

# Adaptive und wandlungsfähige IT-Architektur für Produktionsunternehmen

Von der Graduiertenschule GSaME Graduate School of Excellence  
advanced Manufacturing Engineering der Universität Stuttgart  
zur Erlangung der Würde eines Doktor-Ingenieurs (Dr.-Ing.)  
genehmigte Abhandlung

Vorgelegt von

Stefan Silcher

aus Waiblingen

**Hauptberichter:** Prof. Dr.-Ing. habil. Bernhard Mitschang

**Mitberichter:** Prof. Dr.-Ing. Norbert Ritter

**Tag der mündlichen Prüfung:** 2. April 2014

Institut für Parallele und Verteilte Systeme (IPVS),  
Abteilung Anwendersoftware  
der Universität Stuttgart

2014



---

# Vorwort

---

Diese Arbeit entstand an der Graduate School of Excellence advanced Manufacturing Engineering (GSaME) in Zusammenarbeit mit der Abteilung Anwendersoftware des Institutes für Parallele und Verteilte Systeme (IPVS). Die Graduiertenschule GSaME forscht in einem interdisziplinären Umfeld an der Weiterentwicklung des advanced Manufacturing Engineering, um Produktionsunternehmen für zukünftige Herausforderungen zu wappnen.

Dem Doktorvater meiner Arbeit Prof. Dr.-Ing. Bernhard Mitschang gilt mein besonderer Dank, zum einen für die Möglichkeit diese Forschungsarbeit im Rahmen der Graduiertenschule durchführen zu dürfen und zum anderen für seine intensive, konstruktive und lehrreiche Betreuung während der vergangenen Jahre.

Außerdem möchte ich Prof. Dr.-Ing. Norbert Ritter für seine Bereitschaft zur Übernahme des Mitberichts danken.

Den aktuellen und ehemaligen Kolleginnen und Kollegen der Abteilung Anwendersoftware danke ich für die spannende Zeit mit vielen kritischen aber auch anregenden Diskussionen über meine Dissertation. Nebenbei war die Zusammenarbeit sehr unterhaltsam und freundschaftlich und hat die Zeit schnell vergehen lassen. Ein weiterer Dank gilt allen Kolleginnen und Kollegen der GSaME für die hilfreichen Gespräche über die Forschungsarbeit und gemeinsamen Projekte, aber auch für die unterhaltsamen Zeiten nach getaner Arbeit, die das Leben in der Graduiertenschule abrunden.

Der Administration der GSaME und der Abteilung Infrastruktur des IPVS danke ich für ihre Unterstützung in organisatorischer und technischer Hinsicht, die mich bei Problemen und Anfragen nie im Stich gelassen haben.

Ebenso möchte ich allen Studierenden danken, die zum Gelingen dieser Arbeit durch die Mitarbeit an der Implementierung des Prototypen beigetragen haben. Dies gilt auch für die Kollegin und Kollegen des Institutes für Industrielle Fertigung und Fabrikbetrieb (IFF) und der Lernfabrik, die mir ermöglichten den Prototypen in dieser Umgebung umzusetzen und zu evaluieren.

Meinen Freunden danke ich für die unterhaltsame Zeit außerhalb der Universität, die ich unter keinen Umständen missen möchte. Nicht zuletzt möchte ich mich bei meiner Familie und meiner Freundin bedanken, die mich in meiner Arbeit bestärkt und mir Ihre Unterstützung, Ihr Verständnis und Ihre Geduld in den vergangenen Jahren entgegengebracht haben.

Weinstadt, im November 2013

Stefan Silcher

---

# Inhaltsverzeichnis

---

<b>Abkürzungsverzeichnis</b>	<b>11</b>
<b>Kurzfassung</b>	<b>17</b>
<b>Abstract</b>	<b>23</b>
<b>1 Einleitung</b>	<b>27</b>
1.1 Motivation . . . . .	28
1.1.1 Turbulentes Umfeld . . . . .	30
1.1.2 Globalisierung . . . . .	31
1.1.3 Wachsende Komplexität . . . . .	32
1.2 Problemstellungen . . . . .	33
1.3 Vision, Ziele und Forschungsfragen . . . . .	36
1.3.1 Vision . . . . .	36
1.3.2 Ziele der Arbeit . . . . .	38
1.3.3 Forschungsobjekte . . . . .	39
1.4 Aufbau der Arbeit . . . . .	42
<b>2 Grundlagen</b>	<b>45</b>
2.1 Managementkonzepte für Produktionsunternehmen . . . . .	45
2.1.1 Computer-integrated Manufacturing . . . . .	48
2.1.2 Produktdatenmanagement . . . . .	51
2.1.3 Digitale Fabrik . . . . .	52
2.1.4 Product Lifecycle Management . . . . .	53

2.1.5	Factory Lifecycle Management . . . . .	55
2.1.6	Supply Chain Management . . . . .	57
2.1.7	Das Stuttgarter Unternehmensmodell . . . . .	58
2.2	Informationstechnische Grundlagen . . . . .	60
2.2.1	Enterprise Application Integration . . . . .	61
2.2.2	Serviceorientierte Architektur . . . . .	62
2.2.2.1	Web Service . . . . .	64
2.2.2.2	WSDL . . . . .	66
2.2.2.3	SOAP . . . . .	66
2.2.2.4	Serviceverzeichnis . . . . .	67
2.2.2.5	Modellierungssprachen für Geschäftsprozesse . . . . .	69
2.2.3	Nachrichtenvermittlung . . . . .	70
2.2.3.1	Einheitliches Nachrichtenformat . . . . .	70
2.2.3.2	Content-based Router . . . . .	71
2.2.3.3	Message Queues . . . . .	71
2.2.4	Enterprise Service Bus . . . . .	72
2.3	Zusammenfassung . . . . .	73
<b>3</b>	<b>IT-Lösungen für den Produktlebenszyklus und verwandte Arbeiten</b>	<b>77</b>
3.1	Phasenmodell des erweiterten Produktlebenszyklus . . . . .	78
3.2	Serviceorientierte Architektur für das Product Lifecycle Management . . . . .	80
3.2.1	PLM Services 2.0 . . . . .	80
3.2.2	Enterprise Service Bus zur Integration des Product Lifecycle Management . . . . .	83
3.3	CHAMPAGNE . . . . .	85
3.4	Engineering Service Bus (EngBus) . . . . .	88
3.5	Manufacturing Service Bus (MSB) . . . . .	89
3.6	Softwarelösungen für PLM, FLM und SCM . . . . .	92

---

3.7	Zusammenfassung . . . . .	99
<b>4</b>	<b>Serviceorientierte IT-Architektur für wandlungsfähige Produktionsunternehmen</b>	<b>103</b>
4.1	Adaptivität und Abhängigkeiten von Unternehmenselementen	104
4.2	Anforderungen an eine adaptive und wandlungsfähige IT-Architektur . . . . .	107
4.2.1	Technische Anforderungen . . . . .	111
4.2.2	Organisatorische Anforderungen . . . . .	114
4.3	Adaptive und wandlungsfähige IT-Architektur für Produktionsunternehmen . . . . .	118
4.3.1	Eine adaptive und wandlungsfähige IT-Architektur (ACITA) . . . . .	118
4.3.2	Umsetzung der ACITA zur Integration des Produktlebenszyklus . . . . .	121
4.3.2.1	Datenaustauschformate . . . . .	121
4.3.2.2	Serviceverzeichnisse . . . . .	122
4.3.2.3	Dynamische Nachrichtenvermittlung . . . . .	133
4.3.2.4	Prozessunterstützung . . . . .	137
4.4	Ganzheitliches Managementmodell für Produktionsunternehmen . . . . .	137
4.4.1	Lebenszyklen und Managementmodelle . . . . .	138
4.4.1.1	Produktlebenszyklus . . . . .	138
4.4.1.2	Fabriklebenszyklus . . . . .	140
4.4.1.3	Lieferkette . . . . .	142
4.4.2	Ganzheitliches Managementmodell . . . . .	142
4.5	Zusammenfassung . . . . .	146
<b>5</b>	<b>Prototypische Implementierung der ACITA im Produktlebenszyklus</b>	<b>149</b>
5.1	Integrationsumgebung Lernfabrik . . . . .	149
5.1.1	Die digitale Lernumgebung . . . . .	151

5.1.2	Die physische Fabrik . . . . .	152
5.1.3	Integration der digitalen Werkzeuge und der physischen Fabrik . . . . .	154
5.2	Der Production-planning Service Bus (PPSB) . . . . .	157
5.3	Implementierung der serviceorientierten ACITA für das PLM	161
5.3.1	Implementierung der ACITA für den Produktlebenszyklus . . . . .	162
5.3.2	Implementierung der Serviceverzeichnisinfrastruktur	166
5.3.3	Dynamisches Routing in der ACITA . . . . .	167
5.3.4	Globales Nachrichtendatenformat . . . . .	167
5.3.5	Übersetzungsservices . . . . .	169
5.4	Zusammenfassung . . . . .	170
<b>6</b>	<b>Anwendung und Evaluation der ACITA</b>	<b>173</b>
6.1	Anwendungsszenarien . . . . .	173
6.1.1	Anwendungsszenario 1: Serviceorientierte Integration der Produktionsplanung . . . . .	174
6.1.2	Anwendungsszenario 2: Phasenübergreifende Integration von Produktionsplanung und Produktion . .	177
6.1.3	Anwendungsszenario 3: Wandlungsfähigkeit der ACITA durch Hinzufügen oder Entfernen von phasenspezifischen Integrationsumgebungen . . . .	179
6.1.4	Anwendungsszenario 4: Integration von externen Partnern in die ACITA . . . . .	182
6.2	Bewertung der ACITA gegenüber anderen Lösungen . . . .	184
6.2.1	Bewertungskriterien . . . . .	184
6.2.2	Vergleich mit anderen Systemen . . . . .	189
6.3	Erweiterung der ACITA auf andere Forschungsgebiete . . .	192
6.3.1	Integration von mobilen Endgeräten in die Unternehmensinfrastruktur . . . . .	193
6.3.2	SOA für die Maschineninbetriebnahme . . . . .	196



---

6.4	Evaluierung der ACITA . . . . .	197
6.4.1	Erfüllung der Forschungsobjekte . . . . .	200
6.5	Zusammenfassung . . . . .	202
<b>7</b>	<b>Fazit und Ausblick</b>	<b>205</b>
7.1	Zusammenfassung und Diskussion . . . . .	205
7.2	Ausblick . . . . .	209
7.2.1	Einsatz von mobilen Endgeräten in Produktionsun- ternehmen . . . . .	210
7.2.2	Analyse von Planungs- und Produktionsdaten . . . . .	211
7.2.3	Umsetzung der ACITA in der Cloud . . . . .	212
<b>A</b>	<b>Codebeispiele</b>	<b>213</b>
A.1	XML-Schema des einheitlichen Nachrichtenformates der Pro- duktionsplanung . . . . .	213
A.2	XML-Schema des global einheitlichen Nachrichtenformates .	216
	<b>Abbildungsverzeichnis</b>	<b>221</b>
	<b>Tabellenverzeichnis</b>	<b>223</b>
	<b>Literaturverzeichnis</b>	<b>225</b>



---

# Abkürzungsverzeichnis

---

- ACITA** adaptable and changeable IT-Architecture. 18–20, 24–26, 107, 118–122, 124–128, 130, 133, 144–147, 149, 157, 158, 161, 162, 166, 170, 173, 174, 177, 179, 182–184, 189–192, 195, 197–205, 207–212, 216, 221–223
- aIE** advanced Industrial Engineering. 20, 26, 56, 149, 150, 155, 158, 161, 173, 177, 190
- AWF** Ausschuss für wirtschaftliche Fertigung. 49
- BPEL** Business Process Execution Language. 65, 66, 69, 73, 75, 85, 89, 90, 108, 137, 147
- BPMN** Business Process Model and Notation. 69, 75, 137, 178, 222
- CAD** Computer-aided Design. 46–48, 50, 51, 93, 112, 182
- CAM** Computer-aided Manufacturing. 46–48, 50
- CAP** Computer-aided Planning. 47, 50
- CAQ** Computer-aided Quality Assurance. 47, 50
- CAx** Computer-aided Everything. 46, 47, 49, 73
- CBR** Content-based Router. 19, 25, 71, 73, 75, 90, 91, 108, 109, 119, 124, 132–136, 146, 160, 163, 165–167, 169, 171, 174, 176, 180, 182, 195, 200, 203, 207, 208, 212
- CEC** Common Engineering Client. 88

- CIM** Computer-integrated Manufacturing. 46, 48–51, 73, 197, 221
- CRM** Customer Relationship Management. 96, 97
- DAF** Datenaustauschformat. 163, 164, 166, 179, 180, 182
- DDSB** Design & Development Service Bus. 179, 182
- DMU** Digital Mock-up. 47
- DPE** Delmia Process Engineer. 85, 95, 96, 100, 151, 154–157, 159, 161, 175, 176, 191, 198, 213
- EAI** Enterprise Application Integration. 61, 62, 64, 73, 74, 189
- ECAD** Electronic Computer-aided Design. 94
- ECC** ERP Central Component. 96, 97
- EDV** Elektronische Datenverarbeitung. 49
- EngBus** Engineering Service Bus. 20, 26, 78, 88, 100, 190, 191
- ERP** Enterprise Resource Planning. 57, 58, 96–98, 119, 200
- ESB** Enterprise Service Bus. 18, 19, 24, 25, 37, 38, 40, 45, 61, 64, 72, 73, 75, 77, 78, 80, 83–85, 89, 90, 92, 99, 100, 103, 104, 108–114, 116–122, 124, 133, 134, 137, 144–147, 149, 157–159, 161, 162, 167, 187, 189, 197, 199–201, 207, 208, 212, 221
- FEM** Finite Element Methode. 81
- FIFO** First In - First Out. 71
- FLM** Factory Lifecycle Management. 46, 48, 55, 56, 74, 78, 94, 99, 100, 103, 137, 142, 147
- GPS** Global Positioning System. 195

- 
- HTTP** Hypertext Transfer Protocol. 65, 67
- IE** Industrial Engineering. 150
- IFF** Institut für Industrielle Fertigung und Fabrikbetrieb. 149
- IKT** Informations- und Kommunikationstechnologie. 33
- ISO** International Organization for Standardization. 82
- IT** Informationstechnik. 17, 18, 21, 23, 24, 26–29, 31–34, 36–42, 44, 58, 67, 70–72, 74, 75, 77, 79, 88, 89, 98, 100, 103–108, 111, 114–118, 130, 138, 143, 144, 146, 150, 158, 170, 171, 176, 179, 181, 186, 188, 190, 194, 195, 197, 199, 201–203, 206–209, 222
- iTRAME** intelligent Transformable Assembly and Manufacturing Equipment. 152–155, 161, 204, 222
- JMS** Java Message Service. 65
- JT** Jupiter. 169
- KMU** kleine und mittlere Unternehmen. 36, 41, 97, 98, 114, 201
- MCAD** Mechanical Computer-aided Design. 94
- MES** Manufacturing Execution System. 92, 154, 155
- MQ** Message Queue. 71, 72, 159
- MSB** Manufacturing Service Bus. 20, 26, 78, 89–92, 100, 108, 118, 149, 157, 159–161, 164, 167, 168, 170, 174, 177, 184, 190, 191, 195, 197, 200, 207, 221
- MSMQ** Microsoft Message Queue. 159, 176
- OASIS** Organization for the Advancement of Structured Information Standards. 65, 68

- OEM** Original Equipment Manufacturer. 142
- OMG** Object Management Group. 82, 99
- PC** Personal Computer. 153, 193, 194
- PDM** Product Data Management. 47, 48, 51, 53, 56, 74, 84, 93, 94, 179, 182
- PLM** Product Lifecycle Management. 18–20, 24–26, 34, 46, 48, 53–56, 74, 77–80, 82–85, 93–100, 103, 110, 114, 118–124, 127, 129, 131, 134, 135, 137, 138, 142, 146, 147, 149, 157, 161, 163, 164, 167, 169, 170, 173, 177, 179, 180, 182, 190–192, 198–201, 203, 207, 208, 213, 216, 222
- PPR** Produkt, Prozess und Ressourcen. 48, 85, 87, 154–156, 159, 160, 173, 175
- PPS** Produktionsplanung und -steuerung. 47, 49, 50, 73
- PPSB** Production-planning Service Bus. 20, 26, 149, 157–161, 163, 164, 167, 170, 173, 174, 176, 200, 201, 203, 213
- RPC** Remote Procedure Call. 61
- SCC** Supply Chain Council. 57
- SCM** Supply Chain Management. 46, 48, 57, 74, 78, 96, 97, 99, 100, 104, 137, 139, 140, 142, 143, 147
- SCOR** Supply Chain Operations Reference. 57
- SFB** Sonderforschungsbereich. 85, 152, 154
- SMTP** Simple Mail Transfer Protocol. 67
- SOA** Serviceorientierte Architektur. 37–39, 61–64, 66, 67, 73–75, 77, 78, 83, 85, 88, 89, 94, 98, 108, 111, 115, 116, 120, 123, 133, 146, 170, 176, 190, 193, 197, 221
- SQL** Structured Query Language. 153, 159

**STEP** Standard for the Exchange of Product Model Data. 82, 168, 169

**SUM** Stuttgarter Unternehmensmodell. 48, 58, 74, 150, 206

**TCP** Transmission Control Protocol. 67

**TOSCA** Topology and Orchestration Specification for Cloud Applications.  
212

**UBR** UDDI Business Registry. 68, 69

**UDDI** Universal Description, Discovery and Integration. 66, 68, 75, 124, 127

**VBA** Visual Basic for Applications. 85, 154, 156, 176, 191

**VDI** Verein Deutscher Ingenieure. 52

**W3C** World Wide Web Consortium. 64, 65

**WinCC** Windows Control Center. 153

**WS** Web Service. 64–66

**WSDL** Web Services Description Language. 64–66, 75, 82

**XML** Extensible Markup Language. 64–67, 69, 86, 87, 134, 135, 161, 168, 169,  
195, 216

**XPDL** XML Propagation Definition Language. 86

**XSL** Extensible Stylesheet Language. 86

**XSLT** XSL Transformation. 86





---

## Kurzfassung

---

Die Herausforderungen, denen sich Produktionsunternehmen heutzutage stellen müssen, nehmen kontinuierlich zu. Diese umfassen insbesondere die Globalisierung, die wachsende Komplexität und das heute vorherrschende turbulente Umfeld [JWW09]. Durch die Globalisierung muss sich jedes Unternehmen dem Wettbewerb und den vielfältigen Herausforderungen der unterschiedlichen Märkte stellen. Die zunehmende Komplexität wird nicht nur durch eine steigende Anzahl an Produktvarianten hervorgerufen, sondern nimmt auch auf der Prozessebene kontinuierlich zu. Die Probleme vergrößern sich durch das turbulente Umfeld, in dem interne und externe Einflüsse auf die Produktionsunternehmen einwirken, wie z. B. Maschinenausfälle, Bestellungsschwankungen oder sich ändernde gesetzliche Regelungen, die zu einem kontinuierlichen Anpassungsbedarf führen [WZ09].

Zur Unterstützung der Mitarbeitenden und zur effizienteren Verwaltung und Beherrschung der Komplexität kommt in Produktionsunternehmen zunehmend Informationstechnik (IT) zum Einsatz. Die Vielzahl an Softwaresystemen und deren oft proprietäre Integration führen jedoch schnell zu einer komplexen IT-Landschaft, deren Wartungsaufwand kontinuierlich steigt. Zusätzlich sind sowohl die Softwareanwendungen als auch deren Integration unflexibel [KWW03], weshalb Änderungen und Erweiterungen nur mit großem Aufwand durchführbar sind. Die in den Anwendungen implementierten Prozesse werden damit ebenfalls starr und können aufgrund dessen nicht schnell genug an die sich häufig ändernden Anforderungen angepasst werden. Zudem sind Integrationslösungen weitestgehend auf eine Domäne

beschränkt und ermöglichen keinen unternehmensweiten Datenaustausch oder domänen- und anwendungsübergreifende Prozessdefinitionen.

Aus diesen Gründen wird eine neue IT-Architektur für Produktionsunternehmen benötigt, welche die Adaptivität sowohl der Anwendungen und deren Integration als auch der Prozesse unterstützt. Die vorliegende Arbeit beschreibt eine solche adaptive und wandlungsfähige IT-Architektur (ACITA) für Produktionsunternehmen [SMM11]. Deren initiale Anwendungsdomäne ist der Produktlebenszyklus bzw. das Product Lifecycle Management (PLM), sie kann jedoch relativ einfach auf weitere Domänen ausgeweitet werden. Zur Integration der Anwendungen werden einheitliche Serviceschnittstellen verwendet, die in der standardisierten Web Service Technologie umgesetzt sind. Die lose und damit flexible Kopplung der Services erfolgt über einen angepassten Enterprise Service Bus (ESB). Die Unterstützung der Prozesse geschieht durch die flexible Komposition der Services in Workflows, die einzelne Geschäftsservices oder ganze Geschäftsprozesse unterstützen können. Jede Domäne wird durch dieses Vorgehen getrennt voneinander über einen angepassten ESB integriert. Dies erlaubt die technischen Anforderungen der Domäne zu berücksichtigen, wie z. B. Datenmenge, Datenstruktur und Latenz, die ein ESB verarbeiten muss, wodurch eine leistungsfähigere IT-Umgebung erreicht wird. Die Integration der einzelnen domänenspezifischen ESBs erfolgt über einen weiteren ESB, dem sogenannten PLM-Bus. Dieser sorgt für eine wandlungsfähige IT-Architektur, indem phasenspezifische ESBs einfach hinzugefügt oder entfernt werden können.

Die Umsetzung der ACITA erfordert eine Reihe verschiedener Komponenten. Die zu integrierenden Anwendungen benötigen Serviceschnittstellen, um auf deren Funktionalität oder Daten zugreifen zu können. Die Verwaltung dieser Schnittstellen bzw. Services wird durch mehrere Serviceverzeichnisse bewerkstelligt, deren Anordnung der Hierarchie der ACITA entsprechen [SKRM13]. Innerhalb einer Domäne werden lokale Serviceverzeichnisse eingesetzt, die Serviceinformationen der entsprechenden Phase speichern. Auf der über-

geordneten Ebene des PLM-Busses wird ein Meta-Verzeichnis verwendet, um Serviceinformationen für den phasenübergreifenden Nachrichtenaustausch bereitzustellen. Das Meta-Verzeichnis speichert nur die notwendigsten Serviceinformationen und gewährleistet damit die Vertraulichkeit, z. B. bei Einbindung von Services externer Unternehmen.

Die lose Kopplung der Services erfolgt über einen Content-based Router (CBR), der anhand von Regeln dynamisch die Ziele jeder Nachricht bestimmt und anschließend deren Serviceinformationen über das entsprechende lokale Serviceverzeichnis abfragt. Hierfür wird in jedem ESB ein CBR implementiert, der eine Nachricht an die lokale Anwendung bzw. deren Serviceschnittstelle zustellt oder an den CBR des PLM-Busses sendet. Dieser leitet die Nachricht über den entsprechenden CBR der Zielphase an den Zielservice weiter.

Die Unabhängigkeit von proprietären Datenformaten der integrierten Anwendungen wird durch einheitliche Nachrichtenaustauschformate sichergestellt. In jeder Phase und dem entsprechenden ESB wird ein solches einheitliches Nachrichtenformat umgesetzt, das alle Daten modelliert, die innerhalb dieser Phase zwischen Anwendungen ausgetauscht werden. Der phasenübergreifende Nachrichtenaustausch erfolgt über ein globales Nachrichtenformat, das wiederum nur Daten beschreibt, deren Austausch zwischen den Phasen geschieht. Durch die Aufteilung in mehrere einheitliche Datenformate reduziert sich die Komplexität der einzelnen Datenmodelle, wodurch die Wartbarkeit und Erweiterbarkeit vereinfacht wird.

Um eine Nachricht zwischen unterschiedlichen Phasen zu übertragen, sind Übersetzungsservices notwendig, die jede Nachricht vom lokalen in das globale Nachrichtenformat übersetzen. Anschließend werden diese vom globalen in das lokale Nachrichtenformat der Zielphase transformiert.

Der modulare Aufbau der ACITA erlaubt, neben der Integration der Phasen des Produktlebenszyklus, eine einfachere Erweiterung um weitere Phasen von Produktionsunternehmen. Ein ganzheitliches Managementmodell be-

schreibt ein solches Phasenmodell, das neben dem Produktlebenszyklus auch den Fabriklebenszyklus und die Lieferantenkette enthält [SSZM13]. Darüber hinaus ist die Einbindung von externen Integrationsumgebungen nach dem gleichen Muster möglich, wodurch der Daten- und Informationsaustausch mit Partnern verbessert wird. Zudem ermöglicht dies eine durchgehende Prozessunterstützung zwischen den kooperierenden Parteien.

Die prototypische Implementierung der ACITA erfolgte in der Lernfabrik aIE, die aus einer digitalen Lerninsel und einer physischen Modellfabrik besteht [RKK<sup>+</sup>07]. Zur Integration wurden die beiden domänenspezifischen Integrationsumgebungen des Production-planning Service Bus (PPSB) und des Manufacturing Service Bus (MSB) über den PLM-Bus verbunden, um den nahtlosen Datenaustausch zwischen den entsprechenden Phasen zu demonstrieren [SKRM13].

Die Evaluation der ACITA wird in vier Anwendungsszenarien durchgeführt. Die Vorteile dieser verbesserten Adaptivität und Wandlungsfähigkeit der Integrationsarchitektur und der flexibleren Prozessunterstützung werden hier aufgezeigt. Im Anschluss wird die ACITA anhand von sechs Kriterien mit Integrationslösungen für den Produktlebenszyklus verglichen, die in der Forschung entwickelt wurden oder als kommerzielle Softwarelösungen in der Industrie zum Einsatz kommen. Gegenüber den kommerziellen Lösungen zeigt die ACITA eine höhere Flexibilität auf. Im Vergleich zu Integrationsumgebungen wie dem Engineering Service Bus (EngBus) [KS11] oder Manufacturing Service Bus (MSB) [Min12] ist die bessere domänenübergreifende Integration der ACITA durch deren Modularität entscheidend.

Die ACITA kann in einem global aufgestellten Unternehmen durch die Integration von verteilten Anwendungen bzw. Services den Datenaustausch zwischen allen Standorten sicherstellen. Der durchgängige Einsatz von Softwaresystemen in Kombination mit der flexiblen Prozessunterstützung macht die wachsende Komplexität der Produkte und Prozesse besser beherrschbar. Der kontinuierliche Anpassungsbedarf, der durch das turbulente Umfeld

hervorgerufen wird, ist durch die Adaptivität und Wandlungsfähigkeit der IT-Architektur einfacher durchführbar. Damit sind Unternehmen bestens für zukünftige Herausforderungen gewappnet.



---

## Abstract

---

The challenges manufacturing companies are confronted with nowadays are increasing steadily. They include growing globalization, increasing complexity and prevailing turbulent environments [JWW09]. Due to globalization, every company has to face up to the competition as well as a diversity of markets in a distributed environment. The increasing complexity is not only caused by the rising number of product variants, but is also rising equally on the process level. The problems are amplified even more by the turbulent environment, which is characterized by internal and external influences, e. g., machine failures, order fluctuations, or changing legal regulations. Manufacturing companies thus need to continuously adapt to the current situation [WZ09].

In order to support employees and to make administration more efficient, more and more information technology (IT) is introduced in manufacturing companies. The multitude of software systems and their usually proprietary integration are quickly resulting in complex IT landscapes, which continuously increases the maintenance effort. Additionally, the software applications as well as their integration are inflexible [KWW03], therefore, changes and extensions can only be conducted with great effort. The processes implemented in the applications are inflexible as well, which is why changes that are necessary due to frequently changing requirements cannot be performed as fast as needed. Furthermore, integration solutions are typically limited to a single domain and do not provide enterprise-wide data exchange or cross-domain process definitions.

Because of the above-mentioned reasons, a new IT architecture, which supports the integration of applications and processes in an adaptable way, is necessary. This thesis describes such an adaptable and changeable IT-Architecture (ACITA) for manufacturing companies [SMM11]. Its initial application domain is the product life cycle, or Product Lifecycle Management (PLM), though it can easily be extended to support other domains. The integration of software applications is based on uniform service interfaces, which are implemented in the standardized web service technology. The loose coupling of services is enabled by an adapted Enterprise Service Bus (ESB). Processes are supported by a flexible composition of services into workflows that execute business services or business processes. Each domain is integrated separately by using such a phase-specific ESB. This way, the technical requirements of each phase, e. g., data throughput, data structure, and latency each ESB has to process, can be taken into account. Consequently, the resulting IT landscape is more efficient. The integration of the single domain-specific ESBs is accomplished with a further ESB, the so-called PLM-Bus. This PLM-Bus provides a changeable IT architecture by enabling the simple addition of phase-specific ESBs.

The implementation of the ACITA requires a set of several components. To access the functionality and data of the applications that are to be integrated, service interfaces are needed. The management of these service interfaces, or services, is accomplished by multiple service registries that follow the hierarchy of the ACITA [SKRM13]. Within each domain, a local service registry manages and stores the service information of the corresponding phase. The PLM-Bus uses a meta service registry to provide service information for cross-phase message exchange. This meta service registry only stores the relevant service information needed for service endpoint retrieval in remote phases. Confidentiality can therefore be guaranteed, e. g., concerning the integration of external services of partners.



A content-based router (CBR) dynamically determines the destinations of each message by using rules. Subsequently, the service information is retrieved from the corresponding local service registries for each destination service. Each phase-specific ESB is therefore associated with a CBR and with a local service registry. The CBR forwards all messages to the locally available applications or to the CBR of the PLM-Bus. This global CBR then sends the messages to the appropriate CBR of the destination phases where the message is finally delivered. Altogether, this leads to a loose coupling of services within the whole ACITA.

The dependencies between applications and their proprietary data formats can be decoupled by introducing common message exchange formats. Each phase is integrated separately by using its own common data format, which only contain data exchanged within the corresponding phase. For cross-domain message exchange, an additional global message format, which only contains data exchanges between the integrated phases, is defined. Compared to a single message format, this separation of common message exchange formats reduces the complexity for the whole ACITA, leading to a reduced effort for maintenance and extensions.

Different common message formats require translation services for cross-domain message exchange. Therefore, each cross-phase message is transformed from the local message format into the global message format and afterwards into the local message format of the destination phase.

The ACITA is initially intended to integrate the heterogeneous phases of the product life cycle in a homogeneous manner. Moreover, the high modularity of this architecture allows for an easy extension to integrate other domains within a production company or even external services of partners. Such a phase model is described in the holistic management model, which, aside from the product life cycle, also integrates the factory life cycle and the supply chain [SSZM13]. This holistic integration approach enables an improved data

and information exchange as well as a continuous process support between cooperation parties.

The prototypical implementation of the ACITA was conducted in the learning factory aIE, which contains a digital learning shell and a physical model factory [RKK<sup>+</sup>07]. The PLM-Bus integrates the domain-specific integration platforms Production-planning Service Bus (PPSB) and Manufacturing Service Bus (MSB) to demonstrate the seamless and flexible data exchange between the corresponding phases [SKRM13].

Four usage scenarios accompany the evaluation of the ACITA and show the benefits of an improved adaptivity and changeability of this integration architecture as well as the higher flexibility of process support. Subsequently, a comparison of the ACITA with other integration approaches in research as well as with commercial software is presented. In comparison to the commercial software, the ACITA allows for a higher flexibility in integration and process support. Integration platforms such as the Engineering Service Bus (EngBus) [KS11] or Manufacturing Service Bus (MSB) [Min12] are mainly restricted to a single domain and do not provide a cross-domain integration such as the one enabled by the modularity of the ACITA.

In conclusion, the ACITA can provide a vital integration of distributed applications and services of all local sites of a globally acting company. The consistent use of software systems combined with flexible process support makes the growing complexity of products and processes more manageable. The necessity to continuously adapt to new situations caused by the turbulent environment can be fulfilled easier through the adaptivity and changeability of the IT architecture. As a result of this, manufacturing companies are well-prepared for challenges in the future.

---

# KAPITEL 1

## Einleitung

---

Produktionsunternehmen stehen heute im globalen Wettbewerb, in dem ein effizienter Ressourceneinsatz in allen Bereichen erforderlich ist. Neben dem optimalen Einsatz der Mitarbeitenden und der maximalen Auslastung der Maschinen steht immer mehr die Unterstützung durch Informationstechnik (IT) im Vordergrund. Dabei ist nicht nur die Unterstützung von Aufgaben oder deren automatische Ausführung wichtig, auch die Wartung und Weiterentwicklung der gesamten IT-Infrastruktur muss mit relativ geringem Aufwand möglich sein.

Die Motivation dieser Arbeit zeigt die Herausforderungen von Produktionsunternehmen in Bezug auf deren turbulentes Umfeld, der Globalisierung und der kontinuierlich wachsenden Komplexität auf. Die Unterstützung durch Softwaresysteme und die daraus resultierenden Möglichkeiten werden im folgenden Abschnitt beschrieben. Die Problemstellungen dieses breiten Softwareeinsatzes wird in Abschnitt 1.2 dargelegt, indem die fehlende Flexibilität in Bezug auf die Integration und Prozessunterstützung aufgezeigt wird. Abschnitt 1.3 beschreibt die Vision und Ziele der Arbeit, um eine verbesserte Integration der IT-Landschaft von Produktionsunternehmen zu erreichen, die eine höhere Flexibilität und eine adaptive Prozessunterstützung bietet. Aus diesen Zielen werden anschließend die Forschungsobjekte der Arbeit

abgeleitet. Das Kapitel schließt, indem der Aufbau der Arbeit in Abschnitt 1.4 beschrieben wird.

## 1.1 Motivation

Unternehmen stehen heutzutage vor großen Herausforderungen, um am Markt bestehen zu können. Aufgrund des ständigen Wandels mit verschiedensten internen und externen Einflüssen müssen sich die Unternehmen mit ihrem Produktportfolio kontinuierlich anpassen. Dieses als turbulent bezeichnete Umfeld führt dazu, dass nicht nur die Produkte, sondern auch die Fabriken in immer kürzer werdenden Abständen anzupassen sind, um einen möglichst maximalen Ertrag bei gleichzeitig niedrigen Kosten zu erzielen. Zusätzlich kommt der Wandel zu einer nachhaltigen Produktion hinzu, der größere Anpassungen in den Unternehmen erfordert [JWW09]. Diese schnellen Anpassungen sind insbesondere durch die digitale Unterstützung der Produktionsunternehmen möglich, die einen wesentlichen Treiber der Wandlungsfähigkeit darstellen, um den zunehmenden Anpassungsbedarf zu unterstützen. Deshalb nimmt in immer mehr Bereichen und über das ganze Unternehmen verteilt der Einsatz von Softwareanwendungen zu, wie z. B. in der Produktentwicklung, der Digitalen Fabrik und der Produktion selbst.

Die Verwendung von IT-Systemen hat nicht nur Vorteile, denn über die Zeit gewachsene Softwarelandschaften können vielfältige Probleme durch die steigende Komplexität mit sich bringen, wie z. B. hohe Wartungskosten, hoher Aufwand für Anpassungen und Erweiterungen, schlechte Integration oder fehlende Adaptivität. Insbesondere die Integration der vielen Anwendungen sorgt für eine enorme Steigerung der Komplexität von IT-Landschaften und erhöht zusätzlich den Aufwand für Wartungen und Erweiterungen. Dies kann dazu führen, dass die erwarteten Vorteile der Softwaresysteme durch deren Anpassungs- und Wartungsaufwand reduziert werden oder im schlimmsten Fall sogar der Aufwand den Nutzen übersteigt. Der Nutzen muss aus diesem

Grund erhöht werden, indem eine vollständige und flexiblere Integration aller Anwendungen durchgeführt wird [Hei11]. Dies ist aufgrund der Heterogenität und dem verteilten Charakter von gewachsenen IT-Landschaften komplex, da oft Altsysteme einen wichtigen Teil der Systemlandschaft bilden und diese nur mit großem Aufwand in eine einheitliche Umgebung zu integrieren sind. Gleichzeitig ist eine Reduzierung der Komplexität notwendig, um den Aufwand für die Wartungen von IT-Infrastruktur zu verringern, was mit einer neuen, unternehmensweit einheitlichen IT-Architektur erreicht werden kann. Die Komplexitätsreduktion und die Vereinfachung von Wartungs- und Weiterentwicklungsarbeit führt schließlich zu einer Kostenreduktion für den Betrieb der IT-Landschaft.

Weitere Treiber für zukünftige Änderungen der Produktion sind die Kosten, mit denen sich Produkte herstellen lassen. Heute lassen sich höhere Produktpreise nur durch eine bessere Qualität am Markt erfolgreich verkaufen, werden aber zunehmend von günstigeren Produkten mit geringerer Qualität verdrängt. Ziel muss deshalb sein, die Produktionskosten zu senken und gleichzeitig hohe Qualität zu liefern, um marktfähige Produkte zu erhalten [VWBZ09]. Ein gesättigter Markt erfordert, dass Produktionsunternehmen zu vergleichsweise günstigen Preisen ihre Produkte verkaufen und dennoch eine bessere Qualität wie Konkurrenzprodukte liefern, denn diese kann letztendlich ausschlaggebend für die Kunden zum Kauf sein. Die Zeit bis zur Markteinführung von neuen Produkten, auch „Time-to-market“ genannt, spielt darüber hinaus eine immer wichtigere Rolle. Ist das Produkt früher am Markt als ein vergleichbares Konkurrenzprodukt, so kann in dieser Zeit der Preis selbst bestimmt und somit bereits Umsätze erzielt werden, bevor ein Preiskampf mit dem Konkurrenzprodukt beginnt. Deshalb versuchen viele Unternehmen durch einen kürzeren Innovationszyklus eine frühere Markteinführung zu erreichen, um Vorteile gegenüber der Konkurrenz zu erzielen.

Unternehmen müssen darüber hinaus die Herausforderungen des bereits heute weitverbreiteten turbulenten Umfelds beherrschen, das kontinuierliche Anpassungen erfordert. Die Globalisierung, die die Märkte international öffnet, erhöht den Konkurrenzdruck auf Unternehmen und fordert eine höhere Diversifizierung der Produkte für den jeweiligen Markt. Dies führt unter anderem zu einer wachsenden Komplexität von Produkten und Prozessen, die ein erfolgreiches Unternehmen effizient beherrschen muss [SI08]. Die Produktionsprozesse dienen dabei zur Steuerung der Fertigung, die Geschäftsprozesse werden zur Unternehmenssteuerung eingesetzt. Auf diese Herausforderungen wird im Folgenden genauer eingegangen.

### 1.1.1 Turbulentes Umfeld

Unternehmen agieren heute in einem turbulenten Umfeld und müssen sich deshalb den kontinuierlichen Veränderungen anpassen, die von innen und außen auf das Unternehmen einwirken. Diese Einflussfaktoren werden Turbulenzkeime genannt. Beispiele hierfür sind schwankende Auftragseingänge, Veränderungen des Absatzmarktes, Auftreten von Konkurrenzprodukten, technologische Verbesserung, Änderungen der Produkte bzw. Prozesse oder Veränderung der personellen bzw. maschinellen Ressourcen [WZ09]. Auch gesetzliche Regulierungen, Unternehmensaufkäufe oder -zusammenschlüsse können Anpassungen von Produktionsunternehmen und Fertigung notwendig machen.

Um diesen Turbulenzen entgegen zu wirken, sind Konzepte zur Wandlungsfähigkeit von Produktionsunternehmen eingeführt worden, die eine kurz- und mittelfristige Anpassung der Organisation und Produktion erlauben [WZ09]. Wandlungsfähige Organisationsstrukturen erlauben Mitarbeitende dort einzusetzen, wo sie benötigt werden. Die Aufgaben, die Mitarbeitende innerhalb einer solchen Organisationsstruktur ausführen können, werden durch eine breite Qualifikation verbessert [KRSS08]. Im Bereich der Produktion kommen

modulare Fertigungssysteme zum Einsatz, die je nach Anforderungen auf einfache Weise erweiter- oder reduzierbar sind. Liegt ein System vor, das sowohl wandlungsfähige Produktionsmethoden und ein Produktionssystem bereit stellt, als auch ein entsprechend auf die Organisation abgestimmtes wandelbares Management bzw. Mitarbeitende besitzt, so spricht man von einem wandlungsfähigen Produktionssystem [NRA08, NHR<sup>+</sup>08, NFPPW09]. Dieses Konzept beschränkt sich jedoch auf die Verwaltung und Organisation von Unternehmen sowie einer aufwandsarmen Anpassung der Fertigung durch wandlungsfähige Maschinen. Eine wandlungsfähige IT-Architektur für Produktionsunternehmen wird in der Literatur nicht beschrieben.

In praktisch allen Unternehmensbereichen, wie z. B. zur schnelleren Planung der Fertigung und Verbesserung der Planungsqualität, werden heute in der Breite Softwarelösungen eingesetzt. Diese Anwendungen unterstützen die Mitarbeitenden bei ihren Aufgaben und erlauben, durch teilweise oder vollständig automatisierte Schritte, Aufgaben schneller abzuarbeiten. Die automatische Ausführung von einzelnen Arbeitsschritten durch IT-Systeme führt zu einer Beschleunigung des Gesamtprozesses und kann durch eine Reduzierung von Fehlern zu einer gleichzeitigen Erhöhung der Qualität führen. Die durch das turbulente Umfeld hervorgerufenen häufigen Anpassungen von Produkten und Produktion führen dazu, dass viele Arbeitsschritte mehrfach auszuführen sind. Somit macht sich die automatische Ausführung von Arbeitsschritten hier besonders bezahlt, da die Vorteile in jeder Iteration des Planungsprozesses genutzt werden.

### 1.1.2 Globalisierung

Die meisten mittleren und großen Unternehmen sind heutzutage international aufgestellt [WZ09]. Dabei geht der Trend weg von der lokalen Entwicklung und Produktion, um die Produkte anschließend in die ganze Welt zu exportieren, hin zu einer weltweiten Entwicklung und Produktion. Dies hat

zur Folge, dass man in jedem der lokalen Märkte gegenüber der Konkurrenz bestehen muss, was eine enorme Herausforderung an die Innovationskraft des Unternehmens darstellt. Insbesondere die lokalen Anforderungen an Qualität und Kosten der Produkte müssen berücksichtigt werden. Der Vorteil von einer lokalen Entwicklung ist jedoch, dass die Produkte besser an den jeweiligen Markt angepasst werden können, indem ein besonders niedriger Preis erzielt oder eine besonders gute Qualität erreicht wird, was zu einer steigenden Akzeptanz bei den Kunden führt. Zugleich ist eine bessere Berücksichtigung von gesetzlichen Vorgaben möglich. Die lokale Produktion sorgt zudem für geringere Logistikkosten und Lagerbestände können besser angepasst werden.

Die Zusammenarbeit über den gesamten Globus erfordert eine bessere Koordination der verteilten Entwicklungsteams und Produktionsstandorten. Dies wird durch eine verteilte IT-Infrastruktur unterstützt, die einen nahtlosen Datenaustausch zwischen allen beteiligten Parteien und somit allen eingesetzten Softwaresystemen innerhalb und zwischen den Abteilungen des Unternehmens ermöglicht. Darüber hinaus muss die IT-Infrastruktur die Einbindung von Partnern, Zulieferern und Kunden ermöglichen.

### **1.1.3 Wachsende Komplexität**

In Produktionsunternehmen ist eine kontinuierliche Zunahme der Komplexität festzustellen. Dies ist auf die steigende Variantenvielfalt der Produkte, die zunehmende Komplexität der Geschäfts- und Produktionsprozesse sowie der Globalisierung der Unternehmen zurückzuführen [An07, Est03, JWW09]. Der Trend von Massenprodukten hin zu individuellen Produkten sorgt für einen Wandel in der Produktion, der zu einer kundenindividuellen Serienfertigung (engl.: mass customization) führt. Diese kundenangepasste Fertigung stellt in einer Produktionslinie mehrere Varianten eines Produktes her, wodurch



mit der steigenden Anzahl an Varianten jeder Produktgeneration die Komplexität der Planung der entsprechenden Produktion kontinuierlich steigert. Die Verwaltung dieser enormen Anzahl an Varianten hat seit langem eine Komplexität erreicht, deren Verwaltung ausschließlich mit Softwarelösungen effizient durchführbar ist [Vol09].

Die Komplexität der Prozesse nimmt aufgrund der wachsenden Anzahl an Produktvarianten ebenso zu, da jede Variante unter Umständen einen eigenen Fertigungsprozess notwendig macht. Ein wachsendes Unternehmen, Kooperationen mit Partnern oder neue Methoden oder Verfahren können zu kontinuierlichen Veränderungen der Prozesse führen, die dadurch über die Zeit an Komplexität zunehmen und effizient verwaltet werden müssen. Insbesondere Softwaresysteme und Methoden helfen diese Komplexität zu beherrschen, indem Abläufe, Zusammenhänge oder Ergebnisse für die Mitarbeitenden automatisch aufbereitet und dargestellt bzw. anschaulich visualisiert werden. Unter Methoden wird in diesem Zusammenhang eine systematische Handlungsvorschrift verstanden, die ein planmäßiges Anwenden und eine begründete Vorgehensweise zur Erreichung von festgelegten Zielen beschreibt [HMF92, SW04]. Eine Methode kann zudem in einem Softwaresystem implementiert werden, um die systematische Ausführung zu unterstützen.

## 1.2 Problemstellungen

Wie im vorangegangenen Abschnitt dargelegt, ist der Einsatz von IT heutzutage von zentraler Bedeutung. Die Softwaresysteme erlauben die Unterstützung der Mitarbeitenden bei der Ausführung ihrer Arbeiten, so dass diese schneller und mit höherer Qualität durchführbar sind. Der Einsatz von Informations- und Kommunikationstechnologie (IKT) erlaubt einen schnellen und weltweiten Austausch von Informationen und Daten durch E-Mail,

Instant Messaging, Dateien Transfer, sowie vielen weiteren Kommunikations- und Datenaustauschprotokollen.

Ein Problem der Anwendungslandschaften von Produktionsunternehmen besteht darin, dass sie aus vielen Anwendungen bestehen, die unterschiedlich umfangreich sind. Betrachtet man Softwaresysteme für das Product Lifecycle Management (PLM), so gibt es wenige große Hersteller am Markt, die versuchen den gesamten Produktlebenszyklus digital zu unterstützen. Zusätzlich gibt es unzählige Anwendungen, die sich auf spezielle Aufgaben in allen Branchen der Produktionsindustrie fokussieren. Diese Anwendungen setzen alle auf ihre eigene proprietäre Datenhaltung und bilden sogenannte Insel-lösungen, da sie in der Regel für die Aufgaben oder Teilaufgaben innerhalb einer Fachabteilung entwickelt wurden [Min12]. Durch die kontinuierlich wachsende Anzahl an Softwaresystemen entstehen in kurzer Zeit Schnittmengen der Daten, d. h. verschiedene Anwendungen benötigen dieselben Daten und Informationen. Um eine wiederholte Eingabe von Daten zu vermeiden, werden Schnittstellen geschaffen, um die entsprechenden Daten zwischen den Anwendungen auszutauschen [Hei11]. Die Integration erfolgt dabei meist über Punkt-zu-Punkt Verbindungen, die mit steigender Anzahl schnell an Komplexität zunehmen und damit einen hohen Aufwand in Bezug auf Verwaltung und Wartung mit sich bringen, so wie frühere Bemühungen zeigen [Gun97, Sny91]. Diese Komplexität verhindert schnelle Änderungen der Anwendungen, da die gegenseitigen Abhängigkeiten bei Änderungen von einer Anwendung sich über die Schnittstellen auf andere Anwendungen auswirken<sup>1</sup>. Diese Abhängigkeiten müssen bei Änderungen berücksichtigt werden, was letztendlich zu einem hohen administrativen Aufwand führt, der mit hohen Kosten für die Verwaltung der IT-Landschaft verbunden ist.

Die Software-Initiative Deutschland hat in einer Studie festgestellt, dass zweidrittel aller Geschäftsprozesse in Großunternehmen nicht mehr von der zentralen IT-Abteilung kontrolliert werden [Sof11]. Der Hauptgrund die

---

<sup>1</sup>[http://www.mckinsey.com/insights/business\\_technology/it\\_architecture\\_cutting\\_costs\\_and\\_complexity](http://www.mckinsey.com/insights/business_technology/it_architecture_cutting_costs_and_complexity)

Verwaltung von Geschäftsprozessen heutzutage nicht mit Standardsoftware durchzuführen, wie z. B. SAP oder Oracle, ist die hohe Geschwindigkeit mit der Prozessanpassungen erfolgen müssen. Dies ist auf das turbulente Umfeld zurückzuführen, in dem die Unternehmen agieren. Zudem enthält die Studie einen negativen Ausblick, der zukünftig davon ausgeht, dass in den nächsten Jahren rund 80 % der Geschäftsprozesse nicht mehr von Standardsoftware gesteuert werden kann.

Die Probleme die sich daraus ergeben sind vielfältig. Der Status, in dem sich die Geschäftsprozesse befinden, ist nicht mehr zugänglich, wenn keine Standardsoftware eingesetzt wird auf die jeder Verantwortliche Zugriff hat. Die Aufgaben bzw. Teilprozesse, die normalerweise durch Software automatisch ausführbar sind, müssen manuell gestartet oder durchgeführt werden, wie z. B. die automatisch generierte Bestellung von neuen Teilen, wenn ein gewisser Lagerbestand unterschritten wird. Außerdem ist die Überwachung von Geschäftsprozessen nicht mehr möglich und somit sind Potentiale für Prozessoptimierungen nur schwer erkennbar. Nicht zuletzt kann die Qualität bei manueller Ausführung von Prozessen darunter leiden, was letztendlich zu höheren Kosten führt.

Die hohe Bedeutung der Wandlungsfähigkeit, um die Auswirkungen von Turbulenzen zu vermindern oder zu vermeiden, beschreibt eine Studie über die Wandlungsfähigkeit von produzierenden Unternehmen [KWW03]. Über 80 % der befragten Unternehmen geben in der Studie an, dass sie eine Verbesserung ihrer Wandlungsfähigkeit unter der Prämisse der Wirtschaftlichkeit für erforderlich erachten. Als Hindernisse für eine Erhöhung der Wandlungsfähigkeit im Unternehmen gibt die Studie z. B. die Ängste der Mitarbeitenden vor Veränderungen, ein Management das den Wandel nicht fordert, Personalmangel und Abhängigkeiten vom Zulieferer an. Bei Unternehmen, die sich bereits als wandlungsfähig einschätzen, sind die größten Hindernisse für eine weitere Verbesserung der Wandlungsfähigkeit die mangelnden personellen

Ressourcen, die Ängste der Mitarbeitenden vor Veränderungen und starre Unternehmenssoftware.

Die Durchdringung mit diesen Softwaresystemen speziell für kleine und mittlere Unternehmen (KMU), sowie der Einsatz über den gesamten Produktlebenszyklus, sind heute noch nicht gegeben. Dies kann nur geschehen, wenn die Implementierung dieser Systeme als ganzheitlicher Prozess betrachtet wird, der die Organisation, die Technik und den Menschen umfasst [ES09].

Eine weitere Schwierigkeit besteht darin, die Vielzahl der Anwendungen über eine einheitliche IT-Architektur bei gleichzeitig geringem Aufwand und Kosten zu integrieren, da insbesondere KMUs zumeist nur geringe Ressourcen für die Entwicklung der IT-Infrastruktur haben [Hei11]. Die eingeschränkten Ressourcen beziehen sich hauptsächlich auf das eingeschränkte Budget für die IT-Abteilungen und ihre oft knappe personelle Besetzung, die durch die starke Einbindung in das Tagesgeschäft sehr geringe Kapazitäten für die Weiterentwicklung der IT-Infrastruktur haben.

## 1.3 Vision, Ziele und Forschungsfragen

Im Folgenden werden die Vision, Ziele und Forschungsfragen der vorliegenden Arbeit beschrieben. Insbesondere die Frage, wie man mithilfe von IT-Systemen die Prozesse in Produktionsunternehmen schneller und flexibler unterstützen kann, steht dabei im Fokus.

### 1.3.1 Vision

Die derzeitigen Softwarelösungen und ihre Integration innerhalb der IT-Landschaft erlauben keine schnelle Adaption an ein sich ständig änderndes Umfeld. Aus diesem Grund wird eine adaptive IT-Architektur benötigt, die

schnelle Anpassungen, sowohl der Integrationsumgebung eines Produktionsunternehmens, als auch der unterstützten Geschäftsprozesse erlaubt. Zusätzlich soll die Architektur eine bestehende heterogene IT-Landschaft integrieren können, um einen nahtlosen Austausch der Daten zwischen Anwendungen zu gewährleisten und Prozesse über Anwendungsgrenzen hinweg unterstützen zu können.

Insbesondere die starre Unternehmenssoftware soll in dieser Arbeit betrachtet werden, indem eine Serviceorientierte Architektur (SOA) in Produktionsunternehmen eingeführt und damit die Adaptivität der IT-Landschaft verbessert wird. Die Begriffe Adaptivität und Flexibilität werden im Rahmen dieser Arbeit synonym verwendet. Zusätzlich soll der Aspekt der Wandlungsfähigkeit auf die IT-Landschaft übertragen werden, um Unternehmen jeder Größe die Vorteile der höheren Adaptivität zu ermöglichen.

Das Paradigma der SOA bietet die geforderten Eigenschaften, eine heterogene IT-Landschaft durch lose Kopplung von Services flexibel zu integrieren. Die Möglichkeit diese Services flexibel in Workflows zu kombinieren, erlaubt eine adaptive Abbildung und Ausführung von Prozessen durch IT-Systeme. Die Herausforderung besteht darin, dieses Paradigma an die Anforderungen der Produktionsunternehmen anzupassen, um deren heterogene IT-Landschaft vollständig und adaptiv integrieren zu können. Die Integration selbst erfolgt über den Enterprise Service Bus (ESB), einem plattformunabhängigen Integrationskonzept, das über standardisierte Serviceschnittstellen die Kommunikation zwischen Anwendungen der heterogenen und verteilten IT-Landschaft ermöglicht.

Die nahtlose Integration ermöglicht nicht nur anwendungs- und abteilungsübergreifenden Datenaustausch, sondern auch eine darauf aufbauende Prozessdefinition. Diese modellierten Prozesse ermöglichen eine automatische bzw. teilautomatische Ausführung, wodurch sich die Prozessausführungszeit verkürzt. Dies kann gegenüber der Konkurrenz entscheidende Vorteile bringen, wenn neue Produkte schneller am Markt platziert werden.

Die Verwendung einer SOA reduziert die Komplexität der Anwendungsintegration durch die Bereitstellung von standardisierten Serviceschnittstellen. Der ESB hingegen sorgt für die Integration der typischerweise verteilten und heterogenen Anwendungslandschaft über eine einheitliche Plattform. Gleichzeitig wird die Qualität der Entwicklungs-, Planungs- und Produktionsprozesse durch die digitale Unterstützung erhöht. Darüber hinaus wird durch den durchgängigen Einsatz von integrierten Softwaresystemen eine bessere Beherrschung der Prozesskomplexität ermöglicht.

### 1.3.2 Ziele der Arbeit

Wie im vorangegangenen Abschnitt 1.2 dargestellt, bieten die heutigen Softwarelandschaften in Unternehmen nicht die benötigte Adaptivität, um sich neuen Anforderungen schnell genug anzupassen. Aus diesem Grund wird eine neue Architektur benötigt, die die geforderte Adaptivität und Wandlungsfähigkeit besitzt und den Anforderungen der Produktionsunternehmen gerecht wird.

Die Adaptivität ermöglicht Änderungen an IT-Systemen, d. h. den Austausch, das Upgrade oder das Hinzufügen von neuen Anwendungen mit geringem Aufwand durchzuführen und dabei keine oder geringe Änderungen an anderen Systemen vornehmen zu müssen. Die unterstützten Prozesse sollen ebenfalls nur mit minimalem Aufwand an die Änderungen adaptierbar sein. Die Wandlungsfähigkeit dient der Erweiterung und Reduzierung der Gesamtarchitektur nach Anforderungen, wie z. B. Verfügbarkeit, Performanz und Zuständigkeit der Integrationslösung. D. h. die Architektur muss skalierbar sein, so dass bei höheren Anforderungen die Architektur einfach erweitert werden kann, um die Verfügbarkeit und Performanz nicht zu beeinträchtigen. Bei sinkenden Anforderungen muss die Architektur ebenso einfach verkleinert werden können, um unnötige Kosten für Wartung und Hardware zu vermeiden. Somit können Produktionsunternehmen die IT-Architektur

in jeder beliebigen Größe einsetzen und die verwendeten Ressourcen ihren Anforderungen entsprechend flexibel anpassen.

Ein weiteres Ziel ist die Verbesserung der Integration aller Anwendungen, um einen flexiblen Datenaustausch zu gewährleisten. Dazu müssen die bisher häufig zur Integration eingesetzten Punkt-zu-Punkt Schnittstellen ersetzt werden, da sie schnell zu komplexen, wartungsintensiven und deshalb teuren Integrationslösungen führen. Durch die Erweiterung der bisherigen IT-Landschaft um neue Schnittstellen kann den Mitarbeitenden innerhalb der verwendeten Anwendungen oder in der Fertigung mehr Daten und Informationen zur Verfügung gestellt werden, wodurch eine Verbesserung der Entwicklungsqualität, Planung und Produktion entsteht.

Mithilfe der Integration sollen nicht nur der Datenaustausch über Anwendungs- und Abteilungsgrenzen hinweg möglich sein, sondern gleichzeitig Prozesse zwischen den Anwendungen und Abteilungen definieren. Diese Prozessdefinition soll dieselbe Adaptivität aufweisen wie die Integration der Anwendungen selbst. Somit wird sichergestellt, dass Anpassungen schnell durchführbar sind.

Die Architektur soll zudem den Austausch von Daten mit externen Partnern über dieselben Schnittstellen ermöglichen wie bei internen Anwendungen, abgesehen von höheren Sicherheitsanforderungen.

### **1.3.3 Forschungsobjekte**

Die starre Unternehmenssoftware ist einer der Hauptgründe, warum sich Unternehmen nicht schnell genug an das sich wechselnde Umfeld anpassen können. Aus diesem Grund wird im Rahmen dieser Arbeit eine Architektur auf Basis einer SOA entwickelt, die zum einen die Anpassungsfähigkeit der Anwendungsintegration verbessert, wodurch ein flexibler Datenaustausch zwischen beliebigen Anwendungen erfolgen kann (Tabelle 1.1, Z-1.1). Zum

anderen müssen diese IT-Systeme eine Unterstützung von adaptiven Prozessen ermöglichen, um diese bei geänderten Anforderungen schnell anpassen zu können.

Bestehende IT-Landschaften weisen bereits eine hohe Komplexität auf, die durch Integration weiterer Anwendungen dazu führen kann, dass die IT-Infrastruktur nicht mehr wartbar ist. Aus diesem Grund wird eine Middlewarerelösung benötigt, die eine einheitliche Integration aller Anwendungen eines Produktionsunternehmens ermöglicht (Tabelle 1.1, Z-1.2). Somit können auch Anwendungen aus unterschiedlichen Domänen ihre Daten über die neue IT-Architektur aufbauend auf der ESB-Technologie flexibel austauschen.

Die IT-Architektur soll darüber hinaus leicht veränderbar sein, um zum einen bei Anpassungen eines Produktionsunternehmens oder durch Änderungen der integrierten Anwendungen sehr einfach erweiterbar bzw. reduzierbar zu sein (Tabelle 1.1, Z-1.3). Zum anderen sollte die neue IT-Architektur für jede Größe von Produktionsunternehmen geeignet sein. Somit ist die Architektur bereits für kleine Unternehmen geeignet, die nur wenige Anwendungen integrieren müssen, aber auch für größere Unternehmen, bei denen hunderte oder tausende Anwendungen über eine einheitliche Middleware zu integrieren sind.

Nicht nur der Datenaustausch zwischen Anwendungen soll in der neuen IT-Architektur eine höhere Adaptivität aufweisen, auch die Prozesse müssen für zukünftige Anforderungen flexibler durch IT-Systeme unterstützt werden (Tabelle 1.1, Z-2.1). Dabei steht sowohl die Prozessbeschreibung zwischen unterschiedlichen Anwendungen innerhalb einer Domäne im Vordergrund, als auch die domänenübergreifenden Prozesse.

Darüber hinaus ist ein durchgehender Informationsaustausch mit Dienstleistern, Lieferanten und Kunden notwendig, um Daten und Prozesse flexibel über Unternehmensgrenzen hinweg austauschen bzw. ausführen zu können (Tabelle 1.1, Z-3.1). Dies wird erreicht, indem externe Services ebenso



Ziele der Arbeit	Forschungsobjekte
Z-1.1: Flexible Integration der bestehenden heterogenen Anwendungslandschaft im Unternehmen	FO-1: Integration der Anwendungen durch Implementierung von standardisierten Service-schnittstellen verbessern und Datenaustausch zwischen verschiedenen Domänen durch lose Kopplung der Serviceschnittstellen ermöglichen.
Z-1.2: Komplexität der IT-Landschaft reduzieren und somit Wartbarkeit verbessern	FO-2: Einheitliche Integrationsarchitektur über eine serviceorientierte Middleware, um Punkt-zu-Punkt Integrationsverbindungen zu ersetzen. Die reduzierte Komplexität sorgt für einfachere und schnellere Durchführung von Änderungen.
Z-1.3: Einfache Veränderung der IT-Landschaft ermöglichen	FO-3: Den Aspekt der Wandlungsfähigkeit in die IT-Architektur einführen, um die Architektur, sowohl für kleine und mittlere Unternehmen, als auch für große Unternehmen anwendbar zu machen.
Z-2.1: Adaptivität der Prozesse erhöhen	FO-4: Eine adaptive IT-Unterstützung von Prozessen innerhalb und zwischen unterschiedlichen Domänen ermöglichen, um diese einfach und schnell an geänderte Anforderungen anpassen zu können.
Z-3.1: Austausch von Informationen zwischen dem Unternehmen und seinen Partnern	FO-5: Die IT-Architektur soll einen flexiblen Datenaustausch mit anderen Unternehmen ermöglichen und Prozesse über Unternehmensgrenzen hinweg durch Softwaresysteme unterstützen.
Z-3.2: Integration von mobilen Endgeräten in IT-Architektur	FO-6: Einheitliche Schnittstellen zur einfachen Datenbereitstellung für mobile Anwendungen.
Z-3.3: Auswirkungen von Änderungen auf die Organisation berücksichtigen	FO-7: Die Reorganisation, die mit Änderungen an der IT-Landschaft einhergeht, sollte minimiert werden, um Mitarbeitenden den Wandel zu erleichtern und damit weniger Widerstände aufzubauen.

Tabelle 1.1: Ziele der Arbeit und Forschungsobjekte

adaptiv in die IT-Architektur eingebunden werden wie interne Services. Darüber hinaus erlauben diese Services eine ähnliche Unternehmensgrenzen überschreitende Prozessdefinition wie bei internen Prozessen.

Die zunehmende Verbreitung von mobilen Endgeräten wird zukünftig die Arbeit in Produktionsunternehmen beeinflussen. Aus diesem Grund muss eine Integrationsplattform Schnittstellen für den Datenaustausch mit mobilen Anwendungen bereitstellen, um einzelne Prozessschritte auf mobile Endgeräte verlagern zu können (Tabelle 1.1, Z-3.2).

Ein weiterer wichtiger Aspekt bei der Einführung einer neuen IT-Architektur sind die Auswirkungen auf die Organisationsstruktur der Produktionsunternehmen (Tabelle 1.1, Z-3.3). Diese sollten möglichst minimiert werden, um Ängste und Widerstände der Mitarbeitenden zu verringern oder zu vermeiden. Die neue IT-Architektur sollte deshalb gegenüber bestehenden Organisationsstrukturen, insbesondere der IT-Abteilungen, keine grundlegenden Erneuerungen erfordern.

Die Ziele und Forschungsobjekte der vorliegenden Arbeit sind nochmals in Tabelle 1.1 zusammengefasst.

## 1.4 Aufbau der Arbeit

Der Aufbau dieser Arbeit, dargestellt in Abbildung 1.1, gliedert sich wie folgt: Das Kapitel 2 beschreibt die Grundlagen der Arbeit und ist in zwei Teile aufgeteilt. Im ersten Teil sind verschiedene Managementkonzepte beschrieben, die in Produktionsunternehmen zum Einsatz kommen, um Abläufe besser zu strukturieren oder Prozesse einheitlicher zu beschreiben. Die Unterstützung dieser Managementkonzepte erfolgt durch IT-Lösungen, die mithilfe der heute vorhandenen Informationstechnik verbessert werden soll und im zweiten Teil des Grundlagenkapitels beschrieben ist. Mit der steigenden Verbreitung



Abbildung 1.1: Gliederung der Arbeit

von Softwarelösungen in Produktionsunternehmen wird deren effiziente Vernetzung immer wichtiger.

Verwandte Arbeiten sind in Kapitel 3 beschrieben, die zum einen den aktuellen Stand der Forschung aufzeigen und auf den neuesten Technologien aufsetzen, um eine effiziente Integration von Anwendungen einer Domäne zu erreichen. Zum anderen werden die in der Industrie eingesetzten Softwarelösungen beschrieben, die meist auf einer zentralen Datenbank verschiedene Softwarewerkzeuge integrieren. Diese Integration der Werkzeuge und Datenbanken erfolgt überwiegend in einer starren Weise durch proprietäre Schnittstellen, weshalb die notwendige Flexibilität nicht gegeben ist.

Die im Rahmen dieser Arbeit entwickelte IT-Architektur zur adaptiven und wandlungsfähigen Integration von Produktionsunternehmen wird detailliert in Kapitel 4 vorgestellt und diskutiert. Durch den modularen Aufbau kann auf einfache Weise eine domänenübergreifende Integration von Anwendungen erreicht werden, die die geforderte Adaptivität bietet und Prozesse flexibel unterstützt.

Die prototypische Implementierung der entwickelten IT-Architektur in der Lernfabrik ist Bestandteil des Kapitels 5 und integriert flexibel den Leit-rechner der Lernfabrik, den Planungstisch zur Fabriklayoutplanung und das Wartungsportal über einheitliche Serviceschnittstellen.

Das darauf folgende Kapitel 6 beschreibt die Ergebnisse der prototypischen Implementierung und zeigt die Vorteile der adaptiven und wandlungsfähigen IT-Architektur gegenüber anderen Lösungen anhand von vier Anwendungsszenarien auf. Im Anschluss erfolgt die systematische Bewertung der Architektur und anderen Lösungen mithilfe von sechs Kriterien, nachdem diese detailliert beschrieben werden.

In Kapitel 7 wird die Arbeit schließlich zusammengefasst und ein Ausblick auf zukünftige Arbeiten gegeben. Eine Erweiterung der Präsentationsschicht durch die Einbindung von mobilen Endgeräten in die Architektur ist ebenso möglich, wie eine systematische Analyse der Daten, die durch Serviceschnittstellen einfacher verfügbar sind. Zudem ist eine Portierung der entwickelten Architektur in die Cloud denkbar.

---

## KAPITEL 2

# Grundlagen

---

Dieses Kapitel beschreibt die Grundlagen der vorliegenden Arbeit und ist in zwei Teile aufgeteilt. Im ersten Teil sind verschiedene Managementkonzepte für Produktionsunternehmen beschrieben, die im Laufe der Jahre eingeführt wurden, um einzelne Bereiche effizienter zu verwalten. Die einzelnen Managementkonzepte beschreiben hierfür eine Systematik, die ein strukturiertes Vorgehen auf der organisatorischen Ebene darstellen, z. B. bei der Entwicklung von Produkten.

Der zweite Teil dieses Kapitels stellt informationstechnische Grundlagen vor, insbesondere Methoden zur Anwendungsintegration. Hier liegt der Fokus auf der serviceorientierten Architektur und den wichtigsten Bausteinen zu deren Umsetzung. Darüber hinaus wird das Konzept der Nachrichtenvermittlung beschrieben und der Enterprise Service Bus vorgestellt, als zentrale Middlewarelösung zur Integration von Anwendungen.

### 2.1 Managementkonzepte für Produktionsunternehmen

Die Effizienz in der Planung und Herstellung von Produkten ist heutzutage ein wichtiger Erfolgsfaktor für Produktionsunternehmen. Aus diesem Grund

sind über die Zeit Managementkonzepte wie z. B. Product Lifecycle Management (PLM), Factory Lifecycle Management (FLM) oder Supply Chain Management (SCM) entwickelt worden, um Aufgaben besser koordinieren und überwachen zu können. Der Fokus dieser verschiedenen Konzepte liegt dabei in der Betrachtung von Produktlebenszyklen, der Unterstützung von Mitarbeitenden in der Konzeption und Planung der Fabrik oder den Kunden- und Lieferantenbeziehungen.

In der nachfolgenden Abbildung 2.1 sind die wichtigsten Konzepte zur Unterstützung von Produktionsunternehmen nach ihrer zeitlichen Entwicklung aufgetragen [BWG11]. Zu Beginn wurden Textsysteme eingeführt, die die Textverarbeitung im Vergleich zu Schreibmaschinen deutlich vereinfachen sollten und deren Bedienung im Laufe der Zeit komfortabler wurde. Mit Einführung von Computer-aided Design (CAD) Systemen in den 1980er Jahren sind die ersten Ingenieurstätigkeiten durch Rechner unterstützt worden, die in kürzerer Zeit das konventionelle Zeichenbrett fast vollständig verdrängt haben. Der Erfolg von CAD führte nur wenige Jahre später zum Einsatz von Computer-aided Manufacturing (CAM) Systemen, die eine rechnerunterstützte Planung der Fabrik erlaubten. Dabei ist zwischen neuen Anwendungen für die Fabrikplanung, wie z. B. einem Werksstrukturdatenbank-System [Bra84], und der CAD gestützten Fabrikplanung, die bei großen Automobilherstellern für die Hallenlayoutplanung oder als Fabrikplanungs- und Informationssystem verwendet werden, zu unterscheiden [BWG11].

Durch die zunehmende Verbreitung von Rechnern und Anwendungen, sowie dem hohen Potential für Verbesserungen durch die Automatisierung der Produktion, führte dies Mitte der 1980er Jahre zum Konzept des Computer-integrated Manufacturing (CIM). Das Ziel von CIM war die isolierten Softwarelösungen mit ihrem Inselcharakter über ein Netzwerk zu integrieren, um eine durchgängige digitale Informationsverknüpfung sicherzustellen [BWG11]. Dabei dient CIM als Verbindung zwischen den technischen Computer-aided

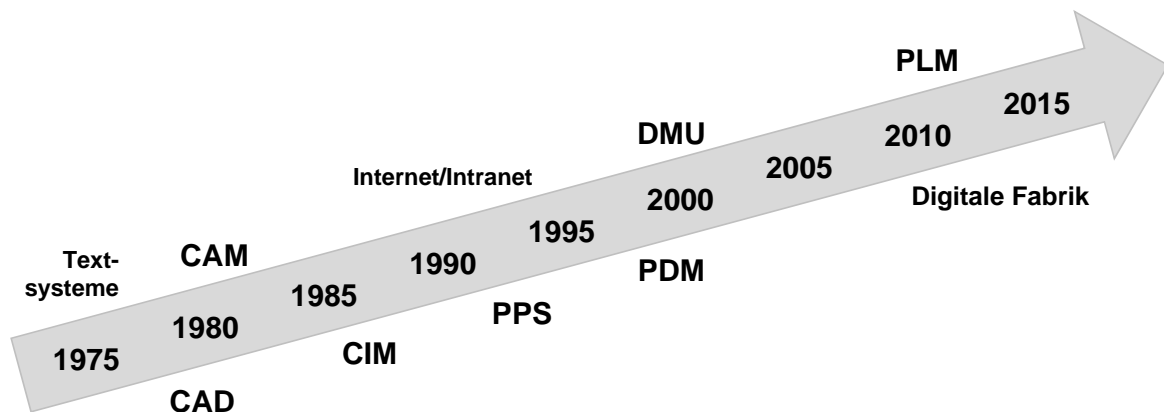


Abbildung 2.1: Historie der Rechnerunterstützung für Geschäfts-, Planungs- und Produktionsprozesse in Unternehmen [BWG11]

Everything (CAx)<sup>1</sup> Systemen und den betriebswirtschaftlichen Funktionen der Produktionsplanung und -steuerung (PPS) [BWG11, GT05]. Der Begriff PPS wurde in den 1980er Jahren eingeführt und fasste damals die Material- und Zeitwirtschaft in der produzierenden Industrie unter einem übergreifenden Konzept zusammen [Sch06].

Die Problematik CAD-Dokumente parallel mit gescannten Papierdokumenten zu verwalten, führte Mitte der 1980er Jahre zur Entwicklung von Product Data Management (PDM)-Systemen. In den 1990er Jahren wurden die PDM-Systeme aufgrund von Produkthaftungsanforderungen erweitert, um Dokumente mit Stamm- und Strukturdaten zu verknüpfen, sowie einfache Änderungsabläufe darin abbilden zu können [ES09]. Die letzte Erweiterung der PDM-Systeme verknüpft Produktionsprozesse und Ressourcen der Digitalen Fabrik mit entsprechenden Produktdaten und stellt deren Konsistenz sicher [BWG11]. Diese Verknüpfung wird in der Praxis jedoch im Rahmen der Digitalen Fabrik durchgeführt und nicht im PDM-System. Wenn das digitale Produkt detailliert genug modelliert ist, kann man von einem Digital Mock-up (DMU) sprechen. Ein DMU erlaubt die Funktionsfähigkeit und Herstellbarkeit eines Produktes mithilfe von verschiedenen Simulationen,

<sup>1</sup>Das x in CAx steht als Platzhalter für CAD, CAM, Computer-aided Planning (CAP) und/oder Computer-aided Quality Assurance (CAQ), die die verschiedenen Einsatzbereiche spezifizieren [VWBZ09]

Animationen und Tests zu verifizieren, bevor es physisch hergestellt wird [VWBZ09].

Die Digitale Fabrik ist eine Weiterführung von CAM und dient als Werkzeug zur digitalen Modellierung einer Fabrik. Im Vordergrund steht, wie bereits erwähnt, die Verknüpfung von Produkt, Prozess und Ressourcen (PPR), um daraus einen Produktionsplan ableiten zu können [VDI08]. Die Verifizierung des digitalen Fabrikmodells erfolgt durch Simulations- und Visualisierungswerkzeuge. Ende der 1990er Jahre ist als Erweiterung der PDM-Systeme das Konzept des PLM entstanden [ES09]. PLM betrachtet und verwaltet das Produkt von der ersten Idee, über die Entwicklung und Herstellung, bis hin zu dessen Recycling oder Entsorgung.

Im Folgenden werden in diesem Abschnitt die für diese Arbeit wichtigsten Konzepte zur Unterstützung von Produktionsunternehmen detaillierter vorgestellt. Bereits mit CIM wurde das Ziel verfolgt, eine vollständige Integration von Softwaresystemen zu erreichen. Die Entwicklung von CAD und PDM stellen die grundlegenden Anwendungen dar, um den Produktlebenszyklus im Rahmen von PLM ganzheitlich zu betrachten. Die Digitale Fabrik ist im Fabriklebenszyklus und dem FLM die zentrale Anwendung für die Fabrikplanung. Das SCM dient zum systematischen Verwalten von Kunden- bzw. Lieferantenbeziehungen und sorgt für eine effiziente Verwaltung von Materialbestellungen und Auftragsmanagement. Das Stuttgarter Unternehmensmodell (SUM) wiederum betrachtet Produktionsunternehmen in einem turbulenten Umfeld und stellt Methoden bereit, um dieses Umfeld besser zu beherrschen.

### **2.1.1 Computer-integrated Manufacturing**

Der zunehmende Einsatz von rechnerunterstützten Systemen, wie z. B. CAD und CAM, führte zu einer erhöhten Effizienz der Ingenieure. Die einzelnen



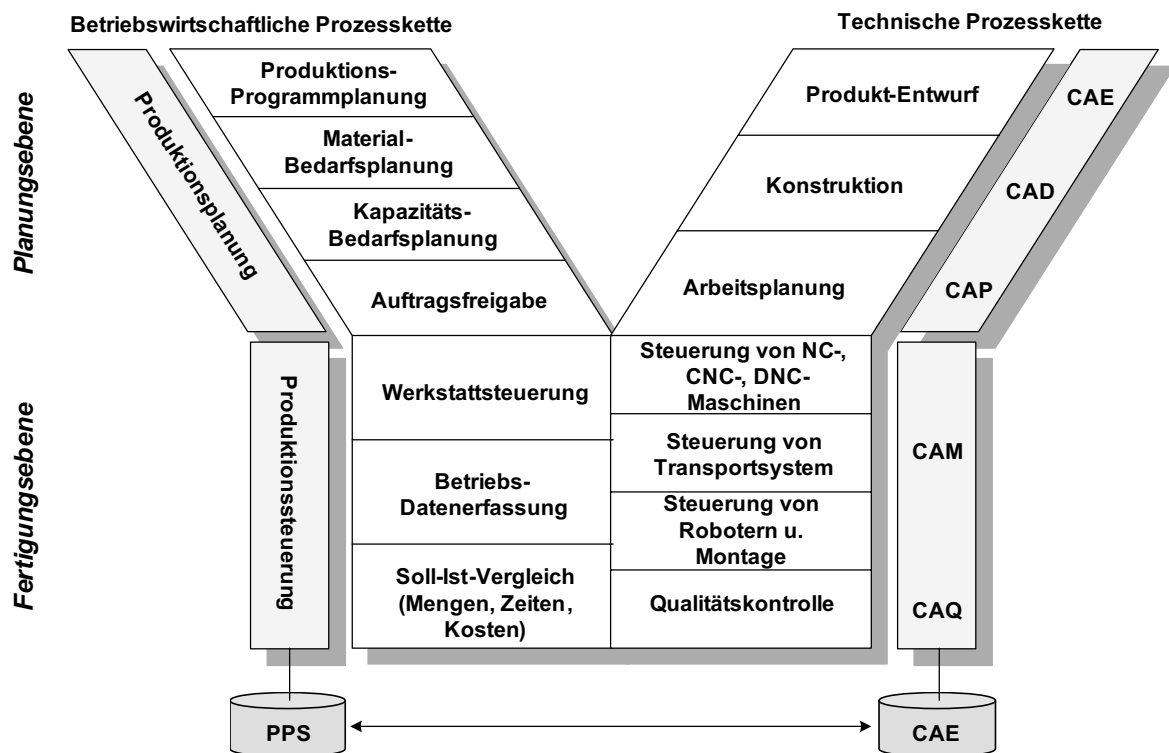


Abbildung 2.2: Y-CIM-Modell für den Industriebetrieb [Sch90, SBM<sup>+</sup>05]

Systeme bildeten dabei Insellösungen, deren Informationsaustausch nur mühsam und oft manuell vonstatten ging. Um schneller auf Änderungen reagieren zu können war deshalb die Integration der einzelnen Systeme von enormer Bedeutung [Gun97]. Aus diesem Grund hat Joseph Harrington im Jahre 1973 das Konzept des Computer-integrated Manufacturing (CIM) als eine Kontroll- und Kommunikationsstruktur beschrieben, um Produktionssysteme zu integrieren [Har73]. Neben den technischen Aspekten beschreibt CIM auch die Angelegenheiten der Organisation, Führung, Strategie und des Betriebs [Gun97]. Dabei integriert CIM nicht nur die technischen CAx-Systeme, sondern bindet auch betriebswirtschaftliche Softwarelösungen wie PPS-Systeme an die Gesamtlösung an, wie durch das Y-CIM-Modell von Scheer in Abbildung 2.2 dargestellt ist [Sch90, SBM<sup>+</sup>05]. Der Ausschuss für wirtschaftliche Fertigung (AWF) definierte CIM 1985 folgendermaßen [Aus85]:

„CIM beschreibt den integrierten EDV-Einsatz in allen mit der Produktion zusammenhängenden Betriebsbereichen. Es umfasst

das informationstechnische Zusammenwirken zwischen CAD, CAP, CAM, CAQ und PPS. Hierbei soll die Integration der technischen und organisatorischen Funktionen zur Produkterstellung erreicht werden. Dies bedingt die gemeinsame Nutzung einer Datenbasis.“

Die Integration der relevanten Betriebsbereiche steht also im Mittelpunkt neben den technischen und betriebswirtschaftlichen Funktionen. Die erfolgreiche Implementierung von CIM soll dazu führen, dass u. a. Produktionssysteme auf einfache Weise miteinander kommunizieren können. Weitere wichtige Aspekte sind eine schnellere Reaktion auf Datenänderungen und gleichzeitig eine höhere Flexibilität bei Einführung von neuen Produkten, sowie eine Verkürzung von Durchlaufzeiten und bessere Qualität der Produkte [NL99].

Diese Vielzahl an potenziellen Vorteilen veranlasste deshalb viele Produktionsunternehmen zur Einführung von CIM. Oft stellten sich die erhofften Vorteile jedoch nicht ein, was unter anderem an Softwareinstallationen von verschiedenen Herstellern, Mangel an vollwertigen CIM-Softwarelösungen, unausgereiften Integrationsprodukten und dem Netzwerkmanagement lag [Sny91]. Des Weiteren wird die erhoffte Flexibilität bei der Integration der IT-Systeme aufgrund von Integrationslücken, Automatisierungsinselfn, ungenügend optimierten Ressourcen und der Schwierigkeiten auf neue Technologien zu migrieren, nicht erreicht [GNK92].

Neben den technischen Problemen, die CIM-Projekte zum Misserfolg führten, waren es vor allem organisatorische Probleme. Die organisatorischen Änderungen, die eine CIM Einführung zwangsläufig mit sich bringt, muss von den Mitarbeitenden mitgetragen werden. Aus diesem Grund sollten Mitarbeitende in solche Projekte eingebunden und Lehrgänge durchgeführt werden, um den Widerstand gegen die Änderungen zu minimieren. Mitarbeitende nicht in solch umfassende Projekte einzubeziehen, kann dazu führen, dass

der Widerstand zu Verzögerungen bei der Umsetzung des Projektes bis hin zum kompletten Scheitern führt [Gun97].

Die Ziele von CIM, eine durchgängige Integration von allen bestehenden Softwaresystemen zu erreichen und somit Prozesse schneller auszuführen, waren vielversprechend. Das Fehlen von Integrationstechnologie und Standards führte jedoch in den meisten Fällen zu individuellen und starren Lösungen, die dem eigentlichen Gedanken von mehr Flexibilität entgegenstanden. Erfolgreiche CIM-Umsetzungen beschränkten sich meist auf einen Teil der Gesamtlösung und konnten somit nicht den maximalen Nutzen erreichen.

## 2.1.2 Produktdatenmanagement

Produktdaten effizient zu verwalten stellt eine enorme Herausforderung dar. Zur Unterstützung von Ingenieuren wurden deshalb CAD-Systeme entwickelt und eingeführt, die den Prozess der Produktentwicklung digitalisieren. Somit wird die Wiederverwendung von Teilen ermöglicht, was wiederum zu einer kürzeren Entwicklungszeit führt. Durch die wachsenden Mengen der digitalen CAD-Daten wurde das Produktdatenmanagementkonzept (engl.: Product Data Management (PDM)) notwendig. PDM verwaltet alle Daten, die während der Produktentwicklung erstellt werden, sorgt für eine konsistente Versionierung der Daten und für das Freigabemanagement im Entwicklungsprozess. PDM wird von Eigner wie folgt definiert [ES09]:

„PDM ist das Management des Produkt- und Prozessmodells mit der Zielsetzung, eindeutige und reproduzierbare Produktkonfigurationen zu erzeugen.“

Das PDM-System verwaltet die in CAD-Anwendungen erstellten Daten und die zugehörigen Prozesse, die für eine reproduzierbare Erstellung, Änderung, Struktur und Freigabe innerhalb der Produktentwicklung sorgen.

Somit können Prozesse effizienter ausgeführt werden und durch die zentrale Datenhaltung werden inkonsistente Daten vermieden.

### 2.1.3 Digitale Fabrik

Der Verein Deutscher Ingenieure (VDI) definiert die Digitale Fabrik in seiner VDI-Richtlinie 4499 Blatt 1 wie folgt [VDI08]:

„Die Digitale Fabrik ist der Oberbegriff für ein umfassendes Netzwerk von digitalen Modellen, Methoden und Werkzeugen - u. a. der Simulation und der dreidimensionalen Visualisierung - , die durch ein durchgängiges Datenmanagement integriert werden.

Ihr Ziel ist die ganzheitliche Planung, Evaluierung und laufende Verbesserung aller wesentlichen Strukturen, Prozesse und Ressourcen der realen Fabrik in Verbindung mit dem Produkt.“

Der Fabrikplaner wird durch die Digitale Fabrik beim Erstellen eines virtuellen Modells unterstützt, sowie bei der anschließenden Optimierung des selbigen. Dieses virtuelle Modell beinhaltet das Produkt, dessen Prozesse und Ressourcen, die miteinander verknüpft die virtuelle Fabrik repräsentieren [Küh06]. Ressourcen können dabei Maschinen, Roboter, Produktions- und Montagelinien, Lager, Puffer, Transporteinheiten und Menschen sein [WFK01]. Die digitale Optimierung soll zu ausgereiften Prozessen führen, bei denen weniger oder im Optimalfall keine Fehler in der realen Umsetzung auftreten [Küh06]. Der Zusammenhang zwischen digitaler, virtueller und realer Fabrik ist in Abbildung 2.3 dargestellt.

Die Digitale Fabrik besteht aus verschiedenen Werkzeugen, die die einzelnen Aufgaben Produktentwicklung, Fabrik- und Produktionsplanung, Inbetriebnahme und Anlauf der Produktion, sowie Produktionsbetrieb und Auftragsmanagement unterstützen [Küh06]. Durch das Modellieren einer Fabrik in

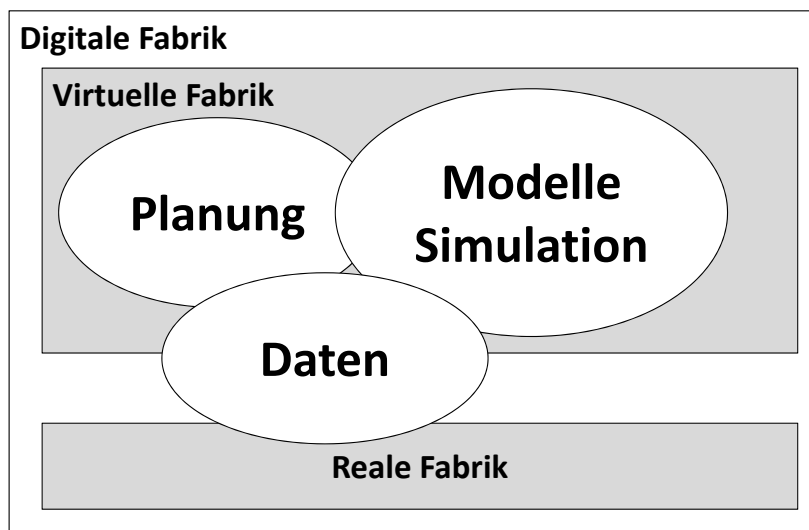


Abbildung 2.3: Zusammenhang zwischen digitaler, virtueller und realer Fabrik [Küh06]

einem digitalen Abbild wird deren Funktions- und Leistungsfähigkeit bereits vor der physischen Umsetzung verifiziert. Somit kann der Zeiteinsatz für fehlerhafte oder ineffiziente Umsetzungen vermieden oder zumindest reduziert werden. Dies führt zu geringeren Produktionskosten, einer höheren Produktivität, Qualität und Produktvielfalt [WFK01]. Zudem werden die Zeiten bis zur Markteinführung, der Lieferung zum Kunden, sowie zum Erreichen der maximalen Produktionsmenge gesenkt [VDI08, Küh06]. Die genannten Vorteile können somit die Wettbewerbsfähigkeit des Unternehmens am Markt verbessern [WFK01].

### 2.1.4 Product Lifecycle Management

Die PDM-Systeme werden heute zunehmend durch das Product Lifecycle Management (PLM) erweitert, das nicht nur die Produktentwicklung abdeckt, sondern Produktinformationen auch früher Phasen mit Markt- und Anforderungsanalysen, der Produktionsplanung und Produktion sowie Postproduktionsphasen mit Nutzung, Wartung und schließlich Recycling bzw. Verschrottung bereit stellt.

Das PLM Development Consortium der Universität von Michigan definiert PLM folgendermaßen:

„PLM is an integrated, information driven approach to all aspects of a product’s life from its design inception, through its manufacture, deployment and maintenance and culminating in its removal from service and final disposal.“

PLM ist laut dieser Definition ein integrierter, datengetriebener Ansatz zur Verwaltung eines Produktes vom ersten Design bis zur Entsorgung. Die Firma CIMdata Inc. sieht PLM hingegen als ein strategisches Vorgehen zur Beschreibung von Produkt bzw. Fabrik und definiert es wie folgt:

„[...] a business approach to solving the problem of managing the complete set of product and plant definition information and the processes through which it passes. The PLM process includes creating and changing that information, managing it through its life and disseminating and using it throughout the lifecycle of the product.“<sup>2</sup>

Im Gegensatz zur ersten Definition, die PLM als eine technische Lösung sieht, geht CIMdata auf PLM als organisatorischen Lösungsansatz ein. Zudem wird PLM nicht nur als Lösung zur effizienten Verwaltung von Produktinformationen gesehen, auch Fabrikinformationen werden hier mit betrachtet. Dieser Aspekt wird in der Literatur nicht einheitlich behandelt, weshalb in Abschnitt 3.1 näher darauf eingegangen wird.

PLM begleitet ein Produkt von den ersten Ideen, über die Entwicklung und Produktion, bis hin zum Recycling oder der Verschrottung. Der Lebenszyklus wird dabei in der Regel in verschiedene Phasen aufgeteilt, um die einzelnen Entwicklungsstufen des Produktes über dessen Existenz hinweg zu verdeutlichen. Die Anzahl bzw. der Inhalt der Phasen unterscheidet sich dabei in der

---

<sup>2</sup><http://www.cimdata.com/plm/definition.html>



Abbildung 2.4: Phasen des Produktlebenszyklus nach Stark (vergleiche [Sta11])

Literatur sehr und wird oft ergänzt durch die Bereiche Produktionsplanung und Vertrieb. Ein Phasenmodell, das ausschließlich auf das Produkt fokussiert ist, wird von Stark in fünf Phasen beschrieben und ist in Abbildung 2.4 dargestellt. Auf die Vielzahl der unterschiedlichen Phasenmodelle wird detailliert in Abschnitt 4.4.1.1 eingegangen.

### 2.1.5 Factory Lifecycle Management

Klassische Methoden der Fabrikplanung beschreiben den Prozess von der Planungsinitiative, die die globale Aufgabenstellung der Fabrik umfasst, bis hin zur Fabriknutzung. Der Prozess wird dabei in folgende sechs Abschnitte aufgeteilt: 'Zielplanung', 'Vorplanung', 'Grobplanung', 'Feinplanung', 'Ausführungsplanung' und 'Ausführung' [Gru08]. Die Grobplanung wird weiter in 'Idealplanung' und 'Realplanung' unterteilt. Abschnitte während des Betriebs, wie Instandhaltung und Optimierung, werden dabei nicht betrachtet, genauso wie die Modernisierung oder der Rückbau der Fabrik an deren Lebensende.

Neuere Ansätze sehen die Fabrik als einen neuen und komplexen Typ von Produkt [WCH06]. Diese „Fabrik als Produkt“ durchläuft wie jedes Produkt einen Lebenszyklus, wodurch sich, analog zum Konzept des PLM, in den letzten Jahren das Konzept des Factory Lifecycle Management (FLM) entwickelt hat. Dabei wird die Idee der einzelnen Phasen, die ein Produkt in seinem Leben durchläuft, auf die Fabrik übertragen. Der Fabriklebenszyklus besteht aus dem Design und der Planung der Fabrik, der Konstruktion, dem Betrieb und der Wartung, der Modernisierung oder Veralterung. Dies führt letztendlich zur Demontage und dem Rückbau der Fabrik [WCH06]. Dieser

Fabriklebenszyklus wird im Rahmen des FLM verwaltet. Eine Methode zur Umsetzung des FLM ist, das Konzept des advanced Industrial Engineering (aIE) aus der Fabrikplanung auf alle Phasen des Fabriklebenszyklus zu übertragen [CHW06]. Die Realisierung erfolgt hier durch die Entwicklung einer kollaborativen, offenen und standardisierten Plattform. Der Fokus liegt bei diesem Ansatz auf der technischen Integration der einzelnen Phasen, der über ein Referenz-Informationsmodell den Datenaustausch zwischen den einzelnen Anwendungen des Lebenszyklus ermöglichen soll. Dabei setzt dieser auf standardisierte Sprachen zur Beschreibung und zum Austausch der Daten.

Die Digitale Fabrik kann bei der Umsetzung des FLM als zentrale Komponente eingesetzt werden, die die Verwaltung aller für die Fabrikplanung relevanten Daten übernimmt. Dies ist vergleichbar mit der Rolle des PDM-Systems im PLM, das alle Produktdaten verwaltet und für alle Anwendungen des Produktlebenszyklus zur Verfügung steht.

Wie das Zusammenspiel Produkt- und Fabriklebenszyklus aussehen kann, zeigt das sogenannte Kanonenmodell, das beide Konzepte kombiniert und deren Abhängigkeiten beschreibt [HW07]. Insbesondere die der Produktion vorgelagerten Phasen der Produktentwicklung und Produktionsplanung können durch eine bessere Abstimmung Aufgaben dieser Phasen parallel abarbeiten. Dies spart Zeit ein und die Umsetzung bis zum Start der Produktion kann schneller erfolgen. Diese parallele Entwicklung wird mit dem Konzept des Simultaneous Engineering bzw. Concurrent Engineering beschrieben, auf das im Rahmen dieser Arbeit nicht weiter eingegangen wird [Kus93, BW95, FGYO95]. Noch frühere Phasen können durch bessere Abstimmung untereinander ebenso Zeit und Ressourcen sparen, wie z. B. durch parallele Entwicklung von Bauteilen und der zugehörigen Werkzeuge, die die Teile schließlich herstellen.



### 2.1.6 Supply Chain Management

Das Supply Chain Management (SCM) beschäftigt sich mit der Verwaltung der Lieferkette. Die Verwaltung umfasst die Planung, Umsetzung und Kontrolle des effizienten Materialflusses. Dies beginnt bei der Lagerung von Rohmaterial und setzt sich über den Lagerbestand im Produktionsprozess fort, bis schließlich das fertige Produkt mit den zugehörigen Dienstleistungen erstellt ist [Pol10]. Die Erfüllung der Kundenwünsche ist dabei das oberste Ziel der Supply Chain, zu der alle beteiligten Parteien gehören [CM12]. Die beteiligten Parteien sind nicht nur Zulieferer, Produktionsunternehmen und Kunden, sondern z. B. auch Transportunternehmen und Einzelhändler.

Mit dem Supply Chain Operations Reference (SCOR) wurde vom Supply Chain Council (SCC) ein systematisches und standardisiertes Modell geschaffen, um unternehmensinterne und unternehmensübergreifende Geschäftsprozesse in der Supply Chain zu beschreiben [BRP08]. Das SCOR-Modell beschreibt fünf Prozess-Elemente der Lieferkette – das Planen, Beschaffen, Herstellen, Liefern und Rückliefern. Zudem erlauben Metriken des SCOR-Modells eine stabile Grundlage, um die Leistungsfähigkeit der Supply Chain Prozesse zu messen und Prioritäten für Verbesserungen zu identifizieren<sup>3</sup>. Diese Metriken umfassen die Funktionsfähigkeit, Reaktionsfreudigkeit, Agilität, Kosten und Anlagegüter der Supply Chain und deren Prozesse.

Die oben genannten Parteien müssen durch das SCM effizient koordiniert werden. Dazu gehört den Informationsaustausch über die gesamte Lieferkette hinweg zu verwalten [Pol10]. Dies geschieht in der Regel über Enterprise Resource Planning (ERP) Systeme, deren wichtigste Vertreter auf dem Markt in Abschnitt 3.6 beschrieben sind. Problematisch ist jedoch, dass es sehr viele ERP-Lösungen gibt, wodurch ein einfacher Datenaustausch zwischen unterschiedlichen Parteien oft erschwert oder verhindert wird. Des Weiteren sind unternehmensübergreifende Prozesse für eine effiziente Lieferkette

---

<sup>3</sup><http://supply-chain.org/scor>

notwendig, um einen durchgängigen Informations- und Datenaustausch zu gewährleisten. IT-gestützte Lösungen für das Verwalten der Lieferkette werden mit eSupply Chain Management Systemen angeboten, die webbasierte Portale bereitstellen, auf die alle Parteien zugreifen können [Wan05]. Diese beschränken sich zumeist auf Kommunikationsmöglichkeiten, eine zentrale Verwaltung des Informationsverzeichnisses und einer Anbindung von eProcurement- und eSales-Lösungen, insbesondere elektronische Produktkataloge. Damit wird der Produktdatenaustausch über die gesamte Lieferkette hinweg und für alle Parteien ermöglicht. Ein automatischer Datenaustausch mit den jeweiligen ERP-Systemen der Unternehmen erfolgt jedoch meist nicht.

### **2.1.7 Das Stuttgarter Unternehmensmodell**

Die zunehmend schnelleren Änderungen, die heutzutage Produktionsunternehmen beeinflussen, führen zu immer häufigeren Anpassungen der Produkte, Geschäfts- und Produktionsprozesse, Organisation und unterstützenden Softwarelandschaften. Dieses als turbulent bezeichnete Umfeld erfordert von Unternehmen eine Wandlungsfähigkeit in jeder der eben genannten Bereiche, um sich schnell an geänderte Anforderungen anpassen zu können. Das Stuttgarter Unternehmensmodell (SUM) beschreibt, wie Produktionsunternehmen in einem turbulenten Umfeld erfolgreich operieren können [WZ09]. Die Einführung des SUM ermöglicht auf der Führungs-, Organisations-, Planungs-, Informations- und Produktionsebene die Adaptivität und Wandelbarkeit zu erhöhen und damit adäquat, in kurzer Zeit und mit geringem Aufwand auf Turbulenzen zu reagieren.

Adaptivität beschreibt hierbei die Fähigkeit eines Produktionssystems, bei Veränderungen seine Materialzufuhr und Produktionsmenge zu verändern, ohne dabei das System selbst zu ändern [ALW06, ZMV05]. Diese Änderungen sollten dabei schnell durchführbar und nur mit einem geringen finanziellen

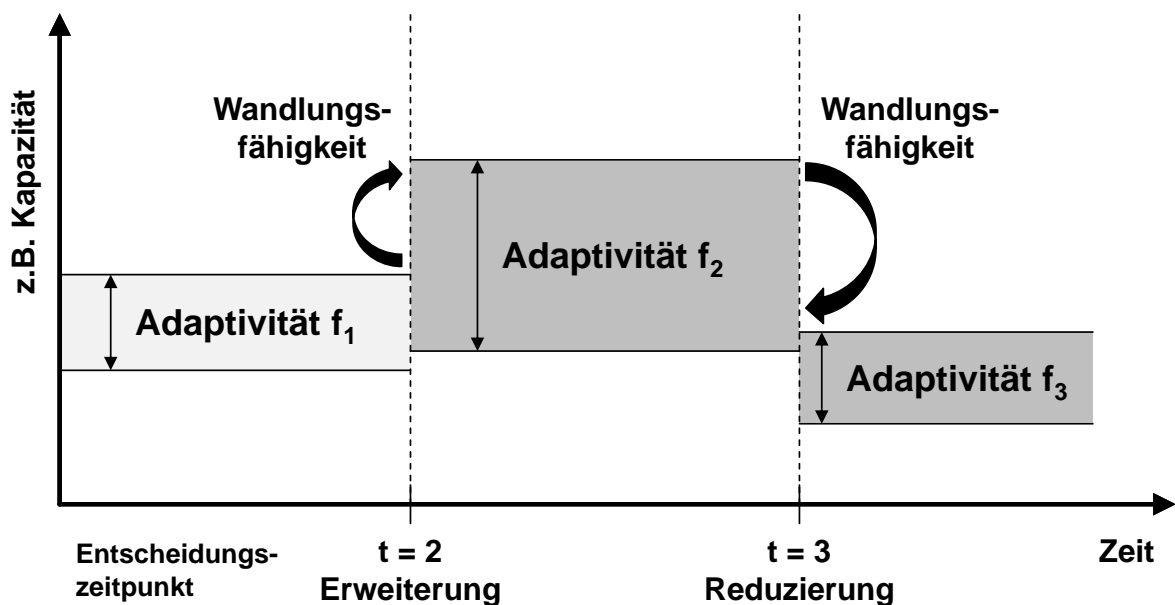


Abbildung 2.5: Unterschied zwischen Adaptivität und Wandelbarkeit (vergleiche [ZMV05])

Aufwand verbunden sein. Die Grenzen der Adaptivität sind durch einen Korridor beschrieben, der die obere und untere Grenze der Produktivität in der vorhandenen Konfiguration des Produktionssystems darstellt, abgebildet in Abbildung 2.5 [NHR<sup>+</sup>08]. Die untere Kapazitätsgrenze gibt die minimale Produktionsmenge an, bei der das Produktionssystem noch wirtschaftlich ist, die obere Kapazitätsgrenze die maximale Produktionsmenge, die mit dem Produktionssystem in der aktuellen Konfiguration erreichbar ist.

Die Wandlungsfähigkeit ist das Potential, den Korridor durch technische und organisatorische Veränderungen verschieben zu können und somit neue Grenzen, in denen sich die Flexibilität des Produktionssystems bewegt, festzulegen [RBES02, ZSMA04]. Eine Kapazitätserweiterung verschiebt die Grenzen und somit den Korridor des Produktionssystems nach oben. Eine Kapazitätsreduzierung hingegen verschiebt diese Grenze nach unten, wie in Abbildung 2.5 bei den Entscheidungszeitpunkten  $t=2$  und  $t=3$  zu sehen ist [ZMV05]. Ein Produktionssystem muss bereits zur Planungszeit entsprechend konzipiert werden, damit es mit geringem Aufwand erweitert und verkleinert werden kann und dadurch den Anforderungen der Wandlungs-

fähigkeit entspricht. Aus diesem Grund besitzen wandlungsfähige Produktionssysteme keine echten Grenzen, da der Korridor durch technische und organisatorische Änderungen jederzeit anpassbar ist [CHN02]. Im Vergleich zu nicht wandlungsfähigen Systemen fallen bei einer Änderung zwar genauso Kosten und Zeitaufwand an, diese sind jedoch deutlich geringer, nachdem das Produktionssystem bereits für einfache und schnelle Änderungen ausgelegt ist und Teile der bestehenden Anlage weiterhin genutzt werden können [NHR<sup>+</sup>08]. Die Wandlungsfähigkeit dient nicht nur der kurzfristigen Reaktion auf Turbulenzen, es ist auch ein systematischer und strategischer Ansatz, um auf langfristige Veränderungen und zukünftige Entwicklungen einzugehen [NFPFW09].

Die Fähigkeit von Produktionsunternehmen aufgrund ihrer Adaptivität und Wandlungsfähigkeit auf allen Ebenen schnell auf Einflüsse von innen und außen reagieren zu können führt zu einer verbesserten Wettbewerbsfähigkeit. Um dies zu erreichen werden auf der Produktionsebene modulare Produktionssysteme eingesetzt, die durch standardisierte Schnittstellen einfach und schnell erweitert oder verkleinert werden können und somit die Wandlungsfähigkeit ermöglichen. Die Mobilität der Module erhöht zudem die schnelle Rekonfiguration der Produktionslinie [EHM03]. Auf der organisatorischen Ebene können selbstregulierende bzw. teilautonome Arbeitsgruppen eingerichtet werden. Eine Arbeitsgruppe ist hierbei verantwortlich für die Erstellung eines Produktes, Teilproduktes oder einer Dienstleistung und organisiert sich selbstständig, indem Aufgaben flexibel einem Mitarbeitenden zugewiesen werden [WZ09].

## 2.2 Informationstechnische Grundlagen

Der zweite Teil dieses Kapitels beschreibt Methoden und entsprechende Technologien zur Integration von Anwendungen. Allgemein kann man drei

Ebenen der Integration unterscheiden: Daten-, Funktions- und Oberflächenintegration. Über die Integration dieser drei Ebenen kann darüber hinaus eine Prozessintegration erreicht werden. Erste Integrationsversuche beschränken sich auf die Daten- oder Anwendungsintegration, indem gemeinsame Datenbanken oder Dateien für den Datenaustausch eingesetzt werden bzw. Remote Procedure Calls (RPCs) für den entfernten Prozeduraufruf.

Das Enterprise Application Integration (EAI) erlaubt sowohl Daten, als auch Funktionen zu integrieren und ermöglicht damit eine systematischere Anwendungsintegration im Vergleich zu früheren Ansätzen und wird in Abschnitt 2.2.1 vorgestellt. Die Weiterentwicklung des EAI-Konzeptes stellt die Serviceorientierte Architektur (SOA) dar. Die Idee hinter SOA ist, durch lose Kopplung von Services eine hohe Flexibilität der Integrationsumgebung zu erreichen. SOA und die zugehörigen Technologien und Standards sind in Abschnitt 2.2.2 beschrieben. Die Nachrichtenvermittlung, auf der das SOA-Konzept beruht, werden in Abschnitt 2.2.3 detailliert beschrieben. In Abschnitt 2.2.4 ist abschließend der ESB beschrieben, der eine allgemeingültige Middleware zur Integration einer Serviceumgebung auf Basis von Nachrichtenvermittlung darstellt.

### **2.2.1 Enterprise Application Integration**

Die ersten Bemühungen die Vielzahl an Anwendungen in Unternehmen systematisch zu integrieren, kam mit dem Konzept des Enterprise Application Integration (EAI) auf. Das Ziel von EAI ist es, mittels einer zentralen Integrationsplattform Daten und Prozesse zwischen den isolierten Anwendungen zu teilen [Lin00]. Bisher genutzte Lösungen zur Integration von Anwendungen sind Message Queuing oder RPCs, die jedoch ohne eine zentrale Integrationsplattform zu Punkt-zu-Punkt Verbindungen führen. Dies führt bei zunehmender Anzahl an Anwendungen schnell zu einer sehr komplexen Integrationsschicht.

EAI definiert eine Lösung, um eine Vielzahl an Anwendungen in einer einheitlichen Form auf der Daten-, Anwendungs- und Prozessebene zu integrieren, aber verzichtet auf eine konkrete Beschreibung, wie die Integration umgesetzt werden kann [CHKT05, HW03]. Typischerweise wird ein sogenannter Message Broker eingesetzt, der die zentrale Integrationsmiddleware in einer EAI-Lösung bildet. Aus diesem Grund wird dieses Architekturmuster auch „Hub-and-Spoke“ genannt, da der zentrale Message Broker von den Anwendungen wie Speichen in einem Rad umgeben ist. Ein Problem dabei ist, dass EAI-Systeme nur wenige oder keine Abstraktion gegenüber den integrierten Anwendungen bieten, weshalb sie nur wenig Vorteile gegenüber dem Punkt-zu-Punkt Integrationsmuster bringen [DSRR08]. Die Anbindung der Anwendungen basiert dabei weiterhin auf proprietären Schnittstellen, weshalb die Komplexität nur ein Stück weit reduziert werden kann [HW03]. Darüber hinaus sind EAI-Lösungen auf Abteilungen oder Anwendungsdomänen beschränkt, weshalb sich diese Integrationslösung auf wenige Softwaresysteme beschränkt [DSRR08].

### 2.2.2 Serviceorientierte Architektur

Die Serviceorientierte Architektur (SOA, engl.: service-oriented architecture) ist ein abstraktes Konzept und adressiert Probleme von EAI-Lösungen, wie z. B. das Integrieren von heterogenen Anwendungen oder der hohe Aufwand zur Wartung von Schnittstellen. Eine SOA bietet auf Basis von verteilten und wiederverwendbaren Services bzw. Diensten, die eine lose Kopplung eingehen können, eine adaptive Integration von heterogenen Anwendungen. Zudem sind die Services selbstbeschreibend, plattformunabhängig und auffindbar [Erl05].

Dies erlaubt Services in einer SOA anzubieten, zu suchen und zu nutzen. Dieses Vorgehen wird im sogenannten SOA-Dreieck beschrieben, dessen Ablauf in Abbildung 2.6 dargestellt ist [Mel10]. Zum dynamischen Auffinden eines

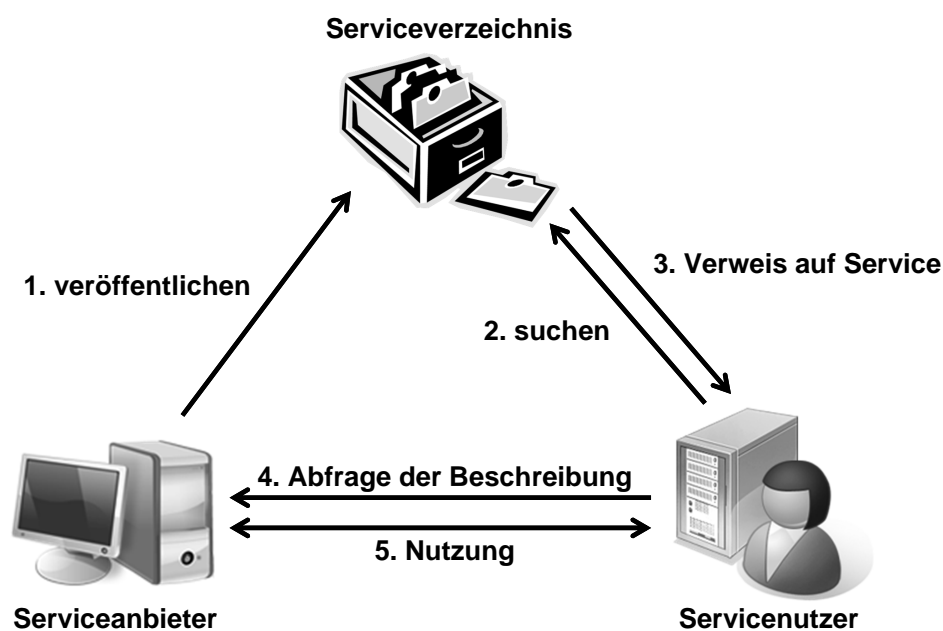


Abbildung 2.6: Das magische Dreieck einer SOA (vergleiche [Mel10])

Services sind drei Parteien notwendig, der Servicenutzer, der Serviceanbieter und das Serviceverzeichnis. Das Serviceverzeichnis speichert alle verfügbaren Services und dazugehörige Metadaten und wird in Abschnitt 2.2.2.4 genauer beschrieben. Der Serviceanbieter veröffentlicht seine Services im Serviceverzeichnis und stellt Ressourcen bereit, um die Services auszuführen. Der Servicenutzer sucht Services, die seinen Anforderungen entsprechen, im Serviceverzeichnis. Anschließend wählt er einen Service aus, der die von ihm gewünschte Qualität liefert und ruft diesen beim Serviceanbieter auf. Darüber hinaus können auch Kosten für die Servicenutzung definiert werden, die ein weiteres Entscheidungskriterium des Servicenutzers für die Auswahl eines Services bietet.

Eine weitere wichtige Eigenschaft einer SOA ist die Unabhängigkeit von der jeweiligen Implementierung. D. h. die Serviceschnittstelle ist getrennt von der Implementierung des entsprechenden Services, weshalb die Implementierung eines Services unabhängig von seiner Schnittstelle geändert werden kann [Mel10]. Melzer definiert eine SOA folgendermaßen:

„Unter einer SOA versteht man eine Systemarchitektur, die vielfältige, verschiedene und eventuell inkompatible Methoden oder Applikationen als wiederverwendbare und offen zugreifbare Dienste repräsentiert und dadurch eine plattform- und sprachenunabhängige Nutzung und Wiederverwendung ermöglicht.“

Der Begriff Dienste wird dabei in dieser Arbeit Synonym zum Begriff Service verwendet. Im Gegensatz zu EAI-Lösungen erlaubt eine SOA die Integration über Abteilungs- und Domänengrenzen hinweg, wobei die Wahl der Technologien zur Umsetzung einer SOA und Abstraktionsebenen entscheidend ist. Die Umsetzung erfolgt typischerweise durch Web Services, die im folgenden Abschnitt 2.2.2.1 beschrieben sind. Als Integrationsmiddleware kommt meistens ein ESB zum Einsatz, der als abstraktes Integrationskonzept die lose Kopplung von Services ermöglicht und in Abschnitt 2.2.4 erklärt ist.

### 2.2.2.1 Web Service

Die Web Service (WS) Technologie ist standardisiert und plattformunabhängig und unterstützt den Informationsaustausch zwischen beliebigen Anwendungen, indem diese Serviceschnittstellen bereit stellen. Web Services kommen in der Regel bei der Umsetzung einer SOA zum Einsatz [WCL<sup>+</sup>05].

Das World Wide Web Consortium (W3C) ist verantwortlich für die Standardisierung von Web Services und den dazugehörigen Spezifikationen Web Services Description Language (WSDL) zur Schnittstellenbeschreibung und SOAP als Transportprotokoll [W3C04]. Die Beschreibung einer Web Service Schnittstelle basiert auf der Extensible Markup Language (XML), die eine offene maschinenlesbare Beschreibungssprache ist. Das W3C definiert Web Services in Zusammenhang mit den zugehörigen Standards folgendermaßen:



„A Web service is a software system designed to support interoperable machine-to-machine interaction over a network. It has an interface described in a machine-processable format (specifically WSDL). Other systems interact with the Web service in a manner prescribed by its description using SOAP-messages, typically conveyed using HTTP with an XML serialization in conjunction with other Web-related standards.“<sup>4</sup>

Neben den bisher genannten Standards besteht die Web Service Architektur aus weiteren Standards, die die Standardisierungsorganisationen *W3C* und *Organization for the Advancement of Structured Information Standards (OASIS)* veröffentlichen [WCL<sup>+</sup>05]. Auf der Nachrichtenebene kommt das bereits erwähnte SOAP zum Einsatz [W3C07a], aber auch Java Message Service (JMS) und weitere Verfahren, um Web Service Nachrichten zwischen Schnittstellen zu übertragen. Diese Schnittstellen werden durch WSDL auf der Beschreibungsebene genau festgelegt, um Endpunkte, Funktionen und deren Parameter zu veröffentlichen [W3C07b].

Weitere Details wie z. B. Servicequalität und Sicherheitsanforderungen können mit WS-Policy beschrieben werden [W3C07c]. Auf der Ebene der Servicequalität sind WS-ReliableMessaging, WS-Security, WS-AtomicTransaction und WS-BusinessActivity spezifiziert. WS-ReliableMessaging sorgt für eine garantierte Zustellung von Nachrichten am Zielservice, indem bei Nachrichtenverlust eine Nachricht erneut gesendet wird [OAS09c]. WS-Security dient zur sicheren Übertragung von Nachrichten, indem z. B. Verschlüsselung eingesetzt wird [OAS06]. WS-AtomicTransaction und WS-BusinessActivity erlauben Transaktionen über Web Services auszuführen, wobei WS-BusinessActivity auch langlaufende Transaktionen unterstützt [OAS09a, OAS09b]. Die oberste Ebene der Web Service Architektur definiert zusammengesetzte Services, die mit Business Process Execution Language (BPEL) oder WS-Choreography beschrieben werden [OAS07, W3C05]. Über

<sup>4</sup><http://www.w3.org/TR/2004/NOTE-ws-gloss-20040211/#webservice>

alle Ebenen hinweg wird Universal Description, Discovery and Integration (UDDI) als Serviceverzeichnis spezifiziert, um Servicebeschreibungen zu speichern, die von Servicenutzenden abgefragt werden können [OAS05]. All diese Web Service Spezifikationen werden oft mit *WS-\** zusammengefasst.

Im Folgenden sind mit WSDL, SOAP, dem Serviceverzeichnis UDDI und BPEL die wichtigsten Standards für die Umsetzung einer SOA detaillierter beschrieben.

### **2.2.2.2 WSDL**

Die Web Services Description Language (WSDL) ist ein auf XML basierender Standard zur Beschreibung von Web Service Schnittstellen [WCL<sup>+</sup>05]. Die Trennung von Schnittstelle und Implementierung ermöglicht die lose Koppung von Services umzusetzen, die in der SOA gefordert wird. Damit wird die Adaptivität erhöht, wodurch Änderungen einfacher und unter Umständen voneinander unabhängig durchführbar sind.

Eine WSDL Beschreibung besteht aus zwei Teilen, einem abstrakten und einem konkreten Teil [W3C07b]. Der abstrakte Teil beschreibt das Verhalten des Web Services, indem Typen und Nachrichten definiert werden. In der aktuellen Version WSDL 2.0 werden zusätzlich Porttypen bzw. Schnittstellen hinzugenommen. Der konkrete Teil enthält Informationen, über Bindings und Services, die der Web Service bereitstellt. Zusätzlich kommt die Beschreibung von Ports bzw. Endpunkten in der aktuellen WSDL Spezifikation hinzu.

### **2.2.2.3 SOAP**

SOAP stand früher als Abkürzung für Simple Object Access Protocol und wird heute als eigenständiger Name verwendet. SOAP spezifiziert das Datenformat und die Regeln, die die Verarbeitung einer SOAP-Nachricht entlang

eines Nachrichtenpfades beschreiben [WCL<sup>+</sup>05]. Eine SOAP-Nachricht besteht aus einem SOAP-Envelope, der keinen oder mehrere SOAP-Header enthalten kann, sowie einem SOAP-Body [W3C07a]. Der SOAP-Header enthält Informationen für die Empfänger über die Verarbeitung der Nachricht. Der SOAP-Body hingegen beinhaltet die Nutzdaten bzw. Geschäftsinformationen, die in XML beschrieben sind. Die Nutzdaten können z. B. einen Serviceaufruf mit entsprechenden Eingabewerten enthalten. Dadurch wird ein Service aufgerufen, der die Eingabewerte der Nachricht verarbeitet.

Der SOAP Standard erlaubt eigentlich nur Zeichen, die auch in XML erlaubt sind, zu versenden. Über ein SOAP-Attachment können aber trotzdem Binärdaten innerhalb einer SOAP-Nachricht versendet werden, indem eine Konvertierung von Binärdaten in Zeichen mittels einer Base64-Codierung durchgeführt wird.

Das Versenden einer SOAP-Nachricht kann direkt oder innerhalb eines anderen Nachrichtenformates erfolgen, z. B. über standardisierte Protokolle, wie Hypertext Transfer Protocol (HTTP), Simple Mail Transfer Protocol (SMTP) oder Transmission Control Protocol (TCP), aber auch über proprietäre Transportprotokolle, wie IBM WebSphereMQ Nachricht. Dies geschieht über ein SOAP-Binding, das die Regeln beschreibt, wie die SOAP-Nachricht in den Nutzdaten des übergeordneten Protokolls eingebettet ist.

SOAP dient als Transportprotokoll für Web Services und bildet ein Rahmenwerk für den Datenaustausch in heterogenen Umgebungen, in dem unterschiedliche Protokolle zum Einsatz kommen. Dies erlaubt eine hohe Flexibilität für den Einsatz von Web Services, die für die Integration von heterogenen IT-Infrastrukturen notwendig ist.

#### 2.2.2.4 Serviceverzeichnis

Die zentrale Speicherung aller Servicebeschreibungen erfolgt nach dem Gedanken des SOA-Dreiecks in einem Serviceverzeichnis. Für diese Art der

Serviceverzeichnisse wurde von OASIS Universal Description, Discovery and Integration (UDDI) entwickelt, das eine standardisierte Verzeichnisstruktur für die Verwaltung von Web Service-Metadaten spezifiziert [OAS05]. Durch die zentrale Speicherung der Services und ihrer Beschreibung können diese zur Laufzeit gefunden und in verschiedenen Prozessen eingebunden werden, was die lose Kopplung und Wiederverwendbarkeit von Services ermöglicht [WCL<sup>+</sup>05].

UDDI ist vergleichbar mit einem Telefonbuch, in dem Informationen über Telefonnummern und deren Besitzer gelistet sind. Das Telefonbuch wird von einem Anrufer genutzt, um die Nummer einer Person zu bestimmen, die er anschließend anruft. UDDI besteht aus vier Teilen, die im Folgenden kurz beschrieben sind [Mel10]:

1. White Pages: Dieses Verzeichnis speichert Informationen über Serviceanbieter.
2. Yellow Pages: Diese Seiten geben Informationen über die Branche wieder, zu der der Service gehört.
3. Green Pages: Dieses Verzeichnis speichert Informationen über einzelne Services.
4. Service Type Registration: Die Serviceinformationen, die in den Green Pages in menschenlesbarer Form vorgehalten werden, sind im Service Type Registration in dieser Form gespeichert.

Microsoft, IBM und SAP stellten mit dem UDDI Business Registry (UBR) ein öffentliches Serviceverzeichnis zur Verfügung, in dem jeder Serviceanbieter seine Servicebeschreibungen veröffentlichen konnte und Servicenutzende diese im Anschluss suchen. Im Jahr 2005 verkündeten die drei Unternehmen, ihre öffentlichen UBRs abzuschalten<sup>5</sup>. In Folge dessen kommt UDDI nur

---

<sup>5</sup><http://www.infoworld.com/d/architecture/microsoft-ibm-sap-discontinue-uddi-registry-effort-777>

noch innerhalb von Firmennetzen zum Einsatz, so dass es keine öffentlichen Serviceverzeichnisse von ähnlichem Umfang wie UBR mehr gibt.

### 2.2.2.5 Modellierungssprachen für Geschäftsprozesse

Einzelne, feingranulare Services können mit Hilfe der Prozessmodellierungssprache BPEL flexibel zu einem Workflow zusammengesetzt werden [OAS07]. Dies erlaubt nicht nur, die Reihenfolge von einzelnen Serviceaufrufen festzulegen, sondern beschreibt auch das Zusammenspiel der einzelnen Parameter zwischen den Services [WCL<sup>+</sup>05]. Ein BPEL Workflow kann als Service zur Verfügung stehen, was zu höherwertigen bzw. aggregierten Services führt, die dann wie andere Services aufgerufen werden können. Diese Tatsache, dass ein Service aus kleineren Services zusammengesetzt ist, die wiederum aus noch feingranulareren Services bestehen, usw., nennt man rekursive Servicekomposition.

BPEL ist nicht nur eine Modellierungssprache für Workflows, diese können auch von einer entsprechenden Workflow Engine interpretiert und ausgeführt werden. Die Ausführung geschieht, indem eine Workflow Engine die einzelnen Services in der modellierten Reihenfolge automatisch aufruft.

Business Process Model and Notation (BPMN) ist eine weitere Modellierungssprache, um Prozesse digital zu beschreiben. BPMN hat gegenüber BPEL eine größere Ausdrucksmächtigkeit, war aber in der ersten Version des Standards nicht ausführbar [OMG09a]. Dies änderte sich mit der Version BPMN 2.0, in der u. a. eine operationale Semantik und eine Modellbeschreibung in XML hinzugefügt wurde, um das Modell direkt ausführen zu können<sup>6</sup>. Es ist aber auch möglich, ein Modell, beschrieben in BPMN, in einen BPEL Workflow zu überführen, sofern die modellierten Eigenschaften in BPEL abbildbar sind [OMG11].

---

<sup>6</sup><http://www.bpmn.org/>

### 2.2.3 Nachrichtenvermittlung

Um Anwendungen lose zu koppeln und somit die Unabhängigkeit der Anwendungen in der integrierten IT-Landschaft zu erhalten, hat sich der Einsatz der Nachrichtenvermittlung (engl.: messaging) bewährt. Bisherige Möglichkeiten zur Integration, wie Dateitransfer oder geteilte Datenbanken, können nur Daten und keine Funktionalitäten berücksichtigen. Der entfernte Prozeduraufruf hingegen verbindet Anwendungen bei der Integration starr. Die Nachrichtenvermittlung bietet im Gegensatz dazu die Möglichkeit, sowohl Daten, als auch Funktionalität zu teilen und gleichzeitig die Unabhängigkeit der Anwendungen zu erhalten [HW03]. Dies wird durch die Trennung von Anwendungsentwicklung und Integrationsentscheidungen erreicht. Der asynchrone Nachrichtenaustausch erlaubt es, dass beim Versenden der Nachricht der Empfänger nicht empfangsbereit sein muss oder umgekehrt.

#### 2.2.3.1 Einheitliches Nachrichtenformat

Verschiedene Anwendungen besitzen in der Regel unterschiedliche Datenformate. Bei der Integration von Anwendungen müssen diese Datenformate deshalb paarweise mithilfe einer Übersetzerkomponente ineinander umgewandelt werden. Bei wachsender Anzahl an integrierten Anwendungen steigt deshalb die Anzahl der Übersetzer exponentiell, wodurch die Integrationslandschaft schnell nicht mehr verwaltbar ist. Insbesondere die Integration von neuen Anwendungen erfordert einen Übersetzer für jede Anwendung, mit der kommuniziert werden soll. Um dies zu verhindern und die Abhängigkeiten zwischen den proprietären Datenformaten von Anwendungen aufzulösen, kann ein einheitliches Nachrichtenformat (engl.: canonical data model) eingeführt werden [HW03]. Durch dieses Nachrichtenformat entsteht eine neue Abstraktionsschicht zwischen den individuellen Datenformaten der Anwendungen. Damit ist für jede Anwendung nur ein Übersetzerpaar vom

eigenen proprietären Datenformat in das einheitliche Nachrichtenformat und zurück erforderlich. Dies reduziert die Anzahl an benötigten Übersetzern auf jeweils einen pro Anwendung in jede Richtung und steigert somit die Wartbarkeit. Beim Hinzufügen einer neuen Anwendung ist unabhängig von der Anzahl der bereits integrierten Anwendungen nur ein neues Übersetzerpaar notwendig.

### 2.2.3.2 Content-based Router

Die dynamische Vermittlung von Nachrichten geschieht mittels eines Content-based Router (CBR) [HW03]. Der CBR bestimmt zu diesem Zweck dynamisch das Ziel einer Nachricht. Dieses erhält er, indem er den Inhalt der Nachricht untersucht und abhängig vom Inhalt und vorgegebenen Regeln das Zielsystem ableitet. Durch das Serviceverzeichnis kann im Anschluss der Endpunkt des Zielsystems bzw. der Zielsysteme bestimmt werden, an die schließlich die Nachricht weiter geleitet wird. Somit können Sender und Empfänger voneinander entkoppelt werden, wodurch u. a. die Auswirkungen von Änderungen im Gesamtsystem reduziert und dadurch die Wartbarkeit verbessert wird.

### 2.2.3.3 Message Queues

Die Robustheit einer IT-Infrastruktur kann mithilfe von Nachrichtenkanälen, auch Message Queue (MQ) genannt, verbessert werden. Dies geschieht, indem die MQs das Versenden einer Nachricht, von einer Anwendung zur Anderen, zeitlich entkoppeln. Eine Nachricht wird vom Quellsystem an die gewünschte MQ geschickt, bei der die Nachricht zwischengespeichert wird, bis das Zielsystem bereit ist die Nachricht zu empfangen [HW03]. Der Nachrichtenkanal arbeitet dabei üblicherweise nach dem First In - First Out (FIFO) Prinzip, d. h. die Nachricht, die zuerst in die MQ gelegt wird, wird als erstes

aus dieser entnommen. Im Vergleich zum direkten Versenden von Nachrichten hat dies verschiedene Vorteile. Zum einen werden die Anwendungen beim Versenden oder Empfangen einer Nachricht nicht blockiert, wenn sie auf die andere Anwendung wartet, da die MQ als Nachrichtenpuffer fungiert. Zum anderen können Nachrichten in der MQ zwischengespeichert werden, wenn die Zielanwendung kurzfristig nicht verfügbar ist. Dadurch kann der Nachrichtenabsender dennoch weiter arbeiten. Darüber hinaus kann eine Nachricht über eine MQ nicht nur an eine Anwendung gesendet werden, sondern kann mehrere Empfänger haben.

Die Warteschlange unterstützt somit die asynchrone Kommunikation zwischen Anwendungen und erhöht die Robustheit der Integrationslandschaft, da Softwareprobleme oder Netzwerkausfälle für eine kurze Zeit überbrückt werden können.

## 2.2.4 Enterprise Service Bus

Der Enterprise Service Bus (ESB) ist ein Ansatz, um ein lose gekoppeltes und weit verteiltes Netzwerk zu integrieren [Cha04]. Ein ESB ist eine auf Standards basierende Integrationsplattform, die Nachrichtenvermittlung, Web Services, Datentransformation und intelligentes Routen kombiniert. Dadurch soll eine große Anzahl an verschiedenen Anwendungen im Unternehmen sowie dessen Partnern mit Transaktionsintegrität, zuverlässig verbunden und koordiniert werden. Die Kombination aus Nachrichtenvermittlung, Web Services und intelligentem Router ermöglicht eine lose Kopplung der Anwendungen. Das einheitliche Nachrichtenformat des ESB trägt zur Unabhängigkeit der Integrationsmiddleware von den proprietären Anwendungen bei und verringert so die Anzahl an Übersetzungsservices deutlich. Das einheitliche Nachrichtenformat und die Datenübersetzungsservices ermöglichen damit die Integration einer heterogenen Systemlandschaft, was insgesamt zu einer Komplexitätsreduktion der IT-Landschaft führt.



Darüber hinaus bringt der ESB auch auf heute etablierten Standards basierende Implementierungen mit, um Anforderungen der Industrie im Bereich der Sicherheit, Zuverlässigkeit, Transaktionsverwaltung und Geschäftsprozessmanagement zu unterstützen.

Ein CBR verarbeitet dabei alle Nachrichten die am ESB ankommen und verhält sich dabei transparent für den Sender und Empfänger, die nichts von dessen Existenz wissen müssen [HW03]. Insbesondere die Tatsache, dass alle Endpunkte vom CBR bzw. dessen zugehörigem Serviceverzeichnis an einer zentralen Stelle zusammengefasst sind, vereinfacht das Ändern von Endpunkten in einer SOA. Die meisten Implementierungen von ESBs kommen bereits mit einer Workflow Engine, so dass BPEL Workflows, die in der ESB-Umgebung installiert werden, direkt ausführbar sind.

Der ESB bildet die verbreitetste Middlewarelösung, wenn es heute darum geht eine Softwarelandschaft nach den Prinzipien einer SOA zu integrieren. Im Vergleich zum klassischen EAI kommt die Integration über einen ESB ohne die starren Verbindungen und Beschränkungen auf die zu Grunde liegenden Nachrichtenvermittlungsverfahren und das physische lokale Netzwerk aus.

## 2.3 Zusammenfassung

In diesem Kapitel wurden die Grundlagen für die vorliegende Arbeit beschrieben. Der erste Teil des Kapitels ist auf unterschiedliche Managementansätze für Produktionsunternehmen eingegangen. Die frühen Versuche zur Integration in Produktionsumgebungen basierten auf CIM, das nicht nur die technischen CAx-Systeme, sondern auch betriebswirtschaftliche PPS-Systeme in einer Gesamtlösung verbindet. Die meisten CIM-Projekte scheiterten aber an der zu hohen Komplexität. Die wenigen erfolgreichen CIM-Lösungen boten durch die starre Integration aber kaum Möglichkeiten für Weiterentwicklungen und waren nur sehr schwer zu warten. Heute ist

ein Trend von PDM-Systemen hin zu PLM als ganzheitlicheren Ansatz zur Produktverwaltung zu erkennen der ein Produkt nicht nur zur Entwicklungszeit, sondern von der ersten Idee bis hin zur Entsorgung betrachtet und verwaltet. In Bezug auf die Fabrikplanung wird auf Werkzeuge der Digitalen Fabrik zurückgegriffen, die den Produktdaten die entsprechenden Ressourcen und Produktionsprozesse zuordnet. Das FLM erweitert die Digitale Fabrik und führt eine ganzheitliche Betrachtung der Fabrik nach dem Vorbild des PLM ein. In Produktionsunternehmen wird FLM derzeit jedoch noch nicht praktiziert, da unter anderem eine durchgehende Softwareunterstützung für alle Aufgaben des Fabriklebenszyklus fehlt. Das SCM dient zur Verwaltung von Lieferanten und Kundenbeziehungen. Ziel ist die Lieferkette zu optimieren, eine just-in-time Lieferung von Material und Zulieferteilen zu erreichen, wodurch Lagerbestände reduziert werden können. Dies gilt auch bei der Vermarktung, die versucht die Lagerbestände zu reduzieren, indem die Fertigprodukte möglichst schnell abgesetzt werden. Die Bedeutung der Adaptivität und Wandlungsfähigkeit wird durch das SUM verdeutlicht, das ein turbulentes Umfeld beschreibt, in dem Produktionsunternehmen heute agieren müssen. Das Ziel des SUM ist Produktionsunternehmen auf allen Ebenen wandlungsfähig zu machen, damit diese schnell auf innere und äußere Einflüsse reagieren können und somit ihre Ressourcen zu jedem Zeitpunkt möglichst optimal ausschöpfen.

Die Notwendigkeit in Unternehmen die wachsende Anzahl an Softwaresystemen zu integrieren führte zum EAI-Konzept. Durch die Einführung eines zentralen Message Brokers wurde die Komplexität der IT-Landschaften ein Stück weit reduziert. Grundlegende Probleme wie die Inflexibilität, auf Abteilungen begrenzte Integration und proprietäre Schnittstellen, bestehen jedoch weiterhin. Das abstrakte Konzept der SOA erlaubt, auf Basis von verteilten und wiederverwendbaren Services und deren losen Kopplung, eine adaptive Integration von heterogenen Anwendungen. Die Services sind selbstbeschreibend, plattformunabhängig und auffindbar, weshalb auch Altsysteme über eine SOA integriert werden können. Standardisierte Technologien wie Web

Services, WSDL, SOAP und UDDI sowie BPEL und BPMN erlauben die Umsetzung einer SOA und unterstützen die flexible Integration von Anwendungen und reduzieren die Komplexität von IT-Landschaften bei gleichzeitig verbesserter Wartbarkeit. Das einheitliche Nachrichtenformat bildet im Konzept der Nachrichtenvermittlung eine neue Abstraktionsschicht zwischen den proprietären Datenformaten der integrierten Anwendungen. Der CBR sorgt für eine dynamische Nachrichtenvermittlung und löst damit Abhängigkeiten zwischen Anwendungen. Die Nachrichtenwarteschlange verbessert die Robustheit einer IT-Infrastruktur, indem Nachrichten gepuffert werden, bis das Zielsystem bereit ist diese zu empfangen. Damit können auftretende Software- oder Netzwerkfehler kurzfristig überbrückt werden, ohne dass eine Anwendung blockiert wird. Der ESB bildet die zentrale Integrationsmiddleware in einer SOA, indem er als plattformunabhängige Komponente die dynamische Integration von Web Services auf Basis der zuvor genannten Technologien unterstützt.



---

## KAPITEL 3

# IT-Lösungen für den Produktlebenszyklus und verwandte Arbeiten

---

Dieses Kapitel führt in das Thema des Produktlebenszyklus ein und stellt verschiedene verwandte Arbeiten vor, die sich mit der Integration von Anwendungen im Produktlebenszyklus beschäftigen. Insbesondere die Umsetzung der SOA zusammen mit der Web Service Technologie und dem ESB Konzept wird betrachtet, aber auch andere Umsetzungen zur Integration von einzelnen Domänen des Produktlebenszyklus. Zu Beginn des Kapitels wird auf die IT-Unterstützung des PLM eingegangen und ein erweitertes Phasenmodell präsentiert, das im Rahmen der Arbeit für die Entwicklung der IT-Architektur zu Grunde liegt.

Mit *PLM Services 2.0* existiert ein Standard für die Schnittstellenbeschreibung im Produktlebenszyklus zusammen mit einem plattformunabhängigen und einem plattformabhängigen Datenformat. Darauf folgend werden Arbeiten vorgestellt, die die Integration des Produktlebenszyklus über einen ESB beschreiben.

Mit CHAMPAGNE wird eine Plattform zur flexiblen Integration von Anwendungen in der Produktionsplanung vorgestellt, die es erlaubt Änderungen, die

in einer Anwendung erfolgen, sofort in anderen Anwendungen zu propagieren, um die Konsistenz der Daten sicherzustellen. Eine domänenspezifische Integrationsumgebung, genannt Engineering Service Bus (EngBus), wird von der Firma Daimler entwickelt. Dieser basiert auf einem ESB und sorgt auf Basis der SOA für eine flexible Integration der Anwendungen der Produktentwicklung. Der Manufacturing Service Bus (MSB), als phasenspezifische Integrationsumgebung für die Produktion, ist angepasst an das schnelle und flexible Verarbeiten von Events, die durch Maschinen oder Anwendungen erzeugt werden.

Die größten und am verbreitetsten Softwarelösungen zur Unterstützung des PLM, FLM und SCM werden zum Abschluss des Kapitels beschrieben. Die Integrationsmöglichkeiten und deren Adaptivität stehen dabei im Vordergrund.

### **3.1 Phasenmodell des erweiterten Produktlebenszyklus**

Produktionsunternehmen setzen zunehmend PLM ein, um ihre Produkte mit diesem durchgängigen, einheitlichen und ganzheitlichen Managementkonzept effizient zu verwalten. PLM besteht aus mehreren Teilen, wie dem Produkt selbst als zentraler und wichtigster Teil, zudem die Organisation und ihre Mitarbeitenden, Prozesse und Informationssysteme [Sta04]. Diese und weitere Teile müssen für sich, aber auch untereinander richtig verwaltet werden, um Produkte herzustellen, die alle Anforderungen erfüllen, die der Markt fordert. Gleichzeitig müssen gesetzliche Regelungen Berücksichtigung finden.

Der breite Einsatz von PLM erfordert aber den zunehmenden Anforderungen nach mehr Adaptivität und Wandlungsfähigkeit gerecht zu werden. Aus

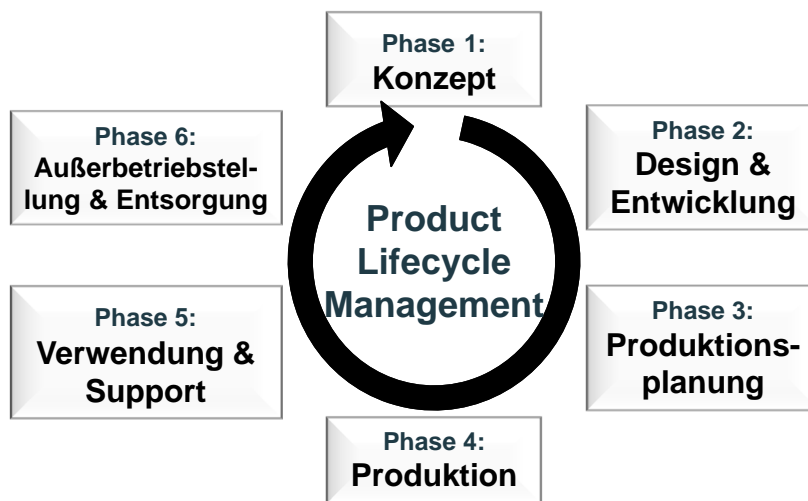


Abbildung 3.1: Erweiterter Produktlebenszyklus mit sechs Phasen (vergleiche [SKRM13])

diesem Grund beschäftigt sich diese Arbeit mit einer Verbesserung der Adaptivität und Wandlungsfähigkeit der unterstützenden IT-Landschaft, die einen immer wichtigeren Teil des PLM-Konzeptes darstellt.

Der Produktlebenszyklus wird in der Literatur typischerweise in Phasen aufgeteilt, um die unterschiedlichen Abschnitte, die ein Produkt durchläuft, hervorzuheben. Diese Arbeit bezieht sich auf den in Abbildung 3.1 dargestellten Lebenszyklus, der in sechs Phasen aufgeteilt ist. Mit der Produktionsplanung enthält dieser Zyklus zwar eine Phase, die typischerweise nicht dem Produktlebenszyklus, sondern dem Fabriklebenszyklus zugehörig ist. In der Praxis trennen die Softwarehersteller in diesem Fall aber nicht konsequent zwischen den beiden Lebenszyklen. Die PLM-Softwarelösungen unterstützen neben den Kernaufgaben der Produktentwicklung und Produktion im Produktlebenszyklus meist auch die Aufgaben der Digitalen Fabrik, also der Produktionsplanung. Aus diesem Grund wird im Folgenden der Produktlebenszyklus zusammen mit der Produktionsplanung als erweiterter Produktlebenszyklus betrachtet, um die im Rahmen dieser Arbeit entwickelte IT-Architektur besser mit den PLM-Softwarelösungen vergleichen zu können. Der Begriff Produktionsplanung wird in dieser Arbeit zur Beschreibung aller organisatorischen und technischen Vorgänge verwendet. Dieser Begriff

beschreibt die Definition, Planung, Simulation und Umsetzung der Fertigung und deckt somit alle Aufgaben der Digitalen Fabrik ab.

Dieser erweiterte Produktlebenszyklus bietet darüber hinaus den Vorteil, dass er alle Aufgaben der Digitalen Fabrik mit beinhaltet. Die Abstimmungen und Prozesse, die zwischen der Produktentwicklung und Produktionsplanung ablaufen, können so besser beschrieben werden. Das Simultaneous Engineering betrachtet genau diese Koordination zwischen Produktentwicklung und Produktionsplanung, um Aufgaben besser zu parallelisieren und dadurch die Gesamtprozesse schneller auszuführen.

Die typischen Aufgaben der einzelnen Phasen innerhalb des erweiterten Produktlebenszyklus sind in der folgenden Tabelle 3.1 genauer beschrieben.

## **3.2 Serviceorientierte Architektur für das Product Lifecycle Management**

Dieser Abschnitt beschreibt den Standard *PLM Services 2.0*, der Schnittstellen für die Integration im Produktlebenszyklus und ein plattformabhängiges und plattformunabhängiges Modell für den Datenaustausch definiert. Zudem wird kurz auf Implementierungen des Standards und deren Ergebnisse eingegangen. Die Vorteile der serviceorientierten Integration des Produktlebenszyklus über einen ESB werden im Anschluss näher beschrieben.

### **3.2.1 PLM Services 2.0**

Die Vielzahl an Anwendungen, die im gesamten Produktlebenszyklus zum Einsatz kommen, erhöhen nicht nur die Heterogenität, sondern erschweren auch deren Integration. Standards helfen die Heterogenität zu reduzieren und die Kompatibilität zwischen Anwendungen zu erhöhen. Aus diesem



Phase 1: Konzept	Phase 2: Design & Entwicklung	Phase 3: Produktionsplanung	Phase 4: Produktion	Phase 5: Verwendung & Support	Phase 6: Außerbetrieb- stellung & Entsorgung
<ul style="list-style-type: none"> <li>• Ideengenerierung</li> <li>• Forschung</li> <li>• Marktanalyse</li> <li>• Anforderungsanalyse</li> <li>• Produktskizzen</li> </ul>	<ul style="list-style-type: none"> <li>• Mechanische, elektrische und elektronische Entwicklung</li> <li>• Softwareentwicklung</li> <li>• Digitales und Physisches Mock-Up</li> <li>• FEM Simulation</li> <li>• Test</li> <li>• Validierung</li> <li>• Dokumentation</li> <li>• Produktzertifizierung</li> </ul>	<ul style="list-style-type: none"> <li>• Maschinenprogrammierung</li> <li>• Werkzeugentwicklung &amp; -herstellung</li> <li>• Produktionsplanung und Terminierung</li> <li>• Layoutplanung</li> <li>• Personaleinsatzplanung</li> <li>• Kapazitätsplanung</li> <li>• Prozessplanung</li> <li>• Produktionsoptimierung</li> <li>• Simulation von Prozessen</li> <li>• Visualisierung</li> </ul>	<ul style="list-style-type: none"> <li>• Herstellung</li> <li>• Montage</li> <li>• Qualitätskontrolle</li> <li>• Betriebsmittelversorgung</li> <li>• Schichtplanerstellung</li> </ul>	<ul style="list-style-type: none"> <li>• Produktnutzung</li> <li>• Verwendung</li> <li>• Service</li> <li>• Support</li> <li>• Wartung</li> </ul>	<ul style="list-style-type: none"> <li>• Recycling</li> <li>• Entsorgung</li> <li>• Wiederverwendung als Ersatzteile</li> <li>• Wiedergewinnung von Rohstoffen</li> </ul>

Tabelle 3.1: Aufgaben der einzelnen Phasen im Produktlebenszyklus

Grund hat die Object Management Group (OMG) den Standard *PLM Services 2.0* entwickelt, der ein plattformunabhängiges Format für den Datenaustausch zwischen Anwendungen innerhalb des Produktlebenszyklus definiert [OMG09b]. Dieser erlaubt Daten über standardisierte Schnittstellen und das plattformunabhängige Datenformat z. B. zwischen der Produktentwicklung und der Produktion auszutauschen. Des Weiteren enthält der Standard ein plattformabhängiges Datenformat, das anwendbar auf Web Service Implementierungen ist, die über entsprechende WSDL Beschreibungen spezifiziert sind. *PLM Services 2.0* ist abgeleitet vom *Standard for the Exchange of Product Model Data (STEP) ISO 10303-214* [Int10].

Die Implementierung der *OMG PLM Services* in einem Demonstrator zeigt aufgrund des plattformunabhängigen Datenformates sowie den standardisierten und einheitlichen Schnittstellen den Vorteil der erhöhten Adaptivität und Transparenz [BCM08]. Zudem verhindert die Verwendung eines solchen Standards die Abhängigkeit von einem bestimmten Hersteller (engl.: vendor lock-in).

Der Demonstrator zeigt zudem, dass den *PLM Services 2.0* die Detailtiefe fehlt. Dies führt dazu, dass kundenspezifische Anpassungen der Schnittstellen bei jeder Implementierung durchzuführen sind, jedoch erschwert dies die Kompatibilität zwischen verschiedenen Umsetzungen. Zudem sind die standardisierten Schnittstellen auf interne Systeme beschränkt und können nicht für den Datenaustausch mit externen Partnern verwendet werden, weshalb der Standard für den Datenaustausch mit Kunden oder Lieferanten nicht zum Einsatz kommen kann.

Eine weitere Implementierung der *PLM Services* von Lämmer und Bugow zeigt, dass der Einsatz des *PLM Services* Standard, insbesondere dem standardisierten Datenformat und den Schnittstellen, im Vergleich zu nicht standardisierten Lösungen die Betriebskosten drastisch senken und die Zeit bis zur Markteinführung reduzieren kann [LB07].

Diese Vorteile sind allerdings in der Praxis nicht nutzbar, da die Hersteller von Softwarelösungen diese standardisierten Schnittstellen nicht implementieren. Stattdessen setzen sie auf proprietäre Lösungen, sofern Schnittstellen angeboten werden, um von außen auf deren Systeme zuzugreifen. Deshalb muss für eine serviceorientierte Integration für jede proprietäre Schnittstelle zusätzlich ein Service implementiert werden, der eine Schnittstellenbeschreibung bereit stellt und die proprietären Daten in das gewünschte Datenformat zum Datenaustausch übersetzt, wodurch der Aufwand deutlich erhöht wird.

### **3.2.2 Enterprise Service Bus zur Integration des Product Lifecycle Management**

Die Integration des PLM auf Basis der SOA und einem ESB beschreiben Credle et al. von der Firma IBM [CBB<sup>+</sup>08]. Im Fokus stehen dabei die Integration der fünf zentralen Disziplinen des Produktlebenszyklus, dem Konstruktionsänderungs-, Produktdaten-, Simulationsdaten-, Produktionsdaten- und Servicedatenmanagement. Es wird ein systematisches Vorgehen beschrieben, das auf Basis von unternehmensspezifischen Geschäftsprozessen die Integration der voneinander abgeschotteten Anwendungen durchführt. Die Geschäftsprozesse werden dabei aus den eben beschriebenen *PLM Services 2.0* gebildet und sorgen für eine standardisierte Schnittstelle zu den Anwendungen des Produktlebenszyklus. Dies wird durch eine Referenzarchitektur unterstützt, die auf der SOA Technologie beruht und über vier Basisintegrationsschichten die verschiedenen Disziplinen des PLM zusammenführt. Die vier Basisintegrationsschichten bestehen aus der Anwendungsintegration, der Datenintegration, dem Geschäftsprozessmanagement und der Geschäftsservicesteuerung. Der ESB fungiert dabei als Integrationskomponente, um die Anwendungen der Anwendungsintegrationsschicht über Schnittstellen und Konnektoren anzubinden. Die Schichten Datenintegration und Geschäftsprozessmanagement sorgen in Form von

Workflows für die Integration der einzelnen Services und Anwendungen. Die Geschäftsservicesteuerung dient zur Verwaltung und Weiterentwicklung der darunterliegenden Schichten, indem Serviceänderungen und Einbindung von neuen Services überwacht und kontrolliert werden. Durch die Verwendung eines einzelnen ESBs wird die Skalierbarkeit dieses Ansatzes erschwert, da in jeder Domäne des Produktlebenszyklus andere Anforderungen an die Integrationsumgebung gestellt werden. Zudem ist die Definition eines globalen Datenaustauschformates zur Integration aller Anwendungen des Produktlebenszyklus aufwendig und geht über das plattformunabhängige Datenformat weit hinaus, das im Standard der *PLM Services 2.0* beschrieben ist.

Keen et al. entwickelten Muster für eine serviceorientierte Architektur im e-Business [KAB<sup>+</sup>04, KAH<sup>+</sup>05]. Ziel der Muster ist es, bei der Analyse und dem Verstehen komplexer Geschäftsprobleme zu helfen, indem diese in kleinere und besser verwaltbare Funktionsbereiche aufgeteilt werden, die dadurch einfacher zu implementieren sind. Zur Umsetzung der Integrationsmuster wird auch hier ein ESB verwendet, der die lose Kopplung der Serviceschnittstellen ermöglicht und durch seine Plattformunabhängigkeit die Integration von heterogenen Anwendungslandschaften unterstützt. Anwendung finden diese Muster für die Ende-zu-Ende Beschreibung von Geschäftsprozessabläufen, die aus sequentiellen Schritten bestehen. Einzelne Schritte können dabei durch Menschen ausgeführt werden. Zudem werden langlaufende Transaktionen durch die Muster unterstützt, die insbesondere für die Interaktion zwischen Menschen und Computern notwendig sind. Die Verwendung von Mustern kann die Integration des Produktlebenszyklus vereinfachen, indem wiederkehrende Mensch- und Computer-Interaktionen durch dieselben Muster umgesetzt werden. Dies reduziert letztendlich die Komplexität der Lösung, wodurch ein geringerer Wartungsaufwand entsteht.

Lee et al. beschreiben die Erweiterung eines PDM-Systems der Firma Samsung Electronics Co. zu einem PLM-System, um die über die Zeit gewachsene

Anzahl an Anwendungen im Produktlebenszyklus zu integrieren und Geschäftsprozesse zwischen Anwendungen auszuführen [LLS<sup>+</sup>07]. Die von Managern und Ingenieuren geforderte Flexibilität der Geschäftsprozesse führte zur Entscheidung, dass das neue PLM-System auf Basis der SOA aufzubauen und als Integrationsmiddleware ein ESB zu verwenden ist. Somit können die Prozesse in Form von BPEL-Workflows flexibel definiert werden. Die Präsentationsschicht ist über webbasierte Portale realisiert, als eine plattform- und ortsunabhängige Lösung. Um die Anzahl der Services übersichtlich zu halten, wurde jedoch nur ein Teil der Anwendungen in die Architektur integriert. Durch die lose Kopplung der Services in Workflows wird die Definition von neuen Prozessen ermöglicht. Ein einheitliches Datenaustauschformat wird in der Arbeit nicht beschrieben, dies führt jedoch bei einer steigenden Anzahl an integrierten Services schnell zu einer großen Anzahl an Übersetzungsservices zwischen den einzelnen proprietären Datenformaten.

### 3.3 CHAMPAGNE

Im Rahmen des Sonderforschungsbereichs (SFBs) 467 wurde die sogenannte *Stuttgarter Integrationsplattform* entwickelt. Der *Change Propagation Manager* (CHAMPAGNE) ist der zentrale Teil dieser Integrationsplattform und beschreibt eine Plattform zur Propagation von Änderungen in autonomen und heterogenen Informationssystemen [RKHW06]. CHAMPAGNE selbst besteht aus drei Schichten. Die einzelnen Schichten und Komponenten der Integrationsplattform sind in Abbildung 3.2 dargestellt.

Die erste Schicht wird Basisebene genannt, sie enthält den Delmia Process Engineer (DPE) der Firma Dassault Systèmes. Der DPE verwendet zur Verwaltung aller PPR-Daten, die in der Produktionsplanung anfallen, eine Datenbank, die PPR-Hub genannt wird. Der Zugriff auf den PPR-Hub erfolgt über die Benutzeroberfläche des DPE oder mittels Visual Basic for Applications

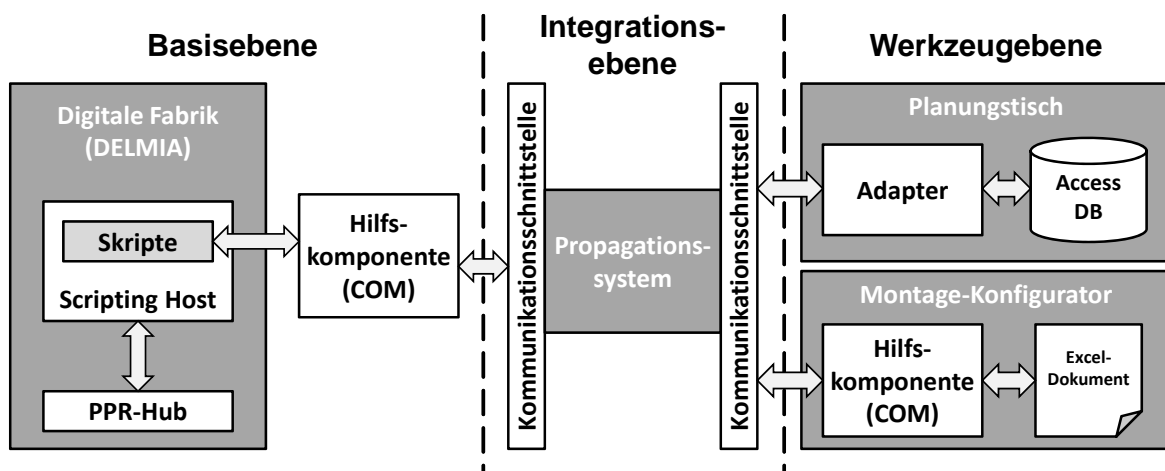


Abbildung 3.2: Architektur der CHAMPAGNE Integrationsplattform [Hei11]

(VBA) Skripte. Diese Skripte starten eine Hilfskomponente, durch die mit der CHAMPAGNE Plattform kommuniziert werden kann [Hei11].

Die zweite Schicht ist die Integrationsebene, die die CHAMPAGNE Plattform enthält. Die Plattform selbst besteht aus drei Komponenten: *Dependency Manager*, *Propagation Manager* und *Repository* [RCHM02]. Der *Dependency Manager* stellt einen Editor bereit, mit dem die Abhängigkeiten zwischen Anwendungen spezifiziert werden können. Diese Informationen nutzt der *Propagation Manager*, um Daten zwischen den Anwendungen zu propagieren. Der Benutzer kann mithilfe des *Dependency Editors* Abhängigkeiten, die im *Repository* gespeichert sind, erstellen, ändern oder löschen. Zusätzlich können mit dem Editor *Propagations-* und *Transformations*skripte geändert werden. Die *Propagations*skripte werden in der Sprache *XML Propagation Definition Language (XPDL)* beschrieben. Die *Transformations*skripte verwenden die standardisierte Sprache *XSL Transformation (XSLT)*, basierend auf der Sprache *Extensible Stylesheet Language (XSL)*, die XML-Daten zwischen unterschiedlichen Datenformaten transformieren kann [HCM05]. Der *Propagation Manager* ist die Laufzeitkomponente der Plattform und überwacht alle Quellsysteme auf Änderungen. Tritt eine Änderung auf, so wird diese durch die *Propagations*skripte zu den Zielsystemen übertragen und dort angewendet. Die Konvertierung erfolgt zwischen den verschiedenen

Datenschemas der Quell- und Zielanwendungen, sowie dem XML Schema zur Datenübertragung innerhalb der CHAMPAGNE Plattform. Um die Daten zwischen den Datenschemas zu konvertieren stehen sogenannte *Transformer* zur Verfügung. Während der Datenkonvertierung können zusätzlich Filter eingesetzt werden, um die Daten auf bestimmte Bedingungen zu überprüfen. Das Repository speichert alle Abhängigkeiten, Propagations- und Transformationsskripte, Informationen über die Anwendungen und ihre Datenschemas, sowie Authentifizierungsinformationen in einer objekt-relationalen Datenbank.

Die dritte Schicht enthält die integrierten Anwendungen des SFB 467, den Planungstisch, Logistikprüfstand und Montagekonfigurator. Jede Anwendung besitzt einen Adapter oder eine Hilfskomponente, um Änderungen in der Datenquelle der Anwendungen zu überwachen und an CHAMPAGNE zu melden. Die integrierten Anwendungen verwenden dabei als Datenquelle die Datenbank Access oder Microsoft Excel [Hei11].

Alle Anwendungen verwenden Warteschlangen, die in Abschnitt 2.2.3.3 detailliert beschrieben sind. Die asynchrone Kommunikation über Nachrichten verhindert, dass der Sender einer Nachricht blockiert wird, sofern der Empfänger nicht erreichbar ist oder zur gleichen Zeit eine andere Nachricht empfängt oder verarbeitet. Der Empfänger kann ankommende Nachrichten durch die eingehende Warteschlange zwischenspeichern, bis er die Nachricht verarbeiten kann. Dies verringert die Abhängigkeit zwischen den Anwendungen, die dank der asynchronen Kommunikation voneinander entkoppelt sind.

Diese Architektur integriert die Anwendungen des SFB 467 flexibel. Durch das Propagationssystem sind die Daten im PPR-Hub immer auf dem aktuellen Stand. Inkonsistenzen zwischen den verschiedenen Datenspeichern werden somit vermieden.

## 3.4 Engineering Service Bus (EngBus)

Die Daimler AG setzt in der Produktentwicklung eine Vielzahl an Softwaresystemen ein, um die Ingenieure bei ihrer Arbeit zu unterstützen. Über die Jahre wurden die voneinander getrennten Anwendungen durch viele heterogene Schnittstellen mit den Kernsystemen verbunden, wodurch ein automatischer Datenaustausch zwischen den Anwendungen entsteht. Dies wiederum führt zu einer Verbesserung der Datenqualität und zu Zeiterparnissen bei Datentransfers zwischen den Anwendungen. Mit steigender Anzahl an Schnittstellen und integrierten Anwendungen steigt jedoch die Komplexität und führt somit zu einem höheren Verwaltungsaufwand. Dies erfordert einen wachsenden Aufwand für die Wartung der IT-Landschaft und behindert, durch diese massive Beeinträchtigung, die Weiterentwicklung der IT-Infrastruktur [KS11]. Somit wandelt sich die IT von einem Treiber in der Produktentwicklung zu einem Hindernis, da sie sich nicht in gleicher Geschwindigkeit mit der schnellen Entwicklung von Produkten und Prozessen weiterentwickeln kann.

Dies führte zu der Entscheidung, die Integration der Anwendungslandschaft, unter Verwendungen einer SOA, flexibler zu gestalten. Die Einführung einer SOA ermöglicht ein stufenweises Vorgehen, das zur Kostenoptimierung und Reduzierung des Risikos genutzt werden kann. Die kleinen Schritte können zudem schnelle Erfolge („Quick Wins“) bei der Umsetzung aufzeigen. Durch jede Iteration zeigen sich somit Vorteile gegenüber der vorherigen Stufe. Die Schwierigkeit in einer solch grundlegenden Erneuerung der IT-Landschaft ist nicht nur die technischen Herausforderungen zu meistern, sondern auch die Organisation und Geschäftsprozesse dem erforderlichen Rahmen dieses Änderungsprozesses anzupassen. Die Frontendintegration basiert hierfür auf einem prozessorientierten gemeinsamen Engineering Client (engl.: Common Engineering Client (CEC)) und die Backendintegration auf dem Engineering Service Bus (EngBus).



Aus organisatorischer Sicht ändern sich durch die Einführung der SOA die Verantwortlichkeiten der IT-Mitarbeitenden, weg von einzelnen Softwaresystemen hin zu Prozessen, deren Ausführung über mehrere Anwendungen hinweg umgesetzt wird. Um diese Aufgabe zu meistern und die Mitarbeitenden in diesem Änderungsprozess mitzunehmen hat die Daimler AG ein Domänenmodell für die Engineering IT definiert. Dieses Modell besitzt Domänen, die zentrale Aufgaben der Produktentwicklung beschreiben, wie z. B. Design, Konstruktion und Absicherung. Außerdem werden unterstützende Domänen verwendet, die domänenübergreifende Leistungen beschreiben, wie z. B. IT-Sicherheit und IT-Qualitätsmanagement. Das Domänenmodell definiert somit klar die Zuständigkeiten und Verantwortungsbereiche für die zugehörigen Prozesse und Softwaresysteme, sowie die Schnittstellen zwischen den einzelnen Domänen.

### **3.5 Manufacturing Service Bus (MSB)**

Die schnelle Reaktion auf Ereignisse ist in der Produktion eine Grundvoraussetzung, um Störungen schnell zu beseitigen und somit Produktionsausfälle zu minimieren. Aus diesem Grund wurde eine servicebasierte Integrationsumgebung für das Produktionsumfeld geschaffen, die einen ESB mit den Fähigkeiten eines Event-Managements erweitert. Diese MSB genannte domänenspezifische Erweiterung des ESB-Konzeptes erfüllt die Anforderung schnell auf Events in der Fertigung zu reagieren [Min12, MRR<sup>+</sup>10, MRMW11].

Die Erweiterungen und Anpassungen des MSB beinhalten u. a. ein Event-Management. Dieses kann Events aus der Produktion automatisch verarbeiten, indem der MSB entsprechende in BPEL modellierte Prozesse ausführt, die die jeweilige Situation adäquat behandeln. Des Weiteren ist der MSB an den Fabrikkontext angepasst, d. h. er kann verschiedene Datenformate aus dem Produktionsumfeld verarbeiten und deren Daten über ein einheitliches Datenformat zwischen verschiedenen Anwendungen austauschen. Zudem

unterstützt der MSB sogenannte *Change Propagation Workflows*, mit deren Hilfe Änderungen in einem Softwareprogramm automatisch an eine oder mehrere Anwendungen gesendet werden, um inkonsistente Daten zu vermeiden. Dieses Verhalten der Änderungsverbreitung entspricht dabei der CHAMPAGNE Plattform.

Darüber hinaus definiert der MSB eine fünfstufige Integrationshierarchie. Diese beschreibt die Einbindung von Anwendungen auf dem untersten Level in Geschäftsprozesse auf dem obersten Level. Diese Integrationshierarchie ist dargestellt in Abbildung 3.3. Im untersten Level 0 befinden sich die Anwendungen und Daten, die integriert werden sollen. Level 1 enthält alle Serviceschnittstellen, über die auf die Anwendungen bzw. Daten aus dem untersten Level zugegriffen wird. Die eigentliche Integration findet über den MSB in Level 2 statt, der mittels eines CBR, der in Abschnitt 2.2.3.2 detailliert beschrieben ist, den Datenaustausch zwischen Services durch deren lose Kopplung ermöglicht. Übersetzungsservices wandeln Nachrichten zwischen proprietären Datenformaten der Anwendungen und dem einheitlichen Datenformat des MSB um. Die im ESB integrierte Workflow Engine sorgt für die Ausführung der installierten BPEL Workflows, die in der Integrationshierarchie in den Levels 3 und 4 eingeordnet sind. Die Geschäftsservices in Level 3 bilden einfache Prozesse ab, z. B. zur Verarbeitung von Events aus der Produktion oder eine Datenübertragung zwischen zwei Anwendungen und sorgen automatisch für den Aufruf der notwendigen Übersetzungsservices. Das oberste Level 4 enthält die digitale Repräsentation der Geschäftsprozesse, die aus den darunter liegenden Services zusammengesetzt sind.

Diese Hierarchie erlaubt somit eine adaptive Integration von Anwendungen des Produktionsumfeldes über standardisierte Serviceschnittstellen. Darüber hinaus ermöglicht die Komposition dieser Services in Workflows eine adaptive Unterstützung von Geschäftsprozessen im Produktionsumfeld.

Die Umsetzung des MSB erfordert die Erweiterung des ESBs um verschiedene Komponenten. Um Daten über ein einheitliches Datenformat zwischen

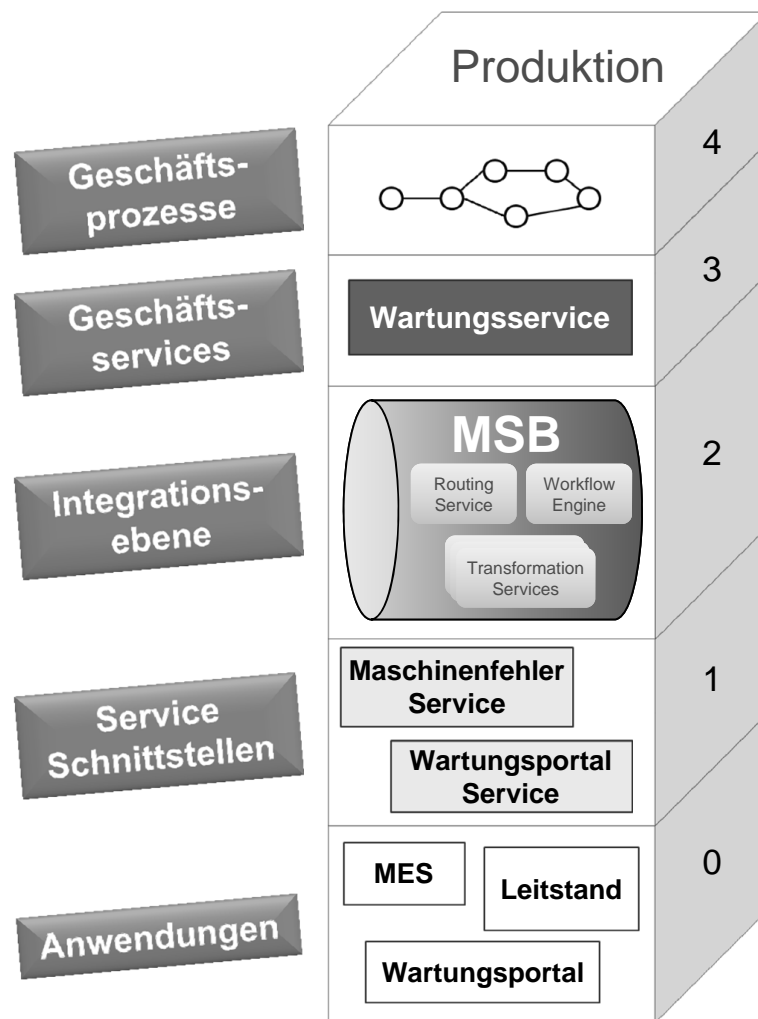


Abbildung 3.3: Integrationshierarchie des Manufacturing Service Bus (vergleiche [Min12])

verschiedenen Services austauschen zu können, wird im MSB ein *Event Canonical Model* definiert. Durch dieses einheitliche Datenaustauschformat kann die Anzahl an Datentransformationen zwischen jedem Paar an proprietären Datenformaten auf jeweils eine Datentransformation zwischen einem proprietären Datenformat und dem Event Canonical Model reduziert werden. Die Transformation wird über *Transformation Services* durchgeführt, die jede Nachricht in das einheitliche Datenformat bzw. zurück in das proprietäre Datenformat der Zielanwendung übersetzen. Die dynamische Zielbestimmung einer Nachricht führt der *Content-based Router* durch. Des Weiteren sind *Event- und Event-Flow-Registries* implementiert, die die Nachrichtenflüs-

se protokollieren und jedem Event eine eindeutige ID zuweisen. Somit ist jeder Event aus der Produktion zu einem späteren Zeitpunkt nachvollziehbar. Als ESB-Produkt wird der quelloffene *OpenESB* eingesetzt [Ope10]. Für die Ausführung der Workflows wird auf die im OpenESB integrierte *Workflow Engine* zurückgegriffen.

Die Benutzerinteraktion mit dem MSB erfolgt über zwei webbasierte Portale, dem Kunden- und Wartungsportal. Über das Kundenportal kann jeder Kunde Bestellungen der Produkte vornehmen, die anschließend über das integrierte Manufacturing Execution System (MES) in die Produktion eingetaktet werden. Zudem kann der Kunde den Fortschritt der Produktion in diesem Portal verfolgen und sieht dort, wenn Störungen der Produktion die Auslieferung der Bestellung verzögern. Jede Störung in der Montagelinie wird in das Wartungsportal übertragen, in dem ein Wartungsingenieur entscheiden kann, wie auf die Störung reagiert wird. Dafür hat er drei Optionen zur Auswahl, indem er den Fehler zeitnah selbst beheben kann, ein Ersatzteil bestellen muss oder den Kundenservice der Maschine benötigt. Gleichzeitig kann er die geschätzte Reparaturzeit angeben, bis die Montagelinie die Produktion fortsetzen kann, die dem Kunden im Kundenportal in Form einer verspäteten Auslieferung angezeigt wird.

## **3.6 Softwarelösungen für PLM, FLM und SCM**

Unternehmen setzen heute für viele Aufgaben Softwareanwendungen ein. Aufgrund dessen beschäftigt sich dieser Abschnitt mit der Unterstützung durch kommerzielle Softwarelösungen in Produktionsunternehmen. Dabei kann allgemein festgestellt werden, dass keine Softwarelösung alle Aufgaben, die in Produktionsunternehmen durchzuführen sind, unterstützt. Sie dient lediglich dazu, dass eine oder mehrere Aufgaben ausgeführt werden.

**PLM** wird heute von immer mehr Softwareherstellern unterstützt. Die meisten PLM-Lösungsanbieter entstanden durch einen Wandel vom PDM-Hersteller. Diese erweitern die PDM-Systeme mit mehr Funktionalität oder durch Anbindung weiterer Anwendungen und erreichen damit die Unterstützung von mehr Aufgaben aus dem Produktlebenszyklus. Im Folgenden werden die drei größten Anbieter von PLM-Lösungen kurz beschrieben.

- **Siemens** stellt mit seiner Softwarelösung *Siemens PLM Software* ein umfangreiches Paket mit unzähligen Anwendungen zur Verfügung, die praktisch alle Aufgaben des Produktlebenszyklus unterstützen. Die Anwendungen wie z. B. NX und Solid Edge sind über Teamcenter integriert, das auch als Manufacturing Backbone bezeichnet wird. Es dient als zentrale Datenbank, um alle Daten des Produktlebenszyklus zu speichern und im Anschluss allen integrierten Anwendungen zur Verfügung zu stellen. Diese sind damit in der Lage, Daten über Teamcenter untereinander auszutauschen<sup>1</sup>. Teamcenter wurde ursprünglich Mitte der 1980er Jahre als PDM-System zur Verwaltung von CAD-Dateien entwickelt und über die Jahre mit Produktentwicklungs- und Servicefunktionen erweitert [CIM10]. Ende der 1990er Jahre bis zum Jahr 2005 wurden die verschiedenen Teamcenter-Produkte grundlegend überarbeitet und an die aktuellen Anforderungen angepasst.
- **Dassault Systèmes** bietet keine explizite Lösung für PLM an. Implizit werden die Hauptaufgaben des Produktlebenszyklus durch mehrere eigenständige Softwarelösungen abgedeckt, die untereinander integriert sind und somit Informationen und Daten austauschen können<sup>2</sup>. Im Detail kommen CATIA und Solidworks als 3D-CAD-Konstruktionssoftware, SIMULIA als Simulationssoftware, 3DVIA zur

---

<sup>1</sup>[http://www.plm.automation.siemens.com/de\\_de/index.shtml](http://www.plm.automation.siemens.com/de_de/index.shtml)

<sup>2</sup><http://www.3ds.com/de/plm/>

Kommunikation, ENOVIA, 3DSWYM für das Innovationsmanagement und weitere Softwarelösungen zum Einsatz.

- **PTC** bietet mit *PTC Windchill* ebenfalls eine PLM-Lösung an, die produktbezogene Inhalte einschließlich Mechanical Computer-aided Design (MCAD), Electronic Computer-aided Design (ECAD) und Dokumente verwaltet und referenziert<sup>3</sup>. Über sogenannte Integrationsmodule können Anwendungen von anderen Herstellern wie AutoCAD, CATIA, Solidwork oder NX angebunden werden, was sich allerdings auf MCAD und ECAD beschränkt. PTC Windchill erweitert die klassische PDM Funktionalität, indem Produktinformationen über den gesamten Lebenszyklus verfügbar sind und somit neue Prozesse, z. B. im Service, ermöglichen. Darüber hinaus liefert PTC Windchill im Vergleich zu den PLM-Lösungen von Siemens und Dassault Systèmes keine Unterstützung für die Produktionsplanung und bietet auch keine Schnittstellen zu Softwaresystemen der Digitalen Fabrik von Drittherstellern.

Die eben aufgeführten Hersteller haben in den letzten Jahren angekündigt ihre PLM-Lösungen um Aspekte der SOA zu erweitern [CIM06]. Insbesondere Web Service Schnittstellen sollen den Datenaustausch zwischen den unterschiedlichen Anwendungen vereinfachen und die Komplexität reduzieren. Dies steht jedoch dem eigentlichen Konzept entgegen, die Vielzahl der Anwendungen über eine zentrale Datenbank zu integrieren. Es bleibt offen, ob diese Web Service Schnittstellen nur zur internen Integration der Anwendungen an die zentrale Datenbank dienen, zum Datenaustausch mit Software von Drittanbietern genutzt werden oder die Möglichkeit schafft, dass Anwendungen direkt Daten austauschen.

Eine Ende-zu-Ende Unterstützung des **FLM** durch kommerzielle Softwarelösungen gibt es derzeit nicht auf dem Markt. Vielmehr beschränkt sich die

---

<sup>3</sup><http://de.ptc.com/product/windchill/>

Unterstützung durch die Digitale Fabrik auf die Produktionsplanung, in der eine Verknüpfung von Produkt, Prozess und Ressourcen stattfindet. Des Weiteren findet eine Abbildung der Fabrik in einem digitalen Modell statt, das genutzt wird, um die Prozesse zu simulieren und anschließend zu optimieren. Die zwei größten Anbieter von Fabrikplanungssoftware sind bereits als PLM-Lösungsanbieter vorgestellt worden. Im Folgenden wird kurz auf deren Softwarelösung zur Unterstützung der Produktionsplanung eingegangen.

- **Siemens:** Bei dem Produkt *Tecnomatix* handelt es sich um ein Portfolio von digitalen Manufacturing-Lösungen, die die Produktionsplanung in Form von Prozessdefinition und -planung und deren Simulation unterstützt<sup>4</sup>. Tecnomatix ist Teil von Siemens PLM Software und bietet Schnittstellen zu den Softwarewerkzeugen für die Produktentwicklung und Fertigung.
- **Dassault Systèmes:** Das Softwarepaket *DELMIA* bildet die Digitale Fabrik Lösung von Dassault Systèmes, um Produktionsprozesse virtuell zu definieren, planen, erstellen, überwachen und steuern<sup>5</sup>. DPE ist dabei der Kern des Softwarepaketes und integriert andere Anwendungen wie GEOVIA oder SIMULIA. Darüber hinaus kann man mit der DELMIA Software die Supply Chain verwalten und überwachen.

Diese Softwarelösungen der Digitalen Fabrik werden zumeist nur in mittleren und großen Produktionsunternehmen eingesetzt, da der Aufwand für die Modellierung der virtuellen Fabrik enorm ist, insbesondere wenn z. B. keine Schnittstellen zum automatischen Einlesen von Produktdaten bestehen.

Die großen Softwarehersteller setzen auf die starre Integration der verschiedenen digitalen Werkzeuge, die über eine zentrale Datenbank integriert sind. Dassault Systèmes bietet mit DELMIA ein umfangreiches Softwarepaket für

---

<sup>4</sup>[http://www.plm.automation.siemens.com/de\\_de/products/tecnomatix/](http://www.plm.automation.siemens.com/de_de/products/tecnomatix/)

<sup>5</sup><http://www.3ds.com/de/products/delmia/>

die Digitale Fabrik an und verwendet daraus den DPE als zentralen Datenspeicher, an den die verschiedenen Werkzeuge angebunden sind. Siemens PLM Software nutzt dagegen die Softwarelösung Teamcenter, die im Jahr 2007 durch die Übernahme der Firma UGS erworben wurde, um alle Daten der Fabrik zu speichern.

Das **SCM** wird meist durch ERP-Systeme unterstützt und ist entweder als Modul vorhanden, oder als eigenständige Softwarelösung Bestandteil eines Softwarepaketes. Die meisten Unternehmen setzen bereits heute zur Unterstützung ihrer Verwaltung auf ERP-Systeme. Je nach Größe und Branche des Unternehmens gibt es eine unterschiedliche Zahl an Anbietern für ERP Softwarelösungen, da viele branchenspezifische Lösungen existieren. Die großen ERP Hersteller bieten dagegen eine generische Lösung an, die vor dem Einsatz an die jeweiligen Branchen angepasst wird. Im Folgenden werden die drei größten ERP Anbieter und ihre Softwarelösungen kurz vorgestellt.

- **SAP:** Die am verbreitetste Softwarelösung im Bereich ERP ist *SAP ERP Central Component (ECC)*. SAP ECC enthält eine Reihe an Modulen zur Unternehmenssteuerung, u. a. ERP, Customer Relationship Management (CRM), SCM, Finanz-, Personal-, Beschaffungs- und Produktmanagement<sup>6</sup>. Darüber hinaus kommen heute Module zur Geschäftsanalyse wie Data Warehousing, Enterprise Information Management, Business Intelligence und weitere Analyselösungen hinzu. Lange Zeit war SAP ECC unter dem Namen SAP R/3 bekannt, bevor es 2003 von mySAP ERP basierend auf NetWeaver abgelöst wurde. Seit 2007 heißt die ERP-Lösung SAP ECC, beinhaltet aber weiter die Kernmodule von R/3.

SAP bietet mit *SAP PLM* ein Modul für die Unterstützung des Produktlebenszyklus an, das im Wesentlichen die ERP-Module Materialwirtschaft, Produktionsplanung und das Projektmanagement-Modul

---

<sup>6</sup><http://www.sap.com/germany/solutions/index.epx>



zusammenfasst<sup>7</sup>. Das Dokumenten- und Workflow-Management findet über die entsprechenden SAP ECC Module statt, die zum einen eine hohe Integration bieten, jedoch die technischen Produkthanforderungen nicht erfüllen. Die Funktionalität, Benutzeroberfläche und das Datenmodell sind nur rudimentär. Somit kann SAP PLM Dokumente über den Produktlebenszyklus verwalten, eine native PLM-Lösung jedoch nicht ersetzen.

- **Microsoft:** Die ERP Software *Microsoft Dynamics AX* für mittelständische Unternehmen und Großkonzerne und *Microsoft Dynamics NAV* für KMUs bieten eine ähnliche Funktionalität wie SAP R/3<sup>8</sup>. Die Trennung in zwei Produkte rührt aus der Historie der Software. Microsoft Dynamics AX geht aus der Übernahme von Axapta hervor, Microsoft Dynamics NAV hingegen aus der Übernahme von Navision. Eine CRM-Lösung vertreibt Microsoft als Modul, das in Dynamics AX und Dynamics NAV integriert ist oder getrennt als eigenständige Lösung unter *Microsoft Dynamics CRM* firmiert.
- **Oracle:** *JD Edwards EnterpriseOne* wird die ERP-Lösung von Oracle genannt, die wie SAP ECC aus verschiedenen Modulen besteht, die sich weitgehend in ihren Einsatzgebieten decken<sup>9</sup>. Es sind neben ERP auch Module für CRM, SCM und PLM vorhanden. Das PLM Modul ist jedoch in Umfang und Funktionalität den zuvor vorgestellten PLM-Softwarelösungen weit unterlegen.

Die Integration der Domänen PLM und ERP kann wesentliche Vorteile für Unternehmen jeder Größe mit sich bringen [CIM05]. Durch die Integration wird die Konsistenz der Produktdaten sichergestellt, wodurch die Zeit neuer Produkte bis zur Markteinführung reduziert wird. Dabei ist eine gleichzeitige Erhöhung der Qualität und Reduzierung der Kosten zu verzeichnen. Darüber

<sup>7</sup><http://www.pdm-infoshop.de/pdimplm/index41.php>

<sup>8</sup><http://www.microsoft.com/de-de/dynamics/default.aspx>

<sup>9</sup><http://www.oracle.com/de/solutions/midsized/business-solutions/erp/index.html>

hinaus entstehen einheitliche produktbezogene Terminologien und Prozesse über die Anwendungsgrenzen hinweg. Die PLM-Module von SAP und Oracle bieten zwar eine gute Integration mit dem jeweiligen ERP System, jedoch fehlt ihnen, im Vergleich mit den PLM-Lösungen von Siemens, Dassault Systèmes oder PTC, die technische Orientierung, ein produktbezogenes Datenmodell und die breite Funktionalität über den gesamten Lebenszyklus.

Um diese Vorteile der PLM/ERP Integration nutzbar zu machen, werden Schnittstellen benötigt, um Daten zwischen den Softwarelösungen austauschen zu können. Bei den vorgestellten Standardsoftwarelösungen waren Schnittstellen bisher nicht die Regel oder von proprietärer Natur. Mit der Ankündigung vieler Softwarehersteller, ihre Anwendungen mehr nach dem Paradigma der SOA auszurichten und stärker auf das Serviceprinzip zu setzen, könnten sie flexibler in die Gesamtarchitektur integriert werden [CIM06]. Jedoch ist nicht klar, ob die Serviceschnittstellen lediglich dazu dienen, die einzelnen Module bzw. Anwendungen der Gesamtlösung flexibler zu verknüpfen oder tatsächlich für den Datenaustausch mit Anwendungen anderer Anbieter vorgesehen sind. Beides ist notwendig, um die Adaptivität der gesamten IT-Architektur zu verbessern.

Die eben genannten Hersteller bieten große und umfangreiche Softwarelösungen, die sehr gut integriert sind, aber hohe Lizenzkosten erfordern. Ein anderer Weg, den oft KMUs wählen, ist der Einsatz von vielen kleinen Werkzeugen, die in der Summe weniger umfangreich sind und geringere Lizenzkosten haben, jedoch nicht oder nur punktuell integriert sind.

Die Unterstützung von Aufgaben und Prozessen im Unternehmensumfeld durch Softwaresysteme steigt stetig. Bei der Entwicklung einer adaptiven und wandlungsfähigen IT-Architektur muss deshalb berücksichtigt werden, dass nicht nur Softwarelösungen für ERP und PLM in Unternehmen zum Einsatz kommen, sondern auch eine Vielzahl weiterer kleiner Anwendungen für spezielle Aufgaben. Zusätzlich kommt noch von Unternehmen selbst entwickelte Software hinzu. Aus diesem Grund müssen sowohl bestehende,

als auch neue Softwarelösungen einfach in eine neue Gesamtarchitektur integrierbar sein.

## 3.7 Zusammenfassung

Dieses Kapitel beschreibt Standards und Arbeiten zur serviceorientierten Integration des Produktlebenszyklus und der kommerziellen Softwarelösungen, die die Verwaltung von PLM, FLM und SCM unterstützen. Einleitend wird das erweiterte Phasenmodell des Produktlebenszyklus beschrieben, das neben den eigentlichen Phasen, die das Produkt durchläuft, noch die Produktionsplanung enthält, um der Funktionalität, die PLM-Softwarelösungen heute enthalten, zu entsprechen.

Der OMG Standard *PLM Services 2.0* definiert ein plattformunabhängiges und -abhängiges Datenmodell für den Datenaustausch im Produktlebenszyklus, sowie zugehörige Web Service Schnittstellen. Versuche zeigen, dass durch die Verwendung des Standards die Komplexität der Integrationslösung gesenkt werden kann, allerdings sind die entsprechenden Schnittstellen in gängigen Softwarelösungen nicht implementiert.

Die Firma IBM beschreibt zur Integration des PLM vier Basisschichten, die aus der Anwendungsintegration, der Datenintegration, dem Geschäftsprozessmanagement und der Geschäftsservicesteuerung bestehen. Der ESB dient dabei als Integrationsplattform, der Nachrichten zwischen den Services vermittelt. Dazu sorgt der ESB für die Ausführung der Workflows, die aus den bereitgestellten Services zusammengesetzt sind. Des Weiteren entwickelte IBM Muster für die Integration des Produktlebenszyklus auf Basis der serviceorientierten Architektur. Mittels der Muster können komplexe Geschäftsprobleme besser analysiert werden und helfen bei der systematischen Umsetzung. Zur Ausführung der Muster wird auch hier ein ESB eingesetzt.

Die CHAMPAGNE Plattform erlaubt durch Verwendung von Propagations- und Transformationsskripten eine flexible Integration von Anwendungen der Produktionsplanung. Der DPE, als zentraler Datenspeicher in dieser Integrationslösung, ist mit seinem domänenspezifischen Datenmodell ein fester Teil der Produktionsplanung. Erweiterungen des Datenmodells sind möglich, jedoch wäre eine vollständige Erweiterung für andere Domänen mit großem Aufwand verbunden.

Der EngBus ist eine domänenspezifische Erweiterung des ESB für die Produktentwicklung, der von der Firma Daimler entwickelt wird. Das Ziel der Bemühungen ist die bestehende IT-Landschaft servicefähig zu machen und letztendlich eine verbesserte und zugleich flexiblere Integration zu erreichen. Dabei steht die Prozessunterstützung durch die IT-Systeme im Vordergrund. Ein Domänenmodell für die Produktentwicklung dient der klaren Abgrenzung der Zuständigkeiten bei der Reorganisation, die im Zuge der Änderungen durchgeführt wird. Ein Change Management Prozess soll ein systematisches Vorgehen hin zu einer serviceorientierten IT-Landschaft vorgeben.

Eine phasenspezifische ESB-Erweiterung für die Produktion stellt der MSB dar. Der MSB stellt ein Event-Management für die Verarbeitung von Ereignissen bereit, die durch Maschinen oder Anwendungen erzeugt werden. Die Implementierung von *Change Propagation Workflows* erlaubt die automatische Verarbeitung von Ereignissen. Die Anpassung an den Fabrikkontext erfolgt durch die Bereitstellung eines *Event Canonical Models*, einem einheitlichen Datenformat, das den Nachrichtenaustausch unabhängig von den proprietären Datenformaten der integrierten Anwendungen macht.

Die Unterstützung des PLM, FLM und SCM durch Softwarelösungen ist weit verbreitet und die Anzahl an entsprechenden Anbietern groß. Dabei gibt es zwar keine Lösung, die alle Bereiche von Produktionsunternehmen unterstützt, die einzelnen Bereiche des Produkt-, Fabriklebenszyklus und der Lieferkette werden hingegen jeweils von wenigen großen Softwarelösungen

---

ganz oder teilweise abgedeckt. Diese großen Softwaresysteme sind entweder auf vielen Modulen aufgebaut, die an einem zentralen Systemkern angebunden sind oder bestehen aus mehreren eigenständigen Softwareanwendungen, die über eine zentrale Datenbank integriert sind. Im Gegensatz dazu findet man viele eigenständige Anwendungen, die spezielle Aufgaben der genannten Bereiche unterstützen. Hier ist das Problem, dass beim Einsatz mehrerer dieser Anwendungen diese gar nicht oder nur punktuell integriert sind. Auch die Integration zwischen den großen Softwaresystemen untereinander oder mit kleineren Anwendungen ist oft nicht gegeben, wodurch ein durchgängiger Datenaustausch nicht möglich ist. Sofern Anwendungen integriert sind, ist die Lösung meist inflexibel und kann nur mit größerem Aufwand geändert werden.



---

## KAPITEL 4

# Serviceorientierte IT-Architektur für wandlungsfähige Produktionsunternehmen

---

In diesem Kapitel wird die Notwendigkeit für eine flexiblere IT-Architektur diskutiert. Die Abhängigkeiten einer unflexiblen IT-Architektur auf andere Elemente in Produktionsunternehmen und deren Auswirkungen aufeinander ist in Abschnitt 4.1 beschrieben.

Die im Rahmen der Arbeit entwickelte adaptive und wandlungsfähige IT-Architektur für das PLM wird in Abschnitt 4.3 vorgestellt, nachdem die Vor- und Nachteile einer Integration mit einem zentralen ESB gegenüber einer Multi-ESB-Integration diskutiert wurden. Für die Etablierung der Integrationsarchitektur, die eine neue Abstraktionsschicht zwischen Anwendungen einführt, werden die Definitionen von Datenaustauschformaten, Serviceverzeichnissen, dynamischem Routing und der Prozessunterstützung im Rahmen der neuen Architektur beschrieben.

Die Anwendung der adaptiven und wandlungsfähigen IT-Architektur auf das gesamte Produktionsunternehmen wird anhand eines ganzheitlichen Managementmodells durchgeführt, das die Aktivitäten des PLM, FLM und

SCM abdeckt. Dabei werden die wichtigsten Domänen der einzelnen Aktivitäten identifiziert, um sie anschließend im ganzheitlichen Managementmodell zusammenzuführen. Die Unterstützung durch IT-Systeme erfolgt durch Implementierung phasenspezifischer ESBs für jede Domäne, die durch einen weiteren ESB integriert werden.

## 4.1 Adaptivität und Abhängigkeiten von Unternehmenselementen

Für die erfolgreiche Herstellung von Produkten müssen Unternehmen verschiedene Elemente beherrschen. Die neun wichtigsten Elemente, die es parallel zum Produktlebenszyklus effizient zu verwalten und beherrschen gilt, um am Markt erfolgreich zu sein, werden von Stark beschrieben [Sta11]. Im Einzelnen sind dies 'Produkte', 'Prozesse', 'Daten und Dokumente', 'Anwendungen', 'Ausrüstung', 'Methoden', 'Organisationsstruktur', 'Menschen' und 'Metriken'. Da die Menschen bzw. Mitarbeitenden des Unternehmens die Organisationsstruktur bilden, werden diese im Folgenden gemeinsam betrachtet. Ebenso zusammengefasst werden die Anwendungen mit ihren Daten und Dokumenten, denn die Anwendungen verwalten und verarbeiten diese und bilden damit meist ein geschlossenes Anwendungssystem. Darüber hinaus wird in dieser Arbeit die Ausrüstung nicht weiter betrachtet, ebenso wie die Metriken, die nachträglich zur Messung des Erfolgs von Maßnahmen dienen. Das Produkt an sich steht im Mittelpunkt aller Bemühungen, das letztendlich erfolgreich hergestellt und verkauft werden soll. Die vier verbleibenden Elemente werden im Folgenden näher betrachtet:

- **Mitarbeitende** sorgen dafür, dass die Aufgaben und somit die Prozesse im Unternehmen ausgeführt werden. Hierfür nutzen die Mitarbeitenden IT-Systeme und Methoden, die sie bei ihren Aufgaben unterstützen



und ihnen erlauben, die Prozesse schneller und mit höherer Qualität auszuführen.

- **IT-Systeme** unterstützen die Nutzenden bei ihren Aufgaben, indem die Verarbeitung von Daten oder das Ausführen von Aufgaben durch Computersysteme vollständig oder teilweise übernommen wird. Dabei kommen eine Vielzahl von Softwareanwendungen zum Einsatz, die auf die jeweiligen Problemstellungen oder Arbeitsgebiete ausgelegt sind und oft spezielle Aufgaben unterstützen.
- **Methoden** sorgen dafür, dass die Ausführung von Prozessen und Vorgängen wiederholbar und die Ergebnisse somit vergleichbar sind. Ein Qualitätsprüfungsprozess beispielsweise erfüllt diese Anforderungen, da ansonsten eine Qualitätsbewertung bei nicht vergleichbaren Ergebnissen hinfällig wäre. Durch die reproduzierbare Ausführung von Methoden können diese Prozesse schneller bearbeitet und zudem die Qualität gesteigert werden. Außerdem können Methoden in Softwareanwendungen implementiert werden, das die zuvor genannten Vorteile der schnelleren Prozessausführung und verbesserten Qualität weiter steigert.
- **Prozesse** sind allgegenwärtig in Unternehmen. Ziel ist es, die Prozesse so zu gestalten, dass die Ausführungszeit so kurz wie möglich gehalten wird. Aus diesem Grund werden Mitarbeitende heute bei vielen Aufgaben durch Softwaresysteme unterstützt oder die Prozesse sind teilweise oder vollständig durch IT-Systeme automatisch ausführbar.

Diese Unternehmenselemente und ihre Abhängigkeiten sind in Abbildung 4.1 nochmals anschaulich illustriert. Dabei wird deutlich, dass Mitarbeitende alle Prozesse ausführen, entweder direkt mithilfe von IT-Systemen oder Methoden. Methoden wiederum können manuell ausgeführt werden oder in einem IT-System implementiert sein.

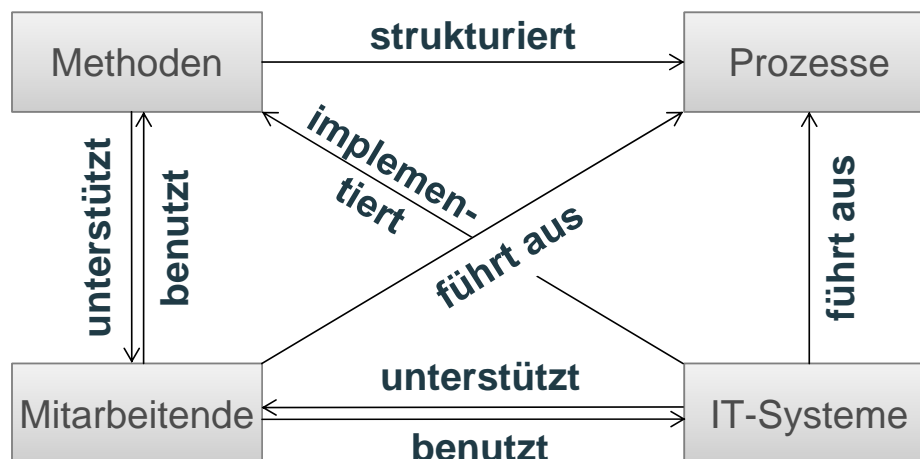


Abbildung 4.1: Wichtige Unternehmenselemente und ihre Abhängigkeiten [SSZM13]

Betrachtet man nun Abbildung 4.1 aus dem Gesichtspunkt der Adaptivität und Wandlungsfähigkeit, die heute wesentliche Voraussetzungen für erfolgreiche Unternehmen sind, so kann man folgende Feststellungen machen. *Mitarbeitende* besitzen die nötige Flexibilität, um verschiedenste Aufgaben auszuführen und können somit für eine hohe Flexibilität der Organisation sorgen. Dies ist jedoch nur möglich, wenn Mitarbeitende eine entsprechend breite Qualifikation besitzen und für mehrere Aufgaben eingesetzt werden können. *Methoden* sind nicht adaptiv. Dies ist jedoch gewünscht, da Methoden reproduzierbare und vergleichbare Ergebnisse liefern sollen, die durch einen festgelegten Prozess erreicht werden. In der Regel wird eine Methode deshalb nicht geändert, sondern durch eine neue Methode ersetzt. *Prozesse* an sich sind flexibel, um sie schnell an neue Anforderungen anpassen zu können, da es sich bei den Prozessen um eine lose Aneinanderreihung von Aufgaben handelt, die nur teilweise voneinander abhängig sind. Das Hauptproblem sind hierbei die *IT-Systeme* und deren starre Integration, die eine schnelle Änderung und Anpassung verhindern. Da viele Prozesse in Softwaresystemen implementiert sind, verlieren diese Prozesse ihre Adaptivität und können nur sehr langsam durch Softwareänderungen an neue Bedingungen angepasst werden. Deshalb werden oft Prozesse durch Softwaresysteme ausgeführt, die diese nicht optimal unterstützen.

Allgemein kann bei Standardsoftware festgestellt werden, dass sich die implementierten Prozesse an bewährten Vorgehensweisen (engl.: best practice) orientieren. Dies wiederum führt dazu, dass Unternehmen ihre Prozesse an die implementierten Prozesse der Standardsoftware anpassen. Das Problem dabei ist, dass in einem turbulenten Umfeld adaptive Prozesse notwendig sind, die einfach und schnell geändert werden können. Dies ist beim Einsatz von Standardsoftware jedoch nicht einfach möglich.

Diese starre Unternehmenssoftware ist ein Hauptgrund, warum Produktionsunternehmen die gewünschte Adaptivität und Wandlungsfähigkeit im gesamten Unternehmen nicht erreichen. Dies wird auch durch die Studie von Kirchner et al. belegt [KWW03]. Diese Unzulänglichkeit muss mit einer neuen adaptiven und wandlungsfähigen IT-Architektur (engl.: adaptable and changeable IT-Architecture (ACITA)) gelöst werden, um schnelle Anpassungen sowohl der Softwaresysteme, als auch der unterstützten Prozesse in einem Unternehmen zu ermöglichen.

## **4.2 Anforderungen an eine adaptive und wandlungsfähige IT-Architektur**

Für die durchgängige Prozessunterstützung durch digitale Werkzeuge ist deren Integration erforderlich, um Daten effizient austauschen zu können. Die Zusammenarbeit der verschiedenen digitalen Werkzeuge ist durch flexible Vernetzung oder starre Integration möglich [SGNWP08]. Die starre Integration besitzt den Vorteil, dass der Initialaufwand gering ist, um zwei Werkzeuge zu verbinden. Anpassungen und Änderungen sind dafür umso aufwendiger. Die flexible Vernetzung besitzt einen hohen initialen Aufwand, um die Integrationsmiddleware aufzusetzen, Anpassungen und Erweiterungen sind dann aber mit geringem Aufwand möglich.

Langfristig gesehen rechnet sich eine flexible Vernetzung der IT-Landschaft, da Änderungen im heutigen Unternehmensumfeld häufiger auftreten und deshalb der Aufwand für Anpassungen möglichst gering zu halten ist. Zur Verbesserung der Adaptivität und Wandlungsfähigkeit wird deshalb das Konzept der SOA auf den Produktlebenszyklus übertragen. Services bilden dabei die standardisierten Schnittstellen, um auf Funktionalität und Daten der Anwendungen zuzugreifen. Die lose Kopplung dieser Services erlaubt eine flexible Integration von Anwendungen und erfolgt durch die Verwendung eines CBR, wodurch die Adaptivität der IT-Architektur erhöht wird. Ein ESB als zentrale Integrationsplattform erzeugt eine neue Abstraktionsebene zwischen den Anwendungen. Damit wird die Kooperation zwischen unterschiedlichen Domänen des Produktlebenszyklus ermöglicht und Daten können flexibel zwischen den integrierten Anwendungen ausgetauscht werden [SMSM10, SMM12].

Darüber hinaus wird die Adaptivität der Prozesse durch die flexible Komposition der Services zu Workflows verbessert. Diese Workflows können automatisch durch ein Workflow-Management-System verwaltet und ausgeführt werden. Die Komposition erfolgt mittels einer standardisierten Workflowsprache, wie z. B. BPEL, die somit von den meisten Workflow-Management-Systemen unterstützt wird. Eine solche adaptive Unterstützung von Prozessen im Produktionsumfeld wird auch durch den MSB und dessen fünfstufige Integrationshierarchie beschrieben [Min12].

In diesem Abschnitt wird diskutiert, wie man alle Services in einem Produktionsunternehmen effizient verwalten kann und ob darüber hinaus eine IT-Architektur nicht nur adaptiv, sondern auch wandelbar sein kann. Dabei steht im Vordergrund, ob ein heterogenes Unternehmensumfeld durch einen ESB integrierbar ist oder ob man mehrere ESBs für die Integration der gesamten IT-Landschaft benötigt. Die folgende Diskussion basiert auf der Veröffentlichung von Silcher et al. [SMM11, MSRM11].

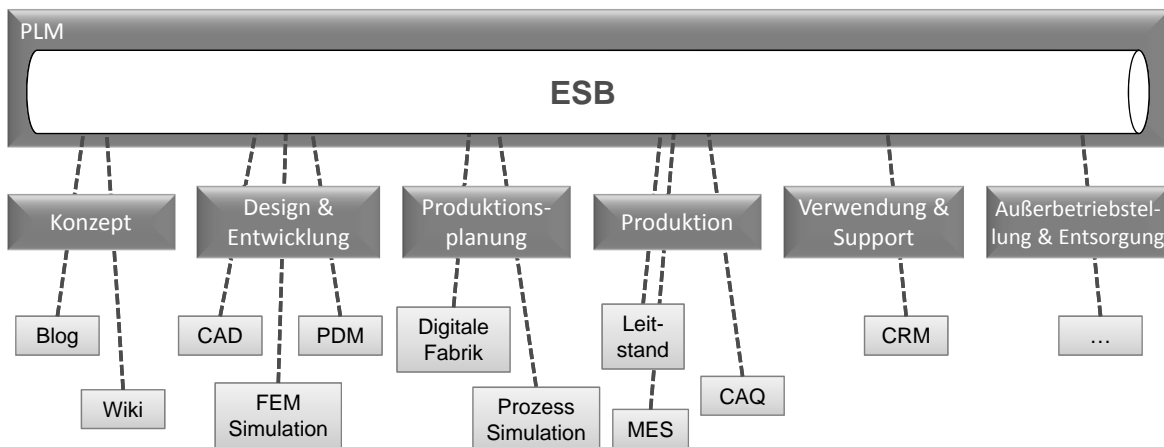


Abbildung 4.2: Integrationsarchitektur mit einem zentralen ESB (vergleiche [SMM11])

Die Umsetzung einer serviceorientierten Integrationsumgebung erfolgt typischerweise auf Basis eines zentralen ESBs. Dabei werden alle Anwendungen mit Serviceschnittstellen versehen und deren Beschreibung über ein Serviceverzeichnis veröffentlicht. Dieses Verzeichnis wird über den CBR des ESB abgefragt, somit können alle am ESB ankommenden Nachrichten dynamisch an die entsprechenden Zielservices zugestellt werden. Die Integration aller Anwendungen des Produktlebenszyklus über einen zentralen ESB ist in Abbildung 4.2 dargestellt. Einzelne Anwendungen sind beispielhaft abgebildet und entsprechend der Zuordnung zur jeweiligen Phase gruppiert. Die tatsächliche Anzahl an Anwendungen, die im Produktlebenszyklus zum Einsatz kommen, hängt vom jeweiligen Unternehmen ab. In der Regel reicht die Anzahl von wenigen dutzend bis zu tausenden Anwendungen.

Die Skalierbarkeit dieser Integrationslösung hängt von der Skalierbarkeit des zentralen ESBs ab. D. h. mit zunehmender Anzahl an integrierten Anwendungen und Services, sowie steigender Anzahl an ausgetauschten Nachrichten, muss der Durchsatz des ESB proportional steigen. Dies kann durch schnellere Hardware erreicht werden, die aber bei sehr hohen Anforderungen entsprechend kostenintensiv ist. Die Skalierung mithilfe von verbesserter Hardware

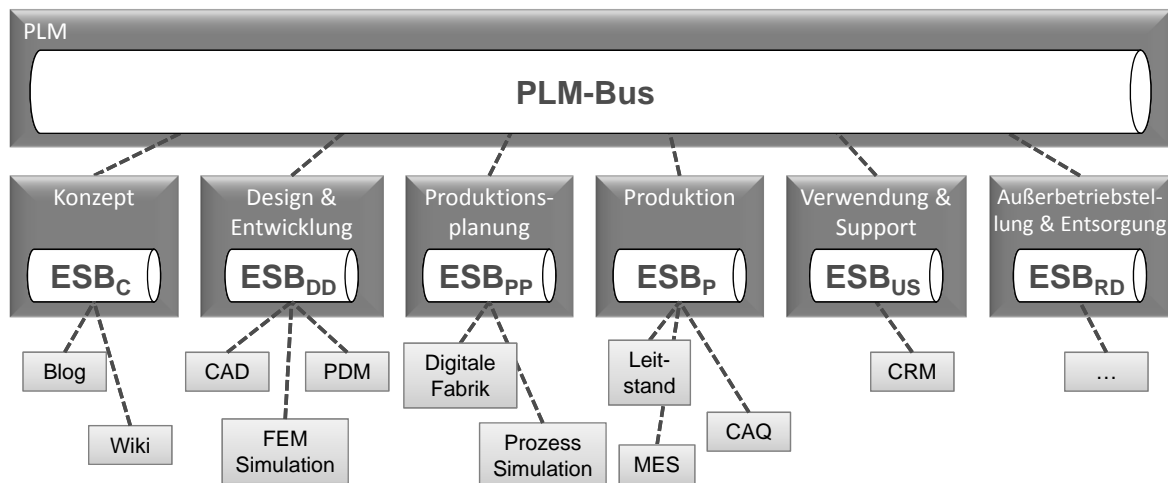


Abbildung 4.3: Integrationsarchitektur mit mehreren ESBs (vergleiche [SMM11])

ist deshalb nur bis zu einem bestimmten Punkt sinnvoll und kann nicht beliebig fortgesetzt werden.

Aufgrund der Heterogenität im Produktlebenszyklus, wie z. B. den unterschiedlichen Datenmengen, Datenformaten oder Echtzeitanforderungen, die zwischen einzelnen Anwendungen bzw. Services ausgetauscht werden, ist eine Integrationsarchitektur mit mehreren ESBs vorstellbar. Dabei erfolgt eine Anpassung der phasenspezifischen ESBs an die technischen Anforderungen der jeweiligen Phase, die wiederum über einen weiteren ESB integriert werden, der als PLM-Bus bezeichnet wird. Dieser Lösungsansatz ist in Abbildung 4.3 dargestellt.

Die phasenspezifischen ESBs binden dabei nur die Anwendungen der entsprechenden Phase an die Integrationsarchitektur an, und verarbeiten Nachrichten und Prozesse innerhalb der Phase lokal, so dass der PLM-Bus, im Vergleich zur Lösung mit einem einzigen zentralen ESB, weniger Nachrichten verarbeiten muss. Der Nachrichtenaustausch zwischen unterschiedlichen Phasen erfolgt daraufhin über den PLM-Bus, der ankommende Nachrichten an die entsprechende Zielphase weiter leitet, bei der sie letztendlich an den Zielservice zugestellt und anschließend verarbeitet werden.

Aus der Heterogenität des Produktlebenszyklus kann man verschiedene Anforderungen an die IT-Architektur ableiten, deren Diskussion getrennt nach technischen und organisatorischen Kriterien erfolgt. Die technischen Anforderungen umfassen den Umgang mit heterogenen Daten, die u. a. in unterschiedlichen Formaten, Mengen und Häufigkeiten auftreten. Des Weiteren stehen die Verfügbarkeit beim Ausfall eines ESBs und die Skalierbarkeit der Gesamtarchitektur zur Diskussion. Die organisatorischen Anforderungen umfassen die Verantwortlichkeiten, wer für welchen Teil der Integrationslösung zuständig ist, sowie deren Administration. Darüber hinaus steht zur Diskussion, wie sich die Einführung einer SOA auf die Organisationsstruktur auswirkt und inwiefern dies Auswirkungen auf die Mitarbeitenden und ihre Arbeit hat.

### **4.2.1 Technische Anforderungen**

Über den Produktlebenszyklus hinweg werden viele Anwendungen eingesetzt, um die Mitarbeitenden bei ihrer Arbeit zu unterstützen, damit diese produktiver arbeiten können. Ein Problem besteht darin, dass jede Anwendung ihr eigenes proprietäres Datenformat verwendet. Ein einheitliches Datenformat, beschrieben in Abschnitt 2.2.3.1, das vom ESB zum Datenaustausch zwischen Anwendungen verwendet wird, muss somit alle Daten der angebotenen Anwendungen abbilden können. Im Fall des zentralen ESBs benötigt man somit ein einziges, global einheitliches Datenformat. Ein solches globales Datenformat zu definieren bedeutet jedoch eine enorme Herausforderung, die bisher nicht gelöst ist. In der Architektur mit mehreren ESBs wird für jede Phase ein solches Datenaustauschformat benötigt. Diese müssen nur Daten der in der jeweiligen Phase verwendeten Anwendungen bzw. ausgetauschten Daten abbilden, wodurch die Komplexität des Datenformats reduziert wird. Der übergeordnete globale ESB besitzt ein weiteres einheitliches Datenschema, das alle Daten enthält, die zwischen den

einzelnen Phasen ausgetauscht werden. Somit bilden diese Datenformate eine Hierarchie, wie sie auch von Davies et al. beschrieben wird [DSRR08].

Die Datenmenge von Nachrichten und deren Häufigkeit, die in den einzelnen Phasen verarbeitet werden, unterscheidet sich deutlich. In den Phasen der Produktentwicklung und Produktionsplanung fallen sehr große Datenmengen an die die Anwendungen untereinander austauschen. Insbesondere dreidimensionale CAD-Zeichnungen und Simulationsdaten sind hierfür verantwortlich. Deren Austausch erfolgt jedoch unregelmäßig. Die Anwendungen der Produktion tauschen dagegen sehr häufig meist viele kleine Datenpakete aus. Wird der Produktlebenszyklus über einen einzigen ESB integriert, so muss dieser sehr viele Datenpakete in kurzer Zeit verarbeiten, deren Datenmenge die ganze Bandbreite von klein bis sehr groß abdeckt. Die Verwendung von phasenspezifischen ESBs ermöglicht deren Anpassung sowohl an ein hohes Datenaufkommen in der Entwicklungs- und Planungsphase, als auch die Verarbeitung vieler kleiner Datenpakete in der Produktion.

Über den gesamten Lebenszyklus findet man verschiedene Datenstrukturen, die von strukturierten bis zu unstrukturierten Daten reichen. Die frühen konzeptionellen Phasen, sowie die Phase der Nutzung des Produktes, müssen viele unstrukturierte Daten analysieren, um Meinungen und Feedback über ein Produkt in Blogs, Foren oder sozialen Netzwerken zu extrahieren, die für Verbesserungen und Weiterentwicklungen von großer Bedeutung sind. In den Phasen der Entwicklung, Planung und Produktion überwiegen strukturierte Daten, die in den verschiedenen Anwendungen gespeichert sind. Darüber hinaus findet man über den gesamten Produktlebenszyklus Dateien zur Dokumentation, die in der Regel viel unstrukturierten Text enthalten. Diese Dokumentationen enthalten eine große Menge an Wissen, das für viele Mitarbeitende nicht zugänglich ist oder sie durch die Unmengen an Dokumentation nicht die benötigte Information finden. Auch hier ist die Verwendung von phasenspezifischen ESBs von Vorteil, denn die ESBs der Konzept- und Nutzungsphase können an die Verarbeitung und Übertragung



von unstrukturierten Daten angepasst, die ESBs der Entwicklung, Planung und Produktion hingegen für die Handhabung von strukturierten Daten optimiert werden. Bei Verwendung eines globalen ESBs ist das Verarbeiten von strukturierten und unstrukturierten Daten ebenso möglich, dadurch wird die Optimierung der spezifischen Datenverarbeitung jedoch erschwert.

Bei einer verteilten IT-Infrastruktur muss die Latenz beim Datenaustausch berücksichtigt werden, insbesondere wenn Echtzeitanforderungen wie in der Produktion erforderlich sind. In den anderen Phasen des Produktlebenszyklus sind im Gegensatz dazu keine Echtzeitanforderungen vorhanden und somit eine geringe Latenz akzeptabel. Die Integration über einen zentralen ESB erfordert somit die Echtzeitverarbeitung von Nachrichten, um die Anforderungen der Produktion zu erfüllen. Zeitgleich muss wie bereits erwähnt die Verarbeitung von großen Datenmengen der Produktentwicklung und Produktionsplanung möglich sein. Aus diesen Gründen ist die Nutzung von phasenspezifischen ESBs vorteilhaft, denn diese können so adaptiert werden, dass sie entweder Nachrichten in Echtzeit oder große Datenmengen in kurzer Zeit verarbeiten können. Dies erlaubt eine bessere Anpassung der Hardware an die jeweiligen Anforderungen, wodurch geringere Kosten für die Gesamtarchitektur entstehen. Zudem können die phasenspezifischen ESBs näher an den einzelnen Standorten zum Einsatz kommen, um die Lokalität besser zu nutzen und die Datenübertragungszeiten zu verkürzen. So kann z. B. in jedem Produktionsstandort ein ESB für die Produktion verwendet werden, um die Echtzeitanforderungen für Maschinensteuerungen zu erfüllen.

Die Anforderungen an die Hardware hängen zudem mit der Skalierbarkeit zusammen. Mit steigender Anzahl an integrierten Anwendungen ist die Verarbeitung von mehr Nachrichten über die ESB-Infrastruktur erforderlich. Bei einem zentralen ESB kann dieser nur skalieren, indem mehr Hardwareressourcen zur Verfügung gestellt werden, um die einzelnen Nachrichten schneller zu verarbeiten. Die Anpassung der Architektur mit mehreren spezifischen ESBs erfolgt durch mehr Hardware oder über die Hinzu- oder Wegnahme

eines phasenspezifischen ESBs. Dies ermöglicht die Gesamtarchitektur im Sinne der Wandlungsfähigkeit zu verändern, indem diese mit vergleichsweise geringem Aufwand je nach Anforderungen wachsen oder schrumpfen kann. Zudem kann dies hohe Kosten verhindern, da unter Umständen ein Verzicht auf hochleistungsfähige Hardware möglich ist. Durch ihre einfache Anpassbarkeit eignet sich diese Architektur sowohl beim Einsatz in KMUs, als auch in großen Unternehmen.

Der letzte Punkt der technischen Anforderungen ist die Verfügbarkeit, die nur eine untergeordnete Rolle spielen sollte, da durch redundante Auslegung der Systeme Ausfälle in der Integrationsinfrastruktur überbrückt werden können. Im Fall einer redundanten Auslegung des zentralen ESBs müssen alle Backup-Systeme die identische, hoch skalierte Hardware besitzen. Jedoch kann diese Hardware, wie bereits diskutiert, sehr teuer sein. Bei einer Infrastruktur mit mehreren ESBs entstehen zwar auch höhere Kosten, jedoch sind die Anforderungen im Vergleich zu einem zentralen ESB geringer und die jeweiligen ESBs kommen mit geringeren Ressourcen aus. Sofern keine redundanten Systeme eingesetzt werden, so ist bei einem Ausfall des zentralen ESBs der gesamte Integrationsbackbone nicht mehr erreichbar und somit kein Datenaustausch zwischen Anwendungen und keine Ausführung von Workflows möglich. Der Ausfall eines phasenspezifischen ESBs hingegen führt zwar dazu, dass in einer Phase keine Nachrichten mehr ausgetauscht werden können, die anderen Phasen jedoch unabhängig davon weiter arbeiten. Beim Ausfall des übergeordneten PLM-Busses ist kein Informationsaustausch zwischen Phasen mehr möglich. Die Arbeit innerhalb einer Phase kann hingegen ungestört weiter laufen.

### **4.2.2 Organisatorische Anforderungen**

Die Umstellung auf eine neue IT-Architektur ist immer verbunden mit organisatorischen Anpassungen. Meistens sind IT-Abteilungen Teil der ein-

zelenen Fachabteilungen oder als Stabsabteilung in der Gesamtorganisation von Produktionsunternehmen zu finden. Entwicklerteams sind häufig für jeweils eine Anwendung zuständig und verantworten deren Wartung und Weiterentwicklung. Die Verantwortlichkeiten für Schnittstellen zwischen den Anwendungen wird auf die entsprechenden Entwicklerteams aufgeteilt, so dass es zu Überschneidungen kommt. Bei einer wachsenden Infrastruktur mit steigender Anzahl an Anwendungen und Schnittstellen wird klar, dass der Koordinationsaufwand drastisch steigt und nur durch eine übergeordnete IT-Abteilung, die für die Gesamtkoordination der IT-Landschaft verantwortlich ist, übernommen werden kann, um einen Wildwuchs an Schnittstellen zu verhindern.

Durch die Einführung einer SOA wandeln sich die Zuständigkeiten von einer Anwendungsverantwortung hin zu einer Prozessverantwortung [KS11]. Dies erfordert ein Umdenken in der Vorgehensweise, wie die Software entwickelt wird. Bei der Entwicklung werden mehrere Softwaresysteme zusammen mit den entsprechenden Schnittstellen betrachtet, die bei der Ausführung des Prozesses beteiligt sind. Dieses Umdenken muss den Entwicklern vermittelt werden, um sie in dem Änderungsprozess mitzunehmen und mögliche Widerstände im Vorfeld zu beseitigen. Widerstände können auftreten, wenn Mitarbeitende sich übergangen fühlen, die Änderungen ihnen aufgezwungen werden oder sie im Änderungsprozess kein Mitspracherecht haben. Dies kann zudem auftreten, wenn Mitarbeitende Angst haben Verantwortung bzw. Macht zu verlieren oder im Extremfall durch die Änderungen überflüssig zu werden.

Darüber hinaus können Verantwortliche in den IT-Abteilungen eine zentrale übergeordnete Einrichtung blockieren, die ihnen die Entscheidungsgewalt über die zu verantwortenden IT-Systeme entzieht. Dieser Widerstand kann so weit führen, dass die Umstellung auf eine neue Architektur scheitert [MSRM11]. Durch phasenspezifische Integrationsumgebungen behalten sie einen Teil dieser Verantwortung, weshalb im Vergleich zur zentralen Lösung

kein oder weniger Widerstand zu erwarten ist. In jedem Fall müssen alle Verantwortlichen und Betroffenen früh in geplante Änderungsprojekte eingeweiht und deren Vorteile aufgezeigt werden. Ebenso wichtig ist die Rolle der Mitarbeitenden in der neuen Organisationsstruktur zu verdeutlichen und ihnen zu zeigen, dass sie dadurch keine Verschlechterung ihrer Position erfahren.

Diese Änderungen erfordern die Zuständigkeiten für die Integrationsumgebung neu zu diskutieren, denn eine IT-Abteilung, die einem Bereich des Produktlebenszyklus zugeordnet ist, kann unmöglich die Koordination des gesamten Lebenszyklus übernehmen. Hierfür ist eine übergeordnete Abteilung wichtig, die nicht nur die IT-Abteilung aus den einzelnen Bereichen koordiniert, sondern auch die übergeordnete Integration umsetzen muss. Insbesondere ein globales Datenaustauschformat muss spezifiziert, sowie phasenübergreifende Prozesse definiert und implementiert werden. Die Firma IBM hat hierzu das *Center of Excellence for SOA* etabliert, einer Organisation die Best-practice-Beispiele, Wissen, sowie pragmatische und wegweisende Lösungen im Bereich der SOA fördert [Sub08]. Insbesondere die Service-Governance und das Service Life Cycle werden durch das Center of Excellence während der Definition und Ausführung unterstützt.

Bei Umsetzung der Architektur mit einem zentralen ESB müssten somit alle IT-Abteilungen über den gesamten Lebenszyklus hinweg koordiniert werden, um ein einheitliches Datenformat zu spezifizieren. Dieses Datenformat muss alle Informationen enthalten, die im Produktlebenszyklus anfallen. Innerhalb der gesamten Organisation einen Konsens für solch ein einheitliches Datenformat zu bekommen, gleicht der Entwicklung von internationalen Standards, die mehrere Jahre bis zu einer Verabschiedung diskutiert werden. Darüber hinaus ist die Pflege und Weiterentwicklung eine Herausforderung, denn das resultierende Datenformat wäre von enormer Komplexität, die mit heutigen Methoden nicht wartbar ist. Zusätzlich müssen in Abstimmung mit allen IT-Verantwortlichen Schnittstellen und Übersetzungsservices entwickelt und die

Prozesse implementiert werden. Allein die Entwicklung eines einheitlichen Datenaustauschformates, das alle Daten des gesamten Produktlebenszyklus abbildet, stellt Unternehmen bisher vor unlösbare Aufgaben.

Die Lösung mit mehreren ESBs profitiert davon, dass jeder ESB ein eigenes, innerhalb der Domäne einheitliches Datenaustauschformat verwendet. Jedes Datenformat muss deshalb nur die Daten modellieren, die innerhalb der jeweiligen Phase zwischen Anwendungen ausgetauscht werden. Somit verringert sich die Größe und Komplexität im Vergleich zu einem einzigen globalen Datenaustauschformat. Einen Konsens für ein phasenspezifisches Datenformat zu erhalten ist deshalb im Vergleich zum globalen Datenmodell einfacher und schneller möglich. Darüber hinaus ist die Kompetenz bei der Modellentwicklung in der entsprechenden Domäne am größten, weshalb dies innerhalb einer Domäne einfacher ist als Domänen übergreifend. Die Pflege und Weiterentwicklung des Datenformates erfordert aufgrund der geringeren Komplexität zudem weniger Aufwand. Zusätzlich können die IT-Verantwortlichen jeder Phase weitgehend unabhängig ihren phasenspezifischen ESB und dessen zugehörige Komponenten entwickeln und müssen sich z. B. nur bei Daten und Prozessen, die mit anderen Phasen geteilt werden, mit der übergeordneten IT-Organisation abstimmen.

Durch die Trennung der Verantwortlichkeiten (engl.: separation of concerns) in phasenspezifische und globale Integrationsumgebungen kann die IT-Architektur gut in hierarchische Organisationsformen oder Matrixorganisationen eingebettet werden. Zudem können Entscheidungen innerhalb einer Domäne schneller getroffen werden, wodurch insgesamt die Flexibilität der Lösung erhöht wird.

## **4.3 Adaptive und wandlungsfähige IT-Architektur für Produktionsunternehmen**

In diesem Abschnitt wird die adaptive und wandlungsfähige IT-Architektur präsentiert und detailliert beschrieben. Dabei steht die flexible Integration des erweiterten Produktlebenszyklus im Vordergrund, deren technische Umsetzung im Detail vorgestellt wird. Die Beschreibung der Datenaustauschformate und Serviceverzeichnisstruktur wird diskutiert, sowie die dynamische Nachrichtenvermittlung in der IT-Architektur und die entsprechende Prozessunterstützung.

### **4.3.1 Eine adaptive und wandlungsfähige IT-Architektur (ACITA)**

Die Entwicklung einer *adaptiven und wandlungsfähigen IT-Architektur für Produktionsunternehmen*, auf englisch *adaptable and changeable IT-Architecture (ACITA)*, erfordert eine einheitliche Integrationsumgebung für den Produktlebenszyklus, im Folgenden auch als PLM-Architektur bezeichnet. Dabei wird der in Abschnitt 4.2 diskutierte Ansatz mit mehreren ESBs umgesetzt, der durch seine hohe Modularität mehr Flexibilität bietet und den Aspekt der Wandlungsfähigkeit in die IT-Architektur aufnimmt. Damit erfüllt die Architektur die Anforderungen des heutigen turbulenten Umfeldes und erlaubt eine schnelle Anpassung auf jeder Ebene, sowohl im Kleinen bei der Anwendungsintegration innerhalb einer Domäne, als auch im Großen bei der domänenübergreifenden Integration.

Die serviceorientierte PLM-Architektur nutzt für jede Phase die aus dem MSB bekannte Integrationshierarchie, detailliert beschrieben in Abschnitt 3.5, um Anwendungen über standardisierte Serviceschnittstellen flexibel in

Geschäftsprozesse zu integrieren. Dies führt zu einer optimalen Adaption der Integrationsumgebung an die jeweiligen Anforderungen der Phasen, die in Abschnitt 4.2 ausgeführt werden und erlaubt die entsprechenden Anwendungen adaptiv einzubinden. Die domänenübergreifende Integration der phasenspezifischen ESBs erfolgt über den PLM-Bus. Dieser dient als zentrale Integrationskomponente für alle phasenspezifischen ESBs und verhindert deren Punkt-zu-Punkt Integration, wodurch eine reduzierte Komplexität der Gesamtarchitektur erreicht wird. Darüber hinaus werden Anwendungen, die in Produktionsunternehmen zum Einsatz kommen und nicht direkt einer Phase des Produktlebenszyklus zuzuordnen sind, wie z. B. ERP-Systeme, Gehalts- oder Steuersoftware, direkt an den PLM-Bus angebunden.

Die daraus resultierende ACITA besitzt drei Dimensionen, um alle Anwendungen des Produktlebenszyklus flexibel zu integrieren: (1) über die fünfstufige Integrationshierarchie wird eine adaptive Integration von Anwendungen bzw. deren Services ermöglicht, indem ein dynamischer Nachrichtenaustausch über einen CBR im entsprechenden ESB erfolgt. Dies erlaubt eine lose Kopplung von Services in Workflows, die die Geschäftsprozesse abbilden und ausführen. (2) Jede Phase wird getrennt über einen phasenspezifischen ESB integriert, um die Integrationsumgebung an die Anforderungen der entsprechenden Phase anpassen zu können. (3) Es wird zwischen den einzelnen Phasen und der übergeordneten phasenübergreifenden Integrationsumgebung des PLM-Busses differenziert, der alle phasenspezifischen ESBs integriert. Dies führt zu der in Abbildung 4.4 dargestellten adaptiven und wandlungsfähigen Architektur.

Der modulare Aufbau erlaubt es nur Teile der Gesamtarchitektur umzusetzen, sofern das Unternehmen nicht über alle Phasen des Produktlebenszyklus operiert oder Teile davon an externe Firmen auslagert [SDMM12, SDMM13]. Bei der Einbindung von externen Kunden, Lieferanten oder Dienstleistern in die Prozesse des Unternehmens ist der modulare Aufbau der Architektur ebenfalls von Vorteil. Vorausgesetzt die externen Unternehmen setzen genauso auf

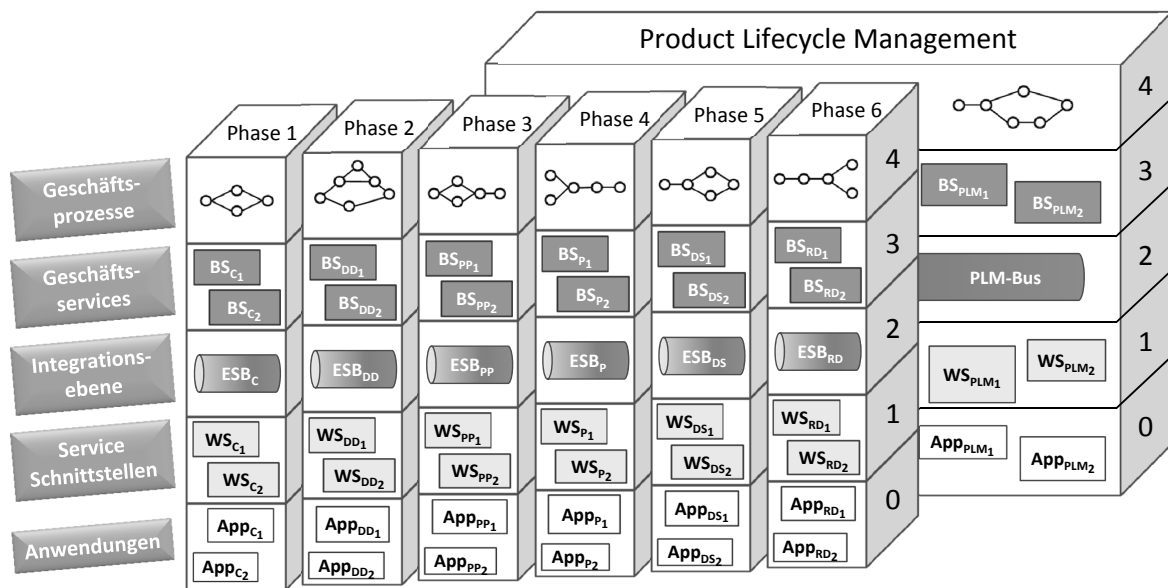


Abbildung 4.4: Servicebasierte Integrationsarchitektur für das Product Lifecycle Management [SDMM12]

eine SOA, die über einen oder mehrere ESBs realisiert ist, dann können die externen ESBs wie phasenspezifische ESBs über den PLM-Bus an die Integrationsumgebung des Unternehmens angeschlossen werden. Dies ermöglicht einen flexiblen Datenaustausch und adaptive, unternehmensübergreifende Prozesse mit den externen Partnern umzusetzen. Der einzige Unterschied bei der Integration über externe ESBs sind zusätzliche Sicherheitsvorkehrungen, um Missbrauch der Schnittstellen vorzubeugen und nur die gewünschten Services und somit Daten an die externen Partner freizugeben. Dies wird im Rahmen dieser Arbeit nicht weiter diskutiert, sollte aber Gegenstand weiterer Arbeiten sein. Somit kann die Anzahl an phasenspezifischen ESBs beliebig variieren. Durch die übergeordnete Integration mittels des PLM-Busses bleibt eine flexible Anpassung der Architektur gewährleistet, wodurch die Gesamtarchitektur wandlungsfähig wird. Die Erweiterung der ACITA auf das gesamte Produktionsunternehmen wird anhand eines ganzheitlichen Managementmodells in Abschnitt 4.4 beschrieben.



### **4.3.2 Umsetzung der ACITA zur Integration des Produktlebenszyklus**

Um den spezifischen Anforderungen jeder Phase gerecht zu werden, findet die Integration der Anwendungen im ersten Schritt in jeder Phase getrennt voneinander statt. Im zweiten Schritt werden daraufhin die einzelnen phasenspezifischen ESBs über den PLM-Bus integriert, um dieselbe Flexibilität beim Datenaustausch und der Prozessunterstützung zu erhalten, wie auf der phasenspezifischen Ebene.

#### **4.3.2.1 Datenaustauschformate**

Die Unabhängigkeit des Nachrichtendatenformates von den proprietären Datenformaten der einzelnen Anwendungen des Produktlebenszyklus wird durch die Verwendung eines einheitlichen Datenformates sichergestellt, das detailliert in Abschnitt 2.2.3.1 beschrieben ist. Die Definition eines einzigen, einheitlichen und unternehmensweiten Datenformates ist jedoch ein bisher ungelöstes Problem [HW03]. Die Größe und Komplexität eines solchen Datenformates, sowie die Anzahl der zu integrierenden Domänen mit ihren unterschiedlichen Datenstrukturen und Termini verhindert bisher die Definition eines solchen übergreifenden Datenschemas.

Die modulare Struktur der ACITA erlaubt es jedoch, für jeden ESB ein einheitliches phasenspezifisches Datenaustauschformat zu definieren, das innerhalb der einzelnen Phasen die Unabhängigkeit des Nachrichtenformates mit den integrierten Anwendungen sicherstellt und damit eine Abstraktionsebene zwischen den einzelnen Anwendungen darstellt. Die Größe und Komplexität der einzelnen Datenformate wird damit reduziert und durch die domänenspezifische Fokussierung die Definition des Datenschemas vereinfacht. Der PLM-Bus benutzt ein weiteres Datenaustauschformat, um Nachrichten zwischen den phasenspezifischen ESBs austauschen zu können. Dieses, im

Folgenden als Global bezeichnetes Datenformat, muss nur Informationen speichern können, die zwischen den Phasen ausgetauscht werden. Somit sind die Größe und Komplexität des globalen Datenformates im Vergleich zu einem einheitlichen und unternehmensweiten Datenformat reduziert, da Informationen, deren Austausch nur innerhalb einer Phase erfolgt, nicht berücksichtigt werden müssen.

Durch den hierarchischen Ansatz der ACITA bilden die spezifischen Datenformate und das globale Datenformat eine identische Hierarchie, die in ähnlicher Form von Davies et al. bereits vorgeschlagen wurde, um eine unternehmensweite Integration zu erreichen [DSRR08]. Dies reduziert die Komplexität bei der Definition der Datenaustauschformate und erhöht dadurch deren Wart- und Erweiterbarkeit.

Da unterschiedliche Datenaustauschformate in der Integrationsarchitektur zum Einsatz kommen, ist es notwendig Übersetzungsservices bereit zu stellen, die Nachrichten aus einem phasenspezifischen Datenformat in das phasenübergreifende Datenformat übersetzen und umgekehrt. Für jede am PLM-Bus angeschlossene Phase wird ein Übersetzungsservice für jede Richtung benötigt, wobei die Nachricht vor dem Absenden bereits in das verwendete Datenaustauschformat des PLM-Busses bzw. der phasenspezifischen ESBs übersetzt wird. Damit müssen die phasenspezifischen ESBs nur einen Übersetzungsservice zur Transformation von Nachrichten in das globale Nachrichtenformat bereitstellen. Der PLM-Bus hingegen benötigt für jede Phase einen entsprechenden Übersetzungsservice.

#### **4.3.2.2 Serviceverzeichnisse**

Die vollständige Integration aller Anwendungen des Produktlebenszyklus erfordert die Bereitstellung von einer oder mehrerer Serviceschnittstellen für jede Anwendung. Diese große Anzahl an Services wird deshalb effizient in einem verteilten Serviceverzeichnismodell verwaltet. Für die optimale

Verteilung aller Services innerhalb der PLM-Architektur sind verschiedene Serviceverzeichnismodelle denkbar. Das resultierende Verzeichnismodell soll entsprechend dem SOA-Dreieck, beschrieben in Abschnitt 2.2.2, für die Serviceanbieter und Servicenutzer transparent sein und wie ein zentrales Serviceverzeichnis erscheinen. Um Services zu veröffentlichen oder abzufragen wird damit nur ein Zugangspunkt in jeder Phase benötigt. Die Verwaltung der Services muss das Modell selbstständig übernehmen, so dass alle Services des Produktlebenszyklus auffindbar sind und wenn gewünscht, abgefragt werden können. Aus diesem Grund werden in diesem Abschnitt verschiedene Modelle für verteilte Serviceverzeichnisse diskutiert und bewertet, um das geeignetste Modell zu ermitteln. Der folgende Abschnitt und die darin enthaltene Diskussion basieren dabei auf der Veröffentlichung von Silcher et al. [SKRM13].

Unabhängig des in Abschnitt 2.2.2.4 vorgestellten allgemeinen Serviceverzeichnisses unterscheidet diese Arbeit zwischen drei verschiedenen Typen von Verzeichnissen, um die unterschiedlichen Verzeichnismodelle zu realisieren:

- Lokales Verzeichnis
- Vollständiges Verzeichnis
- Meta-Verzeichnis

Das lokale Verzeichnis speichert nur Servicebeschreibungen von Services, die in derselben Phase des Produktlebenszyklus verfügbar sind. Somit sind mehrere lokale Verzeichnisse erforderlich, um alle Services registrieren zu können. Im vollständigen Verzeichnis werden im Gegensatz zum lokalen Verzeichnis die Servicebeschreibungen aller Services des gesamten Lebenszyklus registriert. Das Meta-Verzeichnis speichert im Gegensatz zu den zuvor genannten Verzeichnissen nicht die gesamte Servicebeschreibung eines Service, sondern nur Informationen, in welcher Phase sich der jeweilige Service befindet. Dies hat Vorteile, wenn z. B. ein Serviceendpunkt geändert wird.

Diese Information wird nicht im Meta-Verzeichnis gespeichert, weshalb die Änderungen nur im lokalen Verzeichnis durchzuführen sind.

Durch die unterschiedlichen Verzeichnisse ergeben sich verschiedene Möglichkeiten für das verteilte Serviceverzeichnismodell der PLM-Architektur. Zum einen kann die Anzahl der Verzeichnisse variiert werden, zum anderen deren Position in der Architektur. Das Ziel von Nachrichten wird durch den CBR bestimmt, der im ESB integriert und in Abschnitt 2.2.3.2 beschrieben ist. Aus diesem Grund werden die Verzeichnismodelle der Struktur der ACITA bzw. der ESB-Hierarchie entsprechen, um ein effizientes Auffinden der Services sicher zu stellen.

Kang et al. definieren zwei Servicetypen, die in lokale Services und globale Services aufgeteilt sind [KLC06]. Ein Service ist lokal, wenn er nur auf einem Rechner läuft. Ein globaler Service arbeitet hingegen nach dem Client/Server Prinzip und ist somit über mehrere Rechner verteilt. In dieser Arbeit werden die Begriffe des lokalen und globalen Services hingegen anders verwendet. In der ACITA ist ein lokaler Service nur innerhalb einer Phase verfügbar und kann von Services einer anderen Phase nicht aufgerufen werden. Ein globaler Service hingegen ist von jedem Service innerhalb der Integrationsarchitektur verwendbar.

Ad-UDDI ist ein hierarchisches Verzeichnismodell mit drei Ebenen und wurde von Du et al. entwickelt [DHL05]. In den oberen zwei Ebenen befinden sich Ad-UDDI Verzeichnisse, in der untersten Ebene hingegen Services. Die Services veröffentlichen ihre Servicebeschreibung in einem oder mehreren Ad-UDDIs auf der mittleren Ebene, abhängig von deren Typ und Industrieklassifikation. Ad-UDDI Verzeichnisse auf der mittleren Ebene, die dieselbe Industrieklassifikation besitzen, tauschen gegenseitig Informationen über sogenannte Nachbarschaftsbeziehungen aus.

Eine weitere hierarchische Serviceverzeichnisstruktur wird von Xu et al. in Form einer Infrastruktur zum Auffinden von weit verteilten Services beschrie-

ben [XNW01]. Zum Auffinden von Services werden sogenannte Discover Server verwendet, die vergleichbar mit Serviceverzeichnissen sind und einer hierarchischen Organisation folgen. In jeder Domäne wird ein Discover Server eingesetzt, der alle Anfragen aus derselben Domäne verarbeitet. Die Servicebeschreibungen jedes Discover Servers werden in der Hierarchie nach oben propagiert und dort aggregiert. Eine Anfrage von einem Serviceaufrufer an den lokalen Discover Server wird direkt beantwortet, sofern der Service lokal gefunden wird. Ist die Beschreibung lokal nicht verfügbar, so wird die Anfrage an weitere Discover Server auf darüber liegenden Ebenen weitergeleitet bis ein entsprechender Service gefunden wird. Anschließend wird die Anfrage an alle Discover Server auf den darunter befindlichen Ebenen gesendet, um die detaillierten Servicebeschreibungen abzufragen, die letztendlich an den Serviceaufrufer gesendet werden.

Im Serviceverzeichnismodell für den Produktlebenszyklus sollen im Gegensatz zu den eben vorgestellten hierarchischen Verzeichnismodellen die phasenspezifischen Verzeichnisse alle lokalen Anfragen bearbeiten, ohne Nachrichten an andere Domänen weiterzuleiten. Die Frage in welchem Verzeichnis welche Serviceinformationen gespeichert sind, wird in der folgenden Diskussion geklärt.

Zur Speicherung aller Services des Produktlebenszyklus sind verschiedene Modelle entwickelt worden bzw. wurden aus der Literatur abgeleitet und auf die ACITA übertragen. Die Modelle sind in Abbildung 4.5 dargestellt und werden im Folgenden beschrieben, um sie im Anschluss zu vergleichen, zu bewerten und das für die ACITA am besten geeignete auszuwählen.

- **Lokale Verzeichnisse:** Das Modell mit lokalen Verzeichnissen setzt in jeder Phase ein Verzeichnis ein, wie in Abbildung 4.5 (a) dargestellt, um die lokal installierten Services zu registrieren und anschließend abzufragen. Somit können Anfragen nach lokalen Services direkt beantwortet, Serviceanfragen nach Services aus anderen Phasen müssen

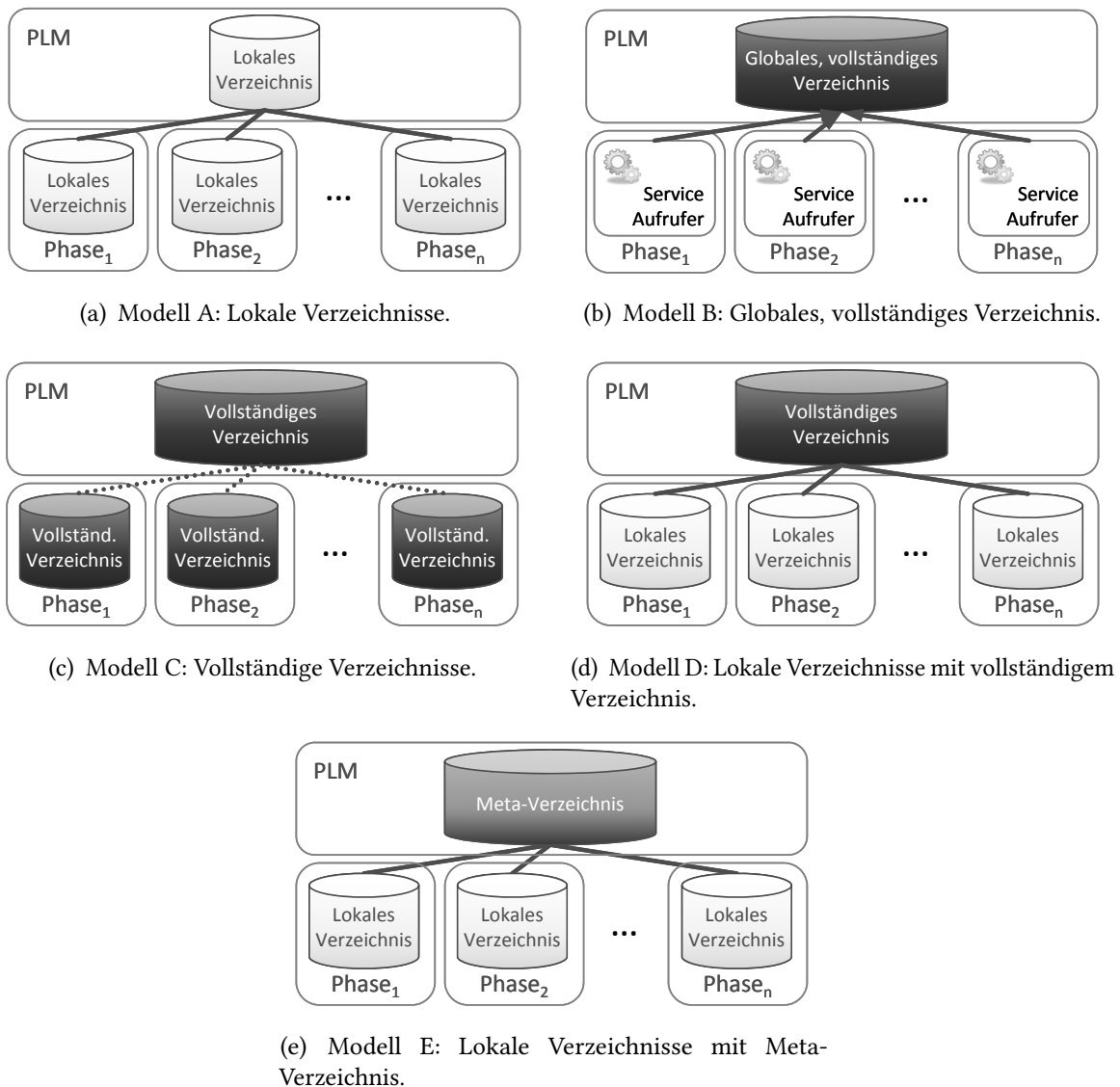


Abbildung 4.5: Verteilte Serviceverzeichnisse für die ACITA (vergleiche [SKRM13])

jedoch an die anderen Phasen weiter geleitet werden. Dies kann entweder sequenziell geschehen, indem eine Anfrage an alle anderen lokalen Serviceverzeichnisse nacheinander nach dem gesuchten Service erfolgt oder durch eine Broadcastnachricht. Die sequentielle Lösung führt zu einer längeren Antwortzeit, wenn der Service erst im letzten Verzeichnis gefunden wird. Bei der Broadcastnachricht wird die Anfrage an alle anderen lokalen Serviceverzeichnisse parallel weitergeleitet, wodurch sich jedoch die Anzahl an Anfragen bei jedem Verzeichnis erhöht und

bei vielen phasenübergreifenden Anfragen zu einer Überlastung von Verzeichnissen führen kann.

Mukkamala et al. präsentieren eine dezentrale Serviceverzeichnisstruktur, die Services beliebig auf mehrere lokale Serviceverzeichnisse verteilt [MAW06]. Eine Serviceanfrage wird in dieser Infrastruktur immer an eines dieser lokalen Verzeichnisse gestellt. Ist die Anfrage nach der Servicebeschreibung erfolgreich, so wird direkt eine Antwort zurückgegeben. Andernfalls wird mittels Multicast die Nachricht an alle anderen lokalen Serviceverzeichnisse innerhalb einer Gruppe weitergeleitet, die im Erfolgsfall mit der gesuchten Servicebeschreibung antworten. Die Redundanz, die durch Speicherung von Serviceinformationen in mehreren Verzeichnissen erreicht wird, bietet Vorteile bei einem Serviceaufruf, da mehrere Verzeichnisse die gewünschte Information liefern können. Bei einer Serviceänderung müssen jedoch alle Verzeichnisse kontaktiert werden, da nicht bekannt ist welches Verzeichnis welche Serviceinformationen speichert. Darüber hinaus verwenden Mukkamala et al. gleichwertige Verzeichnisse und bilden nicht die hierarchische Struktur der ACITA ab.

- **Globales, vollständiges Verzeichnis:** Dieses Verzeichnismodell entspricht dem UDDI, das alle Services in einem zentralen Verzeichnis registriert und bereits in Abschnitt 2.2.2.4 beschrieben ist [OAS05]. In Grafik 4.5 (b) wird das globale, vollständige Verzeichnis in die ACITA eingeordnet. Daher befinden sich in den einzelnen Phasen keine Verzeichnisse und alle Anfragen müssen an das globale Verzeichnis im PLM-Bus weitergeleitet werden. Updates von Servicebeschreibungen sind einfach im globalen Verzeichnis durchzuführen. Als einziges Serviceverzeichnis kann es allerdings zu Performanzproblemen kommen, da alle Serviceanfragen des gesamten Produktlebenszyklus durch dieses Verzeichnis verarbeitet werden müssen. Zudem stellt ein einzelnes Verzeichnis einen Single-Point-of-Failure dar, das die ganze Servicein-

frastruktur zum Erliegen bringt, sollte dieses Verzeichnis nicht mehr erreichbar sein.

- **Vollständige Verzeichnisse:** In diesem Verzeichnismodell kommen statt den lokalen Verzeichnissen im ersten hier beschriebenen Modell, jeweils vollständige Verzeichnisse zum Einsatz, wie in Abbildung 4.5 (c) gezeigt. Deshalb kann jede Anfrage innerhalb dieses Verzeichnismodells lokal beantwortet werden, da jeweils alle Services des gesamten Produktlebenszyklus gespeichert sind. Aufwendiger ist dagegen das Ändern von Servicebeschreibungen, denn dabei müssen alle vollständigen Verzeichnisse aktualisiert werden.

Eine Verzeichnisstruktur mit mehreren vollständigen Verzeichnissen wird von Sun et al. vorgeschlagen, bei der sich die Serviceverzeichnisse zyklisch aktualisieren [SLK04]. Dabei besitzt jedes Verzeichnis einen Vorgänger und Nachfolger. Alle Änderungen, die ein Verzeichnis von seinem Vorgänger erhält werden auf die eigenen Serviceinformationen angewendet. Anschließend wird die Nachricht um Änderungen ergänzt, die von außen in der Zeit seit der letzten gesendeten Nachricht an Servicebeschreibungen durchgeführt wurden. Diese wird dann an den Nachfolger weitergeleitet, der ebenso verfährt. Erfolgt von außen keine Änderungen, so muss eine Änderungsnachricht zweimal im Kreis geschickt werden, bevor alle Änderungen in jedem Verzeichnis übernommen sind. Dieser Ansatz zur Aktualisierung ist sehr einfach. Problematisch wird es jedoch, wenn die Verzeichnisse während der Aktualisierung nicht für Anfragen gesperrt sind, denn dadurch können Inkonsistenzen auftreten. Auch dieser Ansatz bildet den hierarchischen Ansatz der ACITA nicht ab.

- **Lokale Verzeichnisse mit vollständigem Verzeichnis:** Dieses Modell kombiniert die Modelle der lokalen und der vollständigen Verzeichnisse, indem in jeder Phase ein Serviceverzeichnis alle lokalen Services speichert, wie in Abbildung 4.5 (d) dargestellt. Auf der übergeordneten



PLM-Ebene wird dagegen ein vollständiges Serviceverzeichnis verwendet, in dem alle im gesamten Lebenszyklus verfügbaren Services registriert sind. Durch diese Kombination von unterschiedlichen Verzeichnistypen können die Vorteile der zuvor aufgeführten Verzeichnismodelle kombiniert und die Nachteile dadurch beseitigt werden. Dieses Modell erlaubt Serviceanfragen, die sich in derselben Phase befinden, lokal zu verarbeiten, wobei für phasenübergreifende Anfragen eine Abfrage der Serviceinformationen am übergeordneten vollständigen Verzeichnis ausreicht. Ein Update von Serviceinformationen muss nur im entsprechenden lokalen und dem vollständigen Serviceverzeichnis durchgeführt werden.

- **Lokale Verzeichnisse mit Meta-Verzeichnis:** Das letzte hier beschriebene Verzeichnismodell unterscheidet sich zum Vorangegangenen nur durch die Verwendung eines Meta-Verzeichnisses auf der phasenübergreifenden Ebene, anstatt eines vollständigen Verzeichnisses. Dieses Modell ist in Abbildung 4.5 (e) zu sehen. Dies bedeutet, dass auf der lokalen Ebene ausschließlich die in der entsprechenden Phase verfügbaren Services in den lokalen Verzeichnissen gespeichert sind und auf der globalen Ebene im Meta-Verzeichnis alle zur Verfügung stehenden Services registriert sind. Allerdings sind diese nicht mit den kompletten Serviceinformationen, sondern nur mit dem Servicenaamen und der Phase, in der der Service installiert ist gespeichert. Eine phasenübergreifende Serviceanfrage wird somit etwas komplexer, da das Meta-Verzeichnis nicht direkt den Endpunkt eines Services zurückgibt, sondern nur die Phase eines gesuchten Services. Deshalb muss die Nachricht zuerst an die entsprechende Phase weitergeleitet und dort eine Abfrage des lokalen Serviceverzeichnisses durchgeführt werden, um den Endpunkt zu bestimmen und die Nachricht zuzustellen. Das Update eines Services wird hingegen etwas einfacher, da Beschreibungsänderungen, wie z. B. der Endpunkt eines Services, nicht

im Meta-Verzeichnis gespeichert ist und deshalb nur ein Update des lokalen Verzeichnisses notwendig ist.

Die Unterschiede zum Modell mit lokalen Verzeichnissen und einem vollständigen Verzeichnis sind jedoch nicht sehr groß. Der Hauptvorteil des Meta-Verzeichnisses ist, dass es keine Details über Services speichert. Dies wirkt sich insbesondere dann aus, wenn Services von externen Partnern in die ACITA eingebunden werden und die Partner durch das Meta-Verzeichnis keine Details über ihre IT-Infrastruktur bekannt geben müssen. Stattdessen ist nur der Endpunkt ihres eigenen Routers notwendig, hinter dem sich die restliche Infrastruktur verstecken lässt. Auch Änderungen hinter diesem zentralen Zugangspunkt lassen sich somit transparent für andere Unternehmen durchführen.

Einen Ansatz mit mehreren Meta-Verzeichnissen wurde von Kassim et al. entwickelt, bei dem jedes Meta-Verzeichnis nur Serviceinformationen über den Ort des entsprechenden Serviceverzeichnisses speichert [KEMS07]. Die detaillierte Servicebeschreibung wird anschließend aus dem zuvor bestimmten Serviceverzeichnis abgerufen. Dieses Vorgehen stimmt mit dem zuvor genannten Meta-Verzeichnismodell überein. Allerdings wird bisher nur ein Meta-Verzeichnis zur Speicherung aller Services des Produktlebenszyklus verwendet. Zudem wird erst am Meta-Verzeichnis angefragt, wenn zuvor eine Abfrage des lokalen Verzeichnisses ohne Erfolg blieb.

Das am besten geeignete Verzeichnismodell für die ACITA wird anhand von verschiedenen Kriterien ermittelt. Diese Kriterien sind Performanz der Serviceverzeichnisse bei Anfragen, der Aufwand für Serviceänderungen, die Skalierbarkeit des Modells, die Vertraulichkeit bei Einbindung von externen Services und die Erreichbarkeit bei Verzeichnisausfällen oder Netzwerkfehlern. Im Folgenden wird für jedes Verzeichnis anhand der wichtigsten Kriterien aufgezeigt, warum ein Verzeichnis für die Umsetzung nicht ausgewählt wird. Dabei werden nicht für jedes Modell alle Kriterien betrachtet:

- **Das Modell mit lokalen Verzeichnissen** zeichnet sich durch einen sehr geringen Aufwand bei Serviceänderungen aus. Serviceaufrufe können unter Umständen jedoch eine hohe Last erzeugen, da im schlechtesten Fall alle Serviceverzeichnisse angefragt werden müssen, bis ein Service gefunden wird. Dieser Aufwand reduziert die Performanz von Verzeichnissen, in deren Phase sich weder der Serviceaufrufer noch der Service anbietende befindet, wodurch unnötige Last entsteht, die zu vermeiden ist.
- **Das globale, vollständige Verzeichnis** bietet den Vorteil, dass Serviceanfragen und Serviceänderungen nur an einem Verzeichnis durchgeführt werden müssen. Dies bringt gleichzeitig den Nachteil mit sich, dass alle Anfragen des gesamten Produktlebenszyklus von diesem Verzeichnis zu verarbeiten sind, was eine sehr hohe Last bedeutet und deshalb die Performanz bei einer zu hohen Anzahl an Anfragen oder bei einer zu schlecht dimensionierten Hardware leidet. Zudem bildet dieses Verzeichnis einen Single-Point-of-Failure, der bei Nichterreichbarkeit dazu führt, dass keine Services mehr gefunden werden und die Integrationsarchitektur somit vollständig ausfällt. Darüber hinaus ist die geforderte Anpassungsfähigkeit der PLM-Architektur bei diesem Modell nur durch eine Skalierung der Hardware möglich.
- **Das vollständige Verzeichnismodell** bietet eine sehr gute Performance bei der Serviceabfrage, da alle Anfragen lokal behandelt werden können. Fehler eines Serviceverzeichnisses führen nur teilweise zur Unerreichbarkeit von Services, da alle Informationen in jeder Phase repliziert vorliegen. Serviceänderungen hingegen erfordern mehr Aufwand, da die Servicebeschreibungen in allen Verzeichnissen geändert werden müssen. Um Inkonsistenzen während der Verteilung von Serviceänderungen zu vermeiden, sollten Protokolle zur verteilten Synchronisation verwendet werden. Letztendlich ist jedoch fraglich, ob Informationen über alle Services in jeder Phase verfügbar sein müssen,

denn nur ein Teil der Services wird überhaupt aus anderen Phasen aufgerufen. Bei der Integration von externen Services werden zudem vertrauliche Informationen über die gesamte Serviceverzeichnisstruktur verteilt, die die Partner meistens jedoch mit anderen Parteien nicht teilen wollen.

- Die Nachteile der bisher genannten Modelle werden teilweise durch **das Modell mit lokalen Verzeichnissen und einem vollständigen Verzeichnis** überwunden. Zum einen werden Serviceaufrufe wie im lokalen Modell innerhalb der Phase verarbeitet. Sollte ein Service nicht gefunden werden, reicht hingegen eine Anfrage am vollständigen Verzeichnis aus, um die Servicebeschreibung zu erhalten. Diese Anfrage wird automatisch vom lokalen Verzeichnis durchgeführt. Damit ist die Last am vollständigen Verzeichnis im Vergleich zum globalen, vollständigen Verzeichnis geringer. Zum anderen müssen Serviceänderungen nur in zwei Verzeichnissen durchgeführt werden, im entsprechenden lokalen Verzeichnis und dem vollständigen Verzeichnis.
- **Das Modell mit lokalen Verzeichnissen und dem Meta-Verzeichnis** verhält sich fast identisch mit dem zuvor genannten Modell, das statt dem Meta-Verzeichnis ein vollständiges Verzeichnis verwendet. Die Serviceabfrage aus einer anderen Phase ist durch eine weitere Anfrage am lokalen Verzeichnis der Zielphase etwas aufwendiger. Änderungen von Serviceinformationen die nicht im Meta-Verzeichnis gespeichert sind, z. B. eines Endpunktes, werden dagegen einfacher, da sie nur im lokalen Verzeichnis durchzuführen sind. Der große Vorteil bei Verwendung des Meta-Verzeichnisses ist die Vertraulichkeit, die dadurch geschaffen wird. Bei Einbindung von externen Service, z. B. von Zulieferern, Kunden oder Dienstleistern, müssen diese nicht die Details ihrer Services bereit stellen, sondern nur den Endpunkt ihres CBR angeben, hinter dem sich ihre Infrastruktur mit allen Services verbirgt. Insbesondere im Unternehmensumfeld sind

die Sicherheit und Vertraulichkeit von enormer Bedeutung, weshalb nur die nötigsten Daten an andere Unternehmen weitergegeben werden sollten, wie es durch das Meta-Verzeichnis ermöglicht wird.

Alle vorgestellten Modelle bieten einen Single-Point-of-Access für den Serviceaufrufer innerhalb der jeweiligen Phase. Die eigentlich verteilten Serviceverzeichnisse bleiben dem Nutzenden dabei verborgen. Dadurch erscheint diesem das Modell wie ein einziges Verzeichnis, wodurch die Veröffentlichung und Abfrage von Services für den Nutzenden der Infrastruktur wie im SOA-Dreieck beschrieben abläuft.

Die ersten drei vorgestellten Modelle haben jeweils einen großen Nachteil, der in den Bereichen der Serviceabfrage, Performanz und Serviceänderung liegt. Die zwei zuletzt beschriebenen Modelle mit lokalen Serviceverzeichnissen innerhalb der Phasen und einem vollständigen Verzeichnis bzw. dem Meta-Verzeichnis auf der globalen Ebene bieten einen guten Kompromiss in all den geforderten Kriterien. Diese Verzeichnisse werden durch die Verarbeitung von lokalen Anfragen innerhalb der Phase entlastet. Änderungen müssen in maximal zwei Verzeichnissen durchgeführt werden und durch Hinzu- oder Wegnahme von lokalen Verzeichnissen kann das Gesamtmodell entsprechend der ACITA angepasst werden. Da beide Modelle nur geringe Unterschiede aufzeigen, wurde für die Umsetzung in einem Prototyp das Modell mit Meta-Verzeichnis gewählt, da es bei Einbindung von externen Services deren exakten Endpunkt gegenüber dem Meta-Verzeichnis verbirgt. Dies ist heutzutage aus Sicherheits- und Vertraulichkeitsaspekten von enormer Bedeutung.

#### **4.3.2.3 Dynamische Nachrichtenvermittlung**

Die dynamische Nachrichtenvermittlung basiert auf dem Content-based Router (CBR), der in Abschnitt 2.2.3.2 detailliert beschrieben ist. In der ACITA wird in jedem ESB ein CBR implementiert, um Nachrichten dynamisch an das

richtige Ziel bzw. die richtigen Ziele weiterzuleiten. Die CBRs in den phasenspezifischen ESBs sind identisch implementiert, da alle phasenspezifischen CBRs Nachrichten entweder innerhalb einer Phase an Services zustellen oder an den globalen CBR senden, wenn der Service aus einer anderen Phase ist. Dies ermöglicht den CBR in anderen Phasen wiederzuverwenden, ohne Anpassungen der Implementierung vornehmen zu müssen. Lediglich eine Änderung des einheitlichen Nachrichtenformatschemas ist erforderlich. Der CBR auf der globalen Ebene des PLM-Busses funktioniert hingegen etwas anders, da er die Nachrichten, die er von einem phasenspezifischen CBR erhält, an einen anderen phasenspezifischen CBR weiterleiten muss. Somit benötigt er Informationen über alle an den PLM-Bus angeschlossenen phasenspezifischen ESBs, die er in einer zusätzlichen Tabelle seiner Datenbank speichert.

Ein Nachrichtenaustausch zwischen Services läuft so ab, dass der aufrufende Service eine Nachricht an den lokalen CBR derselben Phase sendet. Zur eindeutigen Identifikation jeder Nachricht wird das Nachrichtenverzeichnis (engl.: Message Registry) verwendet. Jede Nachricht, die am CBR ankommt, wird an dieses Nachrichtenverzeichnis gesendet. Es wird dort überprüft, ob die Nachricht bereits eine eindeutige NachrichtenID (engl.: MessageID) besitzt. Sofern dies nicht der Fall ist wird eine neue NachrichtenID generiert und der XML-Nachricht hinzugefügt. Im Anschluss erfolgt eine Überprüfung, ob die Nachricht eine NachrichtenflussID (engl.: MessageFlowID) besitzt und wenn erforderlich eine neue vergeben. Die Nachricht wird nun serialisiert und zusammen mit den IDs in der Datenbank des Nachrichtenverzeichnisses gespeichert, wodurch der gesamte Nachrichtenaustausch, auch über Phasen hinweg, nachverfolgbar ist. Damit lässt sich im Fehlerfall die Suche nach der Ursache vereinfachen. Abschließend sendet das Nachrichtenverzeichnis die Nachricht zurück an den CBR.

Nach erfolgreicher Registrierung durch das Nachrichtenverzeichnis bestimmt der CBR dynamisch das Ziel einer Nachricht mithilfe von Regeln. Die Re-

geln sind in seiner Datenbank hinterlegt, wobei die Bedingung anhand des Inhalts einer Nachricht durch XPath-Ausdrücke ausgewertet wird. Wird ein XPath-Ausdruck positiv ausgewertet, so wird eine Liste mit den entsprechenden Zielsystemen zurückgegeben, für die die Nachricht bestimmt ist. Die konkreten Endpunkte der Ziele werden im Anschluss durch eine Anfrage an das lokale Serviceverzeichnis bestimmt. Das Serviceverzeichnis speichert alle Informationen zu Services der entsprechenden Phase ab und antwortet mit den gesuchten Informationen, sofern ein entsprechender Eintrag in der Datenbank vorhanden ist. Der CBR fügt der XML-Nachricht den vom Serviceverzeichnis erhaltenen Endpunkt hinzu und sendet die Nachricht anschließend an die Zielanwendungen bzw. Zielservices.

Sollte der Service nicht in derselben Phase wie der aufrufende Service verfügbar sein, so findet das lokale Serviceverzeichnis keinen entsprechenden Eintrag in seiner Datenbank. In diesem Fall wird die Anfrage vom lokalen Serviceverzeichnis direkt an das übergeordnete Meta-Verzeichnis weitergeleitet. Im Meta-Verzeichnis sind alle Services der Anwendungen des gesamten Produktlebenszyklus registriert. Das Meta-Verzeichnis liefert im Erfolgsfall die Phase des gesuchten Services zurück an das lokale Verzeichnis, das die Antwort an den CBR zurückgibt. Dieser CBR sendet die Nachricht, in der die Zielphase vorher eingetragen wird, an den globalen CBR des PLM-Busses. Der globale CBR wiederum leitet die Nachricht an den lokalen CBR der Zielphase weiter, der mithilfe seines lokalen Serviceverzeichnisses den Endpunkt des gesuchten Services bestimmt und die Nachricht im Anschluss zustellt. Das Ablaufdiagramm zum Aufruf eines Services in einer anderen Phase zusammen mit den beteiligten Diensten ist in Abbildung 4.6 dargestellt.

Im Meta-Verzeichnis werden alle Services unabhängig davon registriert, ob es sich um lokale oder globale Services handelt, damit global gesehen keine Konflikte der Servicebezeichner entstehen. Eine Anfrage an das Meta-Verzeichnis kann folglich aus zwei Gründen nicht erfolgreich sein. Zum einen kann ein gesuchter Service im Meta-Verzeichnis registriert sein, der jedoch

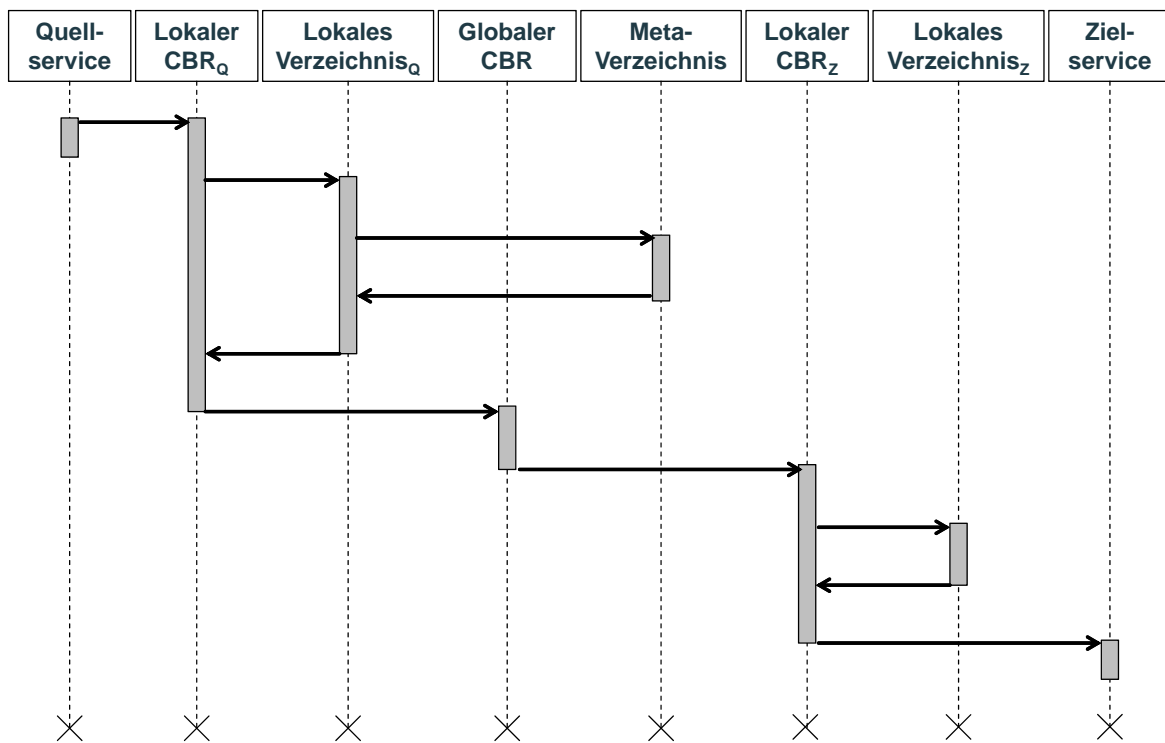


Abbildung 4.6: Aufruf eines Services in einer anderen Phase des Produktlebenszyklus

als lokaler Service gekennzeichnet ist. Da lokale Services nicht aus anderen Phasen aufrufbar sind, wird eine entsprechende Fehlermeldung an das lokale Verzeichnis bzw. den CBR zurückgeschickt, die an den aufrufenden Service weitergeleitet wird. Zum anderen kann es passieren, dass ein gesuchter Service nicht im Meta-Verzeichnis registriert ist. In diesem Fall wird genauso eine entsprechende Fehlermeldung an den aufrufenden Service gesendet.

Für den nahtlosen Nachrichtenaustausch zwischen Phasen müssen die verwendeten unterschiedlichen Nachrichtenformate übersetzt werden. Dies geschieht durch Übersetzungsservices, die an den Phasengrenzen jede Nachricht vom lokalen in das globale Nachrichtenmodell oder umgekehrt übersetzen. Damit ist gewährleistet, dass jeder CBR nur Nachrichten erhält, die dem Datenformat der jeweiligen Phase entsprechen.



#### 4.3.2.4 Prozessunterstützung

Durch die Bereitstellung von Services über den gesamten Produktlebenszyklus hinweg lassen sich Prozesse durch die Definition von Workflows in standardisierten Sprachen wie BPEL oder BPMN unterstützen, die in Abschnitt 2.2.2.5 beschrieben sind. Dabei werden Services aus einzelnen oder unterschiedlichen Phasen zu einem Workflow komponiert und anschließend durch eine Workflow-Engine, die im ESB integriert ist, zur Ausführung gebracht. Der Aufruf eines Workflows erfolgt durch Auftreten eines Events oder manuell durch den Aufruf eines Benutzenden.

In welcher Phase ein Workflow installiert wird, ist theoretisch egal, da jeder Workflow über einen Serviceendpunkt angesprochen wird, der in einem Serviceverzeichnis registriert ist. Dieser ist somit von überall in der Architektur erreichbar. Praktisch sollte jeder Workflow in der Domäne installiert sein, die ihn komponiert hat und die Wartung übernimmt. Bei Workflows die innerhalb einer Domäne ausgeführt werden, ist dies die entsprechende Phase. Bei phasenübergreifenden Workflows sollte der Workflow im PLM-Bus installiert werden, da mehrere Domänen einen Service bereitstellen und somit bei Änderungen nur die lokalen Workflows derselben Phase und des PLM-Busses auf ihre Funktionsfähigkeit getestet werden müssen. Workflows aus anderen Phasen bleiben somit von Änderungen unberührt.

## 4.4 Ganzheitliches Managementmodell für Produktionsunternehmen

Die in Kapitel 2.1 vorgestellten Managementkonzepte PLM, FLM und SCM werden heute in den meisten Unternehmen eingesetzt, um u. a. Informationen und Prozesse zu verwalten. Die Literatur betrachtet diese Konzepte meist nicht gemeinsam. Dies wäre aber notwendig, um klar die Zuständigkeiten,

Abhängigkeiten und Synchronisationspunkte innerhalb und zwischen den Lebenszyklen bzw. der Lieferkette definieren zu können. Darüber hinaus ist die Unterstützung der Aufgaben und Prozesse aller Konzepte durch IT-Systeme und deren Kooperation von zentraler Bedeutung. Die folgende Diskussion beruht auf der Veröffentlichung von Silcher et al. [SSZM13].

Der Produkt- und Fabriklebenszyklus wird typischerweise in Phasen aufgeteilt, die die verschiedenen Zustände des Produktes bzw. der Fabrik im Laufe ihres Lebens durchschreiten. In der Literatur sind diese Phasenmodelle jedoch sehr heterogen, da sie sich in der Anzahl der Phasen bzw. deren Granularität unterscheiden und oft Phasen aus anderen Konzepten enthalten.

#### **4.4.1 Lebenszyklen und Managementmodelle**

Im Bereich der Lebenszyklen und Managementkonzepte ist eine klare Trennung zwischen dem Produktlebenszyklus, Fabriklebenszyklus und der Supply Chain erforderlich. Deren Phasenmodelle sind in der Literatur jedoch sehr unterschiedlich, weshalb die Wichtigsten im Folgenden vorgestellt werden, um daraus ein ganzheitliches Managementmodell für Produktionsunternehmen abzuleiten.

##### **4.4.1.1 Produktlebenszyklus**

Um die Evolutionsstufen, die ein Produkt durchläuft, zu unterscheiden, wird der Produktlebenszyklus typischerweise in unterschiedliche Phasen eingeteilt. Das weit verbreitete Konzept des PLM zeigt bei der Phasenaufteilung die größte Heterogenität in der Literatur in Bezug auf die Anzahl und den Inhalt der einzelnen Phasen.

Eine Unterteilung des Produktlebenszyklus in fünf Phasen stellt Stark vor [Sta11]. Das in Abbildung 4.7 dargestellte Phasenmodell ist mit ebenso fünf



Abbildung 4.7: Phasenmodell des Produktlebenszyklus [SSZM13]

Phasen daran angelehnt. Hervorzuheben ist, dass alle Phasen ausschließlich die Zustände des Produktes beschreiben. Die erste Phase *'Konzept'* umfasst alle Tätigkeiten, die vor der konkreten Umsetzung des Produktes ausgeführt werden, z. B. Forschung, Innovationsmanagement, Marktforschung und Kundenbefragungen. In der zweiten Phase *'Design & Entwicklung'* wird ein konkretes Produkt entwickelt, das ein neues Produkt sein kann oder eine Weiterentwicklung eines bestehenden Produktes. In dieser Phase werden u. a. Aufgaben wie Konstruktion, Elektronik- und Softwareentwicklung, Produktsimulation und Funktionstests durchgeführt. In der dritten Phase *'Produktion'* wird das Produkt hergestellt und die Qualität geprüft. Die Nutzung des hergestellten Produktes erfolgt in der vierten Phase *'Verwendung & Support'*. Darüber hinaus erfolgt die Kundenbetreuung, Reklamationsbearbeitung und Ersatzteilverwaltung in dieser Phase. Die fünfte und letzte Phase *'Außerbetriebstellung & Entsorgung'* sorgt für eine angemessene Wiederverwendung, Recycling oder Entsorgung des Produktes.

Andere Phasenmodelle haben im Vergleich zu Stark nicht die Fokussierung auf das Produkt, sondern erweitern den Produktlebenszyklus um Phasen, wie z. B. der Produktionsplanung. Einen Lebenszyklus aus acht Phasen beschreibt Schuh [Sch06]: *'Produktplanung'*, *'Konstruktion'*, *'Arbeitsvorbereitung'*, *'Produktionsplanung'*, *'Fertigung/Montage'*, *'Vertrieb'*, *'Service'* und *'Recycling'*. Dabei enthält der Lebenszyklus mit der Produktionsplanung nicht nur eine Phase, die dem Fabriklebenszyklus zuzuordnen ist, auch der Vertrieb ist primär ein Bestandteil des SCM. Eigner und Stelzer verwenden für ihren Produktlebenszyklus sieben Phasen, die vergleichbar zum Phasenmodell von Schuh sind [ES09]: *'Anforderungen'*, *'Produktplanung'*, *'Entwicklung'*, *'Prozessplanung'*, *'Produktion'*, *'Betrieb'* und *'Recycling'*. Auch hier findet sich mit der Prozessplanung eine Phase wieder, die dem Fabriklebenszyklus

zuzuordnen ist. Ein weiteres Phasenmodell von Vajna et al. ist mit elf Phasen sehr detailliert [VWBZ09]: 'Forschung', 'Vertrieb/Marketing', 'Formgebung', 'Entwicklung', 'Konstruktion', 'Prozessplanung', 'Fertigungssteuerung', 'Fertigung', 'Versand', 'Nutzung' und 'Recycling/Entsorgung'. Neben der Prozessplanung und Fertigungssteuerung, die dem Lebenszyklus der Fabrik zuzuordnen sind, findet sich in diesem Lebenszyklus der Versand als Teil der Lieferkette.

#### **4.4.1.2 Fabriklebenszyklus**

Das Konzept des Fabriklebenszyklus ist nicht so weit verbreitet, wie der Produktlebenszyklus. Dennoch gibt es für den Fabriklebenszyklus unterschiedliche Phasenmodelle. Westkämper et al. präsentieren ein Phasenmodell aus fünf Phasen: 'Design und Planung', 'Konstruktion', 'Betrieb und Wartung', 'Modernisierung oder Veralterung', 'Demontage und Rückbau' [WCH06].

Ein etwas detaillierterer Fabriklebenszyklus wird in weiteren Arbeiten von Constantinescu et al. beschrieben [CEW09, CEF09]. Im Detail wird der Lebenszyklus in die Phasen 'Produktentwicklung', 'Investitions- und Fabrikleistungsrechnung', 'Standort und Netzwerkplanung', 'Gebäude-, Infrastruktur- und Medienplanung', 'Interne Logistik und Layoutplanung', 'Prozess- und Anlagenplanung und Arbeitsplatzgestaltung', 'Inbetriebnahme und Projektmanagement', 'Fabrikbetrieb' und 'Instandhaltungs- und Anlagenmanagement' unterteilt. Im Vergleich zum Fabriklebenszyklus von Westkämper et al. bindet dieser auch Phasen ein, die sich mit finanziellen, organisatorischen, geographischen und logistischen Aspekten beschäftigen. Die Aspekte überschneiden sich jedoch mit anderen Konzepten, wie z. B. die Logistik mit dem SCM oder die Investitionsrechnung mit der Unternehmenssteuerung.

Daraus kann abgeleitet werden, dass die einzelnen Aufgaben zum einen klar voneinander abzugrenzen sind, zum anderen aber Schnittstellen zwischen



Abbildung 4.8: Phasenmodell des Fabriklebenszyklus [SSZM13]

den einzelnen Phasen und Konzepten benötigt werden, um diese gegenseitig abzustimmen und zu synchronisieren.

Das in Abbildung 4.8 dargestellte Phasenmodell ist an Westkämper et al. angelehnt und besteht aus fünf Phasen. In der Phase *'Fabrikentwicklung'* werden der Fabrikstandort, Gebäude und Medienversorgung geplant und Finanzierungs- und Kostenrechnungen aufgestellt. Die Phase *'Produktionsplanung'* enthält Aufgaben zur Planung der Produktionsprozesse, Zuordnung der benötigten Ressourcen, Simulation des Produktionsablaufes und eine entsprechende Optimierung des selbigen. Die *'Produktion'* der Produkte erfolgt in der dritten Phase. Die Phase *'Service & Wartung'* dient der Instandhaltung und Modernisierung der Fabrik, um möglichst ohne ungeplante Ausfälle eine kontinuierliche Produktion zu erreichen. Die letzte Phase *'Demontage & Rückbau'* bildet das Ende einer Fabrik mit deren Abriss.

Hummel und Westkämper stellen mit der Kombination von Produkt- und Fabriklebenszyklus das sogenannte Kanonenmodell vor [HW07]. Beide Lebenszyklen sind in sieben Phasen eingeteilt, wobei sie sich in der fünften Phase, der Produktion, kreuzen. Der Produktlebenszyklus ist unterteilt in *'Planung'*, *'Entwicklung'*, *'Konstruktion'*, *'Rapid Prototyping'*, *'Produktion'*, *'Gebrauch & Service'* und *'Recycling'*. Der Fabriklebenszyklus wird in *'Investitionsplanung'*, *'Engineering'*, *'Prozessplanung'*, *'Aufbau & Anlauf'*, *'Produktion'*, *'Service & Wartung und Nutzung'* und *'Demontage & Rückbau'* geteilt. Der Vorteil des Kanonenmodells ist die anschauliche Darstellung der Zusammenhänge und Abhängigkeiten der beiden Lebenszyklen bzw. der entsprechenden Phasen.



Abbildung 4.9: Phasenmodell des Designs der Lieferkette [SSZM13]

#### 4.4.1.3 Lieferkette

Die Versorgung der Fabrik mit Material und Teilen, sowie das Liefern der fertigen Produkte an die Kunden, wird im Rahmen des SCM organisiert. Dabei kommen heutzutage oft mehrstufige Lieferketten zum Einsatz, die durch logistische Methoden, wie just-in-time oder just-in-sequence, bis ins Detail geplant sind. Durch die Verkleinerung oder Abschaffung von Material- und Teilelagern kommt der Logistik eine enorme Bedeutung zu, denn diese hat die Prozesse in einem Produktionsunternehmen korrekt zu verwalten und aufeinander abzustimmen.

Die Original Equipment Manufacturers (OEMs) setzen heute auf eine mehrstufige Lieferkette, bei denen ein noch höherer Koordinationsbedarf besteht. Dieser Bedarf ist nicht nur während des Aufbaus des Lieferanten- und Kundennetzwerks vorhanden, sondern auch nach dem Start der Produktion, da es ständig Veränderungen gibt. Dies wird durch das in Abbildung 4.9 dargestellte Phasenmodell für die Lieferkette widergespiegelt.

#### 4.4.2 Ganzheitliches Managementmodell

Erfolgreiche Unternehmen müssen die in den vorangegangenen Abschnitten beschriebenen Konzepte des PLM, FLM und SCM beherrschen, um am Markt bestehen zu können und im Optimalfall der Konkurrenz einen Schritt voraus zu sein. Neben der isolierten Betrachtung der Konzepte besteht die Herausforderung in deren Verzahnung, um die verschiedenen Teilprozesse parallel auszuführen und somit die Dauer des Gesamtprozesses zu verkürzen. Dies

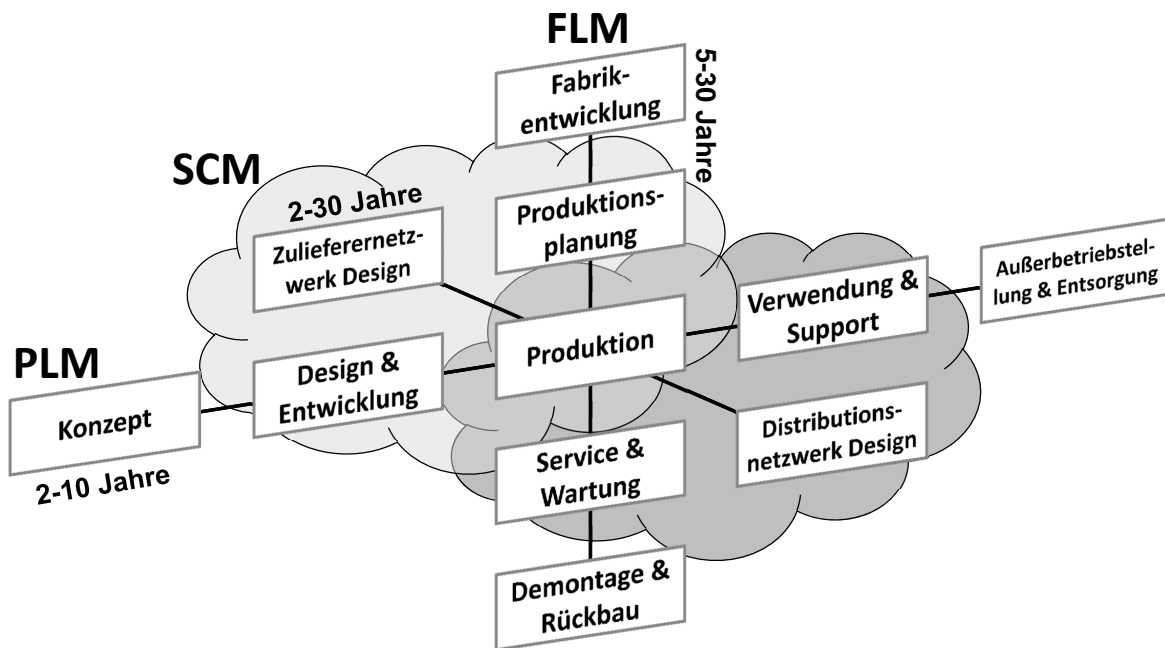


Abbildung 4.10: Ganzheitliches Managementmodell für Produktionsunternehmen [SSZM13]

erfordert jedoch eine klare Trennung der einzelnen Phasen, um Zuständigkeiten und die Übergänge zwischen den Phasen genau definieren zu können. Aus diesem Grund wurde ein ganzheitliches Managementmodell entwickelt, das in Abbildung 4.10 dargestellt ist. Es erweitert das Kanonenmodell von Hummel und Westkämper um den Aspekt des Supply Chain Management, genauer gesagt um das Konzipieren des Zulieferer- und Distributionsnetzwerkes.

Die in der Abbildung eingezeichneten Wolken stellen den Bedarf an Koordination zwischen Phasen des Produkt- und Fabriklebenszyklus und der Lieferkette dar, der durch das ganzheitliche Modell verdeutlicht werden soll. Insbesondere in den Phasen vor und nach der Produktion entsteht ein hoher Abstimmungsaufwand, um nach Möglichkeit die unterschiedlichen Prozesse parallel auszuführen und somit Zeit und Kosten einzusparen. Die Inkonsistenz der Prozesse in isolierten Abteilungen und IT-Systemen erschweren die Koordination jedoch. Verschiedene Mentalitäten der Mitarbeitenden können darüber hinaus die Definition und Ausführung von domänenübergreifenden

Prozessen verhindern. Diese technischen und organisatorischen Grenzen zwischen verschiedenen Abteilungen gilt es zu überwinden, um eine höhere Effizienz und somit Zeit- und Kosteneinsparungen zu erreichen.

Zu beachten sind die unterschiedlichen zeitlichen Dimensionen der Phasenmodelle. Ein Produkt wird in der Regel für zwei bis zu zehn Jahre hergestellt, bevor es durch ein neues oder verbessertes Produkt ersetzt wird. Fabriken hingegen überdauern fünf bis zu dreißig Jahre. Somit wird klar, dass in einer Fabrik bis zu ihrem Abriss unterschiedliche Produkte hergestellt werden. Diese Tatsache muss bei der Planung der Fabrik berücksichtigt werden, um Änderungen der Produktionslinien und Medienversorgung einfach durchführen zu können. Eine Lieferantenbeziehung kann von kurzer Dauer sein und sich evtl. nur auf zwei Jahre erstrecken, sie kann allerdings theoretisch auch ewig bzw. bis zum Rückbau der Fabrik andauern. Unabhängig davon müssen z. B. Liefermengen und Liefertermine für jedes Produkt und jede Fabrik individuell mit jedem Lieferanten abgestimmt werden, dies bringt jedoch einen erheblichen Informationsaustausch mit sich.

Die Unterstützung der Prozesse in diesem ganzheitlichen Managementmodell durch IT-Systeme steht im Fokus dieser Arbeit. So wird eine Vielzahl von Softwareanwendungen zur Unterstützung von Aufgaben in jeder Phase eingesetzt. Die größten Softwarehersteller und ihre Anwendungen sind in Abschnitt 3.6 beschrieben. Keiner dieser Hersteller unterstützt alle Aufgaben, die im ganzheitlichen Managementmodell enthalten sind, vollständig.

Aus diesem Grund wird ein adaptiver Integrationsansatz für das ganzheitliche Managementmodell benötigt, der einen flexiblen Datenaustausch zwischen den Softwaresystemen erlaubt und zudem eine einfach anzupassende Unterstützung von Prozessen ermöglicht. Die Wandlungsfähigkeit der ACITA erlaubt nicht nur den Produktlebenszyklus auf adaptive Weise effizient zu integrieren, sondern kann auch auf das ganzheitliche Managementmodell übertragen werden. Dabei muss eine Vielzahl an domänenspezifischen ESBs für die Integration der unterschiedlichen Phasen entwickelt werden, wie in



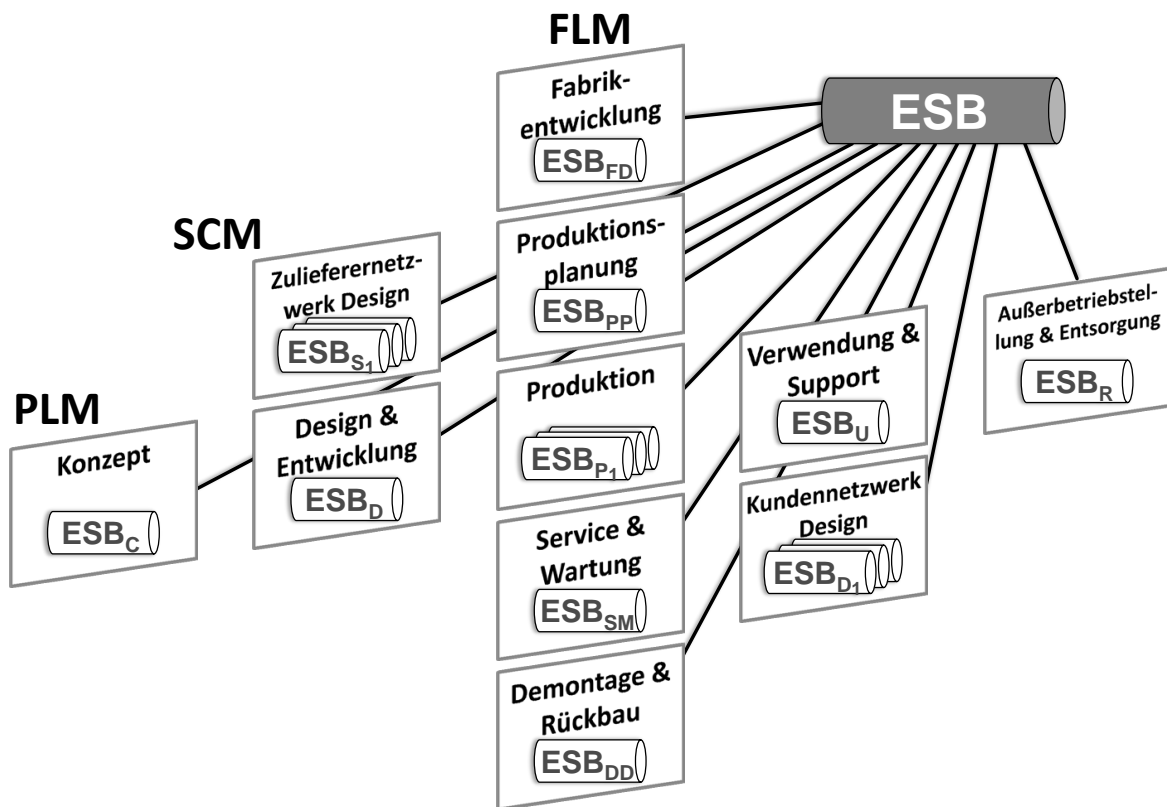


Abbildung 4.11: Unterstützung des ganzheitlichen Managementmodell durch die ACITA [SSZM13]

Abbildung 4.11 dargestellt ist. Die fünfstufige Integrationshierarchie jeder Phase ist aus Gründen der Übersichtlichkeit nicht abgebildet, sondern wird jeweils durch einen entsprechenden ESB repräsentiert.

Die Abbildung verdeutlicht, dass innerhalb einer Domäne mehrere phasen-spezifische ESBs eingesetzt werden können. Dies kann durch die Integration von externen Partnern, wie z. B. Zulieferern oder Kunden der Fall sein, aber auch in Domänen der Produktion, bei der in jeder Fabrik ein ESB eingesetzt wird, um notwendige Echtzeitanforderungen durch die lokale Installation zu erreichen.

## 4.5 Zusammenfassung

In diesem Kapitel ist die Abhängigkeit zwischen den Mitarbeitenden, IT-Systemen, Methoden und Prozessen dargestellt und deren Auswirkungen auf die Adaptivität und Wandlungsfähigkeit erläutert. Insbesondere bei der Abhängigkeit zwischen den Softwareanwendungen und Prozessen zeigt sich, dass die Prozesse, die in unflexiblen IT-Systemen implementiert sind, zusätzlich an Flexibilität verlieren. Um die Adaptivität der Prozesse zu erhalten, müssen die Softwaresysteme flexibler werden. Dies gilt nicht nur für einzelne IT-Systeme, sondern auch für deren Integration.

Zur Verbesserung der Adaptivität von IT-Landschaften wird das Konzept der SOA auf den Produktlebenszyklus übertragen. Der ESB als zentrale Integrationskomponente bietet dabei die gewünschte flexible Integration von Services. Der Ansatz mit mehreren phasenspezifischen ESBs erlaubt, gegenüber der Verwendung eines einzigen zentralen ESBs, eine höhere Adaptivität an die Anforderungen der unterschiedlichen Domänen des Produktlebenszyklus. Dies führt zur ACITA, die durch ihre Modularität und hierarchische Struktur der Integrationslandschaft eine einfache Anpassungsfähigkeit ermöglicht, wodurch die Anforderungen der Wandlungsfähigkeit erfüllt sind.

Die Entkopplung von Anwendungen und Services erfolgt durch die Nutzung von phasenspezifischen Nachrichtenmodellen innerhalb jeder Phase und einem globalen Datenaustauschformat auf der PLM-Ebene. Die dynamische Nachrichtenvermittlung wird durch den Einsatz von einem CBR in jedem ESB gewährleistet. Die CBRs ermöglichen die Abfrage aller Servicebeschreibungen über das jeweilige lokale Serviceverzeichnis. Um Services aus anderen Phasen des Produktlebenszyklus aufzurufen, ist jedes lokale Serviceverzeichnis mit dem Meta-Verzeichnis verbunden, das Informationen über alle installierten Services speichert. Der einfache Zugriff auf Services über Domänengrenzen hinweg erlaubt die flexible und phasenübergreifende Definition von Prozessen auf Basis von standardisierten Workflowsprachen,

wie z. B. BPEL. Diese können über die in ESBs integrierte Workflow Engine ausgeführt werden.

Die ACITA kann durch ihre hohe Modularität nicht nur eine effiziente Integration des PLM unterstützen, sondern auch weitere Domänen von Produktionsunternehmen über diesen Ansatz einbinden, die durch das FLM und SCM abgedeckt werden. Dazu ist aus unterschiedlichen Phasenmodellen des PLM, FLM und SCM ein ganzheitliches Managementmodell abgeleitet worden, das die Domänen identifiziert, unter denen ein hoher Koordinationsbedarf besteht. Dieser Bedarf kann durch die ACITA in Bezug auf Informationsaustausch und domänenübergreifende Prozesse flexibel unterstützt werden.



---

## **KAPITEL 5**

# **Prototypische Implementierung der ACITA im Produktlebenszyklus**

---

Dieses Kapitel stellt die prototypische Implementierung der ACITA in der Lernfabrik aIE vor. Hierfür wird zuerst die Lernfabrik selbst mit ihrer digitalen Lernumgebung und physischen Modellfabrik beschrieben. Im Anschluss wird die prototypische Implementierung des Production-planning Service Bus (PPSB), einem phasenspezifischen ESB für die Produktionsplanung, zur robusten Integration der heterogenen und verteilten Anwendungen der digitalen Lernumgebung gezeigt. Darauf aufbauend erfolgt die Implementierung des PLM-Busses, der die beiden Phasen der Produktion mit dem MSB und der Produktionsplanung mit dem PPSB integriert, um flexibel Daten zwischen beiden Phasen austauschen zu können.

### **5.1 Integrationsumgebung Lernfabrik**

Die Lernfabrik für advanced Industrial Engineering (aIE) am Institut für Industrielle Fertigung und Fabrikbetrieb (IFF) dient zum einen zu Schulungszwecken, um Fabrikplanern Methoden, Werkzeuge und Best-practice-Beispiele zu vermitteln. Zum anderen für die Forschung, um neue Methoden und

Werkzeuge für das aIE in einer realen Umgebung umzusetzen und anschließend evaluieren zu können [HW07]. Das aIE selbst erweitert das klassische Industrial Engineering (IE) um Methoden und Werkzeuge für wandelbare Unternehmensstrukturen zur Unterstützung der variantenreichen Serienproduktion, die im Stuttgarter Unternehmensmodell (SUM) beschrieben sind [RHW06]. Dazu besteht die Lernfabrik aus einer digitalen Lerninsel, in der mit verschiedensten digitalen Werkzeugen die Produktion eines Produktes geplant und optimiert werden kann. Die Umsetzung geschieht in der physischen Montagelinie, die durch ein modulares System schnell anpassbar ist und die Planungen der digitalen Lernumgebung in kurzer Zeit verifizieren kann [RKHW06].

Als Produkt wird in der Lernfabrik ein Schreibtischset verwendet, das den Anforderungen einer hohen Varianz gerecht wird [DRW11]. Das Schreibtischset, das in Abbildung 5.1 dargestellt ist, erreicht eine hohe Anzahl an Varianten, indem eine Grundplatte mit drei identisch großen Bohrungen verwendet wird, in die verschiedene Teile gefügt werden können, wie z. B. Becher, Lampen oder Uhrenaufnahmen. Durch die Varianz der Grundplatte, der Teile und deren Position auf der Grundplatte ergeben sich über 16.000 Varianten des Schreibtischsets [DRW11]. Alle diese Varianten können in der Montagelinie der physischen Fabrik zusammengebaut werden.

Das Ziel der Lernfabrik aIE ist eine kundenindividuelle Serienfertigung in einem turbulenten Unternehmensumfeld umzusetzen. Das Umfeld macht dabei häufige Neuplanungen und Anpassungen notwendig, die Unterstützung durch die digitalen Werkzeuge erfahren. Um diese digitale Unterstützung weiter zu verbessern ist eine nahtlose Integration der heterogenen Softwaresysteme erforderlich, sowie eine Integration der digitalen und physischen Welt mittels einer einheitlichen IT-Architektur. Darüber hinaus wird eine einheitliche IT-Unterstützung des Planungsprozesses in der Lernumgebung benötigt, um die Planungszeiten zu verkürzen, bei gleichzeitig höherer Qualität der Planungsergebnisse [BDS12a]. Dies wird durch die Integration der

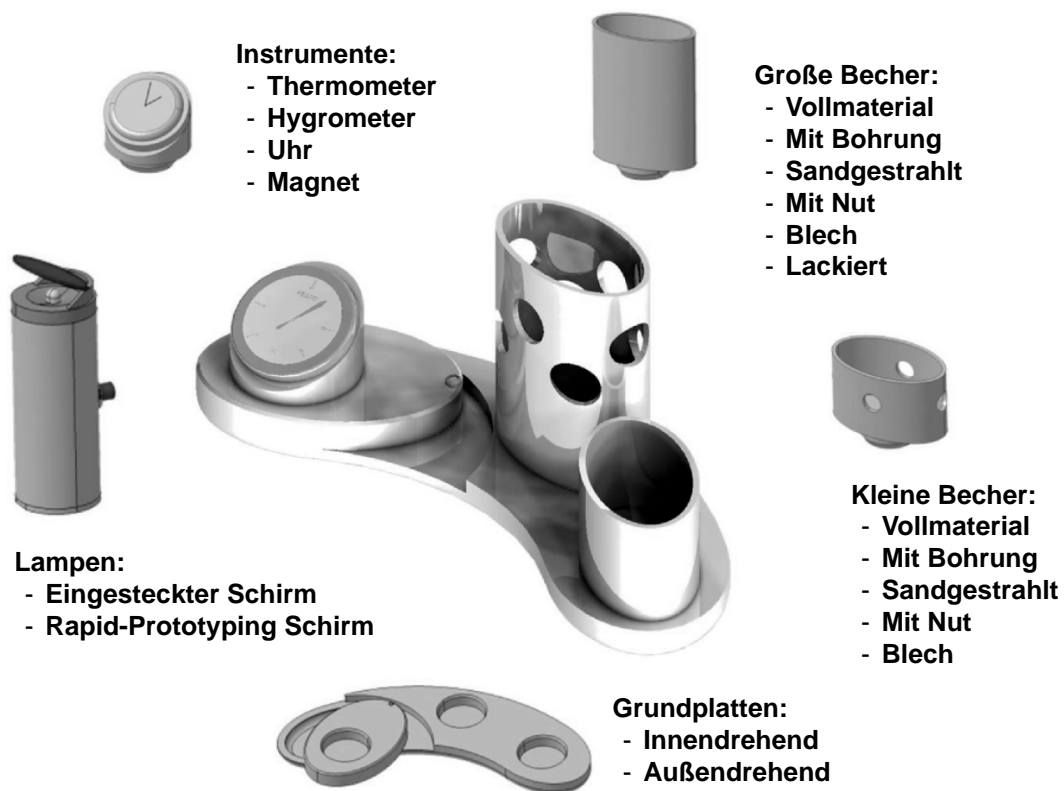


Abbildung 5.1: Teile und Varianten des Schreibtischsets [BDS12a]

verschiedenen digitalen Werkzeugen der Planungsumgebung und der realen Fabrik erreicht [RKHW08]. Diese Integration ist notwendig, um die Planungsmethoden und ihre Werkzeuge effizient einsetzen zu können [BDS12b].

Im Folgenden werden die digitale und physische Fabrik detaillierter beschrieben, um im Anschluss auf deren bisherige Integration einzugehen.

### 5.1.1 Die digitale Lernumgebung

Die digitale Lerninsel besteht aus verschiedensten Werkzeugen, die den Fabrikplaner bei seiner Arbeit unterstützen. Es werden Anwendungen der Firma Dassault Systèmes eingesetzt. Konkret handelt es sich dabei um den DPE, V5 DPM Assembly, Quest, V5 Robotics und V5 Human. Zusätzlich kommen Anwendungen zum Einsatz, die im Rahmen des SFB 467 entwickelt wurden

[RKHW06]. Dabei handelt es sich um den Fabrikplanungstisch, zur partizipativen Layoutplanung der Fabrikhalle [KLBW05], den Logistikprüfstand zur Simulation von Logistikprozessen [KLWW03] und den Montagekonfigurator zur Gestaltung und Optimierung des Arbeitsplatzes [KRHW07].

Die Anwendungen der Firma Dassault Systèmes sind bereits über Schnittstellen fest integriert und können untereinander Daten austauschen. Der Datenaustausch mit Herstellern von Drittsoftware oder selbst entwickelter Software gestaltet sich hingegen aufwendiger, da keine offenen Schnittstellen bestehen. Die Anwendungen des SFB 467 sind über die sogenannte Stuttgarter Integrationsplattform miteinander verbunden, die in Abschnitt 3.3 detailliert beschrieben ist.

### 5.1.2 Die physische Fabrik

Die physische Modellfabrik, genannt intelligent Transformable Assembly and Manufacturing Equipment (iTRAME), besteht aus verschiedenen Modulen, die durch ein Plug-and-Play Prinzip sehr einfach neu angeordnet werden können und in Abbildung 5.2 zu sehen ist. Die einzelnen Module werden in Reihe mittels eines Einheitssteckers verbunden, in den alle benötigten Medien, wie z. B. Strom, Netzwerk und Druckluft, integriert sind. Somit ist sehr schnell die Realisierung von neuen Layouts möglich, um Verbesserungen real testen und verifizieren zu können.

Jedes Modul besitzt zwei Transportebenen, auf der oberen Ebene findet die Bearbeitung des Produktes statt, auf der unteren Ebene der Rücktransport zum Beginn der Linie. An jedem Ende einer Linie befinden sich Lifte, die alle Ladungsträger, egal ob mit oder ohne Produkt, von der oberen auf die untere Ebene heben oder umgekehrt. Die zwei Transportebenen mit den Liften sorgen dafür, dass ein geschlossener Kreislauf entsteht, auf dem die Ladungsträger zirkulieren. Somit können Produkte in einem Montageprozess mehrmals an eine Station gelangen.





Abbildung 5.2: Die physische Modellfabrik iTRAME [BDS12a]

Die iTRAME-Anlage besteht für den Materialtransport aus Transportstrecken, Kurven, Weichen und Liften. Die Montage erfolgt an manuellen Handarbeitsplätzen und vollautomatischen Roboterzellen. Die Einlagerung von fertigen Produkten kann im automatischen Lager erfolgen. Zur Qualitätssicherung gibt es noch eine visuelle Prüfstation.

Die Steuerung der Fabrik erfolgt über einen Industrie-PC, auf dem eine angepasste Version der Software WinCC (Windows Control Center) der Firma Siemens installiert ist. Die Software verwendet zur Speicherung aller Daten der Modellfabrik eine Microsoft SQL-Datenbank, die im Folgenden auch Leitrechnerdatenbank genannt wird. Über diesen Leitrechner erfolgt auch die automatische Erkennung des Layouts der iTRAME-Anlage, das anschließend in dessen eben erwähnten Datenbank abgespeichert wird.

Durch die Modularität der Anlage können in der Zukunft neue Technologien und Verfahren einfach in die Montagelinie integriert werden [DRW11].

Ein Modul, das erst später zur iTRAME hinzukam ist eine Lackierzelle, die Becher in einer gewünschten Farbe beschichtet. Die Einbindung von neuen Technologien und Verfahren erhöht die Varianz der Montagelinie, erlaubt eine höhere Anzahl an Schreibtischsetvarianten und weist damit auch eine höhere Komplexität auf. Die Komplexität gilt es gerade in einem turbulenten Umfeld in zunehmendem Maße zu beherrschen.

### **5.1.3 Integration der digitalen Werkzeuge und der physischen Fabrik**

Die Integration aller Anwendungen der digitalen Fabrik erfolgt über den DPE, der alle PPR-Daten verwaltet und in seiner Datenbank, genannt PPR-Hub, speichert. Alle Delmia-Werkzeuge sind bereits über proprietäre Schnittstellen integriert und greifen über den DPE auf die Daten im PPR-Hub zu. Die Tools des SFB 467 werden über die sogenannte Stuttgarter Integrationsplattform an den DPE und PPR-Hub angebunden [RKHW06]. Diese Integrationsplattform basiert auf dem Change Propagation Manager oder kurz CHAMPAGNE, der heterogene und autonome Anwendungen integrieren kann, wie bereits in Abschnitt 3.3 erklärt ist [CHB<sup>+</sup>05].

Neben der CHAMPAGNE Plattform gibt es zusätzlich die Möglichkeit mittels VBA-Skripten, die im DPE implementiert sind und von dem im DPE integrierten Skripting Host ausgeführt werden, Daten mit anderen Anwendungen auszutauschen. Diese Möglichkeit erlaubt allerdings nur aus dem DPE heraus Daten des PPR-Hub zu lesen oder zu schreiben. Eine offene Schnittstelle von außen, um auf den DPE bzw. den PPR-Hub zuzugreifen, gibt es nicht.

Die physische Fabrik ist direkt mit der digitalen Planungsumgebung verbunden. Als Schnittstelle dient das MES der Firma PSI AG, genannt PSImes. Das PSImes ist wiederum mit dem Leitrechner der physischen Modellfabrik integriert. Somit ist eine durchgängige Datenübertragung zwischen

allen digitalen Werkzeugen der Lerninsel und der physischen Montagelinie gegeben, die den DPE als zentrale Datenaustauschplattform verwendet [RKK<sup>+</sup>07, RKHW08]. Das MES dient zum einen als Schnittstelle zwischen der digitalen und physischen Fabrik, zum anderen führt es die Feinplanung der Produktionsaufträge durch, die in die Montagelinie eingelastet werden.

Der Produktionsablauf wird über den Leitstand der Montagelinie iTRAME protokolliert und in dessen Datenbank gespeichert. Zur Auswertung der Produktion werden die Daten an das PSImes übertragen und dort visualisiert. Dazu können die Informationen in die Lerninsel, genauer gesagt in den PPR-Hub, übertragen werden, wo sie für Umplanungen und Neuplanungen der digitalen Werkzeuge zur Verfügung stehen und somit die Planungsqualität durch Abgleich von Soll- und Istzustand verbessern können.

Die durchgängige Integration der Anwendungen in der Lernfabrik aIE ist beispielhaft und zeigt das große Potential auf, das mithilfe des nahtlosen Datenaustausches erreicht werden kann. Zum einen führt dies zu einer großen Zeiteinsparung, wenn die Daten automatisch von einer Anwendung in eine andere übertragen werden. Zum anderen ist es möglich Informationen aus der Modellfabrik in die Lerninsel zu übertragen, um diese anschließend zu analysieren und die Erkenntnisse in die nächsten Planungsschritte einfließen zu lassen.

Die Lösung den DPE mit dessen PPR-Hub als zentrale Integrationskomponente einzusetzen, ähnelt dem Hub-and-Spoke Konzept mit einer zentralen Datenbank als Integrationsplattform zum Datenaustausch. Der Ansatz verhindert die Verwendung einer Punkt-zu-Punkt Integration, die schnell sehr komplex und somit nicht mehr wartbar wird. Der Zugriff auf diese Datenbank erfolgt allerdings ausschließlich über den DPE. Das Problem bei dieser Architektur ist, dass der DPE in der Integrationslösung der Lernfabrik einen Single-Point-of-Failure darstellt. Des Weiteren erlaubt das Datenmodell, das im PPR-Hub implementiert ist, dass alle Daten der Produktionsplanung dort gespeichert werden. Gleichzeitig können diese Daten mit anderen Anwen-

dungen ausgetauscht werden. Allerdings muss der Datenaustausch zwischen zwei Anwendungen immer über den DPE durchgeführt werden und kann nicht unabhängig von diesem direkt zwischen zwei Anwendungen ablaufen.

Der Datenaustausch erfolgt über individuelle VBA-Skripte, die über die Oberfläche des DPE gestartet werden. Aus diesem Grund können Anwendungen nicht direkt eingebunden werden, sondern tauschen ihre Daten stattdessen über VBA-Skripte aus. Die Daten werden dabei aus der Datenbank des DPE gelesen und in einer Datei oder anderen Datenbank gespeichert, auf die im Anschluss andere Anwendungen zugreifen können. Dadurch ist die Flexibilität der Lösung stark eingeschränkt, was der proprietären Natur des DPE geschuldet ist.

Die Integrationsplattform CHAMPAGNE dient ebenso zum Datenaustausch zwischen dem DPE und den Werkzeugen des SFB 467 [Hei11]. Die Propagationsskripte erlauben Änderungen in einer Anwendung automatisch in andere Anwendungen zu übertragen. Zur Überwachung der Änderungen in den Anwendungen sind Adapter und Hilfskomponenten notwendig. Sowohl die Propagationsskripte, als auch die Adapter und Hilfskomponenten erfordern Programmierkenntnisse und Wissen über die eingesetzten Programmier- und Modellierungssprachen. Dies erhöht den Aufwand bei Änderungen in den Anwendungen, da alle beteiligten Komponenten angepasst werden müssen. Das Fehlen eines einheitlichen Datenaustauschformates führt dazu, dass für jedes Anwendungspaar, das Daten austauschen will, individuelle Transformationsskripte benötigt werden.

Die Abhängigkeit vom DPE führt dazu, dass Änderungen und Erweiterungen nur mit viel Aufwand durchführbar sind. Dies liegt zum einen an der aufwendigen Anpassung des Datenmodells im PPR-Hub und zum anderen an den VBA-Skripten, deren Programmierung Fachkenntnisse des DPE und dessen Attributekonzeptes erfordern. Aus diesem Grund wird eine neue Integrationsplattform benötigt, die offener und flexibler im Gegensatz zur

proprietären und sehr starren DPE-Lösung ist. Auch CHAMPAGNE kommt nicht als Lösung zur Integration des gesamten Produktlebenszyklus in Frage, da diese Plattform ebenfalls stark auf dem DPE aufbaut und zudem nicht die gewünschte Adaptivität bereitstellt, insbesondere bei der Unterstützung der Prozesse. In der physischen Fabrik ist der MSB [Min12] als eine servicebasierte Integrationsumgebung erfolgreich umgesetzt worden. Die Verwendung von offenen Standards und einem Open Source ESB [Ope10] bietet die gewünschte Flexibilität und Unabhängigkeit von proprietärer Software. So kann jeder Fehler in der Modellfabrik über einen Leitrechnerservice aus der Leitrechnerdatenbank ausgelesen und in das Wartungsportal übertragen werden, in dem ein Wartungsingenieur entsprechende Aktionen zur Fehlerbehebung einleiten kann, die durch einen Workflow gesteuert sind.

Die Frage, den MSB zu erweitern, um die ganze Lernfabrik zu integrieren, oder einen Ansatz mit mehreren ESBs zu verwenden, wurde bereits in Abschnitt 4.2 diskutiert. Um die Integrationsplattform offen zu halten und auf weitere Domänen des Produktlebenszyklus erweitern zu können, fiel die Entscheidung auf eine Multi-ESB-Lösung, die in Abschnitt 4.3.1 präsentierte ACITA.

Für die Umsetzung dieser Architektur erfolgt im ersten Schritt die Integration von Anwendungen aus der Produktionsplanung über einen ESB, der PPSB genannt wird. Im zweiten Schritt werden die beiden phasenspezifischen ESBs, der MSB und der PPSB, über den PLM-Bus integriert. Diese Integration ist im Folgenden beschrieben.

## **5.2 Der Production-planning Service Bus (PPSB)**

In der Produktionsplanung kommen viele Anwendungen zum Einsatz, die zum einen heterogen und zum anderen verteilt sind. Aus diesem Grund muss

der Production-planning Service Bus (PPSB) eine robuste Infrastruktur bereitstellen, die die verteilten Anwendungen zuverlässig integriert. Die Anwendungen arbeiten auf teilweise denselben Daten, die sie aber in unterschiedlichen Datenstrukturen speichern. Um diese Daten unter den Anwendungen auszutauschen, wird deshalb ein einheitliches Nachrichtenformat für die Produktionsplanung benötigt. Dies reduziert darüber hinaus die Komplexität bei der Integration von neuen oder der Änderung von bereits angebotenen Anwendungen.

Unternehmen setzen heutzutage viele Anwendungen ein, die von unterschiedlichen Herstellern eingekauft sind oder selbst entwickelt werden. Diese Anwendungen sind oft nicht oder nur teilweise durch unflexible und proprietäre Punkt-zu-Punkt-Verbindungen integriert. Der Grund hierfür sind nicht vorhandene oder proprietäre Schnittstellen der Anwendungen, die eine einheitliche Integration nicht zulassen. Deshalb setzt der PPSB auf standardisierte Technologien wie die Web Services, um einheitliche Schnittstellen für den Datenaustausch bereit zu stellen und die heterogenen Anwendungen zu integrieren.

Die in Kapitel 4 vorgestellte ACITA besitzt gegenüber der bisherigen Lösung den Vorteil mit dem ESB eine plattformunabhängige Integrationsumgebung zu verwenden, die alle gängigen standardisierten Kommunikationsprotokolle unterstützt und somit nicht auf ein spezielles Protokoll beschränkt ist.

Die vollständige Unterstützung und Ausführung aller Prozesse durch IT-Systeme ist nicht möglich. Deshalb muss die Infrastruktur die Einbindung von manuellen Prozessschritten erlauben, die z. B. durch Ingenieure ausgeführt werden können, um ein optimales Zusammenspiel zwischen IT-Systemen und Mitarbeitenden zu ermöglichen.

Die Umsetzung der Integrationsumgebung für die Produktionsplanung erfolgt wie bereits erwähnt in der digitalen Planungsumgebung der Lernfabrik aIE mit ihrer heterogenen Anwendungslandschaft. Die neue Integrationsum-

gebung muss deshalb eine dezentrale Plattform zur Verfügung stellen, die auf standardisierten Technologien aufsetzt, wie es durch den plattformunabhängigen ESB gegeben ist. Für die Umsetzung des PPSB kam wie bereits beim MSB der OpenESB zum Einsatz, der eine Open Source Implementierung des ESB-Konzeptes ist [Ope10]. Zum einen steht der Weiterentwicklung dieses ESB durch seine offene Lizenz nichts im Wege, zum anderen ist der OpenESB kostenlos. Dies macht die Lösung neben den zuvor vorgestellten Vorteilen auch kostengünstig und kann grundsätzlich gegen eine andere ESB-Implementierung relativ einfach ausgetauscht werden.

Die fünfstufige Integrationshierarchie, die der MSB einführt und in Abschnitt 3.5 beschrieben ist, wird für die Umsetzung des PPSB übernommen. Damit wird eine adaptive Integration der Anwendungen über Serviceschnittstellen sichergestellt. Die Serviceschnittstellen ermöglichen wiederum durch ihre flexible Komposition zu Workflows eine anpassungsfähige Unterstützung der Geschäftsprozesse.

Um die verteilte Kommunikation zwischen den einzelnen Web Services und dem PPSB robuster gegen Fehler oder Netzwerkausfälle zu machen, werden Nachrichtenwarteschlangen eingesetzt, die bei der asynchronen Kommunikation zwischen Services die Nachrichten zwischenspeichern. Dadurch wird verhindert, dass Sender und Empfänger von Nachrichten beim Senden oder Empfangen blockiert werden, wenn die Gegenseite nicht verfügbar oder nicht bereit ist. Alle Services sind aus diesem Grund für den Empfang von Nachrichten mit jeweils einer Microsoft Message Queue (MSMQ) versehen. Die MSMQ ist die Implementierung einer Message Queue, deren Funktionsweise in Abschnitt 2.2.3.3 beschrieben ist.

Zur prototypischen Implementierung der Architektur in der Produktionsplanung sind Systeme über Warteschlangen und den PPSB integriert worden. Da der DPE als zentrale Datenspeicherkomponente nicht zur Verfügung stand ist ersatzweise eine SQL-Datenbank bereitgestellt worden, die alle relevanten PPR-Daten speichert. Auf die Daten dieser Datenbank und des

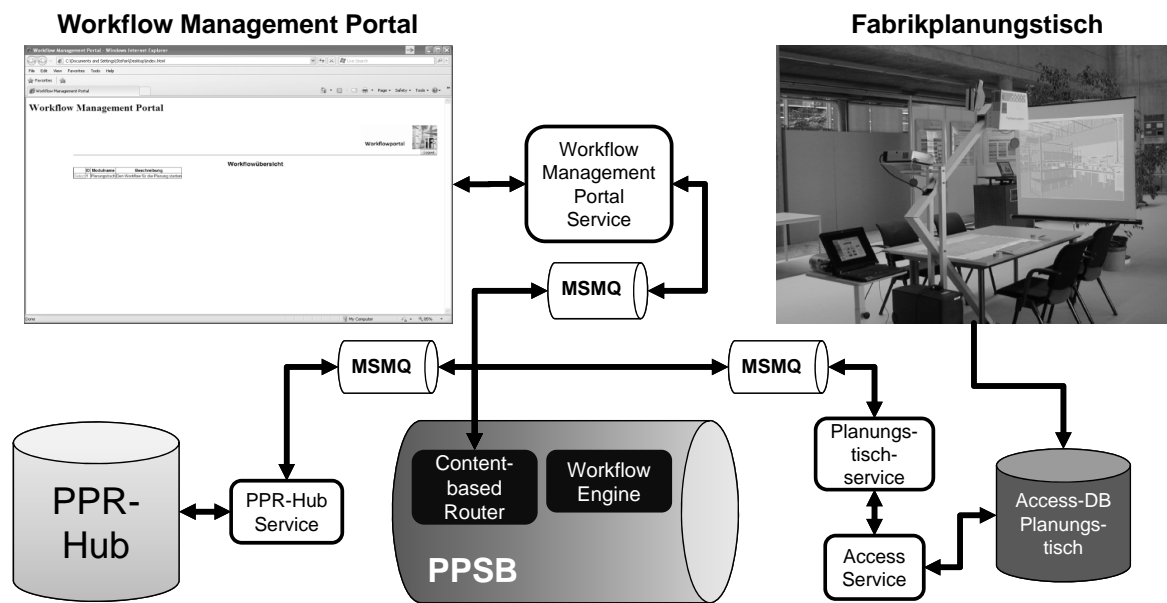


Abbildung 5.3: Architektur zur Integration von Anwendungen in der Digitalen Fabrik (vergleiche [SDMM12])

Layoutplanungstisches kann über neu implementierte Serviceschnittstellen zugegriffen werden. Zusätzlich wurde ein Workflow Management Portal entwickelt, das zum Starten, zur Steuerung und Überwachung der implementierten Workflows dient. Das Portal ist webbasiert und kann somit über jeden Browser aufgerufen werden. Die zentrale Integrationskomponente stellt der PPSB dar, der durch seinen CBR die lose Kopplung der integrierten Anwendungen vollzieht. Die resultierende Architektur mit dem Planungstisch, der PPR-Datenbank und dem Workflow Management Portal, sowie deren Verknüpfung über die Serviceschnittstellen, Warteschlangen und dem PPSB, ist in Abbildung 5.3 illustriert.

Das einheitliche Nachrichtenformat für den Datenaustausch zwischen Anwendungen besteht analog zum MSB aus zwei Teilen. Der erste Teil der Nachricht dient zur Speicherung von Routinginformationen, die das Quellsystem und die Zielsysteme, einen Zeitstempel und eine eindeutige NachrichtenID enthalten. Die NachrichtenID wird vom Nachrichtenverzeichnis generiert und anschließend in die Nachricht eingetragen. Der zweite Nachrichtenteil umfasst die Nutzdaten, die zwischen den Anwendungen der Produktions-



planung ausgetauscht werden. Da der bisher eingesetzte DPE alle Daten in Form von Projekten gruppiert hat, ist dieser Ansatz für das Nachrichtenformat übernommen worden, so dass mehrere Datensätze innerhalb einer Nachricht versendet werden können. Innerhalb eines Projektes sind bisher nur Ressourceninformationen mit iTRAME-Modultypen und den zugehörigen Modulen modelliert. Für jedes Modul wird die aktuelle Position und Rotation gespeichert, die vom Planungstisch für die Generierung des Layouts benötigt werden. Das Nachrichtenmodell enthält bisher keine Produkt- oder Prozessinformationen, da diese für die Layoutplanung nicht benötigt werden. Das Datenaustauschschema ist aber so konzipiert, dass eine einfache Erweiterung um diese Informationen möglich ist. Das XML-Schema dieses Datenaustauschformates ist in Anhang A.1 dargestellt.

## **5.3 Implementierung der serviceorientierten ACITA für das PLM**

In diesem Abschnitt wird die prototypische Implementierung des PLM-Busses beschrieben, der die beiden phasenspezifischen Integrationsumgebungen des PPSB und MSB verbindet und somit einen domänenübergreifenden Datenaustausch ermöglicht. Durch die Implementierung des PLM-Busses können im Anschluss weitere phasenspezifische ESBs auf einfache Weise angebunden werden, um die in Kapitel 4 vorgestellte ACITA für den Produktlebenszyklus vollständig umzusetzen.

Ein Ziel des implementierten Prototypen ist den nahtlosen Datenaustausch zwischen der digitalen und physischen Fabrik der Lernfabrik aIE zu ermöglichen, ohne die beiden bestehenden Integrationsumgebungen in ihren bisher bereitgestellten Funktionalitäten zu beeinträchtigen. Dabei soll der Datenaustausch in beiden Richtungen möglich sein, von der digitalen zur physischen Fabrik, aber auch umgekehrt.

Da Produktionsplanungen oft auf bestehenden Planungsdaten durchgeführt werden, die zu diesem Zeitpunkt eventuell nicht mehr dem aktuellen Stand der Fabrik entsprechen, wurde für die Integration der digitalen und physischen Fabrik ein Prozess umgesetzt, der das aktuelle Layout der physischen Fabrik in den Layoutplanungstisch der digitalen Planungsumgebung überträgt. Somit wird sichergestellt, dass stets das aktuelle Layout der Montagelinie für eine Umplanung in der Digitalen Fabrik bereitsteht. Zur Visualisierung und Umsetzung in der physischen Fabrik wird das geänderte Hallenlayout in das bestehende Wartungsportal übertragen, in dem Änderungen farblich hervorgehoben werden und damit eine schnelle Anpassung erfolgen kann. Zu diesem Zweck ist das Wartungsportal um diese Funktionalität erweitert worden.

### **5.3.1 Implementierung der ACITA für den Produktlebenszyklus**

In diesem Abschnitt werden die erforderlichen Bausteine zur Umsetzung der ACITA erläutert. Ein Modell für verteilte Service Verzeichnisse ist notwendig, um Services effizient in der gesamten Architektur zu registrieren und anschließend zu finden. Des Weiteren werden Router benötigt, um Nachrichten dynamisch an das gewünschte Ziel zu senden. Für diese Nachrichten sind einheitliche Datenformate erforderlich, die innerhalb eines ESBs zur Anwendung kommen. Zudem werden Übersetzungsservices benötigt, um Nachrichten zwischen den verschiedenen Datenaustauschformaten zu übersetzen.

Alle eingesetzten Bausteine zur Integration der Produktionsplanung mit der Produktion sind in Abbildung 5.4 dargestellt. Aus Gründen der Übersichtlichkeit sind die Serviceschnittstellen zu den jeweiligen Anwendungen nicht eingezeichnet. Einige Bausteine aus den bestehenden Implementierungen, wie z. B. der Planungstischservice, konnten größtenteils ohne Änderun-

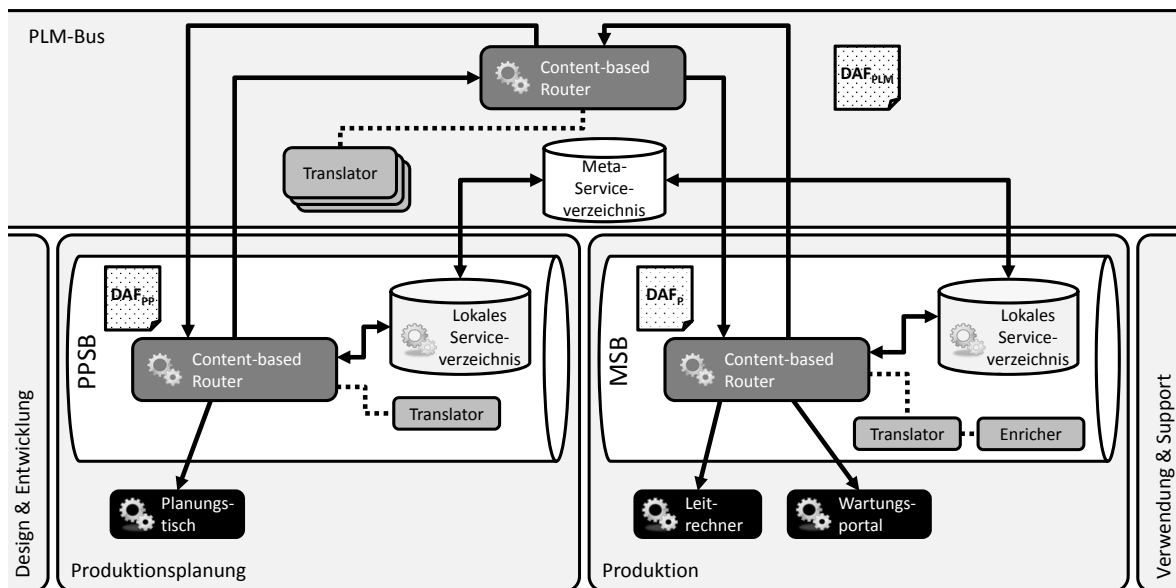



Abbildung 5.4: Komponenten zur Umsetzung der PLM-Architektur (vergleiche [SKRM13])

gen wiederverwendet werden. Allerdings sind der Planungstischservice und Access-Service zur Vereinfachung zusammengefasst worden. Andere Bausteine wurden erweitert, wie das Wartungsportal oder der Leitrechnerservice, der nun zusätzlich alle Module aus der Leitrechnerdatenbank lesen kann. Die CBR-Services wurden angepasst an die Zusammenarbeit mit den neu implementierten Serviceverzeichnissen. Zudem sind die Übersetzungsservices neu implementiert worden, genauso wie die gesamte Serviceverzeichnisinfrastruktur.

Das lokale Datenaustauschformat (DAF) der PPSB-Integrationsumgebung ist im allgemeinen Teil mit Routinginformationen erweitert worden, um eine Nachrichtenübertragung in andere Phasen zu ermöglichen. Zur Übersetzung zwischen den unterschiedlichen Nachrichtendatenformaten ist ein Übersetzungsservice implementiert worden, der zwischen dem lokalen Datenaustauschformat (DAF<sub>PP</sub>) in das globale Datenaustauschformat des PLM-Busses (DAF<sub>PLM</sub>) übersetzt. Dazu wurde der PPSB mit einem lokalen Serviceverzeichnis ergänzt, das Teil des neuen Verzeichnissesmodells ist.

Die MSB-Integrationsumgebung ist ebenfalls mit verschiedenen Bausteinen erweitert worden. Wie der PPSB wird ein lokales Serviceverzeichnis für das Bestimmen der Serviceendpunkte eingesetzt. Ein Übersetzungsservice dient zur Transformation des lokalen Datenaustauschformates ( $DAF_P$ ) in das globale Datenaustauschformat des PLM-Busses ( $DAF_{PLM}$ ). Zusätzlich besitzt dieser Übersetzungsservice eine Enricher-Komponente, die jeder Nachricht, die für den Planungstisch bestimmt ist, zusätzliche Informationen hinzufügt, die in der Datenbank des Leitrechners nicht verfügbar sind. Dabei handelt es sich um die Zuordnung der Modul-IDs der Montagelinie zu den jeweiligen Modultypen. Das Nachrichtenformat wurde entsprechend des Datenformates der Produktionsplanung mit Routinginformationen für den phasenübergreifenden Datenaustausch erweitert. Zudem sind die Nutzdaten der Nachricht um Fabriklayoutinformationen bzw. Ressourcendaten erweitert worden, ohne die bisher enthaltenen Nutzdaten zu verändern. Dies erlaubt bestehende Serviceschnittstellen, wie z. B. die des Kundenportals, ohne Änderungen weiter zu nutzen, da die hinzugefügten Informationen nur optional sind. Nachrichten, die an den Kundenportalservice gesendet werden, haben weiterhin dieselbe Struktur und können deshalb ohne Probleme interpretiert werden. Das Wartungsportal wurde um die Möglichkeit ergänzt, eine Layoutumplanung anzustoßen. Dazu wird das aktuelle Layout der Modellfabrik aus der Leitrechnerdatenbank ausgelesen und an das Wartungsportal und den Planungstisch gesendet. Hierfür wurde der Leitrechnerservice erweitert, um neben den Fehlern der Fertigungslinie auch die Layoutinformationen der Module aus der Datenbank auszulesen. Im Wartungsportal entstand eine neue Ansicht, in der sich ein bestehendes und ein umgeplantes Hallenlayout anzeigen lässt, wobei sich im neuen Layout Änderungen im Vergleich zum bestehenden Layout farblich visualisieren lassen, wie in Abbildung 5.5 dargestellt.

Die Integrationsumgebung des PLM-Busses wurde für den Datenaustausch zwischen der digitalen und physischen Fabrik neu implementiert. Eine Hauptaufgabe ist die Bereitstellung des Meta-Verzeichnisses, um Services aus an-

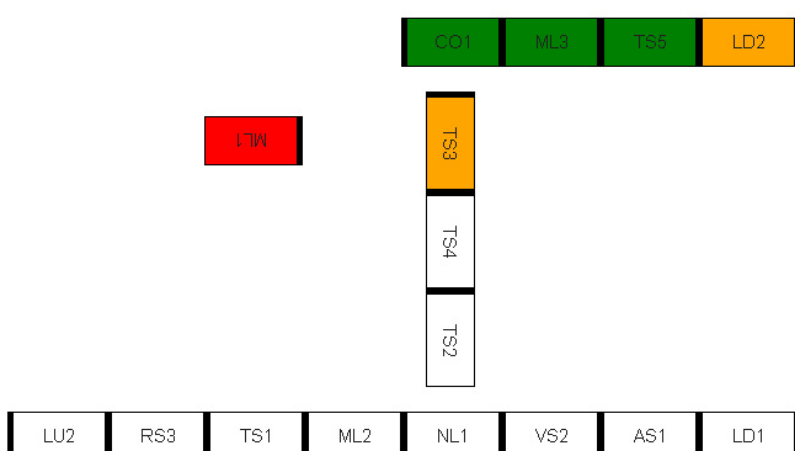
**Maintenance Portal** 

**Current Layout**   Failures   Logout

**Current Layout**

Load current layout and send to planning table   Retrieve redesigned layout from planning table

Original Layout   Redesigned Layout



Overlay **modified**  
 Overlay **added**  
 Overlay **removed**

Abbildung 5.5: Layoutänderungen werden im Wartungsportal farblich visualisiert

deren Phasen auffinden zu können. Eine weitere wichtige Aufgabe ist das Weiterleiten von Nachrichten zwischen unterschiedlichen Phasen. Dies geschieht wiederum über einen CBR, der den lokalen CBRs sehr ähnlich ist. Der Unterschied besteht darin, dass das Nachrichtenverzeichnis des globalen CBR derzeit kein Message Registry besitzt, da alle Nachrichten bereits vom jeweiligen lokalen CBR und dessen Nachrichtenverzeichnis eine eindeutige NachrichtenID erhalten haben. Zur Nachverfolgung der Nachricht besitzt dieses Nachrichtenverzeichnis jedoch ein Message Flow Registry. Wie die lokalen CBRs, verwendet auch der globale CBR Übersetzungsservices. Ab-

hängig vom Ziel wird ein Übersetzungsservice ausgeführt, der die Nachricht vom globalen Datenaustauschformat ( $\text{DAF}_{\text{PLM}}$ ) in das entsprechend von der Zielphase verwendete Datenaustauschformat übersetzt.

Die wichtigsten Bausteine zur Umsetzung der ACITA werden nachfolgend detaillierter beschrieben und basieren auf der Veröffentlichung von Silcher et al. [SKRM13].

### **5.3.2 Implementierung der Serviceverzeichnisinfrastruktur**

Die Implementierung der Service-Verzeichnisse erfolgt entsprechend dem Meta-Verzeichnismodell, das in Abschnitt 4.3.2.2 beschrieben ist. Das Verwalten von Services innerhalb einer Phase wird durch ein lokales Verzeichnis bewerkstelligt. Auf der globalen Ebene wird ein Meta-Verzeichnis verwendet.

Jedes Verzeichnis stellt eine Serviceschnittstelle zur Verfügung, über die Services registriert, geändert oder gelöscht werden können. Die Serviceinformationen sind in einer Datenbank gespeichert. Zur Abfrage von Servicebeschreibungen steht ebenfalls eine Serviceschnittstelle zur Verfügung, über die der CBR den Endpunkt eines Nachrichtenzieles bestimmt.

Das Meta-Verzeichnis speichert die Endpunkte aller lokalen Serviceverzeichnisstellen zusammen mit dem Endpunkt der lokalen CBRs. Um Nachrichten an neue Phasen senden zu können, müssen nur in diesen Tabellen die beiden Endpunkte des neuen lokalen Serviceverzeichnisses und CBRs eingetragen werden. Dies geschieht, indem sich das neue Verzeichnis beim Meta-Verzeichnis registriert. Im Anschluss können die in der Phase installierten Services über das Meta-Verzeichnis bekannt gemacht werden, um von anderen Phasen darauf zugreifen zu können. Umgekehrt wird verfahren, sollte eine lokale Integrationsumgebung aus der Architektur entfernt

werden. Dann erfolgt die Löschung der entsprechenden Services aus dem Meta-Verzeichnis, zusammen mit den Einträgen zu den Endpunkten des lokalen Verzeichnisses und des CBRs dieser Phase.

### 5.3.3 Dynamisches Routing in der ACITA

Die dynamische Zustellung von Nachrichten an das gewünschte Ziel bzw. die Ziele geschieht über die CBRs. In den phasenspezifischen ESBs sind die CBRs identisch. Jede ankommende Nachricht startet einen Routingworkflow. Dieser Workflow startet einen Routing-service, der anhand des Nachrichteninhaltes ermittelt, an welche Zielsysteme die Nachricht gesendet werden muss. Anschließend werden die entsprechenden Serviceendpunkte der Zielsysteme mithilfe des Serviceverzeichnisses bestimmt und die Nachricht dann an die Services gesendet, sofern diese in derselben Phase sind. Andernfalls werden sie an den globalen CBR des PLM-Busses gesendet.

Dieser globale CBR ist etwas einfacher aufgebaut, als die lokalen CBRs, denn der globale CBR kann die Zielphase jeder Nachricht direkt aus deren Metadaten lesen, da diese zuvor von dem lokalen CBR dort eingetragen wird. Im Anschluss muss der globale CBR die Nachricht nur noch an den entsprechenden lokalen CBR der Zielphase senden.

### 5.3.4 Globales Nachrichtendatenformat

Für die beiden Integrationsumgebungen des PPSB und MSB existieren jeweils domänenspezifische Datenaustauschformate, über die Nachrichten innerhalb einer Phase ausgetauscht werden können. Um Nachrichten zwischen den Phasen auszutauschen, wurde ein weiteres globales Nachrichtenformat für den PLM-Bus definiert, das nur Informationen enthält, die zwischen Phasen zum Austausch kommen. Diese hierarchische Form der Datenbeschreibung, die bereits von Davies et al. vorgestellt wurde [DSRR08],

reduziert die Komplexität der einzelnen Datenformate und erhöht somit deren Flexibilität und Wartbarkeit.

Das globale Nachrichtenformat entspricht einer Teilmenge aller vereinigten phasenspezifischen Nachrichtenformate, da nicht alle Informationen, deren Austausch innerhalb einer Phase stattfindet, auch mit anderen Phasen geteilt werden. Somit reduziert sich die Komplexität für das globale Nachrichtenformat gegenüber einem einzelnen unternehmensweiten Nachrichtenformat, das alle Informationen des gesamten Produktlebenszyklus beinhalten würde. Aufgrund der Komplexität eines solchen einzelnen Austauschformates und insbesondere wegen den Problemen bei der Definition von einheitlichen domänenübergreifenden Datenbeschreibungen, sind jedoch bisherige Versuche stets gescheitert. Das Schema des globalen Nachrichtenformates findet sich in Anhang A.2.

Zur Umsetzung des Szenarios, beschrieben in Abschnitt 6.1.2, war eine Erweiterung des im MSB eingesetzten Nachrichtenformates notwendig. Diese Erweiterung um Layoutinformationen der Fabrik konnte so durchgeführt werden, dass die Kommunikation der bereits über den MSB integrierten Anwendungen ohne Änderungen weiter funktioniert. Dies zeichnet die hohe Flexibilität von XML-Nachrichten aus. Lediglich die Schnittstellen des Leitrechners und Wartungsportals wurden erweitert, um die gewünschten Informationen bereitzustellen bzw. zu verarbeiten. Andere Serviceschnittstellen, wie z. B. die des Kundenportals, sind nur insofern verändert worden, dass sie nun das erweiterte Nachrichtenmodell senden und empfangen können.

In dem aktuell implementierten Prototyp werden lediglich Ressourceninformationen zwischen zwei Domänen ausgetauscht. Um die eingesetzten Nachrichtenformate für den industriellen Einsatz bereit zu machen, fehlen noch weitere Informationen, wie z. B. Produkt- und Prozessdaten. Bei einer Erweiterung mit Produktdaten bietet es sich an auf bereits existierende Standards wie STEP zu setzen, das bereits viele Jahre in der Industrie für den Datenaustausch eingesetzt wird [Int94]. Mittlerweile ist eine auf XML basie-



rende Umsetzung von STEP verfügbar, die sich STEP-XML nennt [Int07]. Ein weiterer auf XML aufbauender Industriestandard ist PLM XML [Sie10]. PLM XML bündelt verschiedene Standards wie Parasolid (XT) und JT in einem XML-Schema, um Strukturen, Geometrien und weitere Produkteigenschaften zu beschreiben und einen domänenübergreifenden Produktdatenaustausch zu ermöglichen. Sowohl STEP als auch PLM XML beschränken sich auf die Beschreibung von Produkten bzw. Produktdaten und konnten somit für die Datenaustauschformate des implementierten Prototypen nicht eingesetzt werden, da Ressourceninformationen in diesen Standards nicht abbildbar sind.

### 5.3.5 Übersetzungsservices

Die einheitlichen Nachrichtendatenformate der einzelnen Phasen und des PLM-Busses sind inkompatibel. Deshalb sind Übersetzungsdienste notwendig, um die Nachrichten zwischen verschiedenen Phasen austauschen zu können. Für jede Phase ist ein Übersetzungsservice implementiert, der die Daten aus dem phasenspezifischen einheitlichen Datenformat in das einheitliche Datenformat des PLM-Busses übersetzt. Der PLM-Bus wiederum besitzt für jede Phase einen Übersetzungsdienst, um die Nachricht vom globalen einheitlichen Datenformat in das entsprechende phasenspezifisch einheitliche Datenformat der Zielphase zu übersetzen.

Die Übersetzungsservices werden direkt vom CBR aufgerufen, wenn dieser erkennt, dass die Nachricht an einen Service in einer anderen Phase versendet wird. Die Nachricht wird somit zuerst in das einheitliche Datenformat des PLM-Busses bzw. der Zielphase übersetzt, bevor sie an den entsprechenden CBR gesendet wird. Jeder CBR bekommt deshalb immer die Nachricht im Format seiner Integrationsumgebung und kann diese sofort interpretieren, ohne sie vorher übersetzen zu müssen.

Der Übersetzungsdienst des MSB wird noch um den *Enricher* erweitert, der jeder Nachricht an den Fabrikplanungstisch weitere Informationen hinzufügt.

## 5.4 Zusammenfassung

Dieses Kapitel beschreibt die prototypische Implementierung der ACITA für das PLM. Die Umsetzung erfolgt in zwei Schritten, zuerst wird die Produktionsplanung mittels des PPSB integriert, um einen adaptiven Datenaustausch zu erreichen und eine anwendungsübergreifende Prozessdefinition zu ermöglichen. Anschließend werden die neu implementierte Integrationsumgebung der Produktionsplanung mit dem PPSB und die bestehende Integrationsumgebung der Produktion mit dem MSB über den PLM-Bus verbunden, um einen domänenübergreifenden Datenaustausch zu ermöglichen.

Der PPSB setzt zur Integration der heterogenen und verteilten IT-Landschaft der Produktionsplanung auf eine SOA, wie bereits der MSB, um die benötigte Flexibilität beim Datenaustausch zwischen Anwendungen zu erhalten. Um die Integrationsumgebung gegen Störungen und Netzwerkfehler abzusichern, werden Warteschlangen eingesetzt, die bei der Übertragung die Nachrichten zwischenspeichern, bis der Empfänger diese erfolgreich erhalten hat. Die Unterstützung der Geschäftsprozesse basiert auf derselben fünfstufigen Integrationshierarchie, die bereits beim MSB verwendet wird.

Die adaptive Integration der Produktionsplanung und Produktion erfolgt über den PLM-Bus, der den PPSB und MSB lose koppelt, wodurch ein phasenübergreifender Datenaustausch ermöglicht wird. Der PLM-Bus ist so konzipiert, dass weitere Phasen der Architektur einfach hinzugefügt oder entfernt werden können, wodurch eine hohe Flexibilität der gesamten ACITA zu erreichen ist. Zudem wird durch diese unkomplizierte Änderung der inte-

grierten Domänen die Wandlungsfähigkeit der IT-Landschaft entscheidend verbessert.

Die phasenspezifischen Integrationsumgebungen sind weitestgehend identisch implementiert. Sie verwenden einen CBR für die dynamische Nachrichtenvermittlung und ein lokales Serviceverzeichnis zum Speichern der Servicebeschreibungen. Das Datenaustauschformat ist jedoch unterschiedlich und muss abhängig von den integrierten Anwendungen jeder Domäne und den jeweils ausgetauschten Daten individuell entwickelt werden. Die Übersetzungsservices dienen zur Transformation der phasenspezifischen Nachrichtenformate ins globale Nachrichtenformat und zurück, um Nachrichten nahtlos innerhalb des gesamten Produktlebenszyklus austauschen zu können. Dies wird auch von der Serviceverzeichnisstruktur mit dem Meta-Verzeichnis ermöglicht, die das Auffinden von Services anderer Phasen erlaubt, die im Anschluss aufgerufen werden können.



---

## KAPITEL 6

# Anwendung und Evaluation der ACITA

---

Das Kapitel beschreibt vier Anwendungsszenarien, die die Vorteile der ACITA gegenüber anderen Integrationslösungen oder Softwarepaketen für Produktionsunternehmen aufzeigt. Im Anschluss werden Bewertungskriterien für diese Lösungen festgelegt, die eine systematische Evaluation erlauben und einen Vergleich zulassen. Erweiterungsmöglichkeiten der ACITA um mobile Endgeräte und eine serviceorientierte Maschineninbetriebnahme zeigen weitere Vorteile der entwickelten Architektur auf. Abschließend wird die Erfüllung der Forschungsobjekte durch die ACITA dargestellt.

## 6.1 Anwendungsszenarien

Im Folgenden werden vier Anwendungsszenarien vorgestellt, die die Vorteile der ACITA zur flexiblen Integration von unterschiedlichen Domänen demonstrieren. Die ersten beiden Szenarien wurden in der Lernfabrik aIE umgesetzt und dort erfolgreich getestet. Der erste Anwendungsfall demonstriert die flexible Integration des Planungstisches mit dem PPR-Hub unter Verwendung des PPSB und der Warteschlangen. Die Umsetzung des PLM-Busses zur adaptiven Integration der Produktionsplanung und Produktion

mit den phasenspezifischen PPSB und MSB wird im zweiten Anwendungsszenario beschrieben. Das Laden des aktuellen Layouts der Montagelinie in den Planungstisch zeigt dabei die nahtlose domänenübergreifende Datenübertragung. Die einfache Erweiterung der ACITA durch weitere phasenspezifische Integrationsumgebungen ist im dritten Anwendungsszenario aufgezeigt. Im vierten Anwendungsszenario wird hingegen die Einbindung von externen Partnern in die ACITA beschrieben. Insbesondere auf die höheren Anforderungen an Sicherheit und Vertraulichkeit wird dabei eingegangen.

### **6.1.1 Anwendungsszenario 1: Serviceorientierte Integration der Produktionsplanung**

Die Integration der heterogenen Anwendungslandschaft der Produktionsplanung ist nach dem Vorbild des MSB umgesetzt. Um die Robustheit der Infrastruktur im Vergleich zum MSB zu erhöhen kommen Warteschlangen zum Einsatz. Jeder Service besitzt hierfür eine Eingangswarteschlange zur Zwischenspeicherung von Nachrichten, bis diese vom Service empfangen und verarbeitet werden können. Ein flexibler Nachrichtenaustausch wird über einen CBR realisiert, bei dem Nachrichten dynamisch an das gewünschte Ziel geleitet werden.

In Abbildung 6.1 ist das Anwendungsszenario mit allen beteiligten Komponenten dargestellt. Die Darstellung folgt den fünf Integrationsebenen der ACITA, bei der sich in Ebene 0 die Anwendungen befinden, die ihre Funktionalitäten und Daten bereitstellen. Auf der Ebene 1 bilden die Serviceschnittstellen eine einheitliche Zugriffsschicht auf die Anwendungen. Die mittlere Ebene 2 beinhaltet den PPSB und sorgt für die Integration der Services, die mithilfe des CBR lose gekoppelt werden. Darüber befinden sich die Geschäftsservices in Ebene 3, die domänenspezifische Prozesse abbilden und zur Integration in die Geschäftsprozesse der höchsten Ebene 4 bereit stehen.

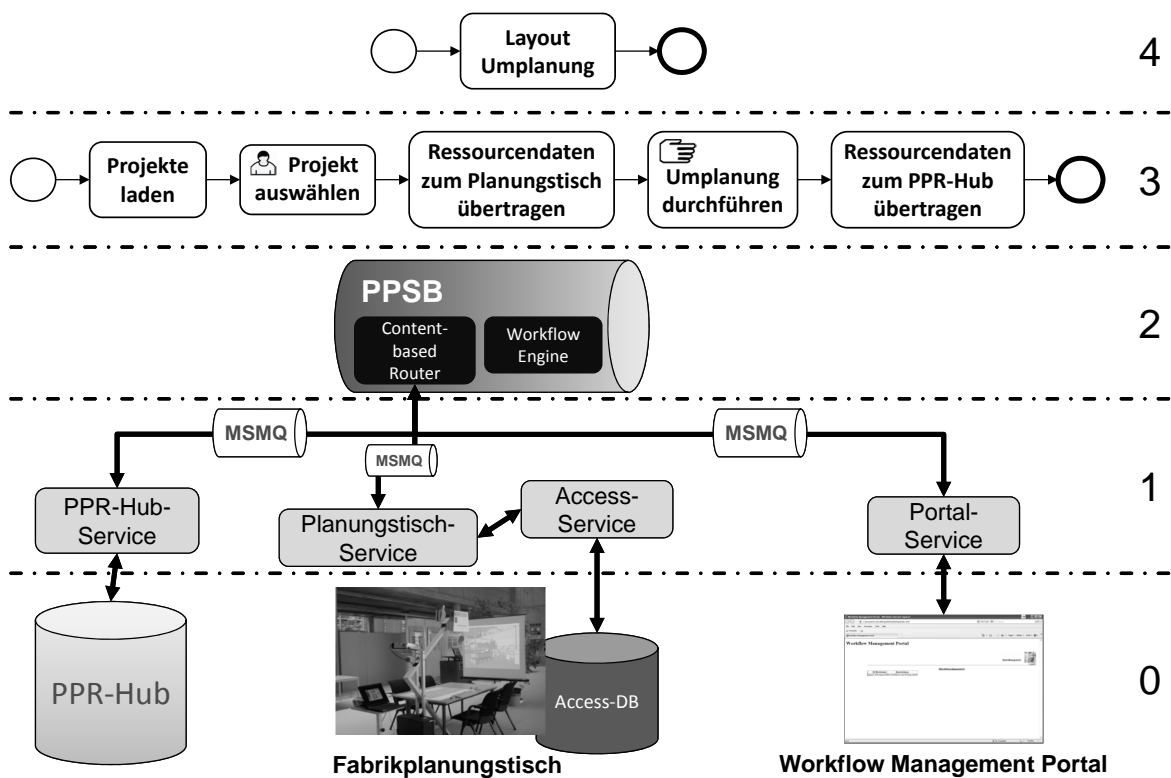


Abbildung 6.1: Umsetzung der fünfstufigen Integrationshierarchie in die Produktionsplanung mit Warteschlangen

Im Rahmen der prototypischen Implementierung wurde ein Hallenlayout-Umplanungsprozess umgesetzt, der in der dritten Ebene von Abbildung 6.1 dargestellt ist. Der Prozess besteht aus fünf Teilschritten und enthält sowohl einen Schritt mit Benutzerinteraktion, dargestellt durch ein schematisches Bild eines Benutzenden in der oberen linken Ecke, als auch eine manuelle Aufgabe, die durch eine Hand gekennzeichnet ist. Dieser Layoutumplanungsprozess könnte eine Aktivität eines Geschäftsprozesses sein, der in der Abbildung in Ebene 4 zu sehen ist. Ein vollständiger Geschäftsprozess könnte alle Prozessschritte zur Planung einer Montagelinie enthalten, deren Teilaufgaben die einzelnen Planungsschritte abbilden. Ein solcher Geschäftsprozess ist im Rahmen der Arbeit für den Prototyp nicht umgesetzt worden.

Auf der untersten Ebene sind drei Anwendungen integriert. Eine Datenbank, die als Ersatz für den DPE dient und alle PPR-Daten in Form von Planungsprojekten speichert und als PPR-Hub bezeichnet ist. Der Fabrik-

planungstisch, der über dessen Access-Datenbank angebunden ist und das neu implementierte Workflow Management Portal, das zum Starten, Steuern und Überwachen von Workflows dient. Auf die Datenbanken bzw. das Portal wird über neu implementierte Web Services zugegriffen, über die Nachrichten empfangen und gesendet werden können. Dabei besitzt jeder Web Service eine MSMQ, die eingehende Nachrichten zwischenspeichert, bis die entsprechende Anwendung für den Empfang bereit ist.

Die Integration dieser Web Services wird durch den PPSB in der zweiten Ebene durchgeführt. Der CBR sorgt dabei für die dynamische Nachrichtenvermittlung zwischen den einzelnen Serviceschnittstellen. Mit der Workflow Engine können die Workflows der beiden darüber liegenden Ebenen ausgeführt werden, nachdem sie zuvor in der PPSB-Umgebung installiert wurden.

Durch die Einführung der SOA, die zur Integration der heterogenen Anwendungslandschaft in der Produktionsplanung dient, konnte eine homogene Integrationsumgebung geschaffen werden, die im Vergleich zu den zuvor in der Lernfabrik eingesetzten Lösungen mit CHAMPAGNE und den VBA-Skripten im DPE eine höhere Adaptivität aufweist. Durch Einführung der fünfstufigen Integrationshierarchie erfolgt eine Entkoppelung die Anwendungen auf der untersten Ebene von deren Schnittstellen auf der darüber liegenden Ebene, wodurch die Abhängigkeiten reduziert werden und somit Änderungen an Anwendungen oder Schnittstellen geringere Auswirkungen aufeinander haben. Das einheitliche Datenaustauschformat bildet eine zusätzliche Abstraktionsebene zwischen den Anwendungen, wodurch der Nachrichtenaustausch unabhängig von proprietären Datenformaten der Anwendungen wird. Zudem reduziert sich die Komplexität der IT-Landschaft, da einheitliche und standardisierte Schnittstellen für die Integration verwendet und die Anzahl der Schnittstellen reduziert werden. Dies hat zur Folge, dass eine verbesserte Wartbarkeit und Erweiterbarkeit erreicht wird. Durch die Abbildung der Prozesse in Workflows können diese besser mit IT-Systemen



unterstützt werden. Zudem steigt durch die flexible Komposition von Services zu Workflows die Adaptivität der Prozesse, was insbesondere im turbulenten Umfeld von Produktionsunternehmen von größter Bedeutung ist.

### **6.1.2 Anwendungsszenario 2: Phasenübergreifende Integration von Produktionsplanung und Produktion**

Der nahtlose Datenaustausch zwischen der digitalen und physischen Fabrik wird veranschaulicht durch die Umsetzung eines Layoutumplanungsprozesses in der Lernfabrik aIE, der in Abbildung 6.2 dargestellt ist. Hierfür wird das aktuelle Layout der physischen Fabrik über die ACITA zum Fabrikplanungstisch übertragen. Dies wird erreicht indem das aktuelle Layout der Modellfabrik aus dessen Leitrechnerdatenbank ausgelesen und über den PLM-Bus an den Planungstisch in der digitalen Lernumgebung gesendet wird. Im Anschluss wird das aktuelle Layout im Planungstisch angezeigt und die Umplanung zur Erzeugung eines neuen und verbesserten Fabriklayouts kann durchgeführt werden. Das geänderte Layout wird im Anschluss über den PLM-Bus zurück an den MSB gesendet und im Wartungsportal zusammen mit dem ursprünglichen Layout gespeichert. Beide Layouts können nun im Wartungsportal betrachtet werden, wobei eine farbliche Hervorhebung der Änderungen im neuen Layout durchgeführt wird, wie in Abbildung 5.5 zu sehen ist. Die Unterscheidung der aus dem Layout entfernten, umplatzierten und hinzugefügten Module erlaubt einen schnelleren Umbau der Montagelinie.

Durch die Übertragung des aktuellen physischen Layouts über die ACITA wird verhindert, dass die Umplanung auf Planungsdaten aufbaut, die unter Umständen gar nicht mehr aktuell sind und sich auf eine alte Version des Hallenlayouts beziehen. Das aufwendige Suchen nach dem aktuellen

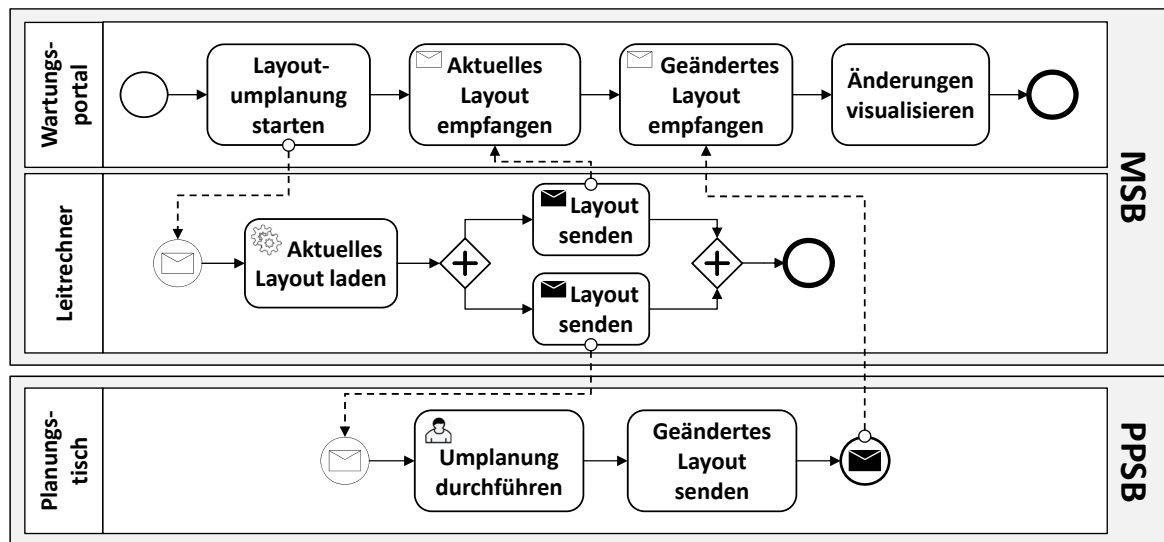


Abbildung 6.2: Prozess zur Layoutübertragung vom Leitreechner zum Planungstisch, modelliert in BPMN (vergleiche [SKRM13])

Planungsstand, wie es häufig bei einer großen Menge an Planungsdaten vorkommt, wird ebenso verhindert. Das Laden des aktuellen Layouts kann zu jedem Zeitpunkt durchgeführt werden.

Um das Laden des aktuellen Fabriklayouts in den Planungstisch anzustoßen, sind im Wartungsportal zwei Möglichkeiten implementiert. In der bestehenden Übersicht, in der alle Fehler der Montagelinie angezeigt werden, kann ein Wartungsingenieur bisher zur Behebung eines Fehlers entscheiden, ob er das betroffene Modul reparieren oder ersetzen will. Diese Möglichkeit wird erweitert um die Option eine Layoutumplanung anzustoßen, in der z. B. das betroffene Modul im Fabrikplanungstisch aus der Montagelinie entfernt und anschließend mit den verbliebenen Modulen wieder ein optimales Layout erzeugt wird. Das Zurückspielen des neuen Layouts kann entweder durch einen Link in der E-Mail durchgeführt werden, in der der Planungsingenieur aufgefordert wird, die Umplanung durchzuführen, oder durch eine neue Schaltfläche im Wartungsportal. Die visualisierten Änderungen im Wartungsportal können im Anschluss in der physischen Fabrik umgesetzt werden. Darüber hinaus wird im Wartungsportal eine Schaltfläche eingefügt, mit der die Layoutumplanung zu jedem Zeitpunkt durchführbar ist,

um präventiv eine Optimierung der Fabrik durch Layoutverbesserungen durchzuführen oder um neue Module in die Anlage zu integrieren. Mittels einer zweiten Schaltfläche kann das geänderte Layout in das Wartungsportal zurück gespielt werden, um dort wie zuvor Änderungen zu visualisieren.

### **6.1.3 Anwendungsszenario 3: Wandlungsfähigkeit der ACITA durch Hinzufügen oder Entfernen von phasenspezifischen Integrationsumgebungen**

Die Wandlungsfähigkeit der ACITA wird durch den PLM-Bus unterstützt, indem phasenspezifische Integrationsumgebungen sehr einfach hinzugefügt oder entfernt werden können. In diesem Anwendungsszenario soll dargestellt werden, wie die Integration einer neuen Domäne in die IT-Architektur funktioniert und welche Veränderungen vorzunehmen sind. Beispielhaft wird deshalb die Integration der Produktentwicklung beschrieben, die den phasenspezifischen *Design & Development Service Bus (DDSB)* zu den bereits über den PLM-Bus integrierten Domänen der Produktionsplanung und Produktion hinzufügt.

Das PDM-System stellt über eine Web Service Schnittstelle Produktinformationen bereit, auf die andere Services über den DDSB zugreifen können. Dies erlaubt Produktdaten in der Produktionsplanung und Produktion flexibel abzurufen. Damit innerhalb der Produktentwicklungsdomäne ein flexibler Datenaustausch möglich ist, wird ein Datenaustauschformat für die Produktentwicklung benötigt, das die Nachrichten unabhängig von den proprietären Datenformaten der integrierten Anwendungen macht ( $DAF_{DD}$ ). Der Nachrichtenaustausch zwischen den Phasen macht weitere Übersetzungsservices notwendig, die Nachrichten vom lokalen in das globale Nachrichtenformat und zurück übersetzen können. Zur Speicherung der Serviceinformationen wird entsprechend der anderen Phasen ein lokales Serviceverzeichnis ver-

wendet, das identisch zu diesen ist und somit keine Änderungen notwendig sind. Als CBR kann ebenso ein lokaler CBR aus einer anderen Phase kopiert werden, der bis auf das verwendete Nachrichtenformat identisch ist. Da der CBR jede Nachricht interpretieren muss und auch unter Umständen Nachrichten verändert, muss diese Anpassung durchgeführt werden. Zusätzlich benötigt der CBR eine Routingtabelle, durch die mittels XPath-Ausdrücken das Ziel einer Nachricht bestimmt werden kann.

Auf der PLM-Bus-Ebene müssen für den Datenaustausch mit der Produktentwicklung ebenso Anpassungen erfolgen. Das globale Nachrichtenformat ( $DAF_{PLM}$ ) benötigt eine Erweiterung, um die Produktdaten aufzunehmen, die mit der Produktionsplanung und Produktion auszutauschen sind. Darüber hinaus muss das lokale Verzeichnis der Produktentwicklung in das Meta-Verzeichnis eingebunden werden, indem man die Endpunkte des lokalen Verzeichnisses und des CBR der Produktentwicklung in dessen Datenbank speichert.

Zu guter Letzt ist eine Erweiterung der phasenspezifischen Datenformate jener Phasen notwendig, die die Daten der neu hinzugekommenen Phase empfangen und verwenden wollen ( $DAF_{PP}$  und  $DAF_P$ ). Jede Phase muss dazu die Teilmenge des globalen Datenformates in ihrem phasenspezifischen Nachrichtenmodell integrieren, dessen Daten mit anderen Phasen ausgetauscht werden. Die resultierende Architektur mit ihren wichtigsten Komponenten ist in Abbildung 6.3 dargestellt.

Wie leicht zu erkennen ist, besteht der größte Aufwand für die Integration neuer Domänen in der Modellierung und Erweiterung der phasenspezifischen Datenaustauschformate. Durch die Verwendung des hierarchischen Ansatzes mit mehreren phasenspezifischen und einem globalen Datenformat müssen zwar mehrere Nachrichtenformate angepasst werden, jedoch wird durch deren geringeren Umfang die Komplexität reduziert und der Gesamtaufwand für die Modellierung der Datenformate verringert.

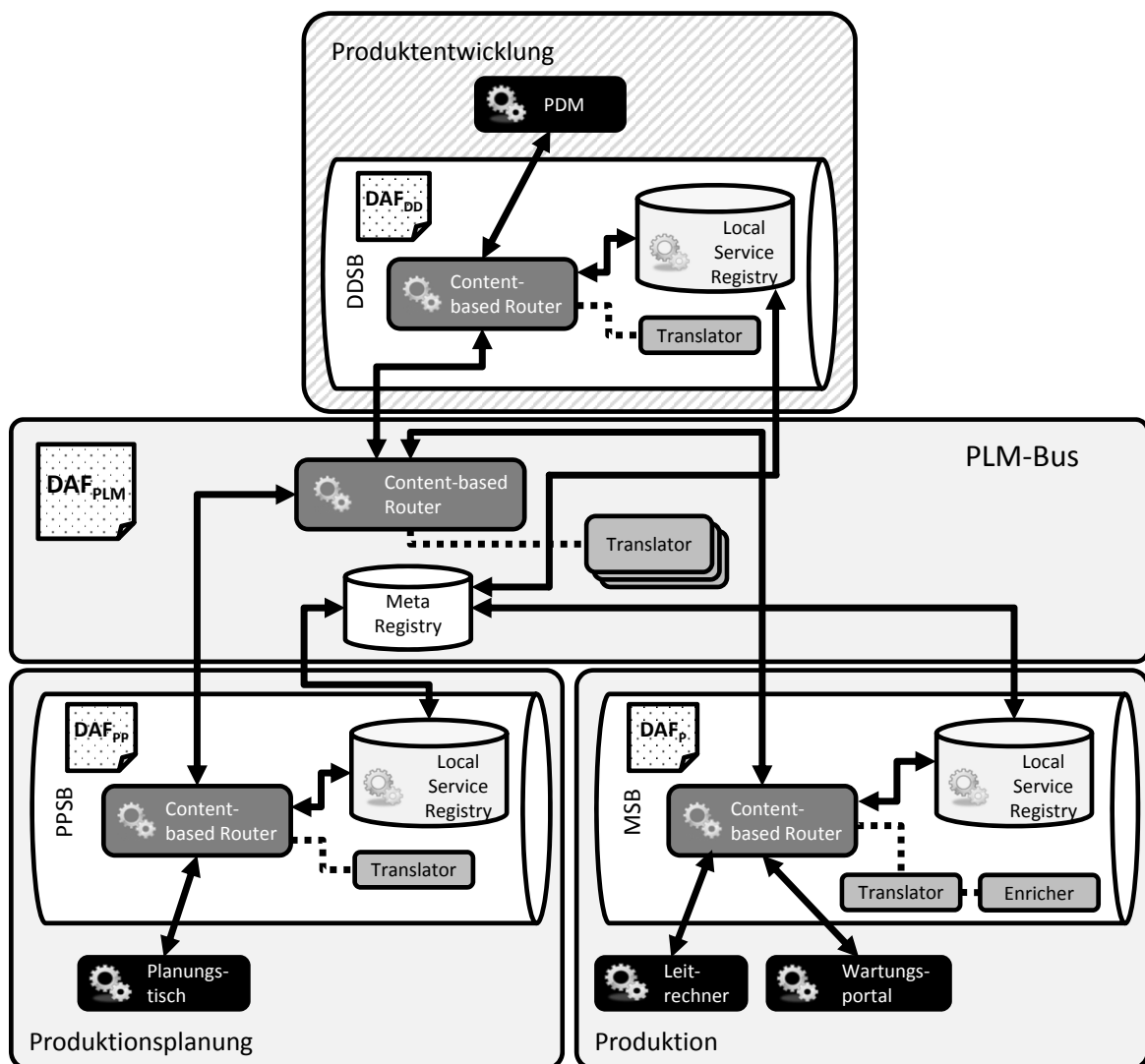


Abbildung 6.3: Wandlungsfähige IT-Architektur durch einfaches Hinzufügen und Entfernen von Integrationsumgebungen

Die Bereitstellung eines einzigen Datenformates für den gesamten Produktlebenszyklus ist aus Sicht von Komplexität, Größe und unterschiedlichen Domänen nicht möglich. Ebenso schwer ist die Definition von allgemeingültigen phasenspezifischen und globalen Datenformaten, da diese abhängig von den eingesetzten Anwendungen sind. Wird ausschließlich Standardsoftware eingesetzt, ist eine Bereitstellung von standardisierten Datenaustauschformaten noch denkbar, doch die übliche Mischung aus Standardsoftware, individuellen branchenspezifischen Lösungen und durch die Unternehmen selbst entwickelter Software macht den Einsatz von standardisierten Nachrichten-

formaten unmöglich. Somit benötigt jedes Unternehmen für seine Anwendungslandschaft maßgefertigte Datenformate, um eine adaptive Integration auf Basis der ACITA zu realisieren. Möglich ist auch eine Bereitstellung von offenen und standardisierten Schemata, die nur noch an die jeweilige Anwendungslandschaft angepasst werden müssen, um den Aufwand zur Erstellung der Datenaustauschformate zu reduzieren. Eine individuelle Adaption ist jedoch unumgänglich.

#### **6.1.4 Anwendungsszenario 4: Integration von externen Partnern in die ACITA**

Die Einbindung von Integrationsumgebungen externer Partner  $S_1$  in die ACITA funktioniert auf dieselbe Weise, wie die im vorangegangenen Abschnitt beschriebene Anbindung des DDSB an den PLM-Bus. Die Integration einer CAD-Anwendung über den DDSB- $S_1$  ist beispielhaft in Abbildung 6.4 dargestellt.

Der Zulieferer entwickelt Teile für ein Produkt in seinem CAD-System und sendet diese nach Fertigstellung über die ACITA an das PDM-System im DDSB. Die Integrationsumgebung DDSB- $S_1$  des Zulieferers kann hierfür dasselbe Datenaustauschformat  $DAF_{DD}$  wie der DDSB verwenden, sofern dieselben Daten innerhalb beider Phasen ausgetauscht werden, wie z. B. 3D-Zeichnungen von Teilen und Produkten. In diesem Fall ist eine Änderung des globalen Datenaustauschformates  $DAF_{PLM}$  nicht erforderlich.

Der Datenaustausch zwischen den CBRs bzw. den lokalen Serviceverzeichnissen und dem Meta-Verzeichnis muss bei der Einbindung von externen Integrationsumgebungen zur Integritäts- und dem Schutz der Daten verschlüsselt erfolgen, da als Kommunikationskanal zwischen Unternehmen in der Regel das Internet eingesetzt wird. Außerdem ist eine Authentifizie-

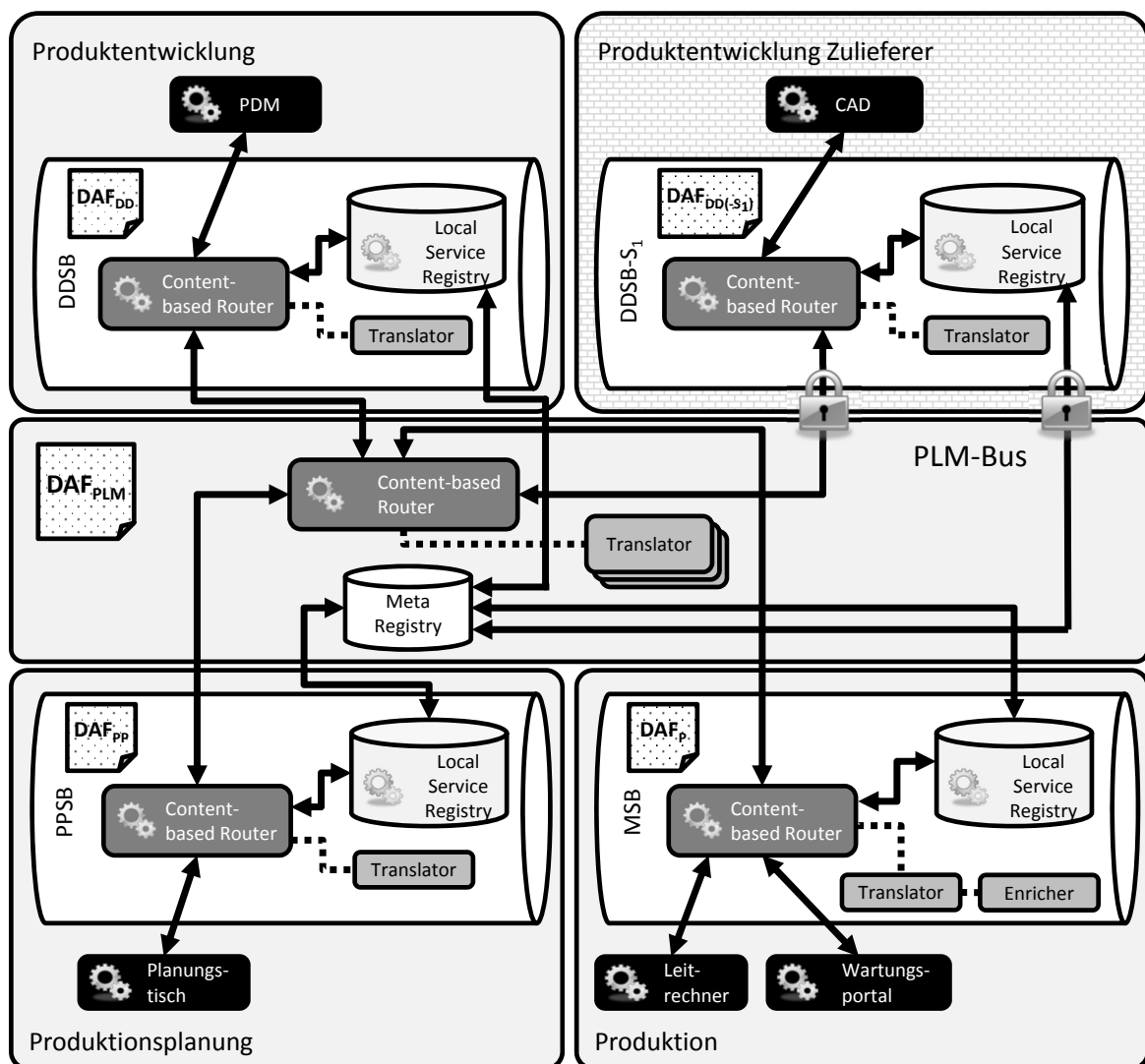


Abbildung 6.4: Integration von externen Partnern in die ACITA

Die Vertraulichkeit von Informationen beider Parteien muss gewährleistet sein. Deshalb werden im Meta-Verzeichnis nur die notwendigsten Serviceinformationen gespeichert. Dies erlaubt dem Zulieferer seine eigene Serviceinfrastruktur unabhängig zu verwalten und flexibel zu ändern, ohne bei jeder Serviceänderung eine Folgeanpassung an das Meta-Verzeichnis senden zu müssen.

## 6.2 Bewertung der ACITA gegenüber anderen Lösungen

Im Folgenden werden sechs Kriterien vorgestellt, die jeweils in fünf Stufen unterteilt sind. Im Anschluss werden verschiedene in der Forschung und Industrie eingesetzte Anwendungen und Integrationsplattformen, sowie die im Rahmen dieser Arbeit entwickelte ACITA, anhand der Kriterien bewertet, um sie im Anschluss vergleichen zu können.

Die Kriterien überdecken sich teilweise mit den von Minguez verwendeten Kriterien Interoperabilität, Adaptivität, Servicevermittlung und Integration zur Evaluation des MSB [Min12]. Dies ist der Tatsache geschuldet, dass sich auch die Anforderungen für beide Integrationskonzepte in Bezug auf Adaptivität, Integration einer heterogenen und teilweise proprietären Anwendungslandschaft, sowie flexible Prozessunterstützung decken. Die einzelnen Kriterien sind teilweise voneinander abhängig, wie z. B. die Interoperabilität bzw. Adaptivität und Integration oder die Servicevermittlung und Prozessunterstützung, bewerten aber grundsätzlich unterschiedliche Eigenschaften der Integrationsarchitektur. Die Bewertung der Interoperabilität bzw. Adaptivität ist nur möglich, wenn Anwendungen überhaupt integriert sind. Zudem führt eine Orchestrierung von Services in der Regel zu einer flexiblen Prozessunterstützung.

### 6.2.1 Bewertungskriterien

Das erste Kriterium zur Bewertung der Systeme ist die **Interoperabilität**. Hier wird unterschieden, welche Möglichkeiten jedes System zur Kommunikation mit anderen Anwendungen bietet. Die erste Stufe weist ausschließlich eine proprietäre Kommunikation mit anderen Systemen auf, d. h. es existieren nur proprietäre Schnittstellen und der Datenaustausch setzt ebenso auf








Bewertung der Interoperabilität	
	Proprietäre Kommunikation
	Einsatz von Standards
	Interoperabilität auf der Datenebene
	Interoperabilität auf der Anwendungsebene
	Unterstützung von mehreren Protokollen

Tabelle 6.1: Interoperabilitätskriterium

ein proprietäres Kommunikationsprotokoll. Standardisierte Schnittstellen und Protokolle kommen auf der zweiten Stufe zum Einsatz und ermöglichen einen einfacheren Datenaustausch zwischen zwei Anwendungen, die auf dieselben Standards setzen. Die dritte Stufe erlaubt Interoperabilität auf der Datenebene, d. h. mehrere Anwendungen können flexibel Daten austauschen. Der entfernte Aufruf von Funktionalität wird mit der vierten Stufe beschrieben, die Interoperabilität auf der Anwendungsebene erlaubt. Die fünfte und höchste Stufe setzt gleichzeitig verschiedene standardisierte Protokolle zum Datenaustausch ein und kann Funktionen in anderen Anwendungen aufrufen. Das Kriterium der Interoperabilität und dessen fünf Stufen sind in Tabelle 6.1 zusammengefasst. Über die Füllmenge des Kreises wird die Stufe der Interoperabilität bei der Bewertung beschrieben, die ein System erreicht.

Das zweite Kriterium beschreibt die **Adaptivität** eines Systems bzw. die Kopplung von Anwendungen und deren Flexibilität, um eine Änderung dieser Kopplungen durchzuführen. Die erste Stufe bedeutet, dass keine Kopplung zwischen Anwendungen vorhanden ist. Eine enge Kopplung zwischen Anwendungen ist gleichbedeutend mit Stufe zwei des Kriteriums. Integrationsumgebungen, deren Anwendungen lose gekoppelt sind und sich damit einfach und schnell ändern lassen, werden in Stufe drei eingeordnet. Die Wiederverwendung von Serviceschnittstellen bei der Integration wird durch Stufe vier repräsentiert. Die höchste Stufe fünf bedeutet eine dynamische






Bewertung der Adaptivität	
	Keine Kopplung
	Eng gekoppelte Komponenten
	Lose Kopplung
	Wiederverwendung von Services
	Dynamische Bindung

Tabelle 6.2: Adaptivitätskriterium

Bindung von Services, d. h. die Zielanwendung einer Nachricht wird nicht statisch, sondern erst zur Laufzeit bestimmt. Tabelle 6.2 zeigt das Adaptivitätskriterium in der Übersicht.

Die **Servicevermittlung** ist das dritte Kriterium zur Bewertung der unterschiedlichen Systeme. Keine Servicevermittlung findet dabei auf der ersten Stufe statt, d. h. integrierte Anwendungen müssen ihre Nachrichten in einer Form versenden, so dass die Zielanwendung diese interpretieren kann. Die zweite Stufe bietet Übersetzungsservices an, die Nachrichten zwischen der versendenden und empfangenden Anwendung in ein Datenformat übersetzt, das die Zielanwendung interpretieren kann. Routingservices dienen der dynamischen Zielbestimmung und können zum Verfolgen aller Nachrichten eingesetzt werden, die mit Stufe drei beschrieben sind. Eine Orchestrierung der Services zu Workflows ist gleichbedeutend mit Stufe vier, bei der Prozesse flexibel durch IT-Systeme unterstützt werden. In der fünften Stufe wird die statusbehaftete Serviceunterstützung erreicht. Dies bedeutet, dass Services einen Status besitzen und so Informationen über mehrere Nachrichten hinweg speichern können. Diese fünf Stufen des Kriteriums der Servicevermittlung sind übersichtlich in Tabelle 6.3 zusammengefasst.

Das vierte Bewertungskriterium ist die **Skalierbarkeit** der untersuchten Systeme. Die erste Stufe enthält Systeme, die nicht skalierbar sind. Dies führt bei steigenden Anforderungen schnell zu Performanzverschlechterungen.






Bewertung der Servicevermittlung	
	Keine Servicevermittlung
	Übersetzungsservices
	Routingservices
	Orchestrierung von Services
	Statusbehaftete Serviceunterstützung

Tabelle 6.3: Servicevermittlungskriterium

Sofern eine Anwendung nicht skalierbar ist, kann in der Regel der Einbau von leistungsstärkerer Hardware, auf der die Software ausgeführt wird, eine Verbesserung bringen, was Stufe zwei des Kriteriums entspricht. Dies sorgt für eine verbesserte Arbeitsgeschwindigkeit, wodurch mehr Operationen in der gleichen Zeit durchführbar sind. Diese Skalierung kann auf einem dedizierten System jedoch nicht beliebig gesteigert werden, da ein verteiltes Rechnen auf mehreren Prozessorkernen nicht möglich ist. Dies wird in der dritten Stufe ermöglicht, in der die Skalierung der Software durch ihre Parallelisierbarkeit gekennzeichnet ist, bei der durch gleichzeitige Ausführung von mehreren Instanzen eines Services mehr Operationen pro Zeiteinheit abarbeitbar sind. In der vierten Stufe sind Systeme enthalten, die nicht nur Services mehrfach ausführen, um eine schnellere Abarbeitung von Aufgaben zu ermöglichen, sondern die gesamte Integrationsumgebung wird mehrfach ausgeführt. Dies erfordert die Vervielfachung der Integrationsumgebung mit dem ESB und allen zugehörigen Services. Bei der Integration von neuen Anwendungen oder Domänen ist dabei auch eine Anpassung der einheitlichen Datenformate notwendig. Diese Skalierbarkeit kann weiter verbessert werden, indem alle Services und Integrationsumgebungen in einer Cloudumgebung ausgeführt werden, dies wird durch Stufe fünf repräsentiert. Damit wird erreicht, dass die derzeit höchstmögliche Stufe der Skalierbarkeit erreicht wird und die In-






Bewertung der Skalierbarkeit	
	Keine Skalierbarkeit
	Hardware (Prozessorgeschwindigkeit)
	Hardware (Mehrkernprozessor) & parallelisierbare Software/Services
	Parallelisierbare Integrationsumgebung
	Cloud-Unterstützung

Tabelle 6.4: Skalierbarkeitskriterium

tegrationsarchitektur den Anforderungen der Elastizität genügt. Alle Stufen der Skalierbarkeit sind in Tabelle 6.4 dargestellt.

Die **Prozessunterstützung** stellt das fünfte Kriterium zur Bewertung der unterschiedlichen Systeme dar. Keine Prozessunterstützung definiert Stufe eins, wobei eine Anwendung nur einzelne Aufgaben unterstützt und dabei nicht mit anderen Anwendungen integriert ist. Die zweite Stufe beinhaltet Anwendungen, die einen Prozess fest implementieren und dieser sich nicht durch umkonfigurieren ändern lässt. Eine flexible Prozessunterstützung findet statt, wenn Prozesse sich leicht ändern lassen. Dies wird in größeren Anwendungen oder Softwarepaketen durch integrierte Workflowsysteme realisiert. Diese bilden Stufe drei ab und sind im Unterschied zur Stufe vier auf die Anwendung eines Herstellers fixiert. Diese anwendungsübergreifende Prozessunterstützung kann, unabhängig zwischen beliebigen Anwendungen, Prozesse durchgängig durch IT-Systeme unterstützen und flexibel ändern. Aufgrund der Datenmodelle und Anwendungen ist eine solche Prozessunterstützung in der Regel auf einzelne Domänen beschränkt. Eine domänenübergreifende Prozessunterstützung ist demnach durch Stufe fünf definiert und erlaubt eine flexible Ausführung von Prozessen zwischen beliebigen Anwendungen unterschiedlicher Domänen. Die Übersicht des Kriteriums zur Prozessunterstützung ist in Tabelle 6.5 zusammengefasst.






Bewertung der Prozessunterstützung	
	Keine Prozessunterstützung
	Starre Implementierung der Prozesse
	Flexible Prozessunterstützung
	Anwendungsübergreifende Prozessunterstützung
	Domänenübergreifende Prozessunterstützung

Tabelle 6.5: Prozessunterstützungskriterium

Das sechste und letzte Kriterium beinhaltet die Form der **Integration** von verschiedenen Anwendungen. Die erste Stufe enthält Systeme, die keine Integration aufweisen. Punkt-zu-Punkt-Verbindungen sorgen bei wachsenden Systemlandschaften schnell zu sehr komplexen und schwer wartbaren Architekturen, die in Stufe zwei zusammengefasst sind. Eine zentrale Datenbank kommt meist bei der Integration von mehreren Werkzeugen eines Herstellers zum Einsatz und ist durch Stufe drei repräsentiert. EAI setzt, wie bereits in Abschnitt 2.2.1 beschrieben, zur Integration von Anwendungen einen zentralen Message Broker ein, der zusammen mit den Anwendungen ein „Hub-and-Spoke“ darstellt und in Stufe vier eingeordnet wird. Die Stufe fünf bietet die flexibelste Art der Integration mit einem Service Bus. Das ESB-Konzept erlaubt nicht nur Services lose zu koppeln. Darüber hinaus ist eine adaptive Prozessunterstützung durch die flexible Komposition von Services in Workflows möglich. Diese können direkt von einer Workflow Engine ausgeführt werden, die in den meisten ESB-Lösungen enthalten ist. Eine Zusammenfassung des Integrationskriteriums befindet sich in Tabelle 6.6.

### 6.2.2 Vergleich mit anderen Systemen

Neben der ACITA werden fünf weitere Systeme auf Basis der eben vorgestellten Kriterien bewertet. Dies sind die phasenspezifischen Integrationsumge-






Bewertung der Integration	
	Keine Integration
	Punkt-zu-Punkt Schnittstellen
	Zentrale Datenbank
	Hub-and-Spoke / Message Broker
	Service Bus

Tabelle 6.6: Integrationskriterium

bungen des MSB [Min12] und der EngBus [KS11]. Darüber hinaus wird die Integrationsplattform CHAMPAGNE für Anwendungen der Produktionsplanung in der Lernfabrik aIE bewertet [Hei11]. Zusätzlich wird die *Smart Factory* beurteilt, die auf Basis eines Kontextmodells ein einfaches Auffinden von Daten in unterschiedlichen Anwendungen erlaubt [LW07, LCW08a, LCW08b]. Kommerzielle PLM-Lösungen, beschrieben in Abschnitt 3.6, sind ebenso Teil der Bewertung. Die Bewertung anhand der Kriterien und deren fünf Stufen ist in Tabelle 6.7 dargestellt.

Die beiden phasenspezifischen Integrationsumgebungen MSB und EngBus zeigen, bis auf die Skalierbarkeit und Prozessunterstützung, identische Ergebnisse wie die ACITA. Dies liegt an der Tatsache, dass die ACITA eine konsequente Weiterentwicklung der SOA in Bezug auf eine durchgängige Unterstützung des gesamten Produktlebenszyklus bzw. gesamter Produktionsunternehmen ist. Die Entwicklung einer wandlungsfähigen IT-Architektur erlaubt eine verbesserte und einfachere Anpassbarkeit, so dass alle IT-Ressourcen eines Unternehmens durch eine einheitliche Integrationsplattform unterstützt werden. Hinzu kommt, dass der Informationsaustausch mit Partnern auf derselben Basis durchgeführt werden kann. Die Prozessunterstützung ist deshalb nicht mehr auf Domänen beschränkt und kann auf Basis von standardisierten Prozessbeschreibungssprachen phasenübergreifend definiert und ausgeführt werden.

	MSB	EngBus	CHAM- PAGNE	Smart Factory	Kommerzielle PLM- Lösungen	ACITA
Interopera- bilität	●	●	◐	◐	○	●
Adaptivität	●	●	◑	◐	◑	●
Service- vermittlung	◑	◑	◐	◑	○	◑
Skalierbarkeit	◐	◐	◑	◑	◑	◑
Prozess- unterstützung	◑	◑	◐	◑	◑	●
Integration	●	●	◐	◐	◐	●

Tabelle 6.7: Bewertung und Vergleich der ACITA mit anderen Lösungsansätzen

Im Vergleich zur ACITA bietet CHAMPAGNE keine flexible Prozessunterstützung und ist an den DPE gebunden, der aufgrund seines Datenmodells eine phasenübergreifende Integration erschwert. Die Interoperabilität ist auf die proprietären VBA-Skripte beschränkt, wodurch Änderungen aufwendig sind. Dies führt zu einer engen Kopplung zwischen dem DPE und dem Propagationssystem, das wiederum Adapter und Hilfskomponenten zur Integration der Anwendungen erfordert. Die Propagationsskripte ermöglichen dagegen das dynamische Weiterleiten von Änderungsnachrichten zwischen den integrierten Systemen, allerdings keine flexible Unterstützung von Prozessen. Die Integration selbst beruht auf dem Propagationssystem und dem DPE, wodurch alle Daten immer in der zentralen Datenbank des DPE gespeichert, bevor sie an andere Anwendungen weiter geleitet werden.

Die Smart Factory bietet ein Kontextmodell, um anwendungsübergreifend Daten auffinden zu können. Da dieses Kontextmodell passiv ist, erlaubt es im Gegensatz zur ACITA keinen aktiven Datenaustausch zwischen Anwendungen. Die Definition von anwendungsübergreifenden Workflows ist hingegen

innerhalb der Smart Factory möglich, allerdings nur im Rahmen des Kontextmodells, wodurch eine domänenübergreifende Prozessdefinition über die Digitale Fabrik und Produktion hinaus verhindert wird. Auf Basis des Kontextmodells können Daten sehr einfach gefunden und abgerufen werden, wodurch die Anforderungen für Interoperabilität auf der Datenebene erfüllt sind. Die Anwendungen, über die sich das Kontextmodell ausdehnt, sind nicht starr an diese gebunden und können somit flexibel eine Veränderung erfahren. Die Verknüpfung der Daten, die in Anwendungen gespeichert sind, über das Kontextmodell ermöglicht eine Übersetzung zwischen den proprietären Datenformaten der Anwendungen. Eine Integration ist deshalb über die Datenbank gegeben, in der das Kontextmodell abgelegt ist.

Die kommerziellen PLM-Lösungen bieten aufgrund ihrer zentralen Datenbank, die alle Werkzeuge integriert, einen durchgängigen Datenaustausch innerhalb des Produktlebenszyklus. Die Prozesse sind jedoch starr in den einzelnen Anwendungen implementiert und lassen sich nicht flexibel anpassen. Eine anwendungsübergreifende Prozessdefinition ist nur indirekt durch den Datenaustausch über die Datenbank gegeben. Aufgrund der proprietären Schnittstellen, sofern offene Schnittstellen überhaupt vorhanden sind, kann nur eine unflexible Integration mit der Datenbank realisiert werden. Dies führt wiederum zu einer engen Kopplung der Komponenten, weshalb Anpassungen nur mit größerem Aufwand möglich sind. Daten werden nur über die zentrale Datenbank ausgetauscht, Übersetzungsservices für einen direkten Datenaustausch zwischen Werkzeugen existieren somit nicht.

### **6.3 Erweiterung der ACITA auf andere Forschungsgebiete**

Dieser Abschnitt beschreibt Forschungsergebnisse, die zum einen die Anbindung von mobilen Endgeräten an die ACITA untersuchen, um den Mitar-



beitenden orts- und zeitunabhängig Informationen zur Verfügung stellen zu können. Zum anderen ist der Einsatz einer SOA zur schnelleren Inbetriebnahme von rekonfigurierbaren Maschinen Gegenstand der Untersuchungen.

### 6.3.1 Integration von mobilen Endgeräten in die Unternehmensinfrastruktur

Der Trend weg vom klassischen Personal Computer (PC) hin zu mobilen Endgeräten, wie z. B. Smartphones oder Tablet-Computern, schreitet kontinuierlich voran. Mit *Bring-your-own-Device* erlauben immer mehr Unternehmen, dass private Smartphones oder Tablet-PCs von Mitarbeitenden für ihre tägliche Arbeit im Büro einsetzen. Damit werden neue Möglichkeiten geschaffen in Bezug auf orts- und zeitunabhängiges Arbeiten. Die Einbindung von mobilen Endgeräten in bestehende Prozesse ermöglicht deren Umgestaltung, um die Vorteile der höheren Mobilität zu nutzen. Zudem integrieren alle mobilen Endgeräte eine Vielzahl an Sensoren und Kameras, die neue Möglichkeiten bei einzelnen Aufgaben bieten. Die Fehlererkennung kann z. B. direkt mit einem Bild dokumentiert werden, das im weiteren Verlauf der Fehlerbehebung hilfreich sein kann. Durch diese Geräte entstehen jedoch gleichzeitig neue Risiken, die mit Kontrollverlust über Unternehmensdaten auf privaten Geräten und Sicherheitslücken einhergehen.

Produktionsunternehmen sollten deshalb ihren Mitarbeitenden durch die Bereitstellung von Unternehmens-Apps die benötigten Daten orts- und zeitunabhängig in einer sicheren Form zur Verfügung stellen. Mit Apps werden in diesem Zusammenhang Anwendungen bezeichnet, die speziell für Smartphones oder Tablet-PCs entwickelt und auf diesen ausgeführt werden. Damit lässt sich die Arbeit flexibilisieren und schlussendlich die Produktivität steigern. Deshalb benötigen die Unternehmen ein systematisches Vorgehen, das im Folgenden im Rahmen eines Frameworks in fünf Teilen dargelegt ist [GSWM13]:

- Die Identifikation von Unternehmensprozessen, die für den Einsatz von Apps geeignet sind, indem Prozessschritte orts- und zeitunabhängig durchgeführt oder die Gerätefähigkeiten der mobilen Endgeräte genutzt werden. Durch diese Flexibilisierung kann man eine Verbesserung der Prozessabläufe erreichen.
- Die IT-Infrastruktur muss die notwendigen Schnittstellen bereitstellen, um einen auf die Mobilgeräte abgestimmten Datenaustausch zu ermöglichen. Zudem ist die Reduktion der übertragenen Datenmenge, aufgrund der begrenzten Bandbreite von mobilen Endgeräten, notwendig.
- Die Erstellung muss durch eine Plattform insofern vereinfacht werden, so dass Mitarbeiter selbst, die für ihre Aufgaben relevanten Apps, entwickeln können. Die Eigenentwicklung von Apps kann man nur durch Bereitstellung eines standardisierten Rahmenwerks erreichen, das z. B. Entwicklungsmuster und Datenmodelle definiert. Die Einbindung von Smartphones oder Tablet-PCs in die eigenen Arbeitsschritte erlaubt den Mitarbeitenden diese effizienter zu gestalten, indem jede App ausschließlich relevante Informationen bereitstellt. Zudem wird ein mobiles und zeitunabhängiges Arbeiten ermöglicht.
- Die erstellten Apps müssen im Rahmen einer Plattform zentral durch das Unternehmen verwaltet werden, um die Aktualisierung der Apps zentral zu überwachen und bei Bedarf den Zugriff auf Daten einzuschränken.
- Neben dem Potential für Optimierungen, die die mobile Technologie mit sich bringt, müssen gleichzeitig die Risiken berücksichtigt werden. Bei der Nutzung von Daten im mobilen Bereich ist deren Sicherheit von besonderer Bedeutung, denn durch verlorene Smartphones oder Tablet-PCs können auch Geschäftsgeheimnisse verloren gehen. Die Verschlüsselung der Daten bei der drahtlosen Übertragung und auf

dem Gerät selbst müssen deshalb gewährleistet sein. Zusätzlich kann die Privatsphäre der Mitarbeitenden verletzt werden, indem ständig deren Position überwacht wird, z. B. durch das Auslesen der Global Positioning System (GPS) Sensoren. Dieses Vorgehen ist unter allen Umständen zu verhindern, indem z. B. ein Rechtemanagement auf dem mobilen Endgerät eingesetzt wird, womit der Benutzende mehr Kontrolle über seine kontextsensitiven Daten bekommt. Ein Beispiel für ein solches System ist CRÊPE [CCFZ12].

Im Rahmen dieser Arbeit wird die ACITA als flexible IT-Infrastruktur zur Integration von mobilen Anwendungen untersucht. Zur Validierung wird eine Fabrikplanungstisch-App entwickelt, die zum einen die mobile Layoutplanung direkt in der physischen Fabrik erlaubt, zum anderen den bisher getrennten Fabrikplanungstisch in der Digitalen Fabrik und das Wartungsportal in einer App zusammenfasst. Die App soll eine einfache und berührungsgesteuerte Umplanung des Fabriklayouts erlauben. Die Ausgangssituation zur Umplanung ergibt sich aus der aktuellen Konfiguration der Montagelinie, die über eine Nachricht an den CBR des MSB geladen wird.

Der Benutzende kann über die Fabrikplanungstisch-App ein Update des aktuellen Fabriklayouts initiieren. Hierfür sendet das mobile Endgerät eine XML-Nachricht an den CBR des MSB, der diese an den Leitrechnerservice weiter leitet. Die Layoutinformationen werden über dieselbe Serviceschnittstelle der Leitrechnerdatenbank geladen, die bereits für den Layoutumplanungsworkflow entwickelt wurde. Die Antwortnachricht wird über den CBR zurück an das mobile Endgerät gesendet und dort in dessen lokalen Speicher abgelegt.

Alle gespeicherten Layouts werden in der App in Form einer Liste angezeigt. Der Benutzende kann ein Layout zum Bearbeiten öffnen oder es aus der Liste löschen, wenn es nicht mehr benötigt wird. Wird das Layout geöffnet, so werden die Module entsprechend ihrer Position in einer Grid-Struktur angezeigt.

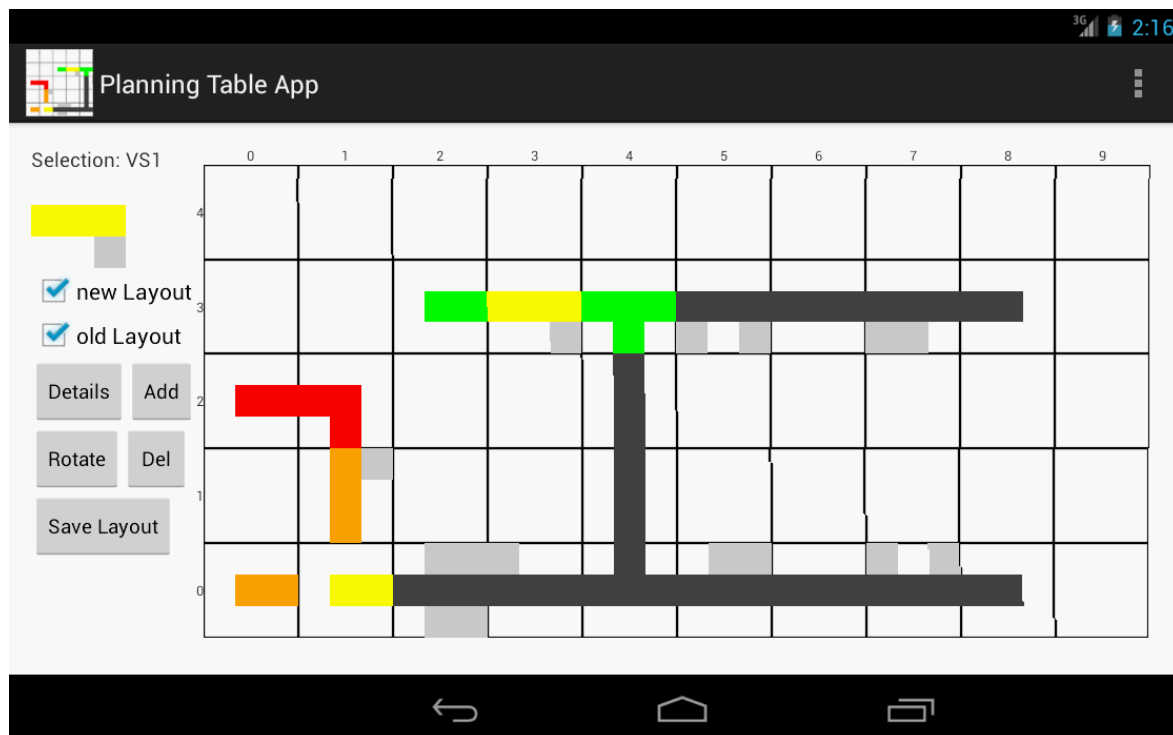


Abbildung 6.5: Screenshot der Planungstisch-App

Zur Unterscheidung der unterschiedlichen Modultypen kommen Piktogramme zum Einsatz. Die Piktogramme können über die berührungsempfindliche Oberfläche des mobilen Endgerätes verschoben, rotiert, gelöscht oder neu hinzugefügt werden, um ein neues Layout zu erhalten. Zur Unterstützung des Umbaus der Montagelinie kann das ursprüngliche und neue Layout getrennt voneinander angezeigt werden, dabei werden Änderungen in beiden Layouts farblich hervorgehoben. So ist eine schnelle Erkennung möglich, welche Module nach dem Umbau nicht mehr benötigt, an eine andere Position verschoben oder neu in die Montagelinie eingebaut werden. Ein Screenshot der Planungstisch-App ist in Abbildung 6.5 dargestellt.

### 6.3.2 SOA für die Maschineninbetriebnahme

Die Inbetriebnahme von rekonfigurierbaren Maschinen ist heutzutage ein aufwendiger Prozess, in dem viele Prozessschritte nacheinander ausgeführt

und die Parameter der einzelnen Maschinenteile und Steuerungen richtig eingestellt werden müssen. Im Rahmen einer Untersuchung ist das Potential einer SOA zur automatisierten Maschinenkonfiguration durchgeführt worden. Ein Konzept zur Umsetzung der SOA wurde entwickelt, um die Inbetriebnahme nach einer Rekonfiguration deutlich zu verkürzen [AKSM11].

Durch Erweiterung der Maschinensteuerungen um das serviceorientierte Konzept könnten diese direkt in den MSB eingebunden und damit eine nahtlose Kommunikation zwischen den Planungs-, Steuerungswerkzeugen und Maschinen ermöglicht werden. Insbesondere die Rückmeldung des Maschinenstatus und der abgearbeiteten Produktionsprozesse lassen sich direkt in Wartungssysteme und die Softwaresysteme der Produktionsplanung übertragen und stehen dort für Wartungsplanungen und weitere Optimierungen der Produktion zur Verfügung. Das Potential von Serviceschnittstellen in Maschinen ist enorm, wie die bisherigen Untersuchungen zur Inbetriebnahme von rekonfigurierbaren Maschinen zeigen.

## 6.4 Evaluierung der ACITA

Die ACITA bietet gegenüber bisherigen Softwarelösungen viele Vorteile, die im Folgenden näher betrachtet werden.

Im Vergleich zu CIM bietet die ACITA zwei Vorteile, um eine durchgängige Integration von Produktionsunternehmen erfolgreich zu erreichen. Zum einen setzt die ACITA auf eine adaptive Integration der heterogenen Anwendungslandschaft und benutzt dazu standardisierte Serviceschnittstellen. Das ESB-Konzept verwendet ebenso standardisierte Technologien und Protokolle zur losen Kopplung der Services. Zum anderen ist die Einbindung von Mitarbeitern in Prozesse durch die ACITA in Form von *Human Tasks* problemlos möglich, wodurch eine durchgängige Zusammenarbeit zwischen Nutzenden und IT-Systemen erreicht wird.

Gegenüber den kommerziellen PLM-Lösungen von Siemens, Dassault Systèmes und PTC bietet die ACITA einen höheren Grad an Flexibilität. Während bei Siemens und Dassault Systèmes die einzelnen Softwaresysteme sehr gut integriert sind, ist das Einbinden von Software von Drittherstellern oder eigen entwickelten Anwendungen aufwendig. Im Falle von PTC können unterschiedliche Softwarelösungen auch von Drittherstellern eingebunden werden, diese sind jedoch auf ausgewählte Anwendungen begrenzt. Dies reduziert die Adaptivität der kommerziellen Lösungen in Bezug auf eine Integration von beliebigen Anwendungen, um alle Prozessschritte, Teilprozesse und Prozesse möglichst digital zu unterstützen.

Die Prozessdefinition selbst ist durch die kommerziellen Anwendungen fest vorgegeben und lässt nur wenige Anpassungen zu. Die heutzutage benötigten adaptiven Prozesse, die sich kontinuierlich ändern und folglich Anpassungen der in Software implementierten Prozesse nach sich ziehen würden, lassen sich mit diesen kommerziellen Lösungen nicht realisieren. Die fünfstufige Integrationshierarchie der ACITA erlaubt dagegen eine flexible Prozessdefinition durch Komposition von Services und kann somit nicht nur Services einer Anwendung flexibel in einen Prozess einbinden, sondern auch Prozesse zwischen verschiedenen Anwendungen lassen sich durchgehend und einheitlich mit der ACITA in einer flexiblen Weise unterstützen. Die Unterstützung von Prozessen zwischen den kommerziellen PLM-Lösungen und Anbietern von Drittsoftware ist, sofern vorhanden, auf starre Schnittstellen beschränkt und damit auch unflexibel. Die ACITA erlaubt hingegen nicht nur eine flexible Prozessdefinition zwischen beliebigen Anwendungen innerhalb einer Domäne, sondern darüber hinaus auch die Unterstützung domänenübergreifender und adaptiver Prozesse über den PLM-Bus.

Die für die Lernfabrik entwickelte Integrationsplattform CHAMPAGNE erlaubt eine flexible Integration von Anwendungen. Diese Flexibilität wird jedoch durch den DPE als zentraler Datenspeicher wieder eingeschränkt, da Änderungen und Erweiterungen aufwendig sind. Die in der CHAMPAGNE-Plattform

eingesetzten Propagations- und Transformationsskripte erfordern detaillierte Kenntnisse über deren Funktionsweise und wie diese implementiert werden müssen. Dies erhöht den Aufwand gegenüber standardisierten Technologien, um Anwendungen über die Plattform integrieren zu können. Darüber hinaus unterstützt CHAMPAGNE nicht die Definition und Ausführung von Prozessen. Diese werden ausschließlich über die integrierten Anwendungen angestoßen.

Aus der Automatisierung des Datenaustausches zwischen Anwendungen und der verbesserten digitalen Unterstützung von Prozessen durch die ACITA, die damit teilweise automatisch ausführbar werden, entsteht eine kürzere Gesamtprozesszeit für die Produktentwicklung und Produktionsplanung. Dies führt zu einem schnelleren Produktionsstart, wodurch die Produkte früher am Markt platzierbar sind und sich Wettbewerbsvorteile des Produktionsunternehmens gegenüber der Konkurrenz ergeben.

Die Einführung der Wandlungsfähigkeit in die IT-Architektur durch Hinzufügen oder Entfernen von phasenspezifischen ESBs ermöglicht eine einfache Anpassungsfähigkeit. Die phasenspezifischen ESBs ermöglichen eine flexible Integration im Kleinen, wobei der PLM-Bus die Integration im Großen vollzieht.

Aus organisatorischer Sicht passt sich die ACITA einer hierarchischen Organisationsstruktur oder Matrixorganisation an. Dies hat zur Folge, dass die organisatorischen Anpassungen durch Einführung der neuen IT-Landschaft mit phasenspezifischen ESBs und dem PLM-Bus, im Vergleich zu einer einzigen zentralen Lösung, geringer sind, da das Prinzip der Trennung von Verantwortlichkeiten (engl.: separation of concerns) beibehalten werden kann.

### 6.4.1 Erfüllung der Forschungsobjekte

Die Implementierung der serviceorientierten Integrationsumgebungen PPSB und des PLM-Busses ermöglichen über standardisierte Serviceschnittstellen die adaptive Integration der heterogenen Anwendungslandschaften in Unternehmen. Zusammen mit dem MSB können Daten über Domänengrenzen hinweg zwischen beliebigen Anwendungen ausgetauscht werden. Die CBRs sorgen für eine dynamische Nachrichtenvermittlung und können über die lokalen Serviceverzeichnisse jeden verfügbaren Service erreichen.

Die Softwarelandschaften in Produktionsunternehmen bestehen heute in der Regel aus ERP- und PLM-Systemen, sowie einer Vielzahl kleinerer Anwendungen für spezielle Aufgaben. All diese Softwaresysteme sind teilweise durch punktuelle Schnittstellen verbunden, erlauben jedoch keine durchgängige Unterstützung der Prozesse. Die ACITA erlaubt diese durchgehende Integration aller Anwendungen über seine Multi-ESB-Architektur. Damit wird die Anzahl der Schnittstellen reduziert, wodurch eine Komplexitätsreduktion erreicht wird.

Der modulare Aufbau der ACITA erlaubt über den PLM-Bus die Anzahl an integrierten phasenspezifischen Integrationsumgebungen zu ändern. Diese Wandlungsfähigkeit der Integrationsarchitektur sorgt für eine einfache Skalierbarkeit, wodurch die ACITA an jede Unternehmensgröße beliebig angepasst werden kann.

Die flexible Komposition der Services zu Workflows erlaubt Geschäftsprozesse adaptiv zu unterstützen und ermöglicht gleichzeitig die Einbindung von Mitarbeitenden in die Prozesse. Diese Workflows können nicht nur anwendungsübergreifend innerhalb einer Phase definiert werden, sondern sorgen mithilfe des PLM-Busses für eine domänenübergreifende Prozessunterstützung.



Ziele der Arbeit	Forschungsobjekte	Beitrag	Erfüllung
Z-1.1: Flexible Integration der bestehenden heterogenen Anwendungslandschaft im Unternehmen	FO-1: Integration der Anwendungen durch Implementierung von standardisierten Serviceschnittstellen verbessern und Datenaustausch zwischen verschiedenen Domänen durch lose Kopplung der Serviceschnittstellen ermöglichen.	Implementierung des PPSB und PLM-Bus	<ul style="list-style-type: none"> <li>• Konzept</li> <li>• Implementierung</li> <li>• Test</li> <li>• Szenario 1 &amp; 2</li> </ul>
Z-1.2: Komplexität der IT-Landschaft reduzieren und somit Wartbarkeit verbessern	FO-2: Einheitliche Integrationsarchitektur über eine serviceorientierte Middleware, um Punkt-zu-Punkt Integrationsverbindungen zu ersetzen. Die reduzierte Komplexität sorgt für einfachere und schnellere Durchführung von Änderungen.	Implementierung des PPSB und PLM-Bus	<ul style="list-style-type: none"> <li>• Konzept</li> <li>• Implementierung</li> <li>• Test</li> <li>• Szenario 1 &amp; 2</li> </ul>
Z-1.3: Einfache Veränderung der IT-Landschaft ermöglichen	FO-3: Den Aspekt der Wandlungsfähigkeit in die IT-Architektur einführen, um die Architektur, sowohl für kleine und mittlere Unternehmen, als auch für große Unternehmen anwendbar zu machen.	Modifikation der phasenspezifischen ESBs in ACITA	<ul style="list-style-type: none"> <li>• Konzept</li> <li>• Implementierung</li> <li>• Szenario 3</li> </ul>
Z-2.1: Adaptivität der Prozesse erhöhen	FO-4: Eine adaptive IT-Unterstützung von Prozessen innerhalb und zwischen unterschiedlichen Domänen ermöglichen, um diese einfach und schnell an geänderte Anforderungen anpassen zu können.	Hierarchie der einzelnen Integrationsumgebungen	<ul style="list-style-type: none"> <li>• Konzept</li> <li>• Implementierung</li> <li>• Test</li> <li>• Szenario 1 &amp; 2</li> </ul>
Z-3.1: Austausch von Informationen zwischen dem Unternehmen und seinen Partnern	FO-5: Die IT-Architektur soll einen flexiblen Datenaustausch mit anderen Unternehmen ermöglichen und Prozesse über Unternehmensgrenzen hinweg durch Softwaresysteme unterstützen.	Modulares und erweiterbares Konzept der ACITA	<ul style="list-style-type: none"> <li>• Konzept</li> <li>• Implementierung</li> <li>• Szenario 4</li> </ul>
Z-3.2: Integration von mobilen Endgeräten in IT-Architektur	FO-6: Einheitliche Schnittstellen zur einfachen Datenbereitstellung für mobile Anwendungen.	Implementierung der Planungstisch-App und Integration in die ACITA	<ul style="list-style-type: none"> <li>• Konzept</li> <li>• Implementierung</li> <li>• Test</li> </ul>
Z-3.3: Auswirkungen von Änderungen auf die Organisation berücksichtigen	FO-7: Die Reorganisation, die mit Änderungen an der IT-Landschaft einhergeht, sollte minimiert werden, um Mitarbeitenden den Wandel zu erleichtern und damit weniger Widerstände aufzubauen.	Hierarchische Struktur der ACITA deckt sich mit Organisationsstruktur	<ul style="list-style-type: none"> <li>• Konzept</li> </ul>

Tabelle 6.8: Erfüllung von Zielen und Forschungsobjekten der Arbeit

IT-Systeme von externen Partnern können auf dieselbe Weise in die ACITA eingebunden werden, wie dies mit phasenspezifischen Integrationsumgebungen durchgeführt wird. Die zu berücksichtigenden höheren Sicherheitsanforderungen sind bereits in Abschnitt 6.1.4 beschrieben. Damit lässt sich nicht nur ein nahtloser Datenaustausch mit den Partnern umsetzen, sondern auch Prozesse zwischen den Parteien können durch die entsprechenden IT-Systeme flexibel unterstützt werden.

Der Datenaustausch zwischen den Backendsystemen eines Unternehmens und den mobilen Endgeräten ist wichtig, wenn die Vorteile der Mobilität ausgenutzt werden sollen. Die Umgestaltung von Prozessen erlaubt die Verlagerung von einzelnen Prozessschritten auf mobile Endgeräte, die deren Vorteile hinsichtlich der Geräteeigenschaften, Mobilität und damit zeitliche Unabhängigkeit, nutzen können.

Die hierarchische Struktur der ACITA erlaubt, im Vergleich zu einer zentralen Lösung, eine bessere Umsetzung in den gängigen hierarchischen Organisationsstrukturen und Matrixorganisationen. Die Trennung der Zuständigkeiten lässt sich mit dieser Integrationsarchitektur besser umsetzen und führt zu einer effizienteren Weiterentwicklung, da Entscheidungen teilweise ohne gegenseitige Auswirkungen schneller getroffen werden können.

Der Beitrag und die Erfüllung der einzelnen Ziele und Forschungsobjekte ist in Tabelle 6.8 anschaulich zusammengefasst.

## **6.5 Zusammenfassung**

Dieses Kapitel beschreibt die Anwendung der ACITA anhand von vier Anwendungsszenarien, um sie im Anschluss gegenüber anderen Integrations- und Softwarelösungen anhand von sechs Kriterien zu evaluieren.

Das erste Anwendungsszenario beschreibt eine flexible und robuste Integration der Produktionsplanung durch den PPSB. Der CBR des PPSB sorgt für eine dynamische Nachrichtenvermittlung zwischen den integrierten Anwendungen und den zugehörigen Warteschlangen. Dadurch können kurze Anwendungsausfälle oder Netzwerkstörungen überbrückt werden, ohne dadurch andere Anwendungen zu blockieren oder zu stören.

Der nahtlose Datenaustausch mithilfe des PLM-Busses zwischen unterschiedlichen Phasen ist im zweiten Anwendungsszenario dargestellt. Die dynamische Nachrichtenvermittlung über die CBRs und das implementierte Serviceverzeichnismodell ermöglicht die lose Kopplung aller Services, die in einem Produktionsunternehmen bereitgestellt werden. Hinzu kommt eine domänenübergreifende Prozessdefinition und Ausführung auf Basis der Services, die für eine verbesserte IT-Unterstützung der Prozesse sorgt.

Das dritte Anwendungsszenario stellt die einfache Einbindung von neuen phasenspezifischen Integrationsumgebungen in die ACITA dar. Diese Wandlungsfähigkeit der Architektur ermöglicht eine einfache Veränderung und kann damit an die jeweiligen Anforderungen angepasst werden.

Die Einbindung von externen Partnern in die eigenen Prozesse ist für eine effiziente Kooperation von großer Bedeutung und ist im Anwendungsszenario vier beschrieben. Insbesondere die Anforderungen an höhere Sicherheit und Vertraulichkeit der Daten wird hier fokussiert.

Die systematische Bewertung der ACITA erfolgt anhand von sechs Kriterien. Diese sechs Kriterien definieren im Bereich der Interoperabilität, Adaptivität, Servicevermittlung, Skalierbarkeit, Prozessunterstützung und Integration jeweils fünf Stufen, die aufeinander aufbauen. Im Anschluss wird die Bewertung der ACITA, sowie der Integrations- und Softwarelösungen aus Forschung und Industrie, durchgeführt. Dabei zeigt sich die ACITA in jedem Kriterium gegenüber den anderen Lösungen gleichwertig oder überlegen.

Über alle Kriterien hinweg ist die ACITA für eine flexible Integration eines Produktionsunternehmens bestens geeignet.

Die Erweiterung der ACITA, zur Integration von mobilen Endgeräten und der direkten Steuerung von Maschinen, zeigt das Potential dieser Lösung auf. Die erste integrierte App, die eine mobile Version des Fabrikplanungstisches darstellt, konnte mit geringem Aufwand an die ACITA angebunden werden, um ein aktuelles Layout vom Leitrechner der iTRAME-Montagelinie zu laden.

Abschließend wird der Beitrag der vorliegenden Arbeit zur Erfüllung der in Kapitel 1, Tabelle 1.1 beschriebenen Ziele und Forschungsobjekte aufgezeigt. Diese sind anschaulich in Tabelle 6.8 zusammengefasst. Es zeigt sich, dass für jedes Ziel eine Verbesserung durch die ACITA erreicht wird.

---

## KAPITEL 7

### Fazit und Ausblick

---

Dieses Kapitel fasst die vorliegende Arbeit zusammen und gibt einen Ausblick auf mögliche Erweiterungen und Verbesserungen der ACITA. Die Erweiterungen umfassen u. a. eine adaptive und sichere Integration von mobilen Anwendungen in die ACITA. Zudem kann die ACITA als Datenplattform die Bereitstellung von Planungs- und Produktionsdaten für Analysezwecke verbessern, um daraus auf Optimierungspotential für Planung und Produktion zu schließen. Für eine vereinfachte Umsetzung der ACITA können die einzelnen Integrationsumgebungen in einer Cloud ausgeführt werden, die zugleich eine Performanzverbesserung durch bessere Skalierungsmöglichkeiten bietet.

#### 7.1 Zusammenfassung und Diskussion

Produktionsunternehmen stehen heute vor vielfältigen Herausforderungen, die sie erfolgreich meistern müssen. Auf alle Bereiche des Unternehmens wirken innere und äußere Einflüsse, die zusammen zu einem turbulenten Umfeld führen. Dieses Umfeld erfordert schnelle Reaktionen auf die aktuelle Situation, um aus den verfügbaren Ressourcen den maximalen Ertrag zu

erzielen [JWW09]. Die größten Herausforderungen stellen die Globalisierung, die zunehmende Komplexität und das turbulente Umfeld dar.

Die Öffnung der Märkte sorgt für mehr Konkurrenz. Um dieser Konkurrenz zu begegnen und ihrerseits in andere Märkte einzutreten, stellen sich immer mehr Produktionsunternehmen international auf. Mit der zunehmenden Globalisierung müssen die Unternehmen ihre Entwicklung, Produktion und den Vertrieb verteilt organisieren, um die lokalen Märkte zielgerichteter bedienen zu können.

Eine weitere Herausforderung besteht in der wachsenden Komplexität, die die Produktionsunternehmen erfolgreich beherrschen müssen. Nicht nur die Produkte bestehen aus immer mehr Varianten, deren Verwaltung deshalb immer komplexer wird. Auch die Komplexität der Prozesse nimmt immer weiter zu, unter anderem durch die global verteilten Unternehmensstandorte und ein immer variantenreicheres Produktsortiment.

Zur schnellen Reaktion auf Turbulenzen, für die Verwaltung des globalen Unternehmens und zur Beherrschung der Komplexität werden heute in der Breite Softwaresysteme eingesetzt. Um effizient Daten zwischen den Anwendungen auszutauschen, werden diese meist über Punkt-zu-Punkt Verbindungen integriert. Dabei entstehen zwei große Probleme. Zum einen führt diese Form der Integration schnell zu komplexen und schwer wartbaren IT-Landschaften, zum anderen leidet die kurzfristige Anpassungsfähigkeit unter dieser Form der starren Integration. Diese unflexible Anwendungslandschaft sorgt dafür, dass die darin implementierten Prozesse genauso unflexibel werden und damit nicht schnell genug an die aktuelle Situation angepasst werden können.

Das Stuttgarter Unternehmensmodell (SUM) beschreibt ein systematisches Vorgehen, wie Produktionsunternehmen in verschiedenen Bereichen adaptiver und wandlungsfähiger werden können, um in einem turbulenten Umfeld erfolgreich eine variantenreiche Serienfertigung zu praktizieren [WZ09].

Diese Adaptivität und Wandlungsfähigkeit wird für die IT-Infrastruktur benötigt, um Prozessanpassungen schneller durchführen zu können. Deshalb ist das Ziel der Arbeit eine unternehmensweit einheitliche Integrationsplattform zu entwickeln, die eine adaptive und wandlungsfähige Unterstützung der verteilten Anwendungsintegration bietet. Zusätzlich soll eine flexible Unterstützung der Unternehmensprozesse gewährleistet sein.

Die im Rahmen dieser Arbeit entwickelte ACITA bietet die geforderten Vorteile gegenüber heutigen Lösungen [SMM12]. Dazu wird die Integration in drei Dimensionen durchgeführt. Die erste Dimension besteht aus einer fünfstufigen Integrationshierarchie, die bereits der MSB verwendet [Min12]. Dabei werden alle Anwendungen über Serviceschnittstellen integriert, die eine neue Abstraktionsschicht bilden und die Implementierung von ihrer Schnittstellenbeschreibung trennen. Die Integrationsschicht wird durch einen phasenspezifischen ESB gebildet, der mittels eines CBRs eine lose Kopplung der verteilten Services durchführt. Die adaptive Prozessunterstützung erfolgt durch die adaptive Komposition von Services zu Workflows, die die digitale Repräsentation der Geschäftsservices und Geschäftsprozesse bilden. Die Ausführung erfolgt schließlich über die im ESB integrierte Workflow Engine.

Die zweite Dimension stellt phasenspezifische ESBs bereit, die jeweils an die spezifischen Anforderungen der entsprechenden Domäne angepasst werden können. Die Anforderungen sind u. a. Durchsatz, Datenmenge, Echtzeitanforderungen, Datenstrukturen und Latenz [SMM11]. Für die Integration der Anwendungen des Produktlebenszyklus kann so für jede Phase ein angepasster ESB bereitgestellt werden.

Die dritte Dimension sorgt für die adaptive Integration der phasenspezifischen ESBs über den PLM-Bus. Durch Veränderung der Anzahl an integrierten domänenspezifischen ESBs kann die Kapazität der Integrationslandschaft einfach verändert werden. Dies entspricht einer wandlungsfähigen

IT-Architektur und erlaubt durch diese Skalierbarkeit die flexible Anpassung an die jeweiligen Anforderungen.

Die Modularität der ACITA ermöglicht eine Skalierung der integrierten Anwendungen im Kleinen, durch die Anpassung der phasenspezifischen Integrationsumgebungen, und im Großen, indem der PLM-Bus die Anzahl an verbundenen lokalen ESBs variiert. Die Ressourcen der Integrationsumgebung können dadurch mit geringem Aufwand geändert werden, wodurch die Anforderungen an eine wandlungsfähige Architektur erfüllt wird. Die einheitliche Struktur und Verwendung von standardisierten Schnittstellen für die Integration der heterogenen und verteilten Anwendungen sorgt darüber hinaus zu einer Komplexitätsreduktion der IT-Landschaft.

Zur effizienten Verwaltung aller bereitgestellten Services eines Produktionsunternehmens wird ein hierarchisches Serviceverzeichnismodell verwendet [SKRM13]. Innerhalb einer Phase werden die Services in einem lokalen Serviceverzeichnis verwaltet. Das übergeordnete Meta-Verzeichnis sorgt dafür, dass Services phasenübergreifend gefunden werden können. CBRs in jeder Phase und dem PLM-Bus erlauben die dynamische Nachrichtenvermittlung innerhalb und über Domänengrenzen hinweg. Die Abstraktion der Datenmodelle der integrierten Anwendungen erfolgt über einheitliche Nachrichtenformate. Zur Komplexitätsreduktion wird für jede Phase ein solches Datenaustauschformat verwendet, das die zwischen den Anwendungen ausgetauschten Daten modelliert. Der PLM-Bus verwendet ein weiteres globales Datenformat, das eine Teilmenge aller phasenspezifischen Nachrichtenformate darstellt und enthält nur die Daten, die zwischen den Phasen ausgetauscht werden.

Die flexible Komposition von Services erlaubt deshalb nicht nur anwendungsübergreifende Prozesse zu unterstützen, sondern bietet durch ihre Hierarchie die geforderte adaptive und domänenübergreifende Prozessunterstützung. Damit können Produktionsunternehmen ihre Prozesse in einem turbulenten Umfeld schneller an neue Anforderungen anpassen. Diese Anpassungen wer-



den immer häufiger notwendig sein. Dies erlaubt eine bessere Ausnutzung der vorhandenen Ressourcen und bietet letztendlich die Möglichkeit Produkte schneller, mit besserer Qualität oder kostengünstiger herzustellen.

Aktuelle Bemühungen die Produktion durch einen breiten Einsatz von IT-Systemen und deren Vernetzung zu optimieren, werden unter dem Begriff der „Industrie 4.0“ zusammengefasst und wird als vierte industrielle Revolution bezeichnet<sup>1</sup>. Die ACITA demonstriert, wie eine solche systematische Integration von heterogenen und verteilten Anwendungen über Domänengrenzen hinweg durchgeführt werden kann und löst damit eine der grundlegenden Probleme für einen durchgängigen Einsatz mit Softwareanwendungen.

## 7.2 Ausblick

Die ACITA bietet neben der flexibleren Anwendungsintegration und Prozessunterstützung Potential für weitere Optimierungen in Produktionsunternehmen, die im Rahmen von Forschungsprojekten untersucht und evaluiert werden können.

Der Einsatz von mobilen Endgeräten in Produktionsunternehmen wird zukünftig zunehmen, weshalb sich die Unternehmen darauf vorbereiten und klare Vorgaben und Vorgehen festlegen müssen. Zusätzlich müssen neue Schnittstellen zu den bestehenden Anwendungen geschaffen werden, um Prozessschritte auf die mobilen Endgeräte zu verlagern und damit neue Möglichkeiten für Mitarbeitende in ihrer täglichen Arbeit zu schaffen. Aus der besseren Verfügbarkeit von Daten innerhalb der ACITA ergeben sich neue Möglichkeiten für eine systematische Analyse der Produktionsdaten. Der modulare Aufbau ermöglicht die einzelnen Integrationsumgebungen in einer Cloud auszuführen. Die Wandlungsfähigkeit kann weiter durch das

---

<sup>1</sup><http://www.acatech.de/industrie4.0> und <http://www.plattform-i40.de>

automatisierte Aufsetzen einer solchen Integrationsumgebung vereinfacht werden.

### **7.2.1 Einsatz von mobilen Endgeräten in Produktionsunternehmen**

Gröger et al. beschreiben Forschungsgebiete, die für einen erfolgreichen Einsatz von mobilen Anwendungen im Unternehmenskontext notwendig sind [GSWM13]. Insbesondere ein systematisches Vorgehen zur Identifikation von Prozessen wird benötigt, die für die Integration von mobilen Endgeräten bzw. Anwendungen geeignet sind. Die Verwaltung der mobilen Endgeräte und Anwendungen muss durch das Unternehmen möglich sein, um zentralisiert Updates von gefährdeten Apps verteilen zu können oder Unternehmensdaten von Endgeräten zu löschen, wenn z. B. Mitarbeitende aus dem Unternehmen ausscheiden. Durch die Bereitstellung einer Entwicklungsplattform mit entsprechenden Mustern und vorgegebene Schnittstellen können Mitarbeitende in die Lage versetzt werden Apps selbst zu entwickeln, die sie bei ihrer täglichen Arbeit unterstützen und damit zu einer Selbstoptimierung ihrer Arbeit beitragen. Nicht zuletzt ist die bereits erwähnte Sicherheit und Vertraulichkeit von Daten und Informationen im mobilen Kontext systematisch zu untersuchen und zu regeln, um Datenspionen keine neuen Zugangsquellen zu bieten und gleichzeitig die Privatsphäre der Mitarbeitenden sicherzustellen.

Die im Rahmen der Arbeit entwickelte Planungstisch-App zur Optimierung eines Fabriklayouts auf einem mobilen Endgerät demonstriert den einfachen Datenaustausch zwischen mobilen Anwendungen und integrierten Softwaresystemen der ACITA. Diese App bietet derzeit aber nicht die von Produktionsunternehmen geforderten Sicherheitsmerkmale, um die Daten bei der Übertragung und auf dem Endgerät selbst sicher vor unberechtigten Zugriffen zu schützen. Zudem kann die ACITA weiter entwickelt werden, um

speziell für die Integration von mobilen Anwendungen kompaktere Nachrichtenformate und sichere Schnittstellen zur Verfügung zu stellen.

### 7.2.2 Analyse von Planungs- und Produktionsdaten

Im Kontext von *Big Data* können die anfallenden Planungs- und Produktionsdaten genutzt werden, um diese zu analysieren und daraus neue Erkenntnisse für Verbesserungen zu ziehen [GNSM12]. Die Daten können in einer föderierten Datenbank oder einem produktionsorientierten Data Warehouse gesammelt werden [GSNM12]. Die Verknüpfung der Planungs- und Prozessdaten für die Analyse ergibt neue Möglichkeiten, um Rückschlüsse auf systematische Fehler in der Fertigung zu ziehen, Prozessverbesserungen aufzuzeigen, Planungen zu verbessern oder ein optimierte präventive Instandhaltung zu etablieren.

Dies kann durch Anpassung von bekannten Analysemethoden, wie z. B. Data Mining, Clustering oder Regressionsanalyse, an die Problemstellungen der Fertigung geschehen [GNM12]. Dadurch können neue Muster erkannt werden, die zu Fehlern oder zu einer präventiven Prozessoptimierung führen. Diese Verbesserungen können durch die Analyse zukünftig in Echtzeit stattfinden, wenn z. B. während der Produktion erkannt wird, dass die prognostizierte Lieferzeit nicht mehr erreichbar ist.

Die vollständige Umsetzung der ACITA erlaubt, die für die Analyse benötigten Daten direkt aus den integrierten Anwendungen und Maschinen über die Serviceschnittstellen abziehen. Durch die einheitliche Architektur können hierfür Daten aus unterschiedlichen Domänen gesammelt und in einer Datenbank integriert werden, wodurch neue Möglichkeiten für Analysen entstehen.

### 7.2.3 Umsetzung der ACITA in der Cloud

Die Anpassungsfähigkeit und Skalierbarkeit der ACITA kann weitere Verbesserungen erfahren, indem die Integrationsumgebungen in einer Cloud ausgeführt werden. Dies ermöglicht eine dynamische Anpassung der Hardwareressourcen an die aktuellen Anforderungen, wodurch die Architektur den Anforderungen der Elastizität einer Cloudumgebung genügt.

Die Wandlungsfähigkeit der Architektur kann durch die automatisierte Installation von phasenspezifischen Integrationsumgebungen durch den Standard *Topology and Orchestration Specification for Cloud Applications (TOSCA)* verbessert werden [OAS13]. Dazu muss die Topologie der Architektur formal beschrieben werden, sodass damit im Anschluss eine automatische Installation in einer Cloudumgebung erfolgen kann. Für die generischen Komponenten, die unverändert bleiben, wie der ESB, CBR und das lokale Serviceverzeichnis, erfolgt die Beschreibung durch ein statisches Servicetemplate. Das Problem ist die Beschreibung des Datenaustauschformates und der Übersetzungsservices, da diese in jeder Integrationsumgebung individuell angepasst bzw. definiert werden. Für eine vollständige TOSCA-Beschreibung ist es deshalb notwendig, das Nachrichtenmodell und den Übersetzungsservice über eine Parametrisierung des Servicetemplates der Integrationsumgebung bei der Installation bereitzustellen.

---

## **ANHANG A**

### **Codebeispiele**

---

Der Anhang enthält Schemas der einheitlichen Datenaustauschformate des PPSB und PLM-Bus, die eine neue Abstraktionsschicht zwischen den proprietären Datenformaten der integrierten Anwendungen darstellen. Beide Nachrichtenmodelle enthalten zwei Teile, im ersten Teil werden Meta-Informationen der Nachricht gespeichert, wie z. B. Absender, Empfänger, Zeitstempel und NachrichtenID, der zweite Teil enthält die Nutzdaten der Nachricht.

#### **A.1 XML-Schema des einheitlichen Nachrichtenformates der Produktionsplanung**

Die integrierten Anwendungen der Produktionsplanung tauschen Ressourceninformationen aus. Diese Positionsdaten der einzelnen Module sind wie bereits im DPE als Projekte gruppiert, um mehrere Layouts gleichzeitig verwalten und übertragen zu können. Zudem ist das Schema so modelliert, dass mit geringem Aufwand weitere Informationen zu entsprechenden Produkten oder Prozessen hinzugeführt werden können.

Listing A.1: Schema des einheitlichen Nachrichtenformates der Produktionsplanung

```

<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="http://tempuri.org/ESB" elementFormDefault="
  qualified" targetNamespace="http://tempuri.org/ESB" xmlns:xs="http:
  //www.w3.org/2001/XMLSchema">
<xs:element name="ProjectMsg" type="tns:ProjectMessage" />
<xs:complexType name="ProjectMessage">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="CommonInfos" type="
      tns:CommonInfos" />
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Projects"
      type="tns:Project" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="CommonInfos">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="MessageId" type="
      xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="MessageFlowId"
      type="xs:string" />
    <xs:element minOccurs="1" maxOccurs="1" name="MessageType" type="
      xs:int" />
    <xs:element minOccurs="1" maxOccurs="1" name="
      MessageIdregistered" type="xs:boolean" />
    <xs:element minOccurs="1" maxOccurs="1" name="
      MessageFlowIdregistered" type="xs:boolean" />
    <xs:element minOccurs="0" maxOccurs="1" name="RoutingInfos" type
      ="tns:RoutingType" />
    <xs:element minOccurs="0" maxOccurs="1" name="Command" type="
      xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="RoutingType">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="OriginSystem" type
      ="xs:string" />
    <xs:element minOccurs="0" maxOccurs="unbounded" name="
      DestinationSystem" type="tns:DestinationSystem" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="DestinationSystem">
  <xs:sequence>

```

```
<xs:element minOccurs="0" maxOccurs="1" name="SystemID" type="
  xs:string" />
<xs:element minOccurs="0" maxOccurs="1" name="SystemURI" type="
  xs:string" />
<xs:element minOccurs="0" maxOccurs="1" name="Queue" type="
  xs:string" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="Project">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="ProjectID" type="
      xs:int" />
    <xs:element minOccurs="0" maxOccurs="1" name="Projectname" type="
      xs:string" />
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Resources"
      type="tns:Resources" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Resources">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="
      ModuleTypes" type="tns:ModuleType" />
    <xs:element minOccurs="1" maxOccurs="1" name="Test" type="xs:int
      " />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ModuleType">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="ModuleID" type="
      xs:int" />
    <xs:element minOccurs="0" maxOccurs="1" name="Modulename" type="
      xs:string" />
    <xs:element minOccurs="1" maxOccurs="1" name="Type" type="xs:int
      " />
    <xs:element minOccurs="0" maxOccurs="1" name="FileName" type="
      xs:string" />
    <xs:element minOccurs="0" maxOccurs="1" name="BitmapName" type="
      xs:string" />
    <xs:element minOccurs="0" maxOccurs="unbounded" name="Module"
      type="tns:Module" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Module">
  <xs:sequence>
```

```
<xs:element minOccurs="1" maxOccurs="1" name="ObjectID" type="
  xs:int" />
<xs:element minOccurs="0" maxOccurs="1" name="Objectname" type="
  xs:string" />
<xs:element minOccurs="0" maxOccurs="1" name="Position" type="
  tns:Position" />
</xs:sequence>
</xs:complexType>
<xs:complexType name="Position">
  <xs:sequence>
    <xs:element minOccurs="1" maxOccurs="1" name="Position_X" type="
      xs:double" />
    <xs:element minOccurs="1" maxOccurs="1" name="Position_Y" type="
      xs:double" />
    <xs:element minOccurs="1" maxOccurs="1" name="Position_Z" type="
      xs:double" />
    <xs:element minOccurs="1" maxOccurs="1" name="Rotation_X" type="
      xs:double" />
    <xs:element minOccurs="1" maxOccurs="1" name="Rotation_Y" type="
      xs:double" />
    <xs:element minOccurs="1" maxOccurs="1" name="Rotation_Z" type="
      xs:double" />
  </xs:sequence>
</xs:complexType>
</xs:schema>
```

---

## A.2 XML-Schema des global einheitlichen Nachrichtenformates

Der Datenaustausch zwischen verschiedenen Phasen erfolgt über ein einheitliches Nachrichtenformat. Dieses globale Nachrichtenformat beinhaltet nur Daten, die zwischen den Phasen ausgetauscht werden und nicht alle Daten, die im gesamten Produktlebenszyklus vorhanden sind. Im Folgenden wird das XML-Schema des global einheitlichen Nachrichtenformates präsentiert, das Daten aufnehmen kann, die für die prototypische Umsetzung der ACITA benötigt werden, um Daten über den PLM-Bus zu senden.



Listing A.2: Schema des global einheitlichen Nachrichtenformates der Produktlebenszyklus

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema xmlns:tns="http://tempuri.org/PLM"
            elementFormDefault="qualified"
            targetNamespace="http://tempuri.org/PLM"
            xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="Message" type="tns:Message" />
  <xs:complexType name="Message">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1"
                  name="Header" type="tns:MessageHeader" />
      <xs:element minOccurs="0" maxOccurs="1"
                  name="Body" type="tns:ArrayOfAPayloadData" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="MessageHeader">
    <xs:sequence>
      <xs:element minOccurs="1" maxOccurs="1"
                  name="MessageType" type="xs:int" />
      <xs:element minOccurs="0" maxOccurs="1"
                  name="MessageID" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1"
                  name="MessageFlowID" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1"
                  name="Routing" type="tns:RoutingData" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RoutingData">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1"
                  name="Origin" type="tns:RoutingSet" />
      <xs:element minOccurs="0" maxOccurs="1"
                  name="Destinations" type="tns:ArrayOfRoutingSet" />
    </xs:sequence>
  </xs:complexType>
  <xs:complexType name="RoutingSet">
    <xs:sequence>
      <xs:element minOccurs="0" maxOccurs="1"
                  name="Endpoint" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1"
                  name="Operation" type="xs:string" />
      <xs:element minOccurs="0" maxOccurs="1"
                  name="SystemID" type="xs:string" />
    </xs:sequence>
  </xs:complexType>
</xs:schema>
```

```

    <xs:element minOccurs="0" maxOccurs="1"
      name="AdditionalFields" type="
        tns:ArrayOfGenericField" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfGenericField">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded"
      name="GenericField" nillable="true" type="
        tns:GenericField" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="GenericField">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="1" name="Value" />
    <xs:element minOccurs="0" maxOccurs="1" name="Name" type="
      xs:string" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfRoutingSet">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded"
      name="RoutingSet" nillable="true" type="
        tns:RoutingSet" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfAPayloadData">
  <xs:sequence>
    <xs:element minOccurs="0" maxOccurs="unbounded" name="
      APayloadData"
      nillable="true" type="tns:APayloadData" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="APayloadData" abstract="true" />
<xs:complexType name="ResourcesData">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:APayloadData">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="ModuleTypes"
          type="tns:ArrayOfModuleType" />
        <xs:element minOccurs="0" maxOccurs="1" name="Modules"
          type="tns:ArrayOfModule" />
        <xs:element minOccurs="1" maxOccurs="1" name="ProjectID"
          type="xs:int" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>

```

```
        <xs:element minOccurs="0" maxOccurs="1" name="ProjectName"
            type="xs:string" />
    </xs:sequence>
</xs:extension>
</xs:complexContent>
</xs:complexType>
<xs:complexType name="ArrayOfModuleType">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="ModuleType"
            "
                nillable="true" type="tns:ModuleType" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ModuleType">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="ID" type="xs:int"
            />
        <xs:element minOccurs="1" maxOccurs="1" name="ObjectType" type="
            xs:int" />
        <xs:element minOccurs="0" maxOccurs="1" name="Name" type="
            xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="FileName" type="
            xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="BitmapFileName"
            type="xs:string" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="ArrayOfModule">
    <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="unbounded" name="Module"
            nillable="true" type="tns:Module" />
    </xs:sequence>
</xs:complexType>
<xs:complexType name="Module">
    <xs:sequence>
        <xs:element minOccurs="1" maxOccurs="1" name="ID" type="xs:int"
            />
        <xs:element minOccurs="1" maxOccurs="1" name="Type" type="xs:int"
            " />
        <xs:element minOccurs="0" maxOccurs="1" name="Name" type="
            xs:string" />
        <xs:element minOccurs="0" maxOccurs="1" name="Position" type="
            tns:Point" />
    </xs:sequence>
</xs:complexType>
```

```
    <xs:element minOccurs="0" maxOccurs="1" name="Rotation" type="
      tns:Point" />
  </xs:sequence>
</xs:complexType>
<xs:complexType name="Point">
  <xs:attribute name="X" type="xs:double" use="required" />
  <xs:attribute name="Y" type="xs:double" use="required" />
  <xs:attribute name="Z" type="xs:double" use="required" />
</xs:complexType>
<xs:complexType name="CommandData">
  <xs:complexContent mixed="false">
    <xs:extension base="tns:APayloadData">
      <xs:sequence>
        <xs:element minOccurs="0" maxOccurs="1" name="Command" type="
          xs:string" />
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
</xs:schema>
```

---

---

# Abbildungsverzeichnis

---

1.1	Gliederung der Arbeit . . . . .	43
2.1	Historie der Rechnerunterstützung für Geschäfts-, Planungs- und Produktionsprozesse in Unternehmen . . . . .	47
2.2	Y-CIM-Modell für den Industriebetrieb . . . . .	49
2.3	Zusammenhang zwischen digitaler, virtueller und realer Fabrik	53
2.4	Phasen des Produktlebenszyklus nach Stark . . . . .	55
2.5	Unterschied zwischen Adaptivität und Wandelbarkeit . . . . .	59
2.6	Das magische Dreieck einer SOA . . . . .	63
3.1	Erweiterter Produktlebenszyklus mit sechs Phasen . . . . .	79
3.2	Architektur der CHAMPAGNE Integrationsplattform . . . . .	86
3.3	Integrationshierarchie des Manufacturing Service Bus . . . . .	91
4.1	Wichtige Unternehmenselemente und ihre Abhängigkeiten .	106
4.2	Integrationsarchitektur mit einem zentralen ESB . . . . .	109
4.3	Integrationsarchitektur mit mehreren ESBs . . . . .	110
4.4	Servicebasierte Integrationsarchitektur für das Product Life- cycle Management . . . . .	120
4.5	Verteilte Serviceverzeichnismodelle für die ACITA . . . . .	126
4.6	Aufruf eines Services in einer anderen Phase des Produktle- benszyklus . . . . .	136
4.7	Phasenmodell des Produktlebenszyklus . . . . .	139
4.8	Phasenmodell des Fabriklebenszyklus . . . . .	141

---

4.9	Phasenmodell des Designs der Lieferkette . . . . .	142
4.10	Ganzheitliches Managementmodell für Produktionsunter- nehmen . . . . .	143
4.11	Unterstützung des ganzheitlichen Managementmodell durch die ACITA . . . . .	145
5.1	Teile und Varianten des Schreibtischsets . . . . .	151
5.2	Die physische Modellfabrik iTRAME . . . . .	153
5.3	Architektur zur Integration von Anwendungen in der Digi- talen Fabrik . . . . .	160
5.4	Komponenten zur Umsetzung der PLM-Architektur . . . . .	163
5.5	Layoutänderungen werden im Wartungsportal farblich vi- sualisiert . . . . .	165
6.1	Umsetzung der fünfstufigen Integrationshierarchie in die Produktionsplanung mit Warteschlangen . . . . .	175
6.2	Prozess zur Layoutübertragung vom Leitrechner zum Pla- nungstisch, modelliert in BPMN . . . . .	178
6.3	Wandlungsfähige IT-Architektur durch einfaches Hinzufü- gen und Entfernen von Integrationsumgebungen . . . . .	181
6.4	Integration von externen Partnern in die ACITA . . . . .	183
6.5	Screenshot der Planungstisch-App . . . . .	196

---

# Tabellenverzeichnis

---

1.1	Ziele der Arbeit und Forschungsobjekte . . . . .	41
3.1	Aufgaben der einzelnen Phasen im Produktlebenszyklus . . .	81
6.1	Interoperabilitätskriterium . . . . .	185
6.2	Adaptivitätskriterium . . . . .	186
6.3	Servicevermittlungskriterium . . . . .	187
6.4	Skalierbarkeitskriterium . . . . .	188
6.5	Prozessunterstützungskriterium . . . . .	189
6.6	Integrationskriterium . . . . .	190
6.7	Bewertung und Vergleich der ACITA mit anderen Lösungs- ansätzen . . . . .	191
6.8	Erfüllung von Zielen und Forschungsobjekten der Arbeit . . .	201





---

## Literaturverzeichnis

---

- [AKSM11] ABEL, Michael ; KLEMM, Peter ; SILCHER, Stefan ; MINGUEZ, Jorge: Start-Up of Reconfigurable Production Machines with a Service-Oriented Architecture. In: *Proceedings of the 21st International Conference on Production Research*, Fraunhofer IAO, 2011, S. 1–5
- [ALW06] ABELE, Eberhard ; LIEBECK, Tobias ; WÖRN, Arno: Measuring Flexibility in Investment Decisions for Manufacturing Systems. In: *CIRP Annals - Manufacturing Technology* 55 (2006), Nr. 1, S. 433–436. – ISSN 0007–8506
- [An07] AN, C.: Hype or Reality: Service Oriented Architecture in Product Lifecycle Management - How IBM Can Help You Achieve Innovation That Matters. In: KRAUSE, Frank-Lothar (Hrsg.): *The Future of Product Development*, Springer Berlin Heidelberg, 2007. – ISBN 978–3–540–69820–3, S. 685–690
- [Aus85] AUSSCHUSS FÜR WIRTSCHAFTLICHE FERTIGUNG (AWF): *Integrierter EDV-Einsatz in der Produktion, CIM - Computer Integrated Manufacturing. Begriffe, Definitionen, Funktionszuordnungen*. Eschborn: AWF, 1985
- [BCM08] BERGSJÖ, Dag ; CATIC, Amer ; MALMQVIST, Johan: Implementing a service oriented PLM architecture using PLM Services 2.0. In: MARJANOVIC, D. (Hrsg.) ; STORGA, M. (Hrsg.) ; PAVKOVIC, N. (Hrsg.) ; BOJCETIC, N. (Hrsg.): *Proceedings of the International Design*

- Conference*. Dubrovnik, Croatia: the Design Society, 2008, S. 271–280
- [BDS12a] BAUERNHANSL, Thomas ; DINKELMANN, Max ; SIEGERT, Jörg: Lernfabrik advanced Industrial Engineering \* - Teil 1. In: *wt Werkstattstechnik online* Jahrgang 102 (2012), Nr. 3, S. 80–83
- [BDS12b] BAUERNHANSL, Thomas ; DINKELMANN, Max ; SIEGERT, Jörg: Lernfabrik advanced Industrial Engineering \* - Teil 2. In: *wt Werkstattstechnik online* Jahrgang 102 (2012), Nr. 3, S. 84–87
- [Bra84] BRACHT, Uwe: *Rechnergestützte Fabrikanalyse und -planung auf der Basis einer flächenbezogenen Werksstruktur-Datenbank*, Universität Hannover, Diss., 1984
- [BRP08] BOLSTORFF, Peter A. ; ROSENBAUM, Robert G. ; POLUHA, Rolf G.: *Spitzenleistungen im Supply Chain Management. Ein Praxishandbuch zur Optimierung mit SCOR*. 1. Auflage. Springer, Berlin, 2008. – ISBN 978–3–540–71183–4 (Print), 978–3–540–71185–8 (Online)
- [BW95] BULLINGER, Hans-Jörg (Hrsg.) ; WARSCHAT, Joachim (Hrsg.): *Concurrent Simultaneous Engineering Systems: The Way to Successful Product Development*. 1. Auflage. Springer, 1995. – ISBN 9783540760030
- [BWG11] BRACHT, Uwe ; WENZEL, Sigrid ; GECKLER, Dieter: *Digitale Fabrik: Methoden und Praxisbeispiele*. 1. Auflage. Springer, Berlin, 2011. – ISBN 978–3–540–89038–6 (Print), 978–3–540–88973–1 (Online)
- [CBB<sup>+</sup>08] CREDLE, Rufus ; BADER, Michael ; BIRKLER, Khirallah ; HARRIS, Martin ; HOLT, Mark ; HYAKUNA, Yoko ; LÄMMER, Dr. L. ; MCCARTY, Estela ; MOMMEJA, Lionel ; NOVAES, Dr. M. ; RAINES, Buddy ; SRINIVASAN, Dr. V.: *SOA Approach to Enterprise Integration*

- for Product Lifecycle Management.* IBM International Technical Support Organization, 2008 (IBM Redbook SG24-7593-00). <http://www.redbooks.ibm.com/redbooks/pdfs/sg247593.pdf>
- [CCFZ12] CONTI, Mauro ; CRISPO, Bruno ; FERNANDES, Earlence ; ZHAUNAROVICH, Yury: CRêPE: A System for Enforcing Fine-Grained Context-Related Policies on Android. In: *IEEE Transactions on Information Forensics and Security* 7 (2012), Nr. 5, S. 1426–1438. – ISSN 1556–6013
- [CEF09] CONSTANTINESCU, Carmen ; EICHELBERGER, Hanno ; FRANK, Thorsten: Flow-based Approach for Holistic Factory Engineering and Design. In: *Proceedings of the 2nd International Researchers Symposium Innovative Production Machines and Systems*, 2009, S. 1–6
- [CEW09] CONSTANTINESCU, Carmen ; EICHELBERGER, Hanno ; WESTKÄMPER, Engelbert: Durchgängige und integrierte Fabrik- und Prozessplanung: "Grid Engineering for Manufacturing". In: *wt Werkstattstechnik online* Jahrgang 99 (2009), Nr. 3, S. 92–98
- [Cha04] CHAPPELL, David A.: *Enterprise Service Bus: Theory in Practice*. 1. Auflage. O'Reilly Media, 2004. – ISBN 978–0–596–00675–4
- [CHB<sup>+</sup>05] CONSTANTINESCU, Carmen ; HEINKEL, Uwe ; BLOND, Jan L. ; SCHREIBER, Stephan ; MITSCHANG, Bernhard ; WESTKÄMPER, Engelbert: Flexible Integration of Layout Planning and Adaptive Assembly Systems in Digital Enterprises. In: *Proceedings of the 38th CIRP International Seminar on Manufacturing Systems (CIRP ISMS)*, CIRP, 2005, S. 10–18
- [CHKT05] CONRAD, Stefan ; HASSELBRING, Wilhelm ; KOSCHEL, Arne ; TRITSCH, Roland: *Enterprise Application Integration: Grundlagen - Konzepte - Entwurfsmuster - Praxisbeispiele*. 1. Auflage. Spektrum Akademischer Verlag, 2005. – ISBN 978–3–8274–1572–1

- [CHN02] CISEK, Robert ; HABICHT, Christian ; NEISE, Patrick: Gestaltung wandlungsfähiger Produktionssysteme. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb ZWF* 09-02 (2002), 441–445. <http://www.zwf-online.de/ZW100566>
- [CHW06] CONSTANTINESCU, Carmen ; HUMMEL, Vera ; WESTKÄMPER, Engelbert: Fabrik Life Cycle Management: Kollaborative standardisierte Umgebung für die Fabrikplanung (KOSIFA). In: *wt Werkstattstechnik online* Jahrgang 96 (2006), Nr. 4, S. 178–182
- [CIM05] CIMDATA: PLM and ERP Integration: Business Efficiency and Value / CIMdata, Inc. 2005. – Forschungsbericht
- [CIM06] CIMDATA: Service-Oriented Architecture for PLM - An Overview of UGS' SOA Approach / CIMdata, Inc. Version: 2006. [http://www.plm.automation.siemens.com/en\\_us/Images/soa\\_for\\_plm\\_tcm1023-46846.pdf](http://www.plm.automation.siemens.com/en_us/Images/soa_for_plm_tcm1023-46846.pdf). 2006. – Forschungsbericht
- [CIM10] CIMDATA: Die einheitliche Umgebung von Teamcenter: "Die PLM-Plattform der nächsten Generation von Siemens PLM Software" / CIMdata, Inc. Version: 2010. [https://www.plm.automation.siemens.com/de\\_de/Images/21848\\_tcm73-100227.pdf](https://www.plm.automation.siemens.com/de_de/Images/21848_tcm73-100227.pdf). 2010. – Forschungsbericht
- [CM12] CHOPRA, Sunil ; MEINDL, Peter: *Supply Chain Management: Strategy, Planning, and Operation*. 5. Auflage. Prentice Hall, 2012. – ISBN 978–0132743952
- [DHL05] DU, Zongxia ; HUAI, Jinpeng ; LIU, Yunhao: Ad-UDDI: An Active and Distributed Service Registry. In: *6th VLDB International Workshop on Technologies for E-Services (TES), Volume 3811 of Lecture Notes in Computer Science*, Springer, 2005, S. 58–71
- [DRW11] DINKELMANN, Max ; RIFFELMACHER, Philipp ; WESTKÄMPER, Engelbert: Training concept and structure of the Learning Facto-

- ry advanced Industrial Engineering. In: ELMARAGHY, Hoda A. (Hrsg.): *Enabling Manufacturing Competitiveness and Economic Sustainability* Band 4, Springer Berlin Heidelberg, 2011 (Proceedings of the 4th International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV2011)). – ISBN 978-3-642-23860-4, S. 624–629
- [DSRR08] DAVIES, Jeff ; SCHOROW, David ; RAY, Samrat ; RIEBER, David: *The Definitive Guide to SOA: Oracle Service Bus (Expert's Voice)*. 2. Auflage. Berkely, CA, USA: Apress, 2008. – ISBN 978-1-4302-1057-3 (Print), 978-1-4302-1058-0 (Online)
- [EHM03] ENDERLEIN, H. ; HILDEBRAND, T. ; MÜLLER, E.: PLUG+PRODUCE: Die Fabrik mit Zukunft aus dem Baukasten. In: *wt Werkstattstechnik online* Jahrgang 93 (2003), Nr. 4, S. 282–286
- [Erl05] ERL, Thomas: *Service-Oriented Architecture: Concepts, Technology, and Design*. 6. Auflage. Prentice Hall International, 2005. – ISBN 9780131858589
- [ES09] EIGNER, Martin ; STELZER, Ralph: *Product Lifecycle Management: Ein Leitfaden für Product Development und Life Cycle Management*. 2. Auflage. Springer, Berlin, 2009. – ISBN 978-3-540-44373-5 (Print), 978-3-540-68401-5 (Online)
- [Est03] ESTREM, William A.: An evaluation framework for deploying Web Services in the next generation manufacturing enterprise. In: *Robotics and Computer-Integrated Manufacturing* 19 (2003), Nr. 6, S. 509–519. – ISSN 0736-5845
- [FGYO95] FOHN, Steffen M. ; GREEF, Arthur ; YOUNG, Robert E. ; O'GRADY, Peter: Concurrent engineering. In: ADELSBERGER, Heimo H. (Hrsg.) ; LAŽANSKÝ, Jiří (Hrsg.) ; MAŘÍK, Vladimír (Hrsg.): *Information Management in Computer Integrated Manufacturing* Band 973.

- Springer Berlin Heidelberg, 1995. – ISBN 978-3-540-60286-6, S. 493–505
- [GNK92] GRANT, Delvin A. ; NGWENYAMA, Ojelanki K. ; KLEIN, Heinz K.: Modeling for CIM information systems architecture definition: an information engineering case study. In: *Computers in Industry* 18 (1992), Nr. 2, S. 199–212. – ISSN 0166-3615
- [GNM12] GRÖGER, Christoph ; NIEDERMANN, Florian ; MITSCHANG, Bernhard: Data Mining-driven Manufacturing Process Optimization. In: Ao, S. I. (Hrsg.) ; GELMAN, L. (Hrsg.) ; HUKINS, D. W. L. (Hrsg.) ; HUNTER, A. (Hrsg.) ; KORSUNSKY, A. M. (Hrsg.): *Proceedings of the World Congress on Engineering 2012 Vol III (WCE 2012)*, Newswood, 2012. – ISBN 978-988-19252-2-0, S. 1475–1481
- [GNSM12] GRÖGER, Christoph ; NIEDERMANN, Florian ; SCHWARZ, Holger ; MITSCHANG, Bernhard: Supporting manufacturing design by analytics, continuous collaborative process improvement enabled by the advanced manufacturing analytics platform. In: *16th International IEEE Conference on Computer Supported Cooperative Work in Design (CSCWD 2012)*, 2012, S. 793–799
- [Gru08] GRUNDIG, Claus-Gerold: *Fabrikplanung: Planungssystematik - Methoden - Anwendungen*. 3. Auflage. Carl Hanser Verlag GmbH & CO. KG, 2008. – ISBN 978-3-446-41411-2
- [GSNM12] GRÖGER, Christoph ; SCHLAUDRAFF, Johannes ; NIEDERMANN, Florian ; MITSCHANG, Bernhard: Warehousing Manufacturing Data. A Holistic Process Warehouse for Advanced Manufacturing Analytics. In: CUZZOCREA, Alfredo (Hrsg.) ; DAYAL, Umeshwar (Hrsg.): *Proceedings of the 14th International Conference on Data Warehousing and Knowledge Discovery (DaWaK 2012)* Band 7448. Berlin, Heidelberg: Springer, 2012 (Lecture Notes in Computer Science), S. 142–155

- [GSWM13] GRÖGER, Christoph ; SILCHER, Stefan ; WESTKÄMPER, Engelbert ; MITSCHANG, Bernhard: Leveraging Apps in Manufacturing. A Framework for App Technology in the Enterprise. In: *Proceedings of the 46th CIRP Conference on Manufacturing Systems (CMS 2013)* Band 7, Elsevier, 2013. – ISSN 2212–8271, S. 664–669
- [GT05] GÜNTHER, Hans-Otto ; TEMPELMEIER, Horst: *Produktion und Logistik*. 6. Auflage. Springer-Verlag GmbH, 2005. – ISBN 978–3–540–23246–9 (Print), 978–3–540–27188–8 (Online)
- [Gun97] GUNASEKARAN, Angappa: Implementation of Computer-Integrated Manufacturing: A Survey of Integration and Adaptability Issues. In: *International Journal of Computer-Integrated Manufacturing* 10 (1997), Nr. 1-4, S. 266–280
- [Har73] HARRINGTON, Joseph: *Computer integrated manufacturing*. Industrial Press, 1973. – ISBN 9780831110963
- [HCM05] HEINKEL, Uwe ; CONSTANTINESCU, Carmen ; MITSCHANG, Bernhard: Integrating Data Changes with Data from Data Service Providers. In: *Proceedings of the 18th International Conference on Computer Applications in Industry and Engineering (CAINE 2005)*, ICSA, 2005, S. 146–151
- [Hei11] HEINKEL, Uwe: *Änderungspropagation für autonome und heterogene Informationssysteme*, Universität Stuttgart, Diss., 2011
- [HMF92] HESSE, Wolfgang ; MERBETH, Günter ; FRÖLICH, Rainer: *Software-Entwicklung. Vorgehensmodelle, Projektführung, Produktverwaltung*. Oldenbourg, 1992. – ISBN 978–3486206937
- [HW03] HOHPE, Gregor ; WOOLF, Bobby: *Enterprise Integration Patterns: Designing, Building, and Deploying Messaging Solutions*. Addison-Wesley Longman, Amsterdam, 2003. – ISBN 9780321200686

- [HW07] HUMMEL, Vera ; WESTKÄMPER, Engelbert: Innovative Learning Environment for Factory Planning and Improvement. In: *Proceeding of the International Conference Competitive Manufacturing - The Challenge of Digital Manufacturing (COMA '07)*, 2007. – ISBN 0-7972-1164-0, S. 473-480
- [Int94] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO): *ISO 10303-1:1994-12: Industrielle Automatisierungssysteme und Integration - Produktdatendarstellung und -austausch - Teil 1: Überblick und grundlegende Prinzipien (STEP)*. 1994
- [Int07] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO): *ISO 10303-28:2007-10: Industrial automation systems and integration - Product data representation and exchange - Part 28: Implementation methods: XML representations of EXPRESS schemas and data, using XML schemas (STEP-XML)*. 2007
- [Int10] INTERNATIONAL ORGANIZATION FOR STANDARDIZATION (ISO): *ISO 10303-214:2010: Industrial automation systems and integration – Product data representation and exchange – Part 214: Application protocol: Core data for automotive mechanical design processes*. 2010
- [JWW09] JOVANE, Francesco ; WESTKÄMPER, Engelbert ; WILLIAMS, David: *The ManuFuture Road: Towards Competitive and Sustainable High-adding-value Manufacturing*. 1. Auflage. Springer, Berlin, 2009. – ISBN 978-3-540-77011-4 (Print), 978-3-540-77012-1 (Online)
- [KAB<sup>+</sup>04] KEEN, Martin ; ACHARYA, Amit ; BISHOP, Susan ; HOPKINS, Alan ; MILINSKI, Sven ; NOTT, Chris ; ROBINSON, Rick ; ADAMS, Jonathan ; VERSCHUEREN, Paul: *Patterns: Implementing an SOA Using an Enterprise Service Bus*. IBM, 2004. – ISBN 9780738490007
- [KAH<sup>+</sup>05] KEEN, Martin ; ADINOLFI, Oscar ; HEMMINGS, Sarah ; HUMPHREYS, Andrew ; KANTHI, Hanumanth ; NOTTINGHAM, Alasdair: *Patterns:*



*SOA With an Enterprise Service Bus in Websphere Application Server V6*. IBM, 2005. – ISBN 9780738490588

- [KEMS07] KASSIM, A. ; ESFANDIARI, B. ; MAJUMDAR, S. ; SERGHI, L.: A Flexible Hybrid Architecture for Management of Distributed Web Service Registries. In: *Proceedings of the 5th Annual Conference on Communication Networks and Services Research (CNSR 2007)*, 2007, S. 311–313
- [Küh06] KÜHN, Wolfgang ; KÜHN, Wolfgang (Hrsg.): *Digitale Fabrik: Fabriksimulation für Produktionsplaner*. 1. Auflage. München; Wien: Carl Hanser Verlag GmbH & CO. KG, 2006. – ISBN 978-3-446-40619-3
- [KLBW05] KAPP, Ralf ; LE BLOND, Jan ; WESTKÄMPER, Engelbert: Fabrikstruktur und Logistik integriert planen: Erweiterung eines kommerziellen Werkzeugs der Digitalen Fabrik für den Mittelstand. In: *wt Werkstattstechnik online* Jahrgang 95 (2005), Nr. 4, S. 191–196
- [KLC06] KANG, Kyuchang ; LEE, Jeunwoo ; CHOI, Hoon: Extended Service Registry for Distributed Computing Support in OSGi Architecture. In: *The 8th International Conference on Advanced Communication Technology (ICACT 2006)* Band 3, 2006, S. 1631–1634
- [KLWW03] KAPP, Ralf ; LÖFFLER, Benno ; WIENDAHL, Hans-Hermann ; WESTKÄMPER, Engelbert: Der Logistik-Prüfstand: Skalierbare Logistiksimulation von der Lieferkette bis zum Arbeitsgang. In: *wt Werkstattstechnik online* Jahrgang 93 (2003), Nr. 1/2, S. 31–38
- [KRHW07] KLUGE, Stefan ; RIFFELMACHER, Philipp ; HUMMEL, Vera ; WESTKÄMPER, Engelbert: Montagesystemplanung - ein Handlungsfeld der Lernfabrik für aIE. In: *wt Werkstattstechnik online* Jahrgang 97 (2007), Nr. 3, S. 150–156

- [KRSS08] KINKEL, Steffen ; RALLY, Peter ; SCHOLTZ, Oliver ; SCHWEIZER, Wolfgang: *Organisatorische Wandlungsfähigkeit produzierender Unternehmen: Unternehmenserfahrungen, Forschungs- und Transferbedarfe*. Fraunhofer IRB Verlag, 2008. – ISBN 978-3-8167-7530-0
- [KS11] KATZENBACH, Alfred ; STEIERT, Hans-Peter: Engineering-IT in der Automobilindustrie – Wege in die Zukunft. In: *Informatik-Spektrum* 34 (2011), S. 7–19. – ISSN 0170-6012
- [Kus93] KUSIAK, Andrew: *Concurrent Engineering: Automation, Tools, and Techniques*. John Wiley & Sons, 1993 (A Wiley-Interscience Publication). – ISBN 9780471554929
- [KWW03] KIRCHNER, Sören ; WINKLER, Ralph ; WESTKÄMPER, Engelbert: Unternehmensstudie zur Wandlungsfähigkeit von Unternehmen. In: *wt Werkstattstechnik online* Jahrgang 93 (2003), Nr. 4, S. 254–260
- [LB07] LÄMMER, L. ; BUGOW, R.: PLM Services in Practice. In: KRAUSE, Frank-Lothar (Hrsg.): *The Future of Product Development*. Springer Berlin Heidelberg, 2007. – ISBN 978-3-540-69820-3, S. 503–512
- [LCW08a] LUCKE, Dominik ; CONSTANTINESCU, Carmen ; WESTKÄMPER, Engelbert: Kontextbezogene Anwendungen in der Produktion: Smart Factory - Gestern, heute und in der Zukunft. In: *wt Werkstattstechnik online* Jahrgang 98 (2008), Nr. 3, S. 138–142
- [LCW08b] LUCKE, Dominik ; CONSTANTINESCU, Carmen ; WESTKÄMPER, Engelbert: Smart factory - a step towards the next generation of manufacturing. In: *Manufacturing Systems and Technologies for the New Frontier: The 41st CIRP Conference on Manufacturing Systems* Volume 41 (2008), S. 115–118

- [Lin00] LINTHICUM, David S.: *Enterprise application integration*. 3. Auflage. Boston: Addison-Wesley, 2000. – ISBN 0–201–61583–5
- [LLS<sup>+</sup>07] LEE, TaeJae ; LIM, JongGyun ; SHIN, JunHo ; MYUNG, SeHyun ; CHOI, MiYoung ; BAEK, SeungSik ; KIM, JuIk ; OH, JaeWook ; LEE, DongSuk ; HAN, YongDuk: An implementation methodology of SOA based PLM system. In: *Proceedings of the International Conference on Product Lifecycle Management - Assessing the industrial relevance (PLM'07)*, 2007, S. 303–310
- [LW07] LUCKE, Dominik ; WIELAND, Matthias: Umfassendes Kontextdatenmodell der Smart Factory als Basis für kontextbezogene Workflow-Anwendungen. In: *Dr. Hut-Verlag 4* (2007), S. 47–51. ISBN 978–3–89963–591–1
- [MAW06] MUKKAMALA, Ravi ; ATLURI, Vijayalakshmi ; WARNER, Janice: A Distributed Service Registry for Resource Sharing Among Ad-Hoc Dynamic Coalitions. In: DOWLAND, Paul (Hrsg.) ; FURNELL, Steve (Hrsg.) ; THURASINGHAM, Bhavani (Hrsg.) ; WANG, X. (Hrsg.): *Security Management, Integrity, and Internal Control in Information Systems* Band 193, Springer Boston, 2006 (IFIP International Federation for Information Processing). – ISBN 978–0–387–29826–9, S. 319–333
- [Mel10] MELZER, Ingo (Hrsg.): *Service-orientierte Architekturen mit Web Services: Konzepte - Standards - Praxis*. 4. Auflage. Spektrum Akademischer Verlag, 2010. – ISBN 978–3–8274–2549–2 (Print), 978–3–8274–2550–8 (Online)
- [Min12] MINGUEZ, Jorge: *A service-oriented integration platform for flexible information provisioning in the real-time factory*, Universität Stuttgart, Diss., 2012
- [MRMW11] MINGUEZ, Jorge ; RIFFELMACHER, Philipp ; MITSCHANG, Bernhard ; WESTKÄMPER, Engelbert: Servicebasierte Integration von

- Produktionsanwendungen. In: *wt Werkstattstechnik online* Jahrgang 101 (2011), Nr. 3, S. 128–133
- [MRR<sup>+</sup>10] MINGUEZ, Jorge ; RUTHARDT, Frank ; RIFFELMACHER, Philipp ; SCHEIBLER, Thorsten ; MITSCHANG, Bernhard: Service-based Integration in Event-driven Manufacturing Environments. In: *WISE 2010 Workshops* Band 6724, Springer, Dezember 2010 (Lecture Notes in Computer Science), S. 295–308
- [MSRM11] MINGUEZ, Jorge ; SILCHER, Stefan ; RIFFELMACHER, Philipp ; MITSCHANG, Bernhard: A Service Bus Architecture for Application Integration in the Planning and Production Phases of a Product Lifecycle. In: *International Journal of Systems and Service-Oriented Engineering* Volume 2 (2011), Nr. 2, S. 21–36. – ISSN 1947–3052
- [NFPFW09] NYHUIS, Peter ; FRONIA, Philip ; PACHOW-FRAUENHOFER, Julia ; WULF, Serjosh: Wandlungsfähige Produktionssysteme. In: *wt Werkstattstechnik online* Jahrgang 99 (2009), Nr. 4, S. 205–210
- [NHR<sup>+</sup>08] NYHUIS, Peter ; HEINEN, Tobias ; REINHART, Gunther ; RIMPAU, Christoph ; ABELE, Eberhard ; WÖRN, Arno: Wandlungsfähige Produktionssysteme: Theoretischer Hintergrund zur Wandlungsfähigkeit von Produktionssystemen. In: *wt Werkstattstechnik online* Jahrgang 98 (2008), Nr. 1/2, S. 85–91
- [NL99] NAGALINGAM, Sev V. ; LIN, Grier C.: Latest developments in CIM. In: *Robotics and Computer-Integrated Manufacturing* 15 (1999), Nr. 6, S. 423–430. – ISSN 0736–5845
- [NRA08] NYHUIS, Peter (Hrsg.) ; REINHART, Gunther (Hrsg.) ; ABELE, Eberhard (Hrsg.): *Wandlungsfähige Produktionssysteme: Heute die Industrie von morgen gestalten*. 1. Auflage. PZH Produktionstechnisches Zentrum, 2008. – ISBN 978–3–939026–96–9

- [OAS05] OASIS: *OASIS UDDI Specification TC v3*. [https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=uddi-spec](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=uddi-spec).  
Version: 2005
- [OAS06] OASIS: *OASIS Web Services Security (WSS) TC*. [https://www.oasis-open.org/committees/tc\\_home.php?wg\\_abbrev=wss](https://www.oasis-open.org/committees/tc_home.php?wg_abbrev=wss).  
Version: 2006
- [OAS07] OASIS: *Web Services Business Process Execution Language Version 2.0*. 2007
- [OAS09a] OASIS: *OASIS Web Services Atomic Transaction Version 1.2*. <http://docs.oasis-open.org/ws-tx/wsac/2006/06>. Version: 2009
- [OAS09b] OASIS: *OASIS Web Services Business Activity Version 1.2*. <http://docs.oasis-open.org/ws-tx/wsba/2006/06>. Version: 2009
- [OAS09c] OASIS: *Web Services Reliable Messaging Policy Assertion (WS-RM Policy) Version 1.2*. <http://docs.oasis-open.org/ws-rx/wsrmp/200702>. Version: 2009
- [OAS13] OASIS: *OASIS Topology and Orchestration Specification for Cloud Applications (TOSCA)*. <http://docs.oasis-open.org/tosca/TOSCA/v1.0/os/TOSCA-v1.0-os.pdf>. Version: 2013
- [OMG09a] OMG: *Business Process Model and Notation (BPMN) Version 1.2*. <http://www.omg.org/spec/BPMN/1.2/>. Version: 2009
- [OMG09b] OMG: *Product Lifecycle Management Services, v2.0*. August 2009
- [OMG11] OMG: *Business Process Model and Notation (BPMN) Version 2.0*. <http://www.omg.org/spec/BPMN/2.0/>. Version: 2011
- [Ope10] OPENESB COMMUNITY: *OpenESB, Version 2.2*. <http://www.open-esb.net/>. Version: 2010

- [Pol10] POLUHA, Rolf G.: *Quintessenz des Supply Chain Managements: Was Sie wirklich über Ihre Prozesse in Beschaffung, Fertigung, Lagerung und Logistik wissen müssen*. 1. Auflage. Springer, Berlin, 2010. – ISBN 978-3-642-01583-0 (Print), 978-3-642-01584-7 (Online)
- [RBES02] REINHART, Gunther ; BERLAK, Joachim ; EFFERT, Christian ; SELKE, Carsten: Wandlungsfähige Fabrikgestaltung. In: *Zeitschrift für wirtschaftlichen Fabrikbetrieb ZWF* 01-02 (2002), 18–23. <http://www.zwf-online.de/ZW100500>
- [RCHM02] RANTZAU, Ralf ; CONSTANTINESCU, Carmen ; HEINKEL, Uwe ; MEINECKE, Holger: CHAMPAGNE: data change propagation for heterogeneous information systems. In: *Proceedings of the 28th international conference on Very Large Data Bases, VLDB Endowment, 2002 (VLDB '02)*, S. 1099–1102
- [RHW06] RIFFELMACHER, Philipp ; HUMMEL, Vera ; WESTKÄMPER, Engelbert: Learning Factory for advanced Industrial Engineering. In: *First CIRP International Seminar on Assembly Systems*, Fraunhofer IRB Verlag, 2006. – ISBN 978-3-81677-213-2, S. 283–288
- [RKHW06] RIFFELMACHER, Philipp ; KLUGE, Stefan ; HUMMEL, Vera ; WESTKÄMPER, Engelbert: Integration der Digitalen Planung und der realen Produktion in einer Lernfabrik für aIE. In: *Vernetzt planen und produzieren (VPP 2006) Volume 12 (2006)*, S. 164–169. – ISSN 0947-2495
- [RKHW08] RIFFELMACHER, Philipp ; KLUGE, Stefan ; HUMMEL, Vera ; WESTKÄMPER, Engelbert: Integration digitaler und virtueller Planungs-umgebungen bei einer realen Fabrik. In: *wt Werkstattstechnik online* Jahrgang 98 (2008), Nr. 3, S. 121–126. – ISSN 1436-4980
- [RKK<sup>+</sup>07] RIFFELMACHER, Philipp ; KLUGE, Stefan ; KREUZHAGE, Rita ; HUMMEL, Vera ; WESTKÄMPER, Engelbert: Learning factory for the manufacturing industry: Digital learning shell and a physical model

- factory - iTRAME for production engineering and improvement. In: *Proceeding of the International Conference on Computer-Aided Production Engineering (CAPE 2007)*, 2007. – ISBN 978-1-905866-14-4, S. 120-131
- [SBM<sup>+</sup>05] SCHEER, August-Wilhelm ; BOCZANSKI, Manfred ; MUTH, Michael ; SCHMITZ, Willi-Gerd ; SEGELBACHER, Uwe: *Prozessorientiertes Product Lifecycle Management*. 1. Auflage. Springer, Berlin, 2005. – ISBN 978-3-540-28402-4 (Print), 978-3-540-28442-0 (Online)
- [Sch90] SCHEER, August-Wilhelm: *CIM Computer Integrated Manufacturing: Der Computergesteuerte Industriebetrieb*. 4. Auflage. Springer-Verlag, 1990. – ISBN 978-3-642-64860-1 (Print), 978-3-642-61510-8 (Online)
- [Sch06] SCHUH, Günther ; SCHUH, Günther (Hrsg.): *Produktionsplanung und -steuerung: Grundlagen, Gestaltung und Konzepte*. 3. Auflage. Springer, Berlin Heidelberg, 2006. – ISBN 978-3-540-40306-7 (Print), 978-3-540-33855-0 (Online)
- [SDMM12] SILCHER, Stefan ; DINKELMANN, Max ; MINGUEZ, Jorge ; MITSCHANG, Bernhard: A Service-based Integration for an improved Product Lifecycle Management. In: CUZZOCREA, Alfredo (Hrsg.) ; MACIASZEK, José Cordeiro L. (Hrsg.): *Proceedings of the 14th International Conference on Enterprise Information Systems*, INSTICC Press, Juni 2012. – ISBN 978-989-8565-10-5, S. 38-47
- [SDMM13] SILCHER, Stefan ; DINKELMANN, Max ; MINGUEZ, Jorge ; MITSCHANG, Bernhard: Advanced Product Lifecycle Management by Introducing Domain-Specific Service Buses. In: CORDEIRO, José (Hrsg.) ; MACIASZEK, Leszek A. (Hrsg.) ; FILIPE, Joaquim (Hrsg.): *Enterprise Information Systems* Band 141. Springer Berlin Heidelberg, 2013. – ISBN 978-3-642-40653-9 (Print), 978-3-642-40654-6 (Online), S. 92-107

- [SGNWP08] SCHUH, Günther ; GOTTSCHALK, Sebastian ; NÖCKER, Jan ; WESCH-POTENTE, Cathrin: Flexible Vernetzung statt starrer Integration - die Zukunft der digitalen Fabrik. In: *wt Werkstattstechnik online* Jahrgang 98 (2008), Nr. 3, S. 127–131
- [SI08] SAAKSUORI, Antti ; IMMONEN, Anselmi: *Product Lifecycle Management*. 3. Auflage. Springer, Berlin, 2008. – ISBN 978–3–540–78173–8 (Print), 978–3–540–78172–1 (Online)
- [Sie10] SIEMENS PLM SOFTWARE: Open product lifecycle data sharing using XML / Siemens Product Lifecycle Management Software Inc. 2010. – White Paper
- [SKRM13] SILCHER, Stefan ; KÖNIGSBERGER, Jan ; REIMANN, Peter ; MITSCHANG, Bernhard: Cooperative service registries for the service-based Product Lifecycle Management architecture. In: *Proceedings of the 17th IEEE International Conference on Computer Supported Cooperative Work in Design (CSCWD '13)*, 2013. – ISBN 978–1–4673–6083–8, S. 439–446
- [SLK04] SUN, Chenliang ; LIN, Yi ; KEMME, Bettina: Comparison of UDDI Registry Replication Strategies. In: *Proceedings of the IEEE International Conference on Web Services*, IEEE Computer Society, 2004 (ICWS '04). – ISBN 0–7695–2167–3, S. 218–225
- [SMM11] SILCHER, Stefan ; MINGUEZ, Jorge ; MITSCHANG, Bernhard: Adopting the Manufacturing Service Bus in a Service-based Product Lifecycle Management Architecture. In: *Proceedings of the 44th International CIRP Conference on Manufacturing Systems (ICMS 2011)*, Online, 2011, S. 1–6
- [SMM12] SILCHER, Stefan ; MINGUEZ, Jorge ; MITSCHANG, Bernhard: A Novel Approach to Product Lifecycle Management based on Service Hierarchies. In: ÖZYER, Tansel (Hrsg.) ; KIANMEHR, Keivan (Hrsg.) ; TAN, Mehmet (Hrsg.): *Recent Trends in Information Reuse and*



- Integration*. Springer Vienna, 2012. – ISBN 978-3-7091-0737-9 (Print), 978-3-7091-0738-6 (Online), S. 343–362
- [SMSM10] SILCHER, Stefan ; MINGUEZ, Jorge ; SCHEIBLER, Thorsten ; MITSCHANG, Bernhard: A Service-Based Approach for Next-Generation Product Lifecycle Management. In: *Proceedings of the 11th IEEE International Conference on Information Reuse and Integration (IRI 2010)*, IEEE, 2010. – ISBN 978-1-4244-8097-5 (Print), 978-1-4244-8098-2 (Digital), S. 219–224
- [Sny91] SNYDER, Charles A.: CIM networking: promise and problems. In: *International Journal of Production Economics* 23 (1991), Nr. 1-3, S. 205–212. – ISSN 0925-5273
- [Sof11] SOFTWARE-INITIATIVE DEUTSCHLAND E.V.: *IT verliert Kontrolle über Geschäftsprozesse*. <http://www.softwareinitiative.de/news/archiv/2011-05-19-ITKontrolle>. Version: 2011
- [SSZM13] SILCHER, Stefan ; SEEBERG, Barbara ; ZAHN, Erich ; MITSCHANG, Bernhard: A Holistic Management Model for Manufacturing Companies and Related IT Support. In: *Proceedings of the 46th CIRP Conference on Manufacturing Systems (CMS 2013)* Band 7, Elsevier, 2013. – ISSN 2212-8271, S. 175–180
- [Sta04] STARK, John: *Product Lifecycle Management: 21st Century Paradigm for Product Realisation (Decision Engineering)*. 1. Auflage. Springer, Berlin, 2004. – ISBN 978-1-85233-810-7 (Print), 978-1-84628-067-2 (Online)
- [Sta11] STARK, John: *Product Lifecycle Management: 21st Century Paradigm for Product Realisation (Decision Engineering)*. 2. Auflage. Springer London, 2011. – ISBN 978-0-85729-545-3 (Print), 978-0-85729-546-0 (Online)

- [Sub08] SUBRAMANIAM, Ravi: *Set up a center of excellence for SOA - Considerations and analysis*. 2008
- [SW04] SCHENK, Michael ; WIRTH, Siegfried: *Fabrikplanung und Fabrikbetrieb: Methoden für die wandlungsfähige und vernetzte Fabrik*. 1. Springer, Berlin, 2004. – ISBN 978-3-540-20423-7
- [VDI08] VDI: *VDI-Richtlinie 4499 Blatt 1 - Digitale Fabrik - Grundlagen*. 2008
- [Vol09] VOLLING, Thomas: *Auftragsbezogene Planung bei variantenreicher Serienproduktion: Eine Untersuchung mit Fallstudien aus der Automobilindustrie (Produktion und Logistik)*. 1. Auflage. Gabler Verlag, 2009. – ISBN 978-3-8349-1477-4 (Print), 978-3-8349-8071-7 (Online)
- [VWBZ09] VAJNA, Sandor ; WEBER, Christian ; BLEY, Helmut ; ZEMAN, Klaus: *CAX für Ingenieure: Eine praxisbezogene Einführung*. 2. Auflage. Springer, Berlin, 2009. – ISBN 978-3-540-36038-4 (Print), 978-3-540-36039-1 (Online)
- [W3C04] W3C WORKING GROUP: *Web Services Architecture*. <http://www.w3.org/TR/ws-arch/>. Version: 2004
- [W3C05] W3C: *Web Services Choreography Description Language Version 1.0*. <http://www.w3.org/TR/ws-cdl-10/>. Version: 2005
- [W3C07a] W3C: *SOAP Version 1.2*. <http://www.w3.org/TR/soap/>. Version: 2007
- [W3C07b] W3C: *Web Services Description Language (WSDL) Version 2.0*. <http://www.w3.org/TR/wsdl20/>. Version: 2007
- [W3C07c] W3C: *Web Services Policy 1.5 - Framework*. <http://www.w3.org/TR/ws-policy/>. Version: 2007

- [Wan05] WANNENWETSCH, Helmut: *Vernetztes Supply Chain Management. SCM-Integration über die gesamte Wertschöpfungskette*. 1. Auflage. Springer, Berlin, 2005. – ISBN 978-3-540-23443-2 (Print), 978-3-540-27507-7 (Online)
- [WCH06] WESTKÄMPER, Engelbert ; CONSTANTINESCU, Carmen ; HUMMEL, Vera: New paradigm in manufacturing engineering: Factory life cycle. In: *Annals of the Academic Society for Production Engineering. Research and Development* 13 (2006), Nr. 1, S. 143–146
- [WCL<sup>+</sup>05] WEERAWARANA, Sanjiva ; CURBERA, Francisco ; LEYMAN, Frank ; FERGUSON, Donald F. ; STOREY, Tony: *Web Services Platform Architecture: Soap, WSDL, WS-Policy, WS-Addressing, WS-Bpel, WS-Reliable Messaging and More*. Prentice Hall International, 2005. – ISBN 9780131488748
- [WFK01] WORN, Heinz ; FREY, Daniel ; KEITEL, Jochen: Digital Factory - Planning and running enterprises of the future. In: *Betonwerk und Fertigteilechnik* 67 (2001), Nr. 4, S. 26–37
- [WZ09] WESTKÄMPER, Engelbert (Hrsg.) ; ZAHN, Hans E. (Hrsg.): *Wandlungsfähige Produktionsunternehmen: Das Stuttgarter Unternehmensmodell*. 2005. Springer, Berlin, 2009. – ISBN 978-3-540-21889-0 (Print), 978-3-540-68890-7 (Online)
- [XNW01] XU, Dongyan ; NAHRSTEDT, Klara ; WICHADAKUL, Duangdao: QoS-aware discovery of wide-area distributed services. In: *Proceedings of the First IEEE/ACM International Symposium on Cluster Computing and the Grid*, 2001, S. 92–99
- [ZMV05] ZÄH, Michael F. ; MÖLLER, Niklas ; VOGL, W.: Symbiosis of Changeable and Virtual Production. In: ZÄH, Michael F. (Hrsg.) ; REINHART, G. (Hrsg.): *Proceedings of the 1st International Conference on Changeable, Agile, Reconfigurable and Virtual Production (CARV 2005)*, Utz Verlag, 2005. – ISBN 978-3831605408, S. 3–10

- [ZSMA04] ZÄH, Michael F. ; SUDHOFF, Wolfgang ; MÖLLER, Niklas ; AULL, Florian: Evaluation of mobile production scenarios based on the Real Option Approach. In: *4. Chemnitzer Tagung - Vernetzt Planen und Produzieren (VPP)*, 2004, S. 300–305

Alle Internetreferenzen in dieser Arbeit wurden zuletzt am  
6. November 2013 geprüft.