Steuerungskonzept der Anlagen betrachtet. Die Analyse der Schnittstelle umfasst die implementierten Protokollfamilien, den Grad der Übereinstimmung mit den die Schnittstelle spezifizierenden Standards bzw. Spezifikationen und etwaige Spezifika für jede betrachtete Anlage in der realen Fabrik. Die Betriebsszenarien beschreiben schließlich für jede Anlage ihre Interaktion mit der MES-Ebene, die für die Umsetzung des im ersten Schritt betrachteten Betriebskonzepts der Fabrik notwendig ist.
- Die Analyse des **Logistiksystems** und der **Integration der Anlagen in das Logistiksystem** ist die dritte wichtige Fragestellung der Systemanalyse. Der erste Teil der Analyse umfasst die Betrachtung der Aspekte des Logistiksystem der Fabrik, die aus Sicht des MES relevant sind. Dazu gehören die zum Materialtransport genutzten Basiskonzepte, deren konkrete Ausprägung und deren Steuerungskonzept. Der zweite Teil der Analyse stellt den Zusammenhang zwischen den Betriebsszenarien einzelner Anlagen und dem Fabriklogistiksystem über die Logistikschnittstellen der Anlagen her. Dabei ist von besonderem Interesse, welches Material einer Anlage in welcher Form zu- bzw. von der Anlage abgeführt wird, in wie weit dieses Material durch die Anlage identifizierbar ist und welche Regelschleifen zwischen Anlagen und dem Logistiksystem implementiert sind. Auch hier werden nur die Aspekte berücksichtigt, die für die Betriebsszenarien der Anlage aus Sicht des MES relevant sind.

### 6.2.3 Konzeption der Modelle der virtuellen Fabrik

Die entstehende virtuelle Fabrik soll die reale Fabrik auf einem Detaillierungsgrad abbilden, der die Ausführung der Fabrik- und Anlagen-Betriebsszenarien in der Interaktion mit dem MES erlaubt, die für die Erreichung der in der ersten Aktivität identifizierten Ziele relevant sind. Dieses Kriterium hat das zu entwickelnde Konzept für die virtuelle Fabrik zu erfüllen. Solange die Systemgrenzen einer Anlage im Sinne von Kapitel 2.1 in der PV-Industrie noch fallweise zu klären sind, hat innerhalb der Konzeption die **Abgrenzung von Anlagen** zu erfolgen. Nach ihrer Abgrenzung sind die identifizierten Anlagen auf Emulatortypen abzubilden. Um den Aufwand für die Modellierung zu minimieren, sollten Emulatortypen so generisch wie für die Unterstützung der formulierten Ziele möglich konzipiert werden. Die **Abbildung des Logistiksystems** muss sich ebenfalls an dem oben genannten Kriterium messen lassen. Für viele Zielstellungen reicht die Abbildung des Logistiksystems in stark abstrahierter Form aus. Die Konzeption des Einsatzmodells wird insbesondere durch den erwarteten Ressourcenbedarf der im

Strukturmodell definierten Emulatoren getrieben, wie deren Bedarf an spezifischen Bibliotheken, die nur auf ausgewählten Knoten zur Verfügung stehen, ihren Speicherbedarf, die zu erwartende CPU-Last und das zu erwartende Kommunikationsaufkommen.

## 6.2.4 Erstellung der virtuellen Anlagen

Im Sinne eines wirtschaftlichen Vorgehens ist, soweit als möglich, die Wiederverwendung verfügbarer virtueller Anlagen anzustreben. Falls für die im Rahmen der Konzeption abgegrenzten Anlagen noch keine adäquaten **Emulatortypmodelle** zur Verfügung stehen, sind diese in Aktivität (4) zu erstellen. Neuimplementierungen von Emulatortypmodellen sind vor ihrer Nutzung zu **validieren**. Dabei wird überprüft, ob ein Emulator an seiner Schnittstelle die im Rahmen der Konzeption spezifizierten Fähigkeiten zur Verfügung stellt und das Verhalten der realen Anlage an ihrer Schnittstelle mit der notwendigen Präzision abbildet. Sind Aufzeichnungen des Nachrichtenverkehrs (Logdaten) der realen Anlage über ihre IT-Schnittstelle verfügbar, bietet sich die Validierung des entsprechenden Emulators über den Abgleich der Aufzeichnungen von Emulator und realer Anlage an. Beide Datensätze werden für ausgewählte Automatisierungsszenarien verglichen, die jeweils zu charakteristischen Ereignisketten in den Aufzeichnungen führen. Die konkrete Ereigniskette hängt, neben dem Automatisierungsszenario, vom gewählten Detaillierungsgrad der Abbildung der realen Anlage ab. Der Abgleich kann manuell oder automatisiert über Verfahren zur Analyse von Ereignisketten durchgeführt werden (siehe Kapitel 4.6.1). Falls kein Zugriff auf die reale Anlage beziehungsweise deren Kommunikationslogs möglich ist, erfolgt die Validierung in Form von Tests auf Basis der Spezifikation aus der Konzeptionsphase.

## 6.2.5 Modellierung/Konfiguration der virtuellen Fabrik

Die Modellierung und Konfiguration einer virtuellen Fabrik greift auf die in Aktivität (4) entstandenen Emulator Typen zurück und ordnet sie unter Berücksichtigung der Ergebnisse der Konzeptionsphase in den Kontext des virtuellen Produktionssystems ein. Mit Hilfe des Strukturmodelleditors (siehe Kapitel 5.4) wird das **Strukturmodell** der virtuellen Fabrik erstellt, das die statischen Zusammenhänge der realen Fabrik abbildet und die Schnittstellen der Emulatoren verdrahtet. Für die Nutzung der virtuellen Fabrik in Experimenten sind darüber hinaus Szenarien im **Szenariomodell** zu definieren. Soweit vorhanden wird für ihre Modellierung auf Zustandsdaten zurückgegriffen, die bereits im Repository verfügbar sind. Anderenfalls sind die Zustandsdaten zunächst mit Hilfe der Emulatoren zu generieren. Parallel zum Szenariomodell wird das **Einsatzmodell** erstellt, das die Verteilung der Emulatoren unter Berücksichtigung der Ressourcen der Emulationsumgebung, in der die virtuelle Fabrik zur Ausführung gebracht werden soll, definiert.

Ähnlich wie einzelne Emulator Typen ist das **Gesamtmodell** der virtuellen Fabrik zu **validieren**, um eine der Zieldefinition angemessene Übereinstimmung des Verhaltens der virtuellen Fabrik mit dem der realen Fabrik sicherzustellen. Die Validierung kann, soweit verfügbar, durch einen Abgleich von Anlagen-Logdaten aus der realen Fabrik

mit Emulator-Logdaten aus der virtuellen Fabrik erfolgen (Vergleich ausgeführter Automatisierungsszenarien). Sind die Emulatortypen, die in den betrachteten Szenarien genutzt werden, validiert, wird von einem definitionsgemäßen Verhalten der Emulatoren an ihren Logistik- und IT-Schnittstellen ausgegangen. Damit sind zur Validierung der virtuellen Fabrik noch die Materialbewegungen im Gesamtsystem zu betrachten und mit der realen Fabrik zu vergleichen. Da Materialbewegungen im Zusammenhang mit der Ausführung logistikrelevanter Automatisierungsszenarien auftreten, lässt ein vergleichbares Profil von Materialbewegungen für die logistikrelevanten Automatisierungsszenarien auf ein vergleichbares Verhalten von virtueller und realer Fabrik schließen. Für diesen Abgleich können Historiendaten aus einem MES, die detaillierte Rückschlüsse auf die Materialbewegungen zwischen den realen Anlagen zulassen, zur Validierung herangezogen werden. Stehen keine Daten aus der realen Fabrik zur Verfügung, ist zumindest eine Plausibilitätsprüfung auf Basis der Festlegungen aus der Konzeptionsphase vorzunehmen.

### 6.2.6 MES Integration

Die Integration virtueller Anlagen mit dem MES erfolgt, wann immer möglich, über dieselben Schnittstellen und Integrationsmechanismen, über die reale Anlagen am MES angebunden werden. Die dabei nutzbaren Mechanismen und Werkzeuge werden vom anbindenden MES bzw. der verwendeten Integrationsumgebung (siehe Kapitel 3.3.2) zur Verfügung gestellt. Bei der Anbindung virtueller Anlagen an das MES sind allenfalls Beschränkungen des Funktionsumfangs des Emulators zu berücksichtigen, die sich aus den in Aktivität (1) definierten Zielen ergeben. Als Ergebnis steht das MES für Experimente im Zusammenspiel mit der virtuellen Umgebung zur Verfügung.

## 6.3 Fazit: Anforderungen an das Verfahren

Soweit zur Umsetzung des emulationsgestützten MES-Engineering-Prozesses die benötigten virtuellen Fabriken nicht fertig konfiguriert zur Verfügung stehen, was den Regelfall darstellen dürfte, ist das in Abschnitt 6.2 beschriebene Verfahren zur Erzeugung virtueller Fabriken mit dem in Abschnitt 6.1 beschriebenen MES-Engineering-Prozess zusammenzuführen. In diesem Fall sind die in Abbildung 6.3 benannten Aktivitäten (1) bis (6) in den Iterationen des MES-Engineering zu berücksichtigen. Viele der in der Abbildung dargestellten Ergebnisse der Aktivitäten können direkt oder durch Transformationen aus den Ergebnissen des MES-Engineering-Prozesses gewonnen werden, insbesondere:

- Systemanalyse. Alle dargestellten Ergebnisse sind zentrale Ergebnisse des MES-Engineering und können direkt für die Entwicklung der virtuellen Fabrik übernommen werden.
- Konzeption der Modelle. Die Abgrenzung von Anlagen als ein Ergebnis der Aktivität (4) wird im Zuge des MES-Engineering bei Analyse/Design Aktivitäten erfolgen und kann für die Entwicklung der virtuellen Fabrik übernommen werden.

Anforderungen an das Verfahren
<b>1. Zielstellung des Verfahrens</b> <ul style="list-style-type: none"><li>a) Kompensation der späten Verfügbarkeit der realen Fabrik <b>[6.1.2]</b></li><li>b) Erweiterung der begrenzten Validierungsmöglichkeiten in der realen Fabrik <b>[6.1.2]</b></li><li>c) Absicherung kritischer Entscheidungen in einer möglichst frühen Phase des Fabriklebenszyklus <b>[6.1.1]</b></li><li>d) Reduktion der Risiken bei Inbetriebnahme und Anlauf <b>[6.1.1]</b></li></ul>
<b>2. Eigenschaften des Verfahrens</b> <ul style="list-style-type: none"><li>a) Der für die Modellbildung zu investierende Aufwand muss in einem wirtschaftlich sinnvollen Verhältnis zum erzielbaren Nutzen stehen <b>[6.1.2, 6.2.1, 6.2.6, 6.3]</b></li><li>b) Das Verfahren muss kompatibel zu den in der Industrie eingesetzten Vorgehensmodellen sein <b>[6.1.1]</b></li><li>c) Verfahren muss in der Projektorganisation implementierbar sein <b>[6.1.1, 6.2, 6.3]</b></li></ul>

Abbildung 6.4: Abgleich des Verfahrens des emulationsgestützten MES-Engineering mit den Anforderungen aus Abbildung 3.8

- Erstellung der virtuellen Anlagen. Werden Automatisierungsszenarien im MES-Engineering formal spezifiziert, können zumindest Teile des Emulatortypmodells daraus abgeleitet werden.
- MES-Integration. Es wird die identische Anlagenanbindung für beide Varianten verwendet. Abhängig vom gewählten Detaillierungsgrad der Emulation ist allenfalls sicherzustellen, dass nur durch die Emulation unterstützte Funktionalitäten genutzt werden.

Indem das Verfahren des emulationsgestützten MES-Engineering diese beiden Prozesse eng verzahnt und auf eine Werkzeugumgebung aufbaut, die den beteiligten Analysten, Lösungsarchitekten, Entwicklern und Testern konzeptionell und technologisch entgegenkommt, können maximale Synergien zwischen beiden Prozessen erzielt werden. Damit wird ein effektiver und effizienter Einsatz von virtuellen Fabriken im MES-Engineering möglich.

Wie im Kapitel 5 zeigt 6.4 den Zusammenhang zwischen den einzelnen Anforderungen aus Kapitel 3 und den Abschnitten dieses Kapitels, die diese Anforderungen jeweils in der Umsetzung aufgreifen.

# 7 Erprobung und Bewertung

Das Verfahren zum emulationsgestützten MES-Engineering für die Photovoltaikindustrie wurde in mehreren Iterationen entwickelt und verfeinert und in der industriellen Anwendung in Zusammenarbeit mit einem MES-Anbieter erprobt. Die Anwendung wird in den folgenden Abschnitten anhand zweier realer Einsatzbeispiele aus der industriellen Praxis beschrieben und verdeutlicht (Fallbeispiel I und II). Die Fallbeispiele werden in der Folge zur Bewertung des Verfahrens anhand des Anforderungsprofils aus Kapitel 3.6 herangezogen.

In Fallbeispiel I kommt das Verfahren beim Aufbau einer Fertigung für Konzentratormodule<sup>1</sup> zur Anwendung. Fallbeispiel II gibt die Erfahrungen der Erprobung beim Aufbau einer Dünnschicht-Linie wieder. Beiden Beispielen liegen sehr unterschiedliche Fertigungsorganisationen zugrunde. Während die Fertigungsorganisation im Fallbeispiel I am ehesten der einer Reihenfertigung entspricht, in der die Herstellung des Endprodukts über mehrere Einzelanlagen und einige Transferstraßen verteilt ist, ist die Fertigung in Fallbeispiel II als getaktete Fließfertigung in Transferstraßen (in der PV-Industrie wird häufig der Begriff der Inline-Fertigung verwendet) organisiert. Beide Fabriken waren jeweils mit einer MES-Lösung auszurüsten, die in Engineering-Projekten an die spezifischen Anforderungen des jeweiligen Produktionssystems anzupassen waren. Die Beschreibung der Fallbeispiele orientiert sich an der Struktur des in Kapitel 6.2 entwickelten Vorgehens.

## 7.1 Fallbeispiel I (Back-end Konzentratormodule)

Das Konzentratorprinzip erfordert im Vergleich zu den in Kapitel 2.2 beschriebenen Technologien einen anderen Aufbau der PV-Module. Für den im Beispiel betrachteten Fall werden mehrere flächenmäßig kleine, hocheffiziente Solarzellen und ein Linsensystem zu einem Modul verbaut. Die betrachtete Back-end Fertigung umfasst in der Produktionskette die Prozessschritte von der Vereinzelung der Solarzellen bis hin zum Test der fertigen Module. Kontext des Fallbeispiels I im Fabriklebenszyklus ist die Phase 'Aufbau und Produktionsanlauf'. Gleichzeitig mit dem Aufbau der realen Fabrik erfolgte in einem MES-Engineering-Prozess die Entwicklung und Konfiguration der MES-Lösung, die die spezifischen Geschäftsprozesse und das Produktionssystem unterstützt. Diese MES-Lösung wurde in der realen Produktionsumgebung installiert und in Betrieb genommen.

---

<sup>1</sup>Einen Einblick in eine solche Fertigung und ein entsprechendes Produkt gibt z.B. [Rutschmann, 2009]

### 7.1.1 Erzeugung der virtuellen Fabrik

**Zieldefinition.** Zielstellung für den Einsatz des Verfahrens zum emulationsgestützten MES-Engineering im Fallbeispiel I ist, frühzeitig im Projekt die Korrektheit und die Nutzbarkeit betriebskritischer Funktionalität bzw. die Einhaltung betriebskritischer nicht-funktionaler Anforderungen nachzuweisen. Besondere Aufmerksamkeit erhält in diesem Zusammenhang das *Traceability*-Konzept, das für die hergestellten Endprodukte die Rückverfolgbarkeit ihrer Fertigungshistorie (Traceability-Kette) mit der vom Hersteller geforderten Präzision sicherstellen muss. Die erste virtuelle Inbetriebnahme erfolgt zu einem Zeitpunkt, zu dem noch nicht alle Anlagenschnittstellen im Detail spezifiziert sind.

**Systemanalyse.** Im Sinne des integrierten Vorgehens erfolgt die Analyse vollständig im Kontext des MES-Projekts. Die zur Entwicklung der virtuellen Fabrik benötigten Informationen werden gleichermaßen für das MES-Engineering benötigt. Abbildung 7.1 zeigt eine konzeptionelle Darstellung des Segments A der Produktion, das einen entscheidenden Einfluss auf das geplante Traceability-Konzept hat. Wichtig für die Traceability-Kette sind in diesem Abschnitt mehrere Wechsel der zu verfolgenden Einheiten: Die im Segment A verfolgten Logistikobjekte werden in Teilobjekte zerlegt, für die Weiterverarbeitung gruppiert, zu Baugruppen montiert und in diesen Gruppen bearbeitet. In weiteren Produktionsschritten werden die gruppierten Baugruppen wieder vereinzelt und in nachfolgenden Liniensegmenten unter Berücksichtigung von Daten, die im Segment A erhoben wurden, verbaut. Zusätzlich erfordern Limitationen einiger Anlagen hinsichtlich der an den Host-Schnittstellen angebotenen Funktionalität die Definition von Automatisierungsszenarien, die die Kompensation dieser Begrenzungen durch das MES erlauben. In diesen Fällen wird die interne Logistik der Anlagen im Detail analysiert. Die Robustheit der Automatisierungsszenarien dieser Anlagen soll im Rahmen der virtuellen Inbetriebnahme im Gesamtkontext mit untersucht werden. Hinsichtlich des Logistiksystems der realen Fabrik sind zwei Fälle zu unterscheiden: Einige Anlagen im Segment A sind über verschiedene Automatisierungslösungen fest verkettet. Die Verkettung ist in Abbildung 7.1 durch einen dickeren Pfeil gekennzeichnet. Im Fehlerfall führt ein Staukonzept zur Abschaltung von vorgelagerten Anlagen. Bei den übrigen Anlagen erfolgt die Be- und Entladung durch menschliche Bediener. Das Material wird, abhängig vom aktuellen Bearbeitungszustand, in unterschiedlichen Materialträgern transportiert, die ebenso im Traceability-Konzept zu berücksichtigen sind. Allen Materialträgern sind eindeutige Bezeichner zugewiesen. Materialpositionen in den Materialträgern sind durch Positionsvektoren eindeutig darstellbar. Zur IT-Integration der Anlagen kommen verschiedene proprietäre und standardisierte Protokollstapel zum Einsatz.

**Konzeption der Modelle.** Zunächst werden die zu berücksichtigenden Anlagen für die Abbildung auf Emulatoren in Rücksprache mit dem MES-Team abgegrenzt. Um den für die Abbildung der internen Anlagenlogistik im Emulatorotyp notwendigen Aufwand für die Anlage A5 zu minimieren, werden ihre Teilsysteme auf einen Emulatorotyp abgebildet, der mehrere Host-Interfaces zur Verfügung stellt. Die übrigen Anlagen in Segment A erfordern die Definition von vier weiteren Emulatorotypen mit jeweils einer Host-Schnittstelle und werden in der virtuellen Fabrik eins zu eins durch Emulatoren repräsentiert. Aufgrund der Vielzahl von IT-Integrationsschnittstellen, von denen einige zum Zeitpunkt der ersten virtuellen Inbetriebnahme noch nicht vollständig spezifiziert

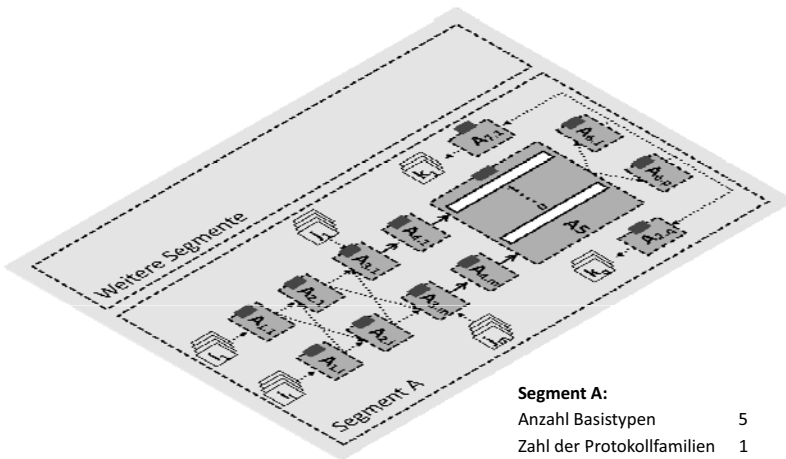


Abbildung 7.1: Fallbeispiel Back-end Konzentratormodule. Konzeptionelle Darstellung des diskutierten Fabrikabschnitts.

sind, werden Emulatoren und MES über eine abstrakte RMI-basierte Schnittstelle integriert, die MES-seitig in einer späteren Phase problemlos auf die tatsächlich verwendeten Schnittstellen umgestellt werden kann. Während der Fall der Verketzung von Anlagen durch Automatisierungssysteme über spezialisierte Ports und Verbindungen abgebildet wird, wird für die Abbildung der manuellen Transporte ein zusätzlicher Emulatortyp eingeführt, der einen unbeschränkten Puffer mit wahlfreiem Zugriff für alle eingesetzten Materialträgertypen bereitstellt und mit allen manuell zu versorgenden Ports verbunden ist. Das statische Modell der Logistikobjekte wird so konzipiert, dass Logistikobjekte, die später in das Endprodukt eingehen, alle Historiendaten kennen, die für die Validierung der Traceability-Kette relevant sind. Darüber hinaus kennen Logistikobjekte objektspezifische Daten, die für die Steuerung der Automatisierungsszenarien nachfolgender Prozessschritte relevant sind. Die Aktualisierung der Historie und der Qualitätsdaten erfolgt durch jeden Emulator entsprechend des repräsentierten Prozessschritts und steht zum Abgleich gegen die MES-Daten zur Verfügung.

**Erstellung der virtuellen Anlagen.** Für die Umsetzung des Modellkonzepts sind sechs Emulatortypmodelle zu erstellen. Der Hauptaufwand liegt in der Modellierung des Logistikkonzepts der Anlage A5, das sehr detailliert abzubilden ist, um die in der Systemanalyse definierten Automatisierungsszenarien mit der virtuellen Fabrik betreiben zu können: Der Materialfluss aus den parallelen Strängen  $A_{4,1}$  bis  $A_{4,m}$  wird synchronisiert und auf die Spuren  $Spur_1$  bis  $Spur_o$  (mit  $o \neq m$ ) aufgeteilt, dort parallel bearbeitet und schließlich in Materialträger überführt. Dieser Ablauf ist in einer Reihe von Interaktionen mit dem MES zu synchronisieren. Auch bei den übrigen Emulatortypen sind vorrangig die Logistikabläufe zu modellieren, die um die Generierung exemplarischer Prozessdaten und die Generierung von Prozessergebnissen ergänzt werden. Letztgenannter Aspekt spielt bei der Generierung von Material-Güteklassen in Messsystemen eine wichtige Rolle,

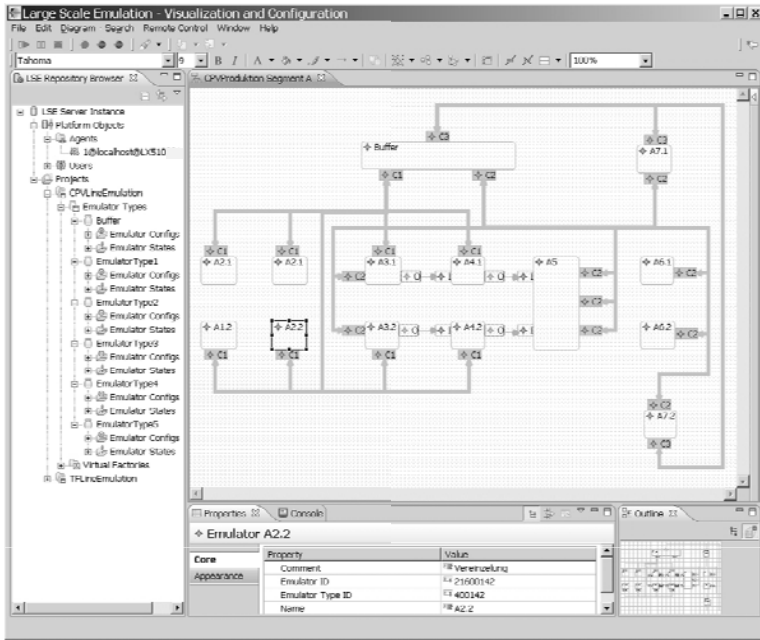


Abbildung 7.2: Beispiel: Überführung der schematischen Darstellung in ein Strukturmodell.

die wiederum die Steuerung anderer Anlagen beeinflussen. Die Generierung dieser Daten erfolgt auf Basis geeigneter statistischer Verteilungen. Alle Emulatoren stellen über ihre Management-Schnittstelle im interaktiven Zugriff Informationen über die aktuell bearbeiteten Logistikobjekte zur Verfügung. Die Implementierung der Host-Schnittstellen greift auf verfügbare API-Pakete des verwendeten MES zurück.

**Modellierung/Konfiguration der virtuellen Fabrik.** Unter Verwendung der sechs Emulatortypmodelle werden im Strukturmodell die benötigten Emulatoren definiert und über das virtuelle Logistiksystem konnektiert. Zu jedem Emulator wird eine Basiskonfiguration erstellt, die alle für die Initialisierung der Host-Schnittstelle relevanten Parameter enthält, und ein Initial-Zustandssatz aufgezeichnet, der eine leere, betriebsbereite Anlage beschreibt. Für Experimente mit der virtuellen Fabrik wird ein Szenario definiert, das die Konfigurationen und Zustandssätze eines leeren, vollständig betriebsbereiten Segments umfasst. Über das in Kapitel 5.3 beschriebene Lebenszyklusmanagement hinaus übernimmt die Szenario-Koordination in diesem Szenario keine weiteren steuernden Funktionen. Stattdessen erfolgt die Steuerung des Szenarios im manuellen Betrieb über den Puffer-Emulator. Die Management-Schnittstelle des Puffer-Emulators gibt dem Benutzer Zugriff auf folgende Funktionen:



- Generieren neuer Rohmaterialien, inklusiv Transfer in Materialträger.
- Auslösen von Transaktionen zum Materialtransfer über das virtuelle Logistiksystem: Vom Puffer zum gewünschten Emulator und umgekehrt.
- Durchsuchen der im Puffer verfügbaren Materialien.

Entsprechend des Ausführungsmodells der Emulationsplattform wird die Verarbeitung verfügbarer Logistikobjekte durch einen Emulator gestartet, sofern alle weiteren aktiven Vorbedingungen erfüllt sind. Ein weiteres Szenario startet ebenfalls von einer leeren, betriebsbereiten Linie, triggert am Puffer aber kontinuierlich Materialtransfers, um den Dauerbetrieb der Linie nachzubilden. Die virtuelle Fabrik wird auf einem Agenten installiert und ausgeführt, da der benötigte Ressourcenbedarf von einem physikalischen Knoten zur Verfügung gestellt werden kann. Als physikalischer Knoten kommt eine zwei-Kern Maschine (Intel® Core™ Duo T2600, 2.16 GHz) zum Einsatz.

**MES-Integration.** Die virtuelle Fabrik wird unter Verwendung der oben beschriebenen Mechanismen an die reale MES-Umgebung gekoppelt, die auf Test-Hardware betrieben wird. Im Rahmen der virtuellen Inbetriebnahme wird in der MES-Umgebung jeweils die Version des MES-Modells ausgeführt, die dem Ergebnis der Iteration entspricht, in der die virtuelle Inbetriebnahme durchgeführt wird.

### 7.1.2 Einsatz der virtuellen Fabrik

Entsprechend der eingangs formulierten Zielstellung für die virtuelle Inbetriebnahme im vorgestellten MES-Engineering-Projekt werden zwei Meilensteine definiert, deren Erreichung durch virtuelle Inbetriebnahmen nachgewiesen wird. Die erste virtuelle Inbetriebnahme umfasst das im Vorfeld diskutierte Segment A; für den zweiten Meilenstein werden weitere Segmente berücksichtigt, die über den Puffer an das hier beschriebene Teilmodell angebunden werden. Die Validierung der virtuellen Fabrik erfolgt über ausgewählte Testfälle, die aus der Spezifikation erzeugt und, soweit bereits verfügbar, mit Informationen aus dem ACM-Prozess abgeglichen werden. Darüber hinaus wird die virtuelle Fabrik im MES-Engineering-Prozess bereits in der Iteration im Vorfeld der Meilensteine für eine Reihe von Integrationstests genutzt. Für diesen Fall spielen die Möglichkeiten zum interaktiven Betrieb eine wichtige Rolle: Ausgewählte Emulatoren werden angehalten, um ihren aktuellen Zustand zu analysieren und mit der Zustandsrepräsentation des MES zu vergleichen. Speziell konfigurierte Materialbestände werden über den Puffer eingeschleust, um die Implementierung von Ausnahmepfaden in den Automatisierungsszenarien zu überprüfen. Der virtuelle Probetrieb erfolgt mit denselben Modellen, allerdings unter Verwendung des kontinuierlichen Materialtransfer-Modus. Für Produkt-Stichproben wird die Traceability-Kette von Produkten mit den im MES aufgenommenen Daten abgeglichen. Der Abgleich erfolgt in diesem Fall manuell zwischen den Historien-Datensätzen von Logistikobjekten der virtuellen Fabrik und Produkt-Trace-Reports des MES. Gleichzeitig werden die durch die virtuelle Inbetriebnahme generierten Testdaten zur Diskussion wichtiger Funktionalitäten am laufenden System mit dem Kunden herangezogen. Dabei wird verifiziert, ob die Modellierung auf Basis der Systemanalyse tatsächlich die Anforderungen der Geschäftsprozesse des Kunden widerspiegelt.

### 7.1.3 Fazit

In Fallbeispiel I wird die betrachtete MES-Lösung zu den beiden definierten Meilensteinen erfolgreich virtuell in Betrieb genommen. Die in der Zielstellung beschriebenen Anforderungen können mit Hilfe der Emulationsplattform erfüllt werden. Allerdings fällt der Modellierungsaufwand höher aus als planmäßig vorgesehen. Die Analyse der Aufwände ergibt, dass der Mehraufwand bei der Modellierung der virtuellen Logistiksysteme entstanden ist. In der im Projekt verwendeten Version der Emulationsplattform ist das virtuelle Logistiksystem der Fabrik, ebenso wie das interne Logistiksystem einzelner Emulatoren, noch programmatisch auf Basis eines objektorientierten Rahmens zu erstellen, das lediglich sehr feingranulare Modellelemente zur Verfügung stellt. Bei steigender Komplexität der erstellten Modelle steigt mit dem Modellierungsaufwand gleichzeitig die Wahrscheinlichkeit für Modellierungsfehler an, die sich störend auf den Prozess der Inbetriebnahme auswirken. Aufgrund dieser Erfahrung wird mit der Folgeversion der Emulationsplattform, die in Fallbeispiel II zum Einsatz kommt, der in Kapitel 5 diskutierte modellgetriebene Ansatz zum Aufbau des Strukturmodells verfolgt. Zur Optimierung der Modellbildung für die Logistik innerhalb des Emulatortyps wird der Übergang von der objektorientierten Modellierung hin zu einer komponentenbasierten Modellierung vorgesehen.

## 7.2 Fallbeispiel II (Dünnschicht Front-end / Back-end)

Im Fallbeispiel II war eine Dünnschicht-Fertigung (eine Beschreibung der Aufgabenbereiche von Front-end und Back-end findet sich in Kapitel 2.2) mit einer MES-Lösung auszurüsten. Kontext dieses Beispiels war die Phase des Aufbaus im Fabriklebenszyklus. Das MES-Projekt war dabei Teil der umfangreichen Projektorganisation zur Errichtung einer neuen Fabrik, mit Abhängigkeiten von und zu anderen Gewerken.

### 7.2.1 Erzeugung der virtuellen Fabrik

**Zieldefinition.** Die Zielstellung für die Nutzung des Verfahrens zum emulationsgestützten MES-Engineering ist die Inbetriebnahme des MES bereits in der frühen Phase des Aufbaus, in der weder die Fabrik noch die Mehrheit der Anlagen fertiggestellt sind und damit unter keinen Umständen für Tests des MES zur Verfügung stehen. Ziele der virtuellen Inbetriebnahme im konkreten Fall sind:

- Die Eintrittswahrscheinlichkeit für Risiken, die aus dem MES-Teilprojekt heraus zu Verzögerungen des Gesamtprojekts führen könnten, soll minimiert werden. Insbesondere sollen Unzulänglichkeiten im angepassten MES-Modell, die Kernfunktionalitäten für Anlauf und Betrieb betreffen, durch die Einführung des Meilensteins “virtuelle Inbetriebnahme” in der Aufbau-Phase vermieden werden, indem diese Funktionalitäten mit Hilfe der virtuellen Fabrik validiert werden.
- Das dynamische Verhalten im Betrieb der statisch ausgelegten Hardware-Infrastruktur, einschließlich des geplanten Fabriknetzes und der Server-Konfiguration

der MES-Ebene, soll überprüft werden. Dabei werden Indikatoren wie die verfügbare Netzwerkbandbreite, die Auslastung von Central Processing Units (CPUs), Transaktionszeiten und die Belegung von Warteschlangen beobachtet.

- Wichtige nicht-funktionale Anforderungen an die angepasste MES-Lösung, wie deren Stabilität im Dauerbetrieb bzw. unter Voll- oder Überlastbedingungen und Mechanismen, die eine ausreichende Verfügbarkeit sicherstellen, sollen im Rahmen der virtuellen Inbetriebnahme unter Bedingungen nachgewiesen werden, die denen der zukünftigen realen Fabrik möglichst ähnlich sind.

**Systemanalyse.** Entsprechend des integrierten Vorgehens zum emulationsgestützten MES-Engineering erfolgt die Systemanalyse vollständig im Kontext des MES-Projekts. Die benötigten Informationen zum Betriebskonzept werden dem MES-Projekt aus der Fabrikplanung zugeführt. Vorgaben zum Betrieb und der Anbindung der Anlagen werden in einem vereinfachten ACM-Prozess, gemeinsam mit der Fabrikplanung, spezifiziert und ermittelt. Abbildung 7.3 zeigt schematisch das Konzept der Inline-Fertigung, in der auf Glassubstraten Dünnschicht-Zellen gefertigt werden, die schließlich zu PV-Modulen aufgebaut werden. Die Fertigung ist in mehreren Segmenten organisiert, die jeweils außen-verkettete Anlagen umfassen. Die Segmente sind ebenfalls über Außenvernetzungen miteinander verbunden und teilweise über Puffersysteme entkoppelt. Durch die verketteten Anlagen werden einzelne Substrate geschleust, die durch eindeutige Kennungen identifizierbar sind. Materialträger spielen für die Betrachtung keine Rolle. 187 Anlagen bzw. Anlagenmodule werden identifiziert, die mit dem MES zu integrieren und damit von der virtuellen Fabrik abzubilden und zu verwalten sind. Zur Integration der Anlagen werden zwei Protokollfamilien genutzt, die jeweils einen Protokollstapel, wie den in Kapitel 2.4.2 vorgestellten PVECI, beschreiben. Aus der Menge der zu integrierenden Anlagen und Module lassen sich vier Basistypen identifizieren, die gleichartige generische Fähigkeiten haben und daher mit gleichartigen Automatisierungsszenarien betrieben werden: Prozessanlagen, Messsysteme, Transportsysteme und Puffer. In den Automatisierungsszenarien für die virtuelle Inbetriebnahme ist der Normalbetrieb der Anlagen zu berücksichtigen, der die Bearbeitung, Prüfung und den Transport von Substraten, inklusiv der Erfassung exemplarischer Prozessdaten, umfasst. Das Routing der Substrate wird auf der Prozessautomatisierungsebene gesteuert.

**Konzeption der Modelle.** In der Konzeptionsphase wird festgelegt, jede aufgrund des Steuerungskonzepts der realen Fabrik einzeln zu integrierende Anlage auf jeweils einen Emulator abzubilden. Aufgrund der formulierten Zielstellung kann, im Gegensatz zu Fallbeispiel I, auf eine detaillierte Abbildung des internen Logistiksystems von Anlagen verzichtet werden. Dafür ist die Anbindung der einzelnen Emulatoren jeweils über die Protokollfamilie zu realisieren, die auch in der realen Fabrik genutzt wird. Als Quelle bzw. Senke für Logistikobjekte werden zwei zusätzliche Emulatoren vorgesehen. Für die Kopplung der Emulatoren über das virtuelle Logistiksystem der Emulationsplattform wird der Standardfluss der Substrate berücksichtigt, der über zwei generische Port-Typen und einen Verbindungstyp modelliert wird.

**Erstellung der virtuellen Anlagen.** Abgeleitet aus den Automatisierungsszenarien der Basistypen wird ein generischer Emulatortyp implementiert, dessen Anlagenmodell konfigurationsgesteuert an das Verhalten aller Basistypen angepasst werden kann. Zur

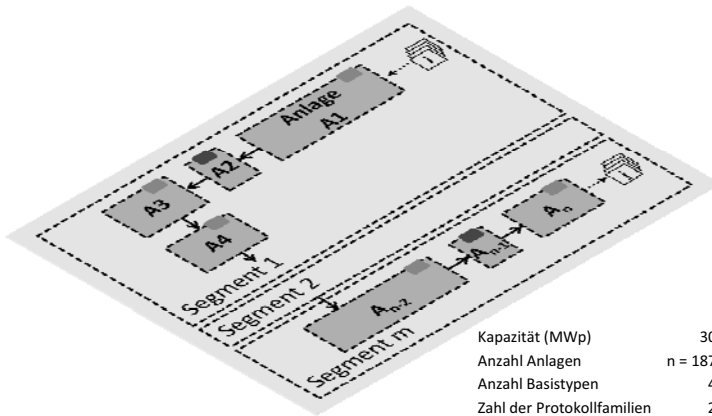


Abbildung 7.3: Fallbeispiel Dünnschicht Front-end / Back-end. Konzeptionelle Darstellung der Fabrik

Beschreibung des Verhaltens der Basistypen werden endliche Zustandsautomaten genutzt, die die Verwendung von Beschleunigungsfaktoren unterstützen. Die exemplarischen Prozessdaten werden über konfigurierbare statistische Generatoren erzeugt. Die spezifische Konfiguration jeder Anlage, zu der die eindeutige Anlagenkennung, IP-Adressen und Ports, protokollfamilien-spezifische Einstellungen und das Anlagenmodell gehören, lässt sich über 'EmulatorKonfigurationen' vornehmen. Für die Emulatoren  $A_1$  bis  $A_{187}$ , die reale Anlagen abbilden, werden Instanzen von 'EmulatorKonfigurationen' durch eine Transformation aus den Ergebnissen des ACM-Prozesses generiert und im Repository verwaltet. Die Implementierung der Host-Schnittstellen basiert auf Referenz-Bibliotheken der beiden Protokollfamilien, die in der realen Produktion MES-seitig und teilweise anlagenseitig zum Einsatz kommen.

**Modellierung/Konfiguration der virtuellen Fabrik.** Auf Basis des generischen Emulatortyps werden im Strukturmodell 189 Emulatoren definiert und über das virtuelle Logistiksystem verkettet. Zu jedem Emulator wird ein 'KonfigurationsSatz' erzeugt, der die Anlage im leeren, betriebsbereiten Zustand beschreibt und auf die zugehörige 'EmulatorKonfiguration' verweist. Das Szenario 'LineStart' dient als Ausgangspunkt für Experimente mit der virtuellen Fabrik und basiert auf diesen Konfigurationssätzen. Es unterstützt sowohl den interaktiven Betrieb, in dem der Benutzer Substrate manuell über die Managementschnittstelle an der Quelle erzeugt, als auch den Dauerbetrieb, in dem die Quelle zeitgesteuert Substrate in die virtuelle Fabrik einschleust. Für die Ausführung dieses Szenarios steht ein physikalischer Knoten (Xeon L5420, 4 Kerne, 2,5 GHz, 8 GByte Arbeitsspeicher, mit 1 GBit Ethernet Anbindungen, Windows 2003 Server) zur Verfügung. Aufgrund implementierungstechnischer Restriktionen werden die Emulatoren auf 6 Agenten verteilt, die jeweils 1 GByte Arbeitsspeicher verwalten, ansonsten aber identische Ressourcen (z.B. die Basisbibliotheken der Protokollfamilien) anbieten. Die Ressourcenbelegung, insbesondere die CPU-Last, der verfügbare freie Ar-

beitsspeicher und die verfügbare Netzwerkbandbreite werden zur Laufzeit überwacht, um etwaige Engpässe und damit einhergehende, mögliche Abweichungen vom Zeitverhalten realer Anlagen zu erkennen.

**MES-Integration.** Die Emulatoren der virtuellen Fabrik sind über GBit-Ethernet an das reale MES-Cluster angebunden, das mit dem realen MES-Modell in der zu evaluierenden Version unter Verwendung der zukünftigen Netzkonfiguration auf der Original-Hardware betrieben wird.

## 7.2.2 Einsatz der virtuellen Fabrik

Entsprechend der formulierten Ziele werden die in Zusammenarbeit mit der Gesamtprojektleitung identifizierten Kernfunktionalitäten des MES in mehreren Testläufen in der Interaktion mit dem skizzierten AnlaufszENARIO der virtuellen Fabrik evaluiert. Der Abgleich des erwarteten Verhaltens mit dem getesteten Verhalten erfolgt MES-seitig über die Werkzeuge, die auch im Betrieb der realen Fabrik eingesetzt werden, wie Visualisierungen des Gesamtzustands der Fabrik oder ausgesuchter Ressourcen, die Darstellung des Zustands ausgewählter Jobs und spezifische Reports. Anlagenseitig kommt der Auswertung der von jedem Emulator aufgezeichneten Kommunikationslogs eine große Bedeutung zu, die auch im realen Betrieb a posteriori zu Analyse Zwecken herangezogen werden und deshalb vom Kunden als Beleg der Vorgänge auf der virtuellen Anlage akzeptiert werden. Mit einer stichprobenartigen Analyse und Korrelation ausgewählter Logfiles zueinander und zu Daten, die das MES zur Verfügung stellt, wird sowohl die Korrektheit der MES-Lösung als auch die Vergleichbarkeit zwischen virtueller und realer Fabrik nachgewiesen. Die Überprüfung der Hardware- und Netzinfrastruktur und der Stabilität unter Vollast und Überlast erfolgt durch den mehrstündigen Dauerbetrieb des MES an der virtuellen Fabrik, wobei Überlastbedingungen über den Beschleunigungsfaktor erzeugt werden, der letztlich zu einem gesteigerten Durchsatz und zu entsprechend erhöhten Transaktionsraten führt. Die Funktionalität der Cluster-Konfiguration als einer der Mechanismen zur Sicherstellung der benötigten Verfügbarkeit wird durch die manuelle Unterbrechung der Netzanbindung einzelner Knoten überprüft.

## 7.2.3 Fazit

In Fallbeispiel II wird die betrachtete MES-Lösung mit der virtuellen Fabrik integriert. Vorgehen und Werkzeuge ermöglichen die erfolgreiche virtuelle Inbetriebnahme des MES in einer frühen Phase des Fabriklebenszyklus, die die eingangs formulierten Ziele vollständig erfüllt. Der zusätzliche Aufwand zur Modellierung und Konfiguration der virtuellen Fabrik bewegt sich in der Größenordnung von etwa drei Manntagen und wird vom Kunde als wirtschaftlich sinnvoll eingestuft. Der modellgetriebene Ansatz zur Erstellung des Strukturmodells stellt einen wichtigen Fortschritt im Vergleich zum Fallbeispiel I dar und liefert einen wesentlichen Beitrag zur Reduktion der Modellierungszeit. Darüber hinaus bietet der Ansatz durch die Konfiguration gut getesteter Komponenten große Vorteile im Hinblick auf die Qualität der erzeugten virtuellen Fabrik. Die Skalierbarkeit für größere Anzahlen von Anlagen kann ebenfalls gezeigt werden. Ein wichtiger Schlüssel

dafür ist der Einsatz des konfigurierbaren, generischen Emulatorstyps im Zusammenspiel mit den Werkzeugen zur automatisierten Generierung der Konfigurationen aus den ACM-Ergebnissen, die in strukturierter Form vorliegen.

### 7.3 Bewertung

Die diskutierten Fallbeispiele zeigen, dass das Verfahren und die zugehörigen Werkzeuge zum emulationsgestützten MES-Engineering für eine große Bandbreite von Produktionssystemen in der PV-Industrie Anwendung finden können und im praktischen Einsatz ein realer Nutzen erzielt werden kann. Bei der Erprobung im Industrieumfeld wurde deutlich, dass die in Kapitel 3.6 abgeleiteten Anforderungen an die Emulationsplattform und das Verfahren tatsächlich die Anforderungen realer MES-Engineering-Projekte berücksichtigen und dass das entwickelte Verfahren und die vorgestellten Werkzeuge diese Anforderungen erfüllen. Für den zukünftigen industriellen Einsatz außerhalb der beschriebenen Evaluationsprojekte sind die bisher prototypisch implementierten Werkzeuge in entsprechende Produktivimplementierungen zu überführen und um Funktionalitäten zu ergänzen, die dem Anwender die konsistente Bedienung erleichtern.

Wie bei allen Engineering-Methoden, die zunächst zusätzlichen Aufwand für Projekte generieren, der nicht unmittelbar in den wertschöpfenden Entstehungsprozess des zu entwickelnden Systems eingeht, ist das Verhältnis des Aufwands zum erwarteten Nutzen genau abzuwägen. Der Zielbeschreibung für den Emulationseinsatz und dem darauf abgestimmten Modellierungskonzept kommt daher eine hohe Bedeutung zu. Die Aktivität der Emulatorstypentwicklung liefert potentiell einen besonders hohen Beitrag zu diesen Zusatzaufwänden. In dem Moment, in dem für Experimente die Generierung realitätsnaher Prozess- und Messdaten für einen komplexen Prozess oder die Verarbeitung realer Prozessprogrammen erforderlich werden, sind hohe Aufwände für die Entwicklung des Emulatorstyps zu erwarten, die für das MES-Engineering mit hoher Wahrscheinlichkeit keinen adäquaten Nutzen bringen werden. Dennoch bietet die Emulationsplattform aufgrund ihrer Offenheit die Möglichkeit, Anlagen mit hohem Detaillierungsgrad mit gut abschätzbarem Aufwand abzubilden, indem externe Systeme, wie HiLSen oder virtuelle Anlagen des Anlagenherstellers, soweit verfügbar, in die virtuelle Fabrik eingebunden werden. In der praktischen Anwendung der Emulationsplattform zur virtuellen Inbetriebnahme von MES-Lösungen in der PV-Industrie hat sich allerdings gezeigt, dass in vielen Fällen andere Eigenschaften wichtiger sind als ein möglichst hoher Detaillierungsgrad bei der Abbildung einzelner Anlagen: Dazu gehören insbesondere eine hohe Flexibilität bei der Abbildung von Logistiksystemen, eine möglichst umfassende native Unterstützung zur Implementierung von Emulatorstypmodellen und die Fähigkeit zur Einbindung verschiedenster Protokollfamilien für die IT-Anbindung.

## 8 Zusammenfassung und Ausblick

Die Umwandlung von nahezu unbegrenzt verfügbarer Sonnenenergie in elektrische Energie gilt als ein vielversprechender Ansatz, um den wachsenden Energiebedarf weltweit, unabhängig von konventionellen Energieträgern, auch in Zukunft zu sichern. Vor diesem Hintergrund und befeuert durch staatliche Subventionsprogramme erlebte die PV-Industrie, die die technischen Systeme zur Realisierung dieses Ansatzes zur Verfügung stellt, weltweit über die letzten Jahre einem steilen Anstieg von Produktionskapazitäten und erzielten Umsätzen. Damit die Erzeugung von Solarstrom in Zukunft auch ohne Subventionsprogramme wirtschaftlich interessant wird, sind signifikante Kostensenkungen unabdingbar, die den Preis einer Kilowattstunde (kWh) Solarstroms auf, oder besser unter, das Niveau einer kWh konventionell erzeugten Stroms drücken. Der Übergang vom Verkäufermarkt zum Käufermarkt setzt die PV-Industrie zusätzlich unter Druck. Während zunächst der Branchenfokus eher auf schnellem Wachstum und hohen Ausbringungsmengen lag, erhalten durch die neue Situation Kriterien wie der Herstellungspreis, die Qualität und die Eröffnung neuer Märkte und Geschäftsmodelle eine zentrale Bedeutung.

Den beschriebenen Herausforderungen sucht die PV-Industrie auf mehreren Ebenen zu begegnen. Eine dieser Ebenen ist die der Produktionssysteme, die entlang der Lieferkette der Branche zum Einsatz kommen. Der Produktions-IT, speziell der Klasse der Manufacturing Execution Systems (MESs), wird für die Optimierung der Produktionssysteme eine wichtige Bedeutung zugemessen. Gleichzeitig sieht die PV-Industrie im verstärkten Einsatz solcher IT-Systeme zusätzliche Risiken im gesamten Fabriklebenszyklus – insbesondere für den Fall, dass solche Systeme für die Aufrechterhaltung des Produktionsbetriebs zwingend notwendig sind. Stellvertretend seien das Risiko der Verzögerung des Anlaufs der Fabrik aufgrund von Softwarefehlern oder Zeitverzug in Software-Teilprojekten, das Risiko von Unterbrechungen des Produktionsbetriebs, die durch die Software ausgelöst werden, und Risiken, die im Rahmen von Fabrikerweiterungen und notwendigen Software-Migrationen entstehen, genannt. Letztere wiegen besonders schwer, wenn Erweiterungen und Migrationen im laufenden Produktivbetrieb durchzuführen sind.

Das Verfahren zum emulationsgestützten MES-Engineering, das im Rahmen dieser Arbeit entwickelt wurde, zielt darauf ab, im Kontext der PV-Industrie angemessene MES-Lösungen frühzeitiger im Fabriklebenszyklus verfügbar und die mit dem MES-Engineering verbundenen Risiken besser beherrschbar zu machen. Damit soll der PV-Industrie der Zugang zu MES-Lösungen erleichtert werden, die die Optimierungsbemühungen der Branche unterstützen. Dazu lag dieser Arbeit der folgende Forschungsansatz zugrunde:

In einem ersten Schritt wurde die Ausgangssituation der PV-Industrie im Hinblick auf die Lieferkette, ausgewählte Fertigungsprozessketten und die genutzten Produktionssysteme diskutiert. Die eher unscharfe Verwendung der Begrifflichkeiten der Produktions-IT und

des MES in Industriepraxis und Literatur erforderten zunächst deren allgemeine Einordnung und Abgrenzung und schließlich ihre Definition im Kontext des Anwendungsgebiets. Aufgrund ihrer Bedeutung für den Ansatz des emulationsgestützten MES-Engineering wurde ein besonderes Augenmerk auf die Beschreibung von Normungsaktivitäten im Umfeld der Produktions-IT gelegt, insbesondere der IT-Schnittstellen von Produktionsanlagen, die der Integration dieser Anlagen mit der MES-Ebene dienen.

Die Aufgabenstellungen und die daraus resultierenden Risiken des MES-Engineering im Kontext des Fabriklebenszyklus wurden in einem zweiten Schritt einer genaueren Analyse unterzogen. Zur Erreichung der Zielstellung wurde ein emulationsgestütztes MES-Engineering-Verfahren vorgeschlagen, das auf einer virtuellen Fabrik basiert, die die reale Fabrik aus der Perspektive der MES-Ebene an der IT-Integrationschnittstelle der Anlagen abbildet und an die reale MES-Lösung angebunden wird. Charakteristische Eigenschaften der virtuellen Fabrik sind insbesondere die vollständig parallelisierte Ausführung der Prozesse auf den integrierten virtuellen Anlagen, die ausgeprägte Skalierbarkeit, die die Verwaltung und den parallelen Betrieb einer großen Zahl von Anlagen entsprechend der Situation in realen Fabriken erlaubt und ein dem realen Projektumfeld angepasstes Maß an Flexibilität zur Modellierung von Produktionssystemen. Im Rahmen der Grobkonzeption wurde aus den Ergebnissen der Analyse das Konzept einer Emulationsplattform entwickelt, das die technische Umgebung zur Modellierung und Ausführung virtueller Fabriken skizziert und das Anforderungsprofil an die Emulationsplattform einerseits und das Engineering-Verfahren andererseits abgeleitet.

Das emulationsgestützte Verfahren des MES-Engineering wurde in der Folge in den Kontext des Stands der Wissenschaft und Technik eingeordnet. Berücksichtigung fanden insbesondere Ansätze zur virtuellen Inbetriebnahme produktionsnaher IT-Systeme und zur Optimierung der Engineering-Aufgaben im Bereich der Produktions-IT. Dabei konnten keine existierenden Werkzeuge und zugehörigen Engineering-Verfahren identifiziert werden, die das Anforderungsprofil zufriedenstellend erfüllten.

Die Weiterentwicklung der Konzeption erfolgte auf drei Ebenen. Die erste Ebene umfasste die Überführung der Konzeption und der Anforderungen an die Emulationsplattform in ein Metamodell zur Beschreibung virtueller Fabriken und eine entsprechende Ausführungsumgebung. Über die Identifikation disjunkter Verantwortungsbereiche wurde ein dreischichtiges abstraktes Architekturmodell entworfen. Für jede Schicht wurden Aufgabenbereich und Abhängigkeiten spezifiziert. Die Beschreibung der Schichten wurde durch die Definition von Komponenten und Diensten und deren Interaktion detailliert. Ausgehend vom Architekturmodell wurden durch eine statische Betrachtung die Primitive des Metamodells definiert, die die Modellierung virtueller Fabriken ermöglichen. Diese Modelle können auf der Emulations ausgeführt werden kann. Eine besondere Rolle in der Diskussion des statischen Modells spielte die Abbildung von Anlagen durch Emulator Typen und Emulatoren, deren Integration in die virtuelle Fabrik über das virtuelle Logistiksystem und die gebündelte Konfiguration der virtuellen Fabrik mit Hilfe von Szenarien. Die statische Betrachtung wurde durch eine dynamische Betrachtung ergänzt, die das Verhalten der Modellprimitive über die Zeit beschreibt. Schwerpunkte der dynamischen Betrachtung waren die Lebenszyklen von Primitiven, die Synchronisation paralleler Prozesse und die Steuerung von Szenarien. Die zweite Ebene umfasste die iterative, prototypische Implementierung der Emulationsplattform auf Basis des Metamodells. Diese



wurde so weit vorangetrieben, dass eine Evaluierung der Emulationsplattform im Kontext von realen MES-Engineering-Projekten möglich war.

Die Entwicklungen auf der dritten Ebene zielten auf die nahtlose Integration des Werkzeugs der Emulationsplattform in die Prozesse des MES-Engineering, also die Entwicklung des Verfahrens zum emulationsgestützten MES-Engineering, ab. Die Integration in einen Engineering-Prozess wurde exemplarisch für ein iteratives, inkrementelles Vorgehen beschrieben, das sich, entsprechend der Ergebnisse der Analyse, besonders gut für die Rahmenbedingungen der PV-Industrie eignet. Die entwickelte Vorgehensweise zur Erzeugung und zum Einsatz virtueller Fabriken lässt sich darüber hinaus in vielen weiteren industrietypischen Vorgehensmodellen einsetzen.

Das emulationsgestützte MES-Engineering wurde im Rahmen realer Projekte mit einem Kooperationspartner aus der Industrie erprobt und für zwei industrielle Fallbeispiele, eine Konzentratoren-PV-Linie (Back-end) und eine Dünnschicht-Linie (Front-end / Back-end), exemplarisch besprochen. In beiden Fällen konnte der vom Emulationseinsatz erwartete Nutzen bestätigt werden.

Über den in der vorliegenden Arbeit beschriebenen Stand hinaus bietet das Konzept des emulationsgestützten MES-Engineering noch verschiedene Ansatzpunkte für zukünftige Weiterentwicklungen:

Weiterer Entwicklungsbedarf besteht im Bereich der Werkzeuge der Emulationsplattform. Im Zuge der Erprobung konnte die grundsätzliche Eignung der vorgeschlagenen Werkzeuge gezeigt werden. Für die Projektpraxis hilfreich wären insbesondere Erweiterungen der Komponenten des virtuellen Logistiksystems und des Emulator-SDK, Werkzeuge zur graphischen Modellierung von Emulatoren und die Optimierung der Werkzeuge zur Modellierung virtueller Fabriken. In gleicher Weise ist der Client zur Steuerung und Überwachung der Emulationsplattform zur Laufzeit zu optimieren und zu erweitern.

Das Potential eines werkzeugunterstützten ACM-Vorgehensmodells für das MES-Engineering wurde bereits in der industriellen Anwendung nachgewiesen (vgl. Kapitel 4.5.2). Eine enge Kopplung des Prozesses und der Werkzeuge mit dem Vorgehen zur Erzeugung virtueller Fabriken lässt zahlreiche Synergien erwarten. Für beide Aufgabenstellungen besitzt eine präzise Spezifikation von Schnittstellen und Betriebsszenarien eine hohe Bedeutung. Würde im ACM-Prozess eine hinreichend formale Beschreibung genutzt, könnte diese bereits zur automatisierten Generierung einfacher Emulatortypen herangezogen werden, die zumindest Experimente mit den spezifizierten Betriebsszenarien erlaubten. Diese Synergien könnten über den gesamten Fabriklebenszyklus optimal genutzt werden, wenn ACM-Prozess und emulationsgestützter MES-Engineering-Prozess über eine gemeinsame IT-Plattform gekoppelt würden. Darüber hinaus würde diese Plattform Inkonsistenzen zwischen virtueller und realer Fabrik vermeiden helfen.

In das Gesamtsystem aus Emulationsplattform und gekoppeltem MES könnte ein Simulationssystem integriert werden, das die nahtlose Verwendung der virtuellen Fabrik und des MES-Datenmodells für verteilte Logistiksimulationen ermöglicht. Damit würden sich weitere Synergieeffekte im Hinblick auf Entwicklung und Pflege dieser Modelle ergeben.



# Summary and Outlook

## **An emulation-supported MES-engineering approach for the PV-industry**

Converting solar energy into electrical energy is considered to be a promising approach to satisfy the growing need for energy in future and to gain independence of the limited reserves of conventional sources of energy. Based on this assumption and fueled by governmental stimulus programs, the global photovoltaic industry (PV-industry) has seen a tremendous growth of production capacities and of total revenue over the last years. The PV-industry supplies technical systems and services required to directly convert solar energy into electrical energy. However, to raise real commercial interest independent of government subsidies and to secure long-term commercial success, the PV-industry has to overcome several challenges: The consumer price for solar-generated power per kWh has to be significantly reduced, at least to the same price, or better below the price, of one kWh of electrical energy from the grid. Furthermore, the industry has to cope with the transition from a seller's market to a buyer's market, which has put the PV-industry under additional pressure. While the industry before the transition rather focused on factors such as quick growth and large output quantities, new criteria, such as manufacturing costs, product quality, new markets and new business models, gain in importance after the transition.

The PV-industry follows different approaches to face these challenges. One approach concerns the optimization of production systems used along the industry's supply chain. Production-IT systems, especially the class of Manufacturing Execution Systems (MESs), are considered to have the potential to significantly contribute to the optimization of these production systems. At the same time, the broader utilization of production-IT systems introduces additional sources of risks along the life cycle of factories, especially for those that explicitly rely on such IT systems for their operation. The following risks are frequently mentioned: Delays of the factory ramp-up due to defective or inappropriate software or due to delays in software-related sub-projects, downtimes of the factory caused by defective software and additional risks for factory extension or software migration projects, especially, if they are executed while the factory is in productive mode.

To exploit the potential of MES solutions and to gain added value in real production environments, appropriate MES solutions need to be available earlier in the factory life cycle and the risks mentioned before need to be controlled more tightly. The emulation-supported MES-engineering approach for the PV-industry developed in this thesis targets these goals. The following scientific approach has been used for this thesis:

Firstly, the initial situation of the PV-industry was discussed considering the supply chain, selected manufacturing process chains and typical production systems used for the implementation of these process chains. Neither the term production-IT nor the term MES are consistently used, both in literature and in industry. For this reason, the

general use of these terms was illuminated and defined for the selected field of application. Due to the relevance of standards for the emulation-supported MES-engineering approach, special attention was turned to standardization in the production-IT environment, especially to the standard frameworks supporting vertical integration of production equipment with the MES layer.

Secondly, tasks and related risks of the MES-engineering process were analyzed in detail in the context of a generic factory life cycle. The life cycle definition used here takes specific aspects of the PV-industry into account. From the analysis results, the concept of the emulation-supported MES-engineering approach was derived. It supports MES-engineering tasks to be executed in several phases of the factory life cycle and helps to mitigate a set of critical risks associated with these tasks. The concept makes use of a virtual factory that reproduces the real factory from the MES-layer's point of view. The virtual factory interacts with the real MES-layer and is created and maintained within the MES-engineering process. The general concept was broken down into two main components: The emulation platform that provides the technical environment to model and to execute virtual factories, and the engineering process that targets to the efficient and effective use of this tool throughout the life cycle of an MES-solution. Both components were specified by corresponding requirements sets. Particular requirements concerning the emulation platform are:

- The ability to reflect the level of parallelism MESs have to cope with in the real factory environment.
- The distinct level of scalability, supporting the operation of a large amount of production equipment to reflect the situation of real PV factories.
- The high level of flexibility related to the production system to be modeled and executed, reflecting the variety of production systems used in PV manufacturing.

Thirdly, the emulation-supported MES-engineering approach was classified with respect to the state-of-the-art of science and technology. Approaches for virtual commissioning of production-IT systems and approaches targeting the optimization of engineering tasks in the production-IT environment were mainly considered for the classification. Neither the available tools nor the available engineering approaches were able to satisfy the requirements identified in the preceding analysis.

Fourthly, the development of the emulation-supported MES-engineering approach was divided into two three sub-projects. The first sub-project established the foundations of the virtual factory environment: The conceptual design was refined, sets of disjoint functionalities were identified and transformed into a basic three-tier architecture model consisting of the execution environment, the platform services layer and the user interaction layer (see figure 1). The model describes both, scope and interdependencies for each tier. The basic architecture model was further broken down into components, services and their interaction model.

Within the framework provided by the architecture, a metamodel was specified that comprises a set of primitives or model elements supporting the description of virtual factories such that they can later be instantiated and executed in the distributed environment of the emulation platform. It comprises static as well as dynamic aspects. The

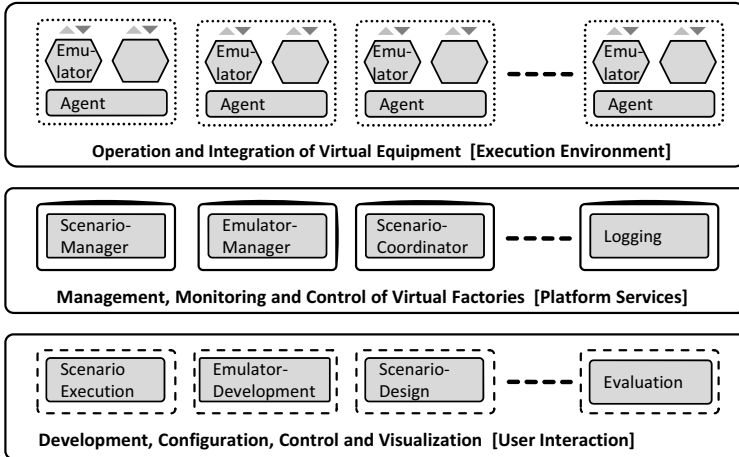


Abbildung 1: The three-tier architecture model of the emulation platform

virtual factory was partitioned into the emulator type model, the structure model, the deployment model and the scenario model. The model elements used within these models were discussed bottom-up, starting from those used to describe virtual equipment, including `EmulatorTypes`, `Emulators`, `EmulatorConfigurations` and `States`. On top of virtual equipment, the virtual equipment environment supplies model elements to integrate virtual equipment into virtual factories, with major elements being the virtual logistics system and logistics objects. In addition to the structure model, virtual factories are linked to scenario models describing consistent configurations of the virtual factory and deployment models specifying the distribution of executable model elements throughout the emulation platform. Both of them lay the foundation for manifold experiments at runtime. The dynamic view complements the static view by describing the behavior of metamodel primitives over time. The synchronization concept for federated emulators is one item of particular interest of the dynamic view.

The second sub-project covered the implementation of the emulation platform in the form of a prototype system to make it available for trials and appraisal in real life projects. An overview on the major use case groups supported by the prototype system is given in figure 2. It distinguishes between the build time use cases groups (1), (2) and (3) and the runtime use case group (4). The repository infrastructure of the emulation platform provides users the ability to manage instances of model elements and promotes re-use of these elements at different levels of granularity (1). If the model elements to reproduce a given real-world scenario are not yet available, the prototype system provides users with tools to model specific types of virtual equipment (2) and virtual factories (3) – including structure models, scenario models and deployment models. At runtime, users may execute scenarios available in the repository (4). To execute a given scenario, the platform deploys model components required by the scenario to the platform execution

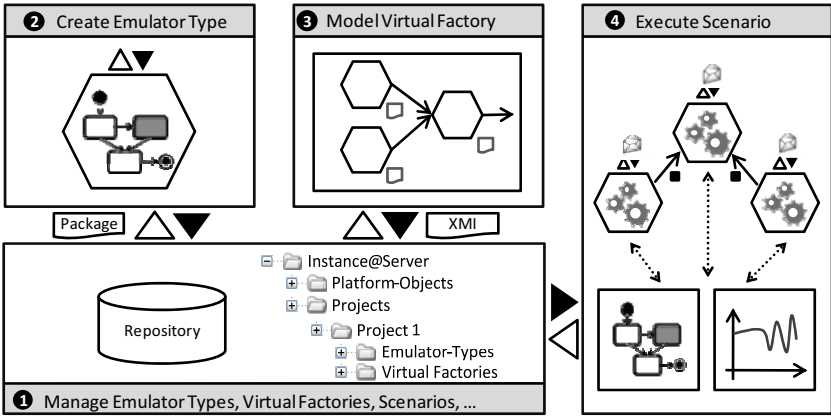


Abbildung 2: Overview: The major use case groups of the prototypical implementation of the emulation platform

environment and runs it according to the specification of the dynamic model.

The third sub-project targeted the efficient and effective integration of the emulation support into the MES-engineering process. The integration methodology takes the aspect of the virtual factory model setup and maintenance as well as the model utilization into account. The engineering approach was exemplified for an iterative, incremental process model, which was considered to be specifically well suited for the current basic conditions of the PV-industry. Figure 3 illustrates the process model and maps the phases of the MES-engineering process (right side, lower process flow) to the phases of the factory life cycle for greenfield factory setup projects (right side, upper process flow). The focus indicator outlined on top of the factory life cycle sketches the weight of each of the five activity types shown in the interaction model on the left side for an iteration  $I_i$  executed at a given point in time or phase in the MES-engineering process, respectively. For non-greenfield projects, the MES-engineering process, including the focus definition, is moved/scaled alongside the factory life cycle. Figure 4 complements figure 3 with regard to the virtual factory model utilization and summarizes the nature of the emulation support available for the different phases of the MES-engineering process.

Fifthly, the emulation-supported MES-engineering approach was implemented and appraised in real life MES-engineering projects in cooperation with an MES-supplier. Two exemplary case studies describing its application in industry projects were discussed in this thesis: The first one in the context of a concentrator-PV back-end, the second one in the context of a thin-film line setup project (front-end and back-end). Both cases confirmed the expected benefits of the emulation-supported MES-engineering approach.

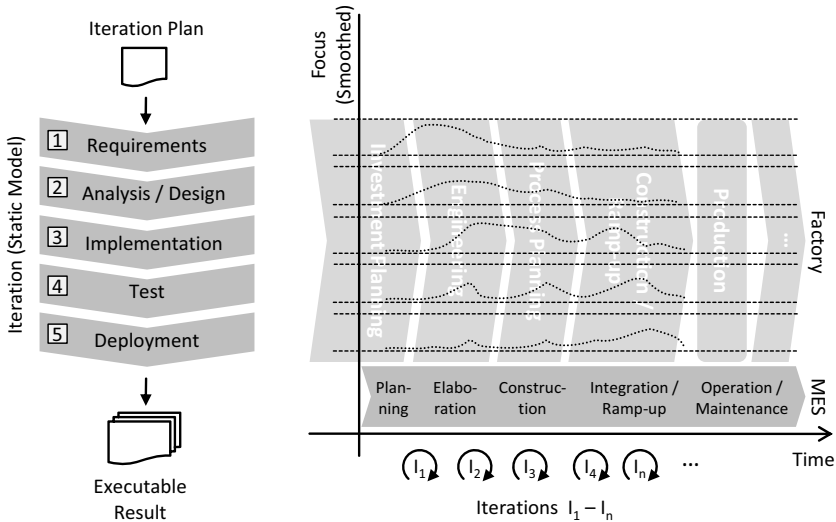


Abbildung 3: An iterative, incremental approach to MES-engineering (based on RUP, see chapter 3.4)

## Outlook

There are several starting points for future work beyond the state of the emulation-supported MES-engineering approach described in this thesis:

- Further research and development efforts are required in the area of the tools and the frameworks of the emulation platform. The application of the platform in real-life projects proved that the tools proposed are generally appropriate for the tasks to be fulfilled. For a comprehensive transfer of the approach into practice, some components of the emulation platform require extensions and improvements: The components of the virtual logistics system, the emulator-SDK, the tools to model virtual factories and the client used to monitor and control the platform at runtime. Furthermore, graphical modeling tools for EmulatorTypes would be helpful.
- The potential of the implementation of a tool-supported ACM process model in the context of the MES-engineering process was already demonstrated in practice (see chapter 4.5.2). Integrating the ACM process and tools closely with the proposed approach to generate virtual factories is expected to yield significant synergistic effects. For both tasks, the precise specification of interfaces and operational scenarios is of vital importance. If the ACM process used a sufficiently formal description model, it could be used to automatically generate simple EmulatorTypes supporting experiments based on the operational scenarios considered in the specification. These synergies could be utilized throughout the factory life cycle in an optimized way, if both, the ACM process as well as the emulation-supported MES-engineering

MES Engineering Phase	Utilization of Virtual Factories
Planning	<ul style="list-style-type: none"> <li>• Support MES assessment/selection process (verify major system properties, including functionality, load behavior, scalability, usability)</li> </ul>
Elaboration	<ul style="list-style-type: none"> <li>• Prototyping-support for analysis and design tasks (early customer feedback)</li> <li>• Architecture evaluation based on prototypes</li> </ul>
Construction	<ul style="list-style-type: none"> <li>• System/integration tests (including functionality, load behavior, availability, usability, early customer feedback)</li> </ul>
Integration / Ramp-up	<ul style="list-style-type: none"> <li>• User/administrator training</li> </ul>
Operation / Maintenance	<ul style="list-style-type: none"> <li>• Safeguard critical engineering-tasks (such as software updates and upgrades)</li> </ul>

Abbildung 4: Nature of emulation support over the different phases of the MES-engineering process

process were integrated by means of a common IT platform. Furthermore, the platform would help to avoid inconsistencies between the real factory and the virtual factory.

- The overall system consisting of the emulation platform and the coupled MES could be extended by a simulation engine that allows for the re-use of equipment models and the virtual logistics system on the one hand and the MES data model on the other hand to enable distributed logistics simulations. This would lead to further synergistic effects with regard to the efforts required to create and maintain factory models.



# Literaturverzeichnis

- [van der Aalst und Hamburg 2008] AALST, L. van der ; HAMBURG, M.: Geschäftsorientiertes Testmanagement. In: SIGS DATACOM GMBH (Hrsg.): *OBJEKTSPEKTRUM Online-Ausgabe 2008 - Schwerpunkt Testing* Bd. 2008. Lindlastr. 2c, 53842 Troisdorf : SIGS DATACOM GmbH, 2008. – URL [http://www.sigs.de/publications/os/2008/testing/hamburg\\_vonalst\\_OS\\_testing\\_08.pdf](http://www.sigs.de/publications/os/2008/testing/hamburg_vonalst_OS_testing_08.pdf). – Zugriffsdatum: 08.10.2008
- [Amans 2008] AMANS, Bob: General Motors Emulation for Virtual Validation. In: SAUER, Olaf (Hrsg.) ; SUTSCHET, Gerhard (Hrsg.): *Karlsruher Leitetchnisches Kolloquium 2008*. Stuttgart : Fraunhofer IRB Verlag, 2008, S. 87–104
- [Ammermann 2007] AMMERMANN, Christoph: Virtuelle Steuerungsmodelle zur Bediener-schulung im Fertigungsanlauf. In: DENKENA, Berend (Hrsg.) ; BRECHER, Christian (Hrsg.): *Ramp-Up/2 - Anlaufoptimierung durch Einsatz virtueller Fertigungssysteme* Bd. 5. Frankfurt am Main : VDMA-Verl., 2007, S. 129–140
- [Ammermann und Lohse 2007] AMMERMANN, Christoph ; LOHSE, Wolfram: Virtueller Fertigungsanlauf. In: DENKENA, Berend (Hrsg.) ; BRECHER, Christian (Hrsg.): *Ramp-Up/2 - Anlaufoptimierung durch Einsatz virtueller Fertigungssysteme* Bd. 5. Frankfurt am Main : VDMA-Verl., 2007, S. 1–14
- [ANSI/ISA-S95 Part I 2000] ANSI/ISA: *ANSI/ISA-S95.00.01-2000: Enterprise-Control System Integration, Part I: Models and Terminology*. 2000
- [ANSI/ISA-S95 Part II 2001] ANSI/ISA: *ANSI/ISA-S95.00.02-2001: Enterprise-Control System Integration, Part II: Object Model Attributes*. 2001
- [ANSI/ISA-S95 Part III 2005] ANSI/ISA: *ANSI/ISA-S95.00.03-2005: Enterprise-Control System Integration, Part III: Activity Models of Manufacturing Operations Management*. 2005
- [AppliedMaterials 2009] APPLIED MATERIALS, Inc: *Applied SunFab Thin Film Line*. 2009. – URL [http://www.appliedmaterials.com/products/solar\\_sunfab\\_3.html](http://www.appliedmaterials.com/products/solar_sunfab_3.html). – Zugriffsdatum: 10.08.2009
- [Aulich 2009] AULICH, Hubert: Solarvalley Mitteldeutschland: Accelerating growth for the PV industry. In: *3rd SEMI Photovoltaic Fab Managers Forum in Dresden*. Brüssel : Semiconductor Equipment and Materials International, 2009
- [Baker 2001] BAKER, Mark: *Cluster Computing White Paper*. 2001. – URL <http://arxiv.org/abs/cs/0004014v2>. – Zugriffsdatum: 13.08.2010

- [Balzert 2001] BALZERT, Helmut: *Lehrbuch der Software-Technik - Bd 1: Software-Entwicklung*. Heidelberg u.a. : Spektrum Akademischer Verlag, 2001 (Lehrbücher der Informatik)
- [Balzert 2008] BALZERT, Helmut: *Lehrbuch der Software-Technik: Softwaremanagement*. 2. Auflage. Heidelberg u.a. : Spektrum Akademischer Verlag, 2008 (Lehrbücher der Informatik)
- [Bär u. a. 2008] BÄR, T. ; MANDEL, S. ; SAUER, O. ; EBEL, M.: Durchgängiges Datenmanagement durch plug-and-work zur virtuellen Linienbetriebnahme. In: SAUER, Olaf (Hrsg.) ; SUTSCHET, Gerhard (Hrsg.): *Karlsruher Leittechnisches Kolloquium 2008*. Stuttgart : Fraunhofer IRB Verlag, 2008, S. 105–121
- [Baylies u. a. 2008] BAYLIES, Win ; BURKHART, Marty ; COOK, Don ; HOCKETT, Dick ; MEIER, Matthias ; WEISS, Bettina: Structuring standards for the photovoltaic manufacturing industry. In: *Photovoltaics International* (2008), Nr. 2, S. 151–154
- [Beck 2001] BECK, Kent ; BEEDLE, Mike ; BENNEKUM, Arie van ; COCKBURN, Alistair ; CUNNINGHAM, Ward ; FOWLER, Martin ; ET. AL.: *Manifesto for Agile Software Development*. 2001. – URL <http://agilemanifesto.org/>. – Zugriffsdatum: 24.06.2009
- [Beitel 2008] BEITEL, Chris: New Standards to Drive Cost. In: *SEMI NA Photovoltaics Advanced Manufacturing Technology Conference 2008 in San Francisco*. San Jose : Semiconductor Equipment and Materials International, 2008
- [Bergholz 2008] BERGHOLZ, Werner: From Lab to Fab using Microelectronics Experience. In: *2nd SEMI Photovoltaic Fab Managers Forum in Dresden*. Brüssel : Semiconductor Equipment and Materials International, 2008
- [Berning 2001] BERNING, Ralf: *Grundlagen der Produktion: Produktionsplanung und Beschaffungsmanagement*. Düsseldorf : Cornelsen, 2001 (Cornelsen Studien-Baustein Wirtschaft)
- [Berwind 2009] BERWIND, Joseph: A seller's market: No More. What is the impact on the PV supply chain? Price declines in a credit constrained world with oversupply, and the consequences of a rigid supply chain. In: *3rd SEMI Photovoltaic Fab Managers Forum in Dresden*. Brüssel : Semiconductor Equipment and Materials International, 2009
- [BMU 2010] : *Energiekonzept der Bundesregierung: Langfristige Strategie für die künftige Energieversorgung*. 2010. – URL <http://www.bmu.de/energieeffizienz/downloads/doc/46394.php>. – Zugriffsdatum: 30.09.2010
- [Bracht und Hagmann 1998] BRACHT, U. ; HAGMANN, M.: Die ganze Fabrik im Simulationsmodell. In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 93 (1998), Nr. 7/8, S. 345–348
- [Brecher und Buchner 2006] BRECHER, C. ; BUCHNER, T.: Modellbasiertes Engineering verteilter Fertigungsleitsysteme. In: *Simulationstechnik in der Produktion, Fortschritt-Berichte VDI, Reihe 2: Fertigungstechnik* 655 (2006), S. 10–31

- [Brecher u. a. 2008] BRECHER, C. ; FAYZULLIN, K. ; POSSEL-DÖLKEN, F.: Intelligent operations control: architecture for seamless integration of scheduling and execution. In: *Production Engineering* 2 (2008), Nr. 3, S. 293–301
- [Brecher 2008] BRECHER, C. Herfs W. Kolster D. Jensen S. Plessow M.: Plug & Play – eine Vision rückt näher: Sind selbstkonfigurierende Visualisierungen von komponentenbasierten Automatisierungsanlagen schon bald Realität? In: *A&D Kompendium 2008/2009* (2008), S. 26–29
- [BSW-Solar 2009a] BSW-SOLAR: *Erneuerbare-Energien-Gesetz (EEG): Kurzinformation zum EEG 2009*. 2009. – URL <http://premium.solarfoerderung.de/download/count.cfm?ID=34>. – Zugriffsdatum: 04.05.2009
- [BSW-Solar 2009b] BSW-SOLAR: *Fotoarchiv*. 2009. – URL <http://www.solarwirtschaft.de/medienvertreter/fotoarchiv.html?L=kfcbgdtknh>. – Zugriffsdatum: 31.07.2009
- [BSW-Solar 2009c] BSW-SOLAR: *Statistische Zahlen der deutschen Solarstrombranche (Photovoltaik)*. 2009. – URL [http://www.solarwirtschaft.de/fileadmin/content\\_files/faktenblatt\\_pv\\_0309.pdf](http://www.solarwirtschaft.de/fileadmin/content_files/faktenblatt_pv_0309.pdf). – Zugriffsdatum: 21.04.2009
- [Buchner 2008] BUCHNER, Tilman: *Virtuelles Engineering von Fertigungsleitsystemen*. Aachen, RWTH Aachen, Dissertation, 2008
- [Centrotherm 2009] CENTROTHERM PHOTOVOLTAICS: *Produkte & Dienstleistungen - Dünnschichtmodul - Produkte - Turnkey-Produktionslinie*. 2009. – URL <http://www.centrotherm.de/de/produkte-dienstleistungen/duennschichtmodul/produkte/turnkey-produktionslinie.html>. – Zugriffsdatum: 10.08.2009
- [Chung und Jeng 2004] CHUNG, Sheng-Luen ; JENG, Muder: Fabulous MESS and C/CS: an overview of semiconductor fab automation systems: Robotics & Automation Magazine, IEEE. In: *Robotics & Automation Magazine, IEEE* 11 (2004), Nr. 1, S. 8–18
- [Cimetrix 2009] CIMETRIX, INCORPORATED: *GEM Testing Software Product for SECS-II & SECS-I Messages*. 2009. – URL <http://www.cimetrix.com/testconnect.cfm>. – Zugriffsdatum: 21.04.2009
- [Constantinescu u. a. 2009] CONSTANTINESCU, Carmen ; EICHELBERGER, Hanno ; WESTKÄMPER, Engelbert: Durchgängige und integrierte Fabrik- und Prozessplanung: "Grid Engineering for Manufacturing". In: *wt Werkstattstechnik online* 99 (2009), Nr. 3, S. 92–98
- [Constantinescu u. a. 2006] CONSTANTINESCU, Carmen ; HUMMEL, Vera ; WESTKÄMPER, Engelbert: Approach for a Service-oriented Factory Planning Environment. In: *Intelligent Computation in Manufacturing Engineering - CIRP ICME '06 / CD-ROM*. Neapel Italien : CIRP, 2006, S. 6 S.
- [Denkena u. a. 2008] DENKENA, Berend ; AMMERMAN, Christoph ; HOPPE, Patryk: Ramp-Up/2 - Verkürzung von Fertigungsanläufen. In: *wt Werkstattstechnik online* 98 (2008), Nr. 3, S. 143–148

- [Dreiss 2007] DREISS, Philipp: *System für die modellbasierte Integration von Anlagen in die Halbleiterfertigung*. Stuttgart, Universität Stuttgart, Dissertation, 2007
- [Duden FWB 2007] Duden - *Das große Fremdwörterbuch: Herkunft und Bedeutung der Fremdwörter*. 4., aktualisierte Auflage. Mannheim, Leipzig, Wien, Zürich : Dudenverlag, 2007
- [Duden UWB 2007] Duden - *Deutsches Universalwörterbuch*. 6., überarbeitete Auflage. Mannheim, Leipzig, Wien, Zürich : Dudenverlag, 2007
- [Ebel u. a. 2007] EBEL, M. ; OKON, M. ; BAUMANN, M.: 'ProduFlexil': Flexible Produktion mit SOA-Architektur und Plug-and-Work-Mechanismus. In: SPATH, D. (Hrsg.) ; WEISBECKER, A. (Hrsg.) ; HÖSS, O. (Hrsg.) ; DRAWWEHN, J. (Hrsg.): *Science meets Business - Ergebnisse aus aktuellen Projekten der Softwareforschung. Stuttgarter Softwaretechnik Forum 2007: Tagungsband, 23. November 2007*. Stuttgart : Fraunhofer IRB Verlag, 2007, S. 65-74
- [Eclipse 2009] THE ECLIPSE FOUNDATION: *Eclipse.org home*. 2009. – URL <http://www.eclipse.org/>. – Zugriffsdatum: 14.05.2009
- [EclipseTPTP 2009] THE ECLIPSE FOUNDATION: *Eclipse Test & Performance Tools Platform Project*. 2009. – URL <http://www.eclipse.org/tptp/>. – Zugriffsdatum: 13.05.2009
- [EPIA 2008] EPIA: *Solar Generation V - 2008: Solar electricity for over one billion people and two million jobs by 2020*. 2008. – URL [http://www.epia.org/fileadmin/EPIA\\_docs/documents/EPIA\\_SG\\_V\\_ENGLISH\\_FULL\\_Sept2008.pdf](http://www.epia.org/fileadmin/EPIA_docs/documents/EPIA_SG_V_ENGLISH_FULL_Sept2008.pdf). – Zugriffsdatum: 22.04.2009
- [EPIA 2009] EPIA: *Global Market Outlook for Photovoltaics until 2013*. 2009. – URL [http://www.epia.org/fileadmin/EPIA\\_docs/publications/epia/Global\\_Market\\_Outlook\\_Until\\_2013.pdf](http://www.epia.org/fileadmin/EPIA_docs/publications/epia/Global_Market_Outlook_Until_2013.pdf). – Zugriffsdatum: 21.04.2009
- [Erdmann und Dreiss 2006] ERDMANN, Barbara ; DREISS, Philipp: Tool Supported Capability Management for Semiconductor High Volume Equipment Integration. In: *5. Requirements Engineering Tagung 2006 / CD-ROM*. München : HOOD GmbH, 2006, S. 41
- [Fath 2009] FATH, Peter: Die Grid-Parity-Fab. In: *VDMA Jahrestagung Photovoltaik-Produktionsmittel*, 2009
- [Foster 2008] FOSTER, Ian: *There's Grid in them thar Clouds*. 2008. – URL <http://ianfoster.typepad.com/blog/2008/01/theres-grid-in.html>. – Zugriffsdatum: 09.09.2009
- [Foster und Kesselman 2007] FOSTER, Ian ; KESSELMAN, Carl: *The grid: blueprint for a new computing infrastructure*. 2. ed., [3.Dr.]. Amsterdam : Elsevier Kaufmann, 2007 (The Elsevier series in grid computing)
- [Freeman u. a. 2004] FREEMAN, Steve ; MACKINNON, Tim ; PRYCE, Nat ; WALNES, Joe: Mock roles, objects. In: *Companion to the 19th annual ACM SIGPLAN conference*

- on *Object-oriented programming systems, languages, and applications*. Vancouver, BC, CANADA : ACM, 2004, S. 236–246. – URL <http://doi.acm.org/10.1145/1028664.1028765>
- [Großmann 2008] GROSSMANN, Stefan: Die virtuelle Werkzeugmaschine von INDEX. In: SCHNITTLER, Volker (Hrsg.): *CIM - aktueller denn je!* Verband Deutscher Maschinen- und Anlagenbau / Abteilung Informatik, 2008, S. Beitrag 3
- [Gundelach 2008] GUNDELACH, Christian: DMG Virtual Machine. In: SCHNITTLER, Volker (Hrsg.): *CIM - aktueller denn je!* Verband Deutscher Maschinen- und Anlagenbau / Abteilung Informatik, 2008, S. Beitrag 2
- [Heidenblut 2001] HEIDENBLUT, V.: Mit Software Software testen. In: *Hebezeuge und Fördermittel* 41 (2001), Nr. 10, S. 444–446
- [IEEE 1516 2000] IEEE: *IEEE Std 1516-2000: Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) — Framework and Rules*. 2000
- [IEEE 1516.1 2000] IEEE: *IEEE Std 1516.1-2000: Standard for Modeling and Simulation (M&S) High Level Architecture (HLA) — Federate Interface Specification*. 2000
- [InFrame 2010] ADVANCED CLEAN PRODUCTION INFORMATION TECHNOLOGY AG (ACP-IT AG): *InFrame Synapse MES*. 2010. – URL <http://www.acp-it.com/Default.aspx?tabid=148>. – Zugriffsdatum: 06.08.2010
- [INSPIRE 2009] INSPIRE TECHNOLOGY: *SWIFT Machine Emulator (SECS/GEM Simulator Freeware)*. 2009. – URL <http://www.insphere.com.sg/swift.php>. – Zugriffsdatum: 21.04.2009
- [ISOC 2009] INTERNET SOCIETY: *Internet Society (ISOC)*. 2009. – URL <http://www.isoc.org/>. – Zugriffsdatum: 24.03.2009
- [ISO/IEC 7498-1 1994] ISO/IEC: *ISO/IEC 7498-1: Information technology – Open Systems Interconnection – Basic Reference Model: The Basic Model*. 1994
- [ISO/IEC 9126 2001] DIN/ISO/IEC: *DIN 66272/ISO/IEC 9126: Bewerten von Softwareprodukten, Qualitätsmerkmale und Leitfaden zu ihrer Verwendung*. 2001
- [Java 2009] ORACLE CORPORATION: *Developer Resources for Java Technology*. 2009. – URL <http://java.sun.com/>. – Zugriffsdatum: 16.08.2010
- [JavaServiceWrapper 2009] TANUKI SOFTWARE LTD.: *Java Service Wrapper - Download Java Service Wrapper*. 2009. – URL <http://wrapper.tanukisoftware.org/doc/english/download.jsp>. – Zugriffsdatum: 13.05.2009
- [JBoss 2009] : *Community driven open source middleware. - JBoss Community*. 2009. – URL <http://www.jboss.org/>. – Zugriffsdatum: 11.05.2009
- [Jindal und Khandelwal 2005] JINDAL, A. ; KHANDELWAL, R.: System centric mission critical simulation model for MES automation: Simulation Conference, 2005 Proceedings of the Winter. In: *Simulation Conference, 2005 Proceedings of the Winter* (2005), S. 5 pp.

- [Kapitza 2007] KAPITZA, Norbert: Herausforderungen bei Entwicklung und Inbetriebnahme von Fertigungsleitsystemen im Wandel der Zeit. In: DENKENA, Berend (Hrsg.) ; BRECHER, Christian (Hrsg.): *Ramp-Up/2 - Anlaufoptimierung durch Einsatz virtueller Fertigungssysteme* Bd. 5. Frankfurt am Main : VDMA-Verl., 2007, S. 141–158
- [Klemm u. a. 2006] KLEMM, P. ; RÜDELE, H. ; WEIMER, T.: Durchgängiges Engineering von mechatronischen Systemen. In: HEISEL, Uwe (Hrsg.): *FtK2006 - Fertigungstechnisches Kolloquium*. Stuttgart : Gesellschaft für Fertigungstechnik, 2006, S. 101–131
- [Kletti 2007] KLETTI, Wolfhard: Konzeption und Einführung von MES-Systemen. In: *PPS Management* 12 (2007), Nr. 3 Serienfertigung, S. 39–42
- [Konrad u. a. 2009] KONRAD, Konstantin ; BÖTTINGER, Fabian ; SEIDELMANN, Joachim: Design for fab scalability. In: *Photovoltaics International* (2009), Nr. 3, S. 12–15
- [Kraemer und Herrmann 2006] KRAEMER, F. A. ; HERRMANN, P.: Service Specification by Composition of Collaborations—An Example. In: *Web Intelligence and Intelligent Agent Technology Workshops, 2006. WI-IAT 2006 Workshops.*, URL doi: 10.1109/WI-IATW.2006.121, 2006, S. 129–133
- [Lamport 1978] LAMPORT, Leslie: Time, clocks, and the ordering of events in a distributed system. In: *Commun. ACM* 21 (1978), Nr. 7, S. 558–565
- [Lechler u. a. 2008] LECHLER, Armin ; KIRCHER, Christian ; VERL, Alexander: Simulationsgestütztes mechatronisches Engineering: Entwurf und Inbetriebnahme mechatronischer Module und flexibler Produktionssysteme. In: *Wt Werkstatttechnik* 98 (2008), Nr. 5, S. 377–383
- [Li-Baboud u. a. 2008] LI-BABOUD, Ya-Shian ; ZHU, Xiao ; ANAND, D. ; HUSSAINI, S. ; MOYNE, J.: Semiconductor manufacturing equipment data acquisition simulation for timing performance analysis. In: *Precision Clock Synchronization for Measurement, Control and Communication, 2008. ISPCS 2008.*, URL doi:10.1109/ISPCS.2008.4659217, 2008, S. 77–82
- [Lillian 2009] LILLIAN, Jessica: PV Fabrication Plants Slowly Embracing Production-Line Software: Obtaining and analyzing multiple layers of data from the factory floor has become increasingly important. In: *Solar Industry* (2009), Nr. October, S. 76–78
- [Lohse u. a. 2007] LOHSE, Wolfram ; HERFS, Werner ; BRECHER, Christian ; EPPLER, Claus: Anwendungsorientierte Mechatroniksimulation mit realen und virtuellen Steuerungen. In: DENKENA, Berend (Hrsg.) ; BRECHER, Christian (Hrsg.): *Ramp-Up/2 - Anlaufoptimierung durch Einsatz virtueller Fertigungssysteme* Bd. 5. Frankfurt am Main : VDMA-Verl., 2007, S. 100–128
- [Maeda u. a. 2007] MAEDA, K+ ; ISHINO, M. ; KOHNO, E. ; KISHIDA, T.: A userland network emulator with packet capture and replay. In: *2007 International Symposium on Applications and the Internet, Hiroshima, Japan, 15-19 Jan. 2007* (2007)
- [Mareck 2009] MARECK, Ute: What did SPC bring up to now and what are the roadblocks? How Deutsche Cell approaches to high volume high quality production.

- In: *3rd SEMI Photovoltaic Fab Managers Forum in Dresden*. Brüssel : Semiconductor Equipment and Materials International, 2009
- [Meier und Homuth 2005] MEIER, Horst ; HOMUTH, Michael: Erschließung der Potenziale der Digitalen Fabrik in heterogener Systemlandschaft. In: *ZWF Zeitschrift für wirtschaftlichen Fabrikbetrieb* 100 (2005), Nr. 1/2, S. 20–24
- [Meier 2006] MEIER, Matthias: A Web Service Technology-based Factory IT Landscape. In: *The Integration of Manufacturing Systems for Mass Manufacture of Miniature / Micro-Products*. Stuttgart : Universität Stuttgart, 2006, S. S. 91–97
- [Meier 2007] MEIER, Matthias: Exploring Potentials of the Shop Floor IT Landscape for Thin Film Manufacturing. In: *Thin Film Photovoltaics*. Stuttgart : FpF - Verein zur Förderung produktionstechnischer Forschung, 2007, S. S. 72–81
- [Meier 2008a] MEIER, Matthias: *Compressed Results: SEMI Standards PV-EIS Task Force Meeting 05.03.2008 in Bruzelles*. 2008
- [Meier 2008b] MEIER, Matthias: Potentials and Limitations of Standardization in PV Manufacturing. In: *PV and Mo(o)re: Energy - Efficiency - Electronics*. Brüssel : SEMI, 2008, S. 9 S.
- [Meier 2009] MEIER, Matthias: Approaching Factory Automation Standards for the PV-Industry: The PV-EIS example and a European perspective. In: *2009 SEMI PV International Standards Workshop*. Taipeh Taiwan : Semiconductor Equipment and Materials International, 2009, S. 23 Folien
- [Meier 2010] MEIER, Matthias: Manufacturing Execution Systems for Micro-Manufacturing. In: QIN, Yi (Hrsg.): *Micro-manufacturing engineering and technology*. Amsterdam u.a. : Elsevier, 2010, S. 377–393
- [Meier u. a. 2007a] MEIER, Matthias ; BÖTTINGER, Fabian ; MUCKENHIRN, Ralf ; WIECHERS, Ortrun ; FISCHMANN, Christian: An Integrated Approach to Production Logistics and Factory IT Optimization for Photovoltaic Manufacturing. In: *3rd Advanced Photovoltaic Manufacturing Technology Conference*. Brüssel : Semiconductor Equipment and Materials International, 2007, S. 9 S.
- [Meier u. a. 2007b] MEIER, Matthias ; DREISS, Philipp ; SEIDELMANN, Joachim: Möglichkeiten und Grenzen der Standardisierung produktionsnaher IT. In: *PPS Management* 12 (2007), Nr. 4, S. 16–19
- [Meier u. a. 2008] MEIER, Matthias ; FISCHMANN, Christian ; BÖTTINGER, Fabian: Synergien nutzen: Wie die MES-Entwicklung von Werkzeugen der Logistikplanung profitieren kann. In: *Wt Werkstattstechnik* 98 (2008), Nr. 5, S. 422–427
- [Meier und Konrad 2009] MEIER, Matthias ; KONRAD, Konstantin: PVECI - a new PV standard supporting efficient and effective production. In: *Photovoltaics International* (2009), Nr. 4, S. 30–35
- [MESA 2009] MESA INTERNATIONAL: *Manufacturing Enterprise Solutions Association – MESA International*. 2009. – URL <http://www.mesa.org>. – Zugriffsdatum: 23.03.2009

- [MESA-WP1 1997] MESA INTERNATIONAL: *The Benefits of MES: A Report from the Field: White Paper Number 1*. 1997
- [MESA-WP2 1997] MESA INTERNATIONAL: *MES Functionalities & MRP to MES Data Flow Possibilities: White Paper Number 2*. 1997
- [Mönch u. a. 2003] MÖNCH, Lars ; ROSE, Oliver ; STURM, Roland: A Simulation Framework for the Performance Assessment of Shop-Floor Control Systems. In: *Simulation* 79 (2003), Nr. 3, S. 163–170
- [Muckenhirn 2005] MUCKENHIRN, Ralf: *Konfigurierbares Leitsystem für modulare Montagezellen am Beispiel von Festplatten*. Stuttgart, Universität Stuttgart, Dissertation, 2005
- [Muckenhirn und Meier 2008] MUCKENHIRN, Ralf ; MEIER, Matthias: A minimal-invasive approach to configuration management supporting collaborative factory configuration scenarios. In: *Journal of Intelligent Manufacturing* 19 (2008), Nr. 6, S. 735–746
- [MySQL 10] ORACLE CORPORATION: *MySQL :: The world's most popular open source database*. 10. – URL <http://www.mysql.com/>. – Zugriffsdatum: 13.08.2010
- [.Net 2009] MICROSOFT: *Microsoft .NET Framework*. 2009. – URL <http://www.microsoft.com/.NET/>. – Zugriffsdatum: 11.05.2009
- [Nikkarinen und Shemyak 2005] NIKKARINEN, S+ ; SHEMYAK, K+: COSIME: real-life cellular network on the desktop. In: *2005 Joint International Conference on Autonomic and Autonomous Systems and International Conference on Networking and Services (ICAS/ICNS), Papeete, Tahiti, French Polynesia, 23-28 Oct. 2005* (2005)
- [Nokia 2009] NOKIA: *S60 Platform SDKs for Symbian OS, for C++*. 2009. – URL <http://www.forum.nokia.com/info/sw.nokia.com/id/4a7149a5-95a5-4726-913a-3c6f21eb65a5/S60-SDK-0616-3.0-mr.html>. – Zugriffsdatum: 04.08.2009
- [Oerlikon 2009] OERLIKON SOLAR: *Fab End to End Solutions*. 2009. – URL [http://www.oerlikon.com/ecomaXL/index.php?site=SOLAR\\_EN\\_fab\\_end\\_to\\_end\\_solutions](http://www.oerlikon.com/ecomaXL/index.php?site=SOLAR_EN_fab_end_to_end_solutions). – Zugriffsdatum: 10.08.2009
- [OMG 2003] MILLER, Joaquin (Hrsg.) ; MUKERJI, Jishnu (Hrsg.): *MDA Guide Version 1.0.1*. 2003. – URL <http://www.omg.org/docs/omg/03-06-01.pdf>. – Zugriffsdatum: 27.05.2009
- [OMG 2005] OBJECT MANAGEMENT GROUP: *A Proposal for an MDA Foundation Model*. 2005. – URL <http://www.omg.org/docs/ormsc/05-04-01.pdf>. – Zugriffsdatum: 27.05.2009
- [OMG 2006] OBJECT MANAGEMENT GROUP: *Meta Object Facility (MOF) Core Specification: Version 2.0*. 2006. – URL <http://www.omg.org/docs/formal/06-01-01.pdf>. – Zugriffsdatum: 27.05.2009



- [OMG-UML-IS 25.02.2009] OBJECT MANAGEMENT GROUP: *OMG Unified Modeling Language (OMG UML), Infrastructure*. 25.02.2009. – URL <http://www.omg.org/spec/UML/2.2/Infrastructure>. – Zugriffsdatum: 11.06.2009
- [OMG-UML-SS 09.03.2009] OBJECT MANAGEMENT GROUP: *OMG Unified Modeling Language (OMG UML), Superstructure*. 09.03.2009. – URL <http://www.omg.org/spec/UML/2.2/Superstructure>. – Zugriffsdatum: 11.06.2009
- [OPC 2009] OPC FOUNDATION: *The OPC Foundation*. 2009. – URL <http://www.opcfoundation.org>. – Zugriffsdatum: 23.03.2009
- [Ow u. a. 2006] OW, Khiam W. ; KOH, Aik G. ; KHOO, Hwee K. ; SHIH, Chih C.: MES Validation using FAB-wide Equipment Simulation: Advanced Semiconductor Manufacturing Conference, 2006. ASMC 2006. The 17th Annual SEMI/IEEE. In: *Advanced Semiconductor Manufacturing Conference, 2006. ASMC 2006. The 17th Annual SEMI/IEEE* (2006), S. 316–321
- [Page u. a. 2005] PAGE, Bernd ; KREUTZER, Wolfgang ; GEHLESEN, Björn: *The Java simulation handbook: Simulating discrete event systems with UML and Java*. Aachen : Shaker, 2005 (Berichte aus der Informatik)
- [Papandreou u. a. 2003] PAPANDEIOU, N+ ; VARSAMOU, M. ; ANTONAKOPOULOS, T.: xDSL systems prototyping using a flexible emulation environment. In: *Proceedings 14th IEEE International Workshop on Rapid Systems Prototyping, San Diego, CA, USA, 9-11 June 2003* (2003)
- [PEERGROUP 2009a] PEERGROUP: *Advanced SECS/GEM Simulator, Semiconductor Equipment Tester*. 2009. – URL [http://www.peergroup.com/PGPFAT.aspx?utm\\_source=SF&utm\\_medium=Email&utm\\_campaign=PFAT%2Bmailer%201.6](http://www.peergroup.com/PGPFAT.aspx?utm_source=SF&utm_medium=Email&utm_campaign=PFAT%2Bmailer%201.6). – Zugriffsdatum: 21.04.2009
- [PEERGROUP 2009b] PEERGROUP: *Automation Software Testing, SEMI Standards (Standards Testing)*. 2009. – URL <http://www.peergroup.com/PGCCSENVOY.aspx>. – Zugriffsdatum: 21.04.2009
- [Perlin 2002] PERLIN, John: *From space to earth: The story of solar electricity*. 1. Harvard Univ. Press ed. Cambridge, Mass. : Harvard Univ. Press, 2002
- [Pfitzner und Schellenberger 2009] PFITZNER, Lothar ; SCHELLENBERGER, Martin: The impact of APC and standards on the semiconductor industry. In: *3rd SEMI Photovoltaic Fab Managers Forum in Dresden*. Brüssel : Semiconductor Equipment and Materials International, 2009
- [Photon-PV-Lexikon 2007] PHOTON - DAS SOLARSTROM-MAGAZIN: *Photovoltaik-Lexikon T-Z*. 2007. – URL <http://www.photon.de/news/Lexikon/lexikon-w.htm>. – Zugriffsdatum: 30.07.2009
- [Pilz 2008] PILZ, J.: Der Benutzer im Zentrum: Last- und Performancetests mit HP LoadRunner. In: SIGS DATACOM GMBH (Hrsg.): *OBJEKTSpektrum Online-Ausgabe 2008 - Schwerpunkt Testing* Bd. 2008. Lindlastr. 2c, 53842 Troisdorf : SIGS DATACOM GmbH, 2008. – URL [http://www.sigs.de/publications/os/2008/testing/pilz\\_08\\_testing\\_08.pdf](http://www.sigs.de/publications/os/2008/testing/pilz_08_testing_08.pdf). – Zugriffsdatum: 08.10.2008

- [Prinz und Liebe 1998] PRINZ, J. ; LIEBE, P.: Integration von Simulationssoftware mit Leitsystemen. In: *Erfahrungen aus der Zukunft, ASIM-Fachtagung Simulation in Produktion und Logistik*, 8. Berlin : Fraunhofer IPK, 1998, S. 63–71
- [PV-EIS 2007] SEMI STANDARDS - PV-EIS TASKFORCE: *Equipment Interface Scope Requirements: WGI Meeting in Alzenau vom 27.11.2007*. 2007
- [PV-EIS 2008] SEMI STANDARDS - PV-EIS TASKFORCE: *Requirements Survey Result: Workshop in Stuttgart, 09.05.2008*. 2008
- [PV-EIS 2009a] SEMI STANDARDS - PV-EIS TASKFORCE: *Questionnaire Summary V.1.2: Workshop in Berlin vom 01.07.2009*. 2009
- [PV-EIS 2009b] SEMI STANDARDS - PV-EIS TASKFORCE: *SNARF for: New Standard: Specification for Single Substrate Tracking for Photovoltaic Equipment. Activity Number: 4804*. 2009
- [Q-Cells-GB 2009] Q-CELLS: *Geschäftsbericht 2008*. 2009. – URL [http://www.qcells.de/medien/ir/berichte/2008/qcells\\_gb2008\\_deutsch.pdf](http://www.qcells.de/medien/ir/berichte/2008/qcells_gb2008_deutsch.pdf). – Zugriffsdatum: 10.08.2009
- [Rauter 2009] RAUTER, Gerhard: Assessment of supply chain by the Fab Managers Group. In: *3rd SEMI Photovoltaic Fab Managers Forum in Dresden*. Brüssel : Semiconductor Equipment and Materials International, 2009
- [Reddig 2009] REDDIG, Kevin: Overview of automation in the photovoltaic industry. In: *Photovoltaics International* (2009), Nr. 4, S. 18–29
- [Rein u. a. 2008] REIN, S. ; KRIEG, A. ; WEIL, A. ; EMANUEL, G. ; GLATTHAAR, M. ; GROHE, A. ; PREU, R. ; SATTLER, B. ; LORCH, P. ; VAYHINGER, K.: Single-wafer tracking in PV production lines. In: *The compiled state-of-the-art of PV solar technology and deployment*. Munich : WIP-Renewable Energies, 2008, S. 1125–1130
- [Rein und Preu 2007] REIN, S. ; PREU, R.: Do QC and in-line inspection systems enable the future mega-fabs? In: *3rd Advanced Photovoltaic Manufacturing Technology Conference*. Brüssel : Semiconductor Equipment and Materials International, 2007
- [Röck 2007] RÖCK, Sascha: *Echtzeitsimulation von Produktionsanlagen mit realen Steuerungssystemen*. Heimsheim, Universität Stuttgart, Dissertation, 2007
- [Rothermel 2001] ROTHERMEL, Kurt: *Introduction To Distributed Systems SS01: Vorlesungsskript. Institut für Parallele und Verteilte Systeme, Abteilung Verteilte Systeme, Universität Stuttgart*. 2001
- [RTI 2002] RESEARCH TRIANGLE INSTITUTE (RTI) ; NATIONAL INSTITUTE OF STANDARDS AND TECHNOLOGY (NIST) (Hrsg.): *The Economic Impacts of Inadequate Infrastructure for Software Testing: Final Report*. 2002. – URL <http://www.nist.gov/director/prog-ofc/report02-3.pdf>. – Zugriffsdatum: 08.10.2008
- [Ruchat 2009] RUCHAT, David: Global overview of current and future manufacturing capacities for crystalline solar cells and thin-film PV panels. In: *Photovoltaics International* (2009), Nr. 3, S. 21–28

- [RUP 2009] IBM: *IBM - Rational Unified Process (RUP)*. 2009. – URL <http://www-01.ibm.com/software/awdtools/rup/>. – Zugriffsdatum: 24.06.2009
- [Rupp u. a. 2007] RUPP, Chris ; QUEINS, Stefan ; ZENGLER, Barbara: *UML 2 glasklar: Praxiswissen für die UML-Modellierung*. München, Wien : Hanser, 2007
- [Rutschmann 2009] RUTSCHMANN, Ines: Mal was anderes: Concentrix Solar weicht die erste Fabrik für Konzentratormodule in Deutschland ein. In: *PHOTON* (2009), S. 58–61
- [Sauer 2007a] SAUER, O.: Digitale Fabrik und MES. In: *MES Wissen Kompakt 2007. Sonderpublikation der Zeitschrift IT&Production* (2007), S. 18–21
- [Sauer 2007b] SAUER, O.: Informationsdrehscheibe MES - Digitale Fabrik und Manufacturing Execution Systeme wachsen zusammen. In: *AuD Kompendium 2007/2008*. München : publish-industry Verlag, 2007, S. 60–61
- [Sauer und Ebel 2007] SAUER, O. ; EBEL, M.: Plug-and-work von Produktionsanlagen und übergeordneter Software. In: HAASIS, Klaus (Hrsg.): *Aktuelle Trends in der Softwareforschung*. Stuttgart : MFG-Stiftung Baden-Württemberg, 2007, S. 24–33
- [Schertl u. a. 2008] SCHERTL, A. ; LÖWEN, U. ; FAY, A. ; DRATH, R. ; GUTERMUTH, G. ; MÜHLHAUSE, M. ; EBEL, M.: Systematische Beurteilung und Verbesserung des Engineerings von automatisierten Anlagen. In: *Automation 2008 - Lösungen für die Zukunft*. Düsseldorf : VDI-Verl., 2008, S. 265–268
- [Schieferdecker 2008] SCHIEFERDECKER, I.: Testautomatisierung leicht gemacht. In: SIGS DATACOM GMBH (Hrsg.): *OBJEKTSpektrum Online-Ausgabe 2008 - Schwerpunkt Testing* Bd. 2008. Lindlastr. 2c, 53842 Troisdorf : SIGS DATACOM GmbH, 2008. – URL [http://www.sigs.de/publications/os/2008/testing/schieferdecker\\_OS\\_testing\\_08.pdf](http://www.sigs.de/publications/os/2008/testing/schieferdecker_OS_testing_08.pdf). – Zugriffsdatum: 08.10.2008
- [Schleipen 2008] SCHLEIPEN, M.: OPC UA supporting the automated engineering of production monitoring and control systems. In: *IEEE International Conference on Emerging Technologies and Factory Automation, 2008*. Piscataway, NJ : IEEE, 2008, S. 640–647
- [Schlögl 2007] SCHLÖGL, Wolfgang: *Automatisierungstechnik lässt die Digitale Fabrik real werden: Siemens-Fachkonferenz "Innovation@Siemens meets Research@TU Berlin", 10. Mai 2007, Berlin*. 2007. – URL [http://www.cki.tu-berlin.de/talentfolder/image/4\\_Schloegl.pdf](http://www.cki.tu-berlin.de/talentfolder/image/4_Schloegl.pdf). – Zugriffsdatum: 24.04.2009
- [Schmahls 2001] SCHMAHLS, Thomas: *Beitrag zur Effizienzsteigerung während Produktionsanläufen in der Automobilindustrie*. Chemnitz : IBF, 2001 (Wissenschaftliche Schriftenreihe des Instituts für Betriebswissenschaften und Fabriksysteme)
- [Schneider 2007] SCHNEIDER, Roland: Entwicklung einer virtuellen Werkzeugmaschinensteuerung. In: DENKENA, Berend (Hrsg.) ; BRECHER, Christian (Hrsg.): *Ramp-Up/2 - Anlaufoptimierung durch Einsatz virtueller Fertigungssysteme* Bd. 5. Frankfurt am Main : VDMA-Verl., 2007, S. 91–99

- [Schwirtlich 2008] SCHWIRTLICH, Ingo: Challenges of the Silicon PV Technology. In: *2nd SEMI Photovoltaic Fab Managers Forum in Dresden*. Brüssel : Semiconductor Equipment and Materials International, 2008
- [SEMI 2009] SEMI: *Semiconductor Equipment and Materials International – SEMI*. 2009. – URL <http://www.semi.org>. – Zugriffsdatum: 23.03.2009
- [SEMI DRAFT 4557 2008] SEMI: *SEMI Draft Document # 4557: Guide for PV Equipment Communication Interfaces (PVECI)*. 2008
- [SEMI E30 2007] SEMI: *SEMI E30-0307E: Generic Model for Communications and Control of Manufacturing Equipment (GEM)*. 2007
- [SEMI E37 2008] SEMI: *SEMI E37-0308: High-Speed SECS Message Services (HSMS) Generic Services*. 2008
- [SEMI E5 2007] SEMI: *SEMI E5-1107: SEMI Equipment Communications Standard 2 Message Content (SECS-II)*. 2007
- [SEMI E81 2000] SEMI: *SEMI E81-0600: Provisional Specification for CIM Framework Domain Architecture*. 2000
- [SEMI PV Group 2008] SEMI PV GROUP: *The Perfect Industry: The Race to Excellence in PV Manufacturing*. 2008. – URL [http://www.pvgroup.org/cms/groups/public/documents/web\\_content/ctr\\_026213.pdf](http://www.pvgroup.org/cms/groups/public/documents/web_content/ctr_026213.pdf). – Zugriffsdatum: 22.04.2009
- [SEMI PV2 2009] SEMI: *SEMI PV2-0709: Guide for PV Equipment Communication Interfaces (PVECI)*. 2009
- [SIGS DATACOM GmbH 2008] SIGS DATACOM GMBH (Hrsg.): *OBJEKTSpektrum Online-Ausgabe 2008 - Schwerpunkt Testing*. Bd. 2008. Lindlaustr. 2c, 53842 Troisdorf : SIGS DATACOM GmbH, 2008
- [Storr und Jost 1997] STORR, A. ; JOST, T.: Anforderungen an ein Simulationssystem zum Test von Leitsteuerungssoftware. In: *ASIM, Simulationstechnik, Symposium im Auftrag der Arbeitsgemeinschaft Simulation, 11*. Wiesbaden : Vieweg, 1997 (Fort-schritte in der Simulationstechnik), S. 442–448
- [Strassacker 1997] STRASSACKER, Dirk: *Testumgebung für die Implementierung und Inbetriebnahme eines adaptierbaren Leitsteuerungssystems*. Stuttgart, Universität Stuttgart, Dissertation, 1997
- [Stumpp 2008] STUMPP, Barbara: Schneller auf Sollzahl: Virtuelle Inbetriebnahme in der Automobilindustrie verkürzt und verbilligt das Ramp-up. In: *A&D - Das Fachmagazin für Industrielle Automation* (2008), Nr. 9, S. 32–35
- [Sturm 2006] STURM, Roland: *Modellbasiertes Verfahren zur Online-Leistungsbewertung von automatisierten Transportsystemen in der Halbleiterfertigung*. Stuttgart, Universität Stuttgart, Dissertation, 2006
- [Sun 2002] SUN MICROSYSTEMS: *Java Message Service Specification*. 2002

- [TestingTechnologies 2008] TESTING TECHNOLOGIES IST GMBH: *Testing Technologies*. 2008. – URL <http://www.testingtech.com/>. – Zugriffsdatum: 13.05.2008
- [V-Modell-XT 2009] DER BEAUFTRAGTE DER BUNDESREGIERUNG FÜR INFORMATIONSTECHNIK (Hrsg.): *Das V-Modell XT*. 2009. – URL <http://www.v-modell-xt.de/>. – Zugriffsdatum: 24.06.2009
- [VDI 2009] VDI: *VDI-Kompetenzfeld Informationstechnik. Fachausschuss 2.5.1: Manufacturing Execution Systems (MES)*. 2009. – URL <http://www.vdi.de/4584.0.html>. – Zugriffsdatum: 24.03.2009
- [VDI 3633 1996] VDI: *VDI 3633 Entwurf 11.1996: Simulation von Logistik-, Materialfluß- und Produktionssystemen: Begriffsdefinitionen*. 1996
- [VDI 3633 1997] VDI: *VDI 3633 Blatt 3: Simulation von Logistik-, Materialfluß- und Produktionssystemen. Experimentplanung und -auswertung*. 1997
- [VDI 3633 2000] VDI: *VDI 3633 Blatt 1 Entwurf 03.2000: Simulation von Logistik-, Materialfluß- und Produktionssystemen - Grundlagen*. 2000
- [VDI 3633 2007] VDI: *VDI 3633 Blatt 8: Simulation von Logistik-, Materialfluss- und Produktionssystemen - Maschinennahe Simulation*. 2007
- [VDI 4499 2008] VDI: *VDI 4499 Blatt 1: Digitale Fabrik - Grundlagen*. 2008
- [VDI 5600 2007] VDI: *VDI 5600, Blatt 1: Fertigungsmanagementsysteme. Manufacturing Execution Systems (MES)*. 2007
- [Verl und Fritsch 2008] VERL, Alexander ; FRITSCH, Dennis: Steuerungsentwicklung mit Simulationssoftware: Entwicklung von Steuerungen für automatisierte Materialflusssysteme. In: *ut Werkstattstechnik online* 98 (2008), Nr. 5, S. 370–376
- [Verl u. a. 2009] VERL, Alexander ; MÜLLER, Verena ; HAUBELT, Andreas: Baukastenbasiertes simulationsgestütztes Engineering. In: *A&D Kompendium 2009/2010*. Publish-Industry Verlag, München, 2009, S. 32–34
- [Versteegen u. a. 2006] VERSTEEGEN, Gerhard (Hrsg.) ; ILLES, Timea (Hrsg.) ; POHLMANN, Horst (Hrsg.) ; ROSSNER, Thomas (Hrsg.) ; U.A. (Hrsg.): *iX Studie Software-Testmanagement: Planung, Design, Durchführung und Auswertung von Tests - Methodenübersicht und Analyse unterstützender Werkzeuge*. Hannover : Heise, 2006
- [Wellington und Kubischta 2003] WELLINGTON, R. J. ; KUBISCHTA, M. D.: Wireless network emulation for distributed processing systems. In: *MILCOM 2003. 2003 IEEE Military Communications Conference, Boston, MA, USA, 13-16 Oct. 2003* (2003)
- [Westkämper 2006] WESTKÄMPER, Engelbert: Digitale Produktion - Herausforderung und Nutzen. In: HEISEL, Uwe (Hrsg.): *FtK2006 - Fertigungstechnisches Kolloquium*. Stuttgart : Gesellschaft für Fertigungstechnik, 2006, S. 469–490
- [Wiechers u. a. 2004] WIECHERS, O. ; ZEHTABAN, M. ; SEIDELMANN, J.: Using simulation as test tool in semiconductor industries: Experiences from the Future. In: MERTINS, Kai (Hrsg.): *Experiences from the future*. Stuttgart : Fraunhofer-IRB-Verl., 2004, S. 197–206

- [Wiendahl u. a. 2007] WIENDAHL, Hans-Hermann ; MUSSBACH-WINTER, Ute ; KIPP, Rolf: *MES - Fertigungssteuerung 2007/2008: Marktspiegel Business Software*. Stuttgart & Aachen : Fraunhofer-Institut für Produktionstechnik und Automatisierung IPA, 2007
- [Williams 1989] WILLIAMS, T. J. (Hrsg.): *A Reference Model for Computer Integrated Manufacturing (CIM): A Description from the Viewpoint of Industrial Automation*. Research Triangle Park, NC, USA : Instrument Society of America, 1989
- [Wissen.Schafft.Arbeit 2008] TECHNISCHE UNIVERSITÄT CHEMNITZ: *wissen.schafft.arbeit* — *Finalisten 2008: Syn@pse MES*. 2008. — URL [http://www.wissenschafttarbeit.de/finalisten\\_08.htm](http://www.wissenschafttarbeit.de/finalisten_08.htm). — Zugriffsdatum: 13.08.2009
- [Wolf und Roock 2008] WOLF, Henning ; ROOCK, Arne: Agilität wird Mainstream: Ergebnisse der Online-Umfrage 2008. In: *OBJEKTSpektrum* (2008), Nr. 3, S. 10–13
- [Zäh und Wünsch 2005] ZÄH, Michael F. ; WÜNSCH, Georg: Schnelle Inbetriebnahme von Produktionssystemen: Qualitätssicherung von automatisierten Maschinen durch Simulation. In: *Wt Werkstatttechnik* 95 (2005), Nr. 9, S. 699–704
- [Zander-Nowicka u. a. 2008] ZANDER-NOWICKA, J. ; BAUER, T. ; ESCHBACH, R. ; RENNOCH A.: Modellbasierte Testentwicklung in der industriellen Anwendung. In: SIGS DATACOM GMBH (Hrsg.): *OBJEKTSpektrum Online-Ausgabe 2008 - Schwerpunkt Testing* Bd. 2008. Lindlastr. 2c, 53842 Troisdorf : SIGS DATACOM GmbH, 2008. — URL [http://www.sigs.de/publications/os/2008/testing/zander\\_08\\_testing\\_08.pdf](http://www.sigs.de/publications/os/2008/testing/zander_08_testing_08.pdf). — Zugriffsdatum: 08.10.2008
- [Zimmermann und Mönch 2004] ZIMMERMANN, Jens ; MÖNCH, Lars: Kopplungsarchitektur für ein .NET-basiertes Multi-Agentensystem und einen diskreten ereignisorientierten Simulator: Experiences from the Future. In: MERTINS, Kai (Hrsg.): *Experiences from the future*. Stuttgart : Fraunhofer-IRB-Verl., 2004, S. 443–450