

Torsten Zimmer (Hrsg.)

Prozessintegration mit SAP NetWeaver® PI 7.1

Eine Einführung in die Architektur
der Prozessintegration anhand von Fallstudien
unternehmensinterner sowie unternehmensübergreifender
Geschäftsprozesse der Logistik

PRAXIS



**VIEWEG+
TEUBNER**

Torsten Zimmer (Hrsg.)

Prozessintegration mit SAP NetWeaver® PI 7.1

Usability Management bei SAP®-Projekten

herausgegeben von P. Abele, J. Hurtienne und J. Prümper

Logistikprozesse mit SAP®

von J. Benz und M. Höflinger

Grundkurs SAP® ERP

von D. Frick, A. Gadatsch und U. Schäffer-Külz

100 Minuten für den kompetenten Auftritt

von S. Horger-Thies

100 Minuten für Anforderungsmanagement

von M. Grande

SAP® ERP – Praxishandbuch Projektmanagement

von H. Gubbels

Management für Ingenieure

von G. Hachtel und U. Holzbaur

Masterkurs IT-Management

herausgegeben von J. Hofmann und W. Schmidt

Testautomation mit SAP®

von A. Vivenzio

Personalwirtschaft mit SAP® R/3

herausgegeben von P. Wenzel

Torsten Zimmer (Hrsg.)

Prozessintegration mit SAP NetWeaver® PI 7.1

Eine Einführung in die Architektur
der Prozessintegration anhand von Fallstudien
unternehmensinterner sowie unternehmensübergreifender
Geschäftsprozesse der Logistik

Mit 191 Abbildungen

PRAXIS



VIEWEG+
TEUBNER

Bibliografische Information der Deutschen Nationalbibliothek
Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der
Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über
<<http://dnb.d-nb.de>> abrufbar.

Das in diesem Werk enthaltene Programm-Material ist mit keiner Verpflichtung oder Garantie irgend-einer Art verbunden. Der Autor übernimmt infolgedessen keine Verantwortung und wird keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials oder Teilen davon entsteht.

Höchste inhaltliche und technische Qualität unserer Produkte ist unser Ziel. Bei der Produktion und Auslieferung unserer Bücher wollen wir die Umwelt schonen: Dieses Buch ist auf säurefreiem und chlorfrei gebleichtem Papier gedruckt. Die Einschweißfolie besteht aus Polyäthylen und damit aus organischen Grundstoffen, die weder bei der Herstellung noch bei der Verbrennung Schadstoffe freisetzen.

„SAP“, R/3®, mySAP®, mySAP.com®, mySAP CRM®, mySAP SCM®, ABAP/4®, SAP-GUI®, SAP APO®, IDES®, BAPI®, BW®, ECC®, SAP Business Information Warehouse®, SAP Business Workflow® sind eingetragene Warenzeichen der SAP Aktiengesellschaft Systeme, Anwendungen, Produkte in der Datenverarbeitung, Neurtotstraße 16, D-69190 Walldorf. Der Herausgeber bedankt sich für die freundliche Genehmigung der SAP Aktiengesellschaft, das Warenzeichen im Rahmen des vorliegenden Titels verwenden zu dürfen. Die SAP AG ist jedoch nicht Herausgeberin des vorliegenden Titels oder sonst dafür presserechtlich verantwortlich. Für alle Screen-Shots des vorliegenden Titels gilt der Hinweis: Copyright SAP AG. Microsoft®, Windows®, Windows NT®, EXCEL®, VISIO®, SQL-Server® sind eingetragene Warenzeichen der Microsoft Corporation. Oracle® ist eingetragenes Warenzeichen der Oracle Corporation. Bei der Zusammenstellung der Informationen zu diesem Produkt wurde mit größter Sorgfalt gearbeitet. Trotzdem sind Fehler nicht vollständig auszuschließen. Verlag, Herausgeber und Autor können für fehlerhafte Angaben und deren Folgen weder eine juristische Verantwortung noch irgendeine Haftung übernehmen.

1. Auflage 2011

Alle Rechte vorbehalten

© Vieweg+Teubner Verlag | Springer Fachmedien Wiesbaden GmbH 2011

Lektorat: Christel Roß | Maren Mithöfer

Vieweg+Teubner Verlag ist eine Marke von Springer Fachmedien.

Springer Fachmedien ist Teil der Fachverlagsgruppe Springer Science+Business Media.

www.viewegteubner.de



Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

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

Umschlaggestaltung: KünkelLopka Medienentwicklung, Heidelberg

Druck und buchbinderische Verarbeitung: AZ Druck und Datentechnik, Berlin

Gedruckt auf säurefreiem und chlorfrei gebleichtem Papier

Printed in Germany

ISBN 978-3-8348-1564-4

Inhalt

1	Einleitung	1
2	SAP NetWeaver®	7
2.1	Einführung Informationssysteme	7
2.2	SAP NetWeaver®-Komponenten	8
2.1.1	SAP NetWeaver® Application Server	9
2.1.2	SAP NetWeaver® Portal	9
2.1.3	SAP NetWeaver® Developer Studio	9
2.1.4	SAP NetWeaver® Master Data Management	10
2.1.5	SAP NetWeaver® Composition Environment	10
2.1.6	SAP NetWeaver® Process Integration	11
3	Innerbetriebliche Geschäftsprozesse	31
3.1	ERP Materialwirtschaft (MM)	31
3.1.1	Organisationsstrukturen in der ERP Materialwirtschaft	31
3.1.2	Stammdaten und Bezugsquellen in der ERP Materialwirtschaft	32
3.1.3	Materialdisposition in der ERP Materialwirtschaft	36
3.1.4	Einkaufsabwicklung in der ERP Materialwirtschaft	39
3.1.5	Bestandsführung in der ERP Materialwirtschaft	42
3.2	ERP Produktion (PPS)	44
3.2.1	Einführung in die ERP Produktion (PPS)	44
3.2.2	Produktions- und Beschaffungsplanung	46
3.2.3	Produktionssteuerung	50
3.2.4	Kanban	54
3.3	ERP Vertrieb (SD)	55
3.3.1	Organisationseinheiten im ERP Vertrieb	55
3.3.2	Stammdaten im ERP Vertrieb	56
3.3.3	Geschäftsprozesse des Vertriebs	58
3.4	ERP Integration	60
3.5	Supplier Relationship Management (SRM)	61
3.5.1	Einführung SRM	61
3.5.2	Einkaufsabwicklung im SRM	64

3.5.3 E-Tendering im SRM	65
3.5.4 Lieferantenmanagement im SRM	67
3.6 Supply Chain Management (SCM)	69
3.6.1 Einführung SCM	69
3.6.2 SCM Produktionsplanung	71
3.6.3 SCM Bestandsplanung und -steuerung	71
3.7 Customer Relationship Management (CRM)	72
3.7.1 Einführung CRM.....	72
3.7.2 Verkaufsabwicklung im CRM.....	72
4 Zwischenbetriebliche Geschäftsprozesse	75
4.1 Integrierte ERP Einkaufs- und Verkaufsabwicklung	75
4.2 Collaborative SCM	76
4.2.1 Grundlagen Collaborative SCM.....	76
4.2.2 Collaborative Planning, Forecasting and Replenishment (CPFR).....	77
4.2.3 Continuous Replenishment Program.....	79
4.3 Integrierte SRM/CRM Einkaufs- und Verkaufsabwicklung.....	81
4.4 Integration von Lieferanten im SRM	83
5 Fallstudie <i>Zentrale Bestellabwicklung</i>	85
5.1 Einleitung	85
5.2 Beschreibung der Szenarien.....	86
5.3 Implementierung des Basisszenarios.....	88
5.3.1 Grundlagen	88
5.3.2 Design	88
5.3.3 Konfiguration.....	91
5.3.4 ABAP-Proxy.....	93
5.3.5 Ausführung.....	94
5.4 Erweitertes Szenario.....	96
5.4.1 Grundlagen	96
5.4.2 Design	97
5.4.3 Konfiguration.....	105
5.4.4 ABAP-Proxy.....	109
5.4.5 Ausführung.....	110
5.5 Fazit.....	111

5.6 Anhang	111
5.6.1 Parameter des Funktionsbausteins BAPI_PO_CREATE	111
5.6.2 ABAP-Quelltext: Aufruf des Basisszenarios	114
5.6.3 ABAP-Quelltext: Definition und Aufruf des Funktionsbausteins	117
5.6.4 ABAP-Quelltext: Aufruf des erweiterten Szenarios.....	118
6 Fallstudie <i>Roh- und Hilfsstoffbestellung</i>	121
6.1 Motivation	121
6.2 Schrittweise Implementierung und Testen des Szenarios.....	123
6.2.1 Funktionsbaustein und Programm in System A	123
6.2.2 Designphase.....	125
6.2.3 Konfigurationsphase	133
6.2.4 Erstellung der DB-Tabelle und Proxy-Generierung in System B	143
6.2.5 Testen des Szenarios.....	146
6.3 Fazit.....	148
6.4 Anhang	148
7 Fallstudie <i>Bestellung über .NET Anbindung</i>	151
7.1 Motivation.....	151
7.2 Technische Prozessbeschreibung	152
7.3 Entwurf.....	153
7.4 Konfiguration.....	157
7.5 Entwicklung der .NET Client-Anwendung.....	162
7.6 Monitoring des Bestellprozesses	167
7.7 Fazit.....	170
8 Fallstudie <i>Bestellung über Java-Proxy</i>	171
8.1 Motivation.....	171
8.2 Konfiguration.....	172
8.2.1 Enterprise Services Repository (ES Repository)	172
8.2.2 Integration Builder.....	178
8.2.3 Java-Programmierung.....	181
8.3 Fazit.....	183
8.4 Anhang	183
8.4.1 Anhang: Source Code 1	183
8.4.2 Anhang: Source Code 2.....	184

9	Fallstudie <i>Internationale Bestellabwicklung</i>	189
9.1	Beschreibung des Szenarios	189
9.2	ABAP-Proxy to SOAP	190
9.2.1	Objekte im Enterprise Services Repository.....	190
9.2.2	Objekte im Integration Directory	193
9.3	ABAP-Proxy to RFC.....	196
9.3.1	Objekte im Enterprise Services Repository.....	196
9.3.2	Objekte im Integration Directory	201
9.4	Generierung der ABAP-Proxys	204
9.5	Ablauf und Monitoring	205
9.6	Anhang: Programmcode	207
10	Fallstudie <i>Produktkatalog eines Shop-Systems</i>	209
10.1	Entwicklung im ERP-System	210
10.1.1	Datenbank	210
10.1.2	Funktionsbausteine	211
10.2	Design des Integrationsszenarios.....	213
10.2.1	Enterprise Services Builder	213
10.2.2	Konfiguration im Integration Builder	218
10.2.3	Testen des Szenarios	220
10.2.4	WSDL-Datei erzeugen	220
10.3	Web Service Consumer.....	221
10.3.1	Web Service Consumer mit Eclipse	221
10.3.2	XML-Nachrichten Monitoring im PI-System	222
10.4	Fazit	222
11	Fallstudie <i>Supplier Finance Process</i>	223
11.1	Technische Prozessbeschreibung	224
11.1.1	Aufbau des Integrationsprozesses.....	224
11.1.2	Übertragung der Rechnungsfreigabedaten	225
11.1.3	Empfang des Acknowledgements	230
11.2	Praxistauglichkeit des beschriebenen Integrationsprozesses.....	236

12 Daten-Austausch mit einem externen Businesspartner	237
12.1 Technische Prozessbeschreibung	238
12.2 Design einer Security-konformen Middleware-Infrastruktur	239
12.2.1 Security konforme Infrastruktur unter Nutzung einer dezentralen AAE.....	240
12.2.2 Implementierung eines Virencanners für eingehende Nachrichten	243
12.3 Nutzung von SFTP zur sicheren Datenübertragung	252
12.3.1 Objekte des Integration Directory	252
12.3.2 Objekte des Enterprise Services Repository	254
13 Testszzenarien mit SAP NetWeaver PI 7.1 und SAP ERP	257
13.1 Motivation	257
13.2 Technische Prozessbeschreibung	258
13.3 Variante 1: Testszzenario ohne ccBPM u. ohne Systemweiche.....	260
13.4 Variante 2: Testszzenario ohne ccBPM mit Systemweiche.....	261
13.5 Variante 3: Testszzenario mit ccBPM und mit Systemweiche	263
13.6 Ausblick und Tipps.....	264
13.7 Anhang	266
14 Umsetzung einer Prozessintegration.....	267
14.1 Kommunikation zwischen SRM und ERP über Schnittstellen	267
14.2 Kommunikation zwischen CRM und ERP über Schnittstellen.....	268
14.3 B2B-Integration von ERP-Systemen	269
14.4 Fazit.....	269
Literaturverzeichnis.....	271
Über die Autoren	275

1 Einleitung

In Unternehmen entstehen derzeit starke heterogene IT-Systemlandschaften. Dabei sind die einzelnen Informationssysteme oftmals nicht integriert, verursacht durch eine einseitige Orientierung an Funktionsbereichen wie Einkauf, Produktion, Verkauf, Finanzbuchhaltung oder durch Verwendung von Legacy-Systemen, d.h. veraltete und proprietäre Informationssysteme. Auch durch Dezentralisierung der Logistik, durch Internationale Ausrichtung des Unternehmens, durch Fusionen, durch Akquisitionen oder durch starke Landesgesellschaften wird diese Heterogenität verstärkt. Probleme entstehen dann bei der Ausführung systemübergreifender Geschäftsprozesse durch sogenannte *Medienbrüche*. Sollen jedoch Geschäftsprozesse über mehrere Systeme durchgängig ausgeführt werden, dann müssen Konzepte und Systeme zur integrierten Ausführung dieser Geschäftsprozesse vorhanden sein. Wie kann ein Unternehmen diesen elektronischen Austausch von Dokumenten und Nachrichten über eine große Anzahl von Schnittstellen zwischen (heterogenen) Komponenten managen? Unterschiedliche Formen von Middleware wurden entwickelt, um Daten und Nachrichten zwischen verschiedenen Systemen austauschen zu können. Eine Klasse von Software, die als Abstraktionsschicht bzw. als Softwareschicht eine Kommunikation zwischen verschiedenen Anwendungen ermöglicht, wird als *Middleware* bezeichnet. Die einfachste Form einer Middleware sind *Remote Procedure Calls* (RPC) bzw. die SAP®-spezifische Ausprägung *Remote Function Calls* (RFC)¹, bei der über punktuelle Verbindungen (Peer to Peer) Funktionen aus anderen Systemen heraus aufgerufen werden (Meinhardt 2002). Im Gegensatz dazu basieren EAI-Systeme auf Topologien wie *Hub & Spoke* oder *Enterprise-Service-Bus* (ESB) und ermöglichen einen Nachrichtenaustausch über eine zentrale Plattform. EAI bedeutet *Enterprise Application Integration*. EAI-Systeme können heterogene Anwendungen eines Unternehmens integrieren, um dadurch die Ausführung aktueller Geschäftsprozesse eines Unternehmens zu unterstützen. Der Begriff EAI entstand durch das Bemühen, viele Anwendungen, die ursprünglich isoliert entworfen wurden und dabei nur Teilaufgaben von Geschäftsprozessen abdecken, zu einem einheitlichen Geschäftsprozess zusammenzufügen (Keller 2002). Die Aufgaben eines EAI-Systems sind dabei, einen vollständig medienbruchfreien Austausch von Nachrichten zu erreichen sowie eine flexible und entkoppelte Ausführung sowohl unternehmensinterner als auch unternehmensübergreifender Geschäftsprozesse zu ermöglichen. Diese vollständige elektronische Integration von Geschäftsprozessen sowohl innerhalb eines Unternehmens als auch zwischen Unternehmen wird als *Prozessintegration* bezeichnet. Ein Geschäfts-

¹ Beispielsweise ist ein SAP BAPI® zur RFC-Verwendung gekennzeichnet.

prozess wird dabei vollständig elektronisch über Nachrichtenaustausch zwischen einem Sendersystem und einem Empfängersystem unter Verwendung von Protokollen² ausgeführt. Die Integration zwischen Anwendungen innerhalb eines Unternehmens wird als *Application-to-Application Integration* (A2A-Integration) bezeichnet. Dabei werden innerbetriebliche Geschäftsprozesse zwischen einem EAI-System und Anwendungen definiert und integriert ausgeführt. Eine *Business-to-Business Integration* (B2B-Integration) liegt vor, wenn Geschäftsprozesse zwischen Unternehmen integriert ausgeführt werden. Der Austausch von Nachrichten kann sowohl synchron als auch asynchron erfolgen. Bei einer synchronen Nachricht wird zu einer gesendeten Anfrage (*request*) eine Antwort (*response*) erwartet. Bei einer asynchronen Kommunikation wird nur eine Anfrage gesendet. Prozessintegration wird insbesondere innerhalb heterogener IT-Systemlandschaften neben der Datenintegration in Zukunft immer mehr an Bedeutung gewinnen.

Durch eine Neugestaltung der Geschäftsprozesse über eine Prozessintegration können verschiedene Potentiale genutzt werden (Nicolescu 2009).

- Höhere Flexibilität bei der Ablauforganisation und bei der Ausführung von innerbetrieblichen Geschäftsprozessen im Vergleich zur bisherigen Standardsoftware
- Verbesserte Kommunikation zwischen Kunden und Lieferanten bzgl. des Nachrichtenaustauschs von z.B. Bestell- und Auftragsdaten, Preis- und Produktinformationen, Verfügbarkeit von Materialien, geplante Abverkaufsdaten sowie Informationen zu Service- und Finanzdiensten
- Potentiale zu Zeit- und Kosteneinsparungen bei der Ausführung von Geschäftsprozessen
- Reduzierung von Latenzzeiten und Fehleingaben in der Informationsbeschaffung und -verarbeitung
- Reduzierung der Entwicklungskosten, z.B. zentrales Repository für gemeinsam genutzte Entwicklungsobjekte

Zur Umsetzung einer Prozessintegration werden oftmals E-Business-Standards verwendet. Ziel des E-Business ist die integrierte Ausführung aller digitalisierbaren Bestandteile ökonomischer Prozesse. Durch die medienbruchfreie, rechnerbasierte und damit weitgehend automatisierte Reaktion auf eingehende und ausgehende Nachrichten von Geschäftsprozessen sollen Potentiale zu Zeit- und Kosteneinsparungen genutzt werden. Dieses Ziel kann nur erreicht werden, indem sowohl eine organisatorische Anpassung als auch eine Vereinheitlichung der Datenstrukturen der beteiligten Anwendungen erfolgt (Thome 2002). Zur Umsetzung dieser Anforderungen werden verbreitete E-Business-Standards wie z.B. ebXML³

² Zur Erläuterung und zur Verwendung von Protokollen siehe (Mandl 2010).

³ <http://www.ebXML.org>

oder RosettaNet⁴ eingesetzt. ebXML basiert auf dem Austausch XML-basierter Nachrichten über SOAP. Jedoch werden nicht nur XML-Nachrichten definiert, sondern ebXML adressiert den elektronischen Geschäftsverkehr als Ganzes. So wird sowohl das Zueinanderfinden von Geschäftspartnern als auch das Ausführen von Geschäftsprozessen über sogenannte *Business Process Specifications* basierend auf den XML-Nachrichten definiert (Beimborn 2002). RosettaNet ist ein im Jahre 1998 auf Initiative von 400 Unternehmen entwickeltes XML-Framework. Dabei werden jedoch nicht nur die XML-Formate zum Nachrichtenaustausch bei der Ausführung von Geschäftsprozessen im sogenannten *RosettaNet Implementation Framework* (RNIF) definiert, sondern zusätzlich werden auch die Abläufe und Regeln dieser Geschäftsprozesse über sogenannte *Partner Interface Processes* (PIPs) festgelegt (Schellenbach 2003).

Aufgabe der Datenintegration ist die Synchronisation bzw. die zentrale Verwaltung von Stammdaten, z.B. von Lieferanten, Kunden, Produkten, Kontrakten oder Konditionen. Insbesondere ist die Datenintegration auch eine wesentliche Voraussetzung der Prozessintegration über ein EAI-System geworden. Sie stellt die Stammdatenbasis bei einer systemübergreifenden Ausführung von Geschäftsprozessen dar. Beispielsweise müssen bei der Übertragung einer Bestellung aus einem SRM-System in ein ERP-System in der Bestellung verwendete Stammdaten innerhalb der beteiligten Systeme synchronisiert sein. Bei einer B2B-Integration können zur Identifikation von Geschäftspartnern auch verschiedene internationale Standards genutzt werden. Beispielsweise kann die sogenannte D-U-N-S® Nummer zur Angabe eines Geschäftspartners beim elektronischen Austausch von Nachrichten verwendet werden. Diese Nummer wird von der Firma Dun & Bradstreet⁵ vergeben und ermöglicht eine weltweit eindeutige Unternehmensidentifikation. Konzepte zur unternehmensweiten Stammdatenintegration werden in (Schmidt 2010) beschrieben.

Ebenso wird derzeit neben der Prozess- und der Datenintegration die konkrete Umsetzung einer *Serviceorientierten Architektur* (SOA) auf Basis eines EAI-Systems unter Verwendung von Webservices sehr stark diskutiert (Abramowicz 2008, Grauer 2010). In einer SOA werden Funktionen betrieblicher Anwendungssysteme wie z.B. eine Kreditkartenprüfung oder eine Bonitätsprüfung als plattformunabhängige Services (z.B. Webservices) in zentralen Verzeichnissen bereitgestellt. Diese autonomen Services können dann von verschiedenen Anwendungssystemen genutzt werden. Um diese Services verschiedener Quellen zu einem kollaborativen Geschäftsprozess zu integrieren, kann ein Enterprise-Service-Bus (ESB) als Backbone einer SOA eingesetzt werden. Standardisierte Protokolle und Workflow-Management-Systeme (WfMS) ermöglichen die sogenannte *Orchestrierung* der

⁴ <http://www.rosettanet.org/>

⁵ <http://www.dnb.com/>

Services zu komplexen Geschäftsprozessen (Grauer 2010). In (SAP 2011) wird betont, dass gerade die Vorstellung, dass Anwendungen aus Services zusammgebaut werden können, die in einer zentralen Registry bereitgestellt werden, die Verwendung von Webservices attraktiv macht. Eine Beschreibung der Entwicklung von sogenannten *Enterprise Services* für das Konzept SAP® Enterprise SOA™ ist in (Huvar 2008) gegeben. Ähnlichkeiten und Unterschiede beim Entwurf traditioneller Anwendungssysteme zum Entwurf serviceorientierter Architekturen unter Verwendung von lose gekoppelten Enterprise Services werden in (Schelp 2008) dargestellt.

Unternehmen, die eine EAI-Lösung einsetzen, nutzen diese technische Plattform jedoch mehr zur Optimierung von unternehmensinternen und unternehmensübergreifenden Geschäftsprozessen als für die Umsetzung einer Serviceorientierten Architektur (Gerg 2010).

Ziel dieses Buches ist es, Möglichkeiten zur Realisierung der prozessorientierten Integration von Anwendungssystemen in homogenen und in heterogenen IT-Systemlandschaften anhand SAP NetWeaver® PI 7.1 darzustellen. Insbesondere sollen Potentiale der Integration von Geschäftsprozessen verdeutlicht werden. Es werden verschiedene E-Business-Standards zur Realisierung einer Prozessintegration erläutert. Weiterhin sollen Modellierungskennnisse für die Beschreibung und Ausführung von Geschäftsprozessen anhand der *Business Process Execution Language for Web Services* (BPEL)⁶ erlangt werden. Über die Fallstudien sollen insbesondere die technischen Möglichkeiten zur Verbesserung der Ausführung von Geschäftsprozessen mittels Prozessintegration aufgezeigt werden. Als Ergänzung dazu sollen auch Möglichkeiten zur Umsetzung einer Serviceorientierten Architektur über ein EAI-System durch Kombination von Webservices aus unterschiedlichen Quellen veranschaulicht werden.

Dieses Buch erläutert zunächst in einem theoretischen Teil die Komponenten von SAP NetWeaver® und dabei insbesondere die EAI-Lösung SAP NetWeaver® PI 7.1. Danach werden bedeutende Geschäftsprozesse der Logistik beschrieben. Dabei werden sowohl innerbetriebliche als auch zwischenbetriebliche Geschäftsprozesse von ERP-, SRM-, SCM- und CRM-Systemen betrachtet, deren Ausführung durch Prozessintegration unterstützt werden kann. Die Beschreibung der Geschäftsprozesse orientiert sich dabei sehr stark an entsprechenden SAP®-Lösungen.

In einem praktischen Teil werden dann Realisierungen zur Umsetzung der Prozessintegration anhand von theoretischen und praktischen Fallstudien vorgestellt. In einer Fallstudie werden dabei zu einem beschriebenen Geschäftsprozess die notwendigen Aktivitäten und die verwendete Integrationstechnik für eine (bestimmte) vorhandene Systemlandschaft dargestellt. Ziel ist es dabei, möglichst viele unterschiedliche Adapter-Typen und Proxy-Techniken vorzustellen sowie Möglich-

⁶ <http://www.oasis-open.org/specs/#wsbpelv2.0>

keiten zur Entwicklung von zusammengesetzten Anwendungen (*Composite Applications*) über die Einbindung von Webservices zu veranschaulichen. Dabei wird meist der Geschäftsprozess der Bestellabwicklung verwendet. Jedoch ist die Übertragung auf andere logistische Geschäftsprozesse wie z.B. die Verkaufsabwicklung durch Austausch der entsprechenden Entwicklungsobjekte auf einfache Weise möglich. Alle Fallstudien wurden auf SAP NetWeaver® Process Integration Version 7.1 entwickelt.

Herr Stefan Kellnar beschreibt eine Fallstudie *Zentrale Bestellabwicklung*, bei der zunächst eine Integration der Einkaufsabwicklung zwischen zwei ERP-Systemen mittels eines SOAP-Adapters realisiert wird. Eine Erweiterung des Szenarios erfolgt dann über einen BPEL-Integrationsprozess unter Einbindung eines File- und eines Mail-Adapters.

Die Fallstudie *Roh- und Hilfsstoffbestellung* von Herrn Kern beschreibt die Entwicklung und die Ausführung eines Integrationsprozesses, der auf der Modellierungssprache BPEL basiert. Dabei wird aus einem ERP-System unter Verwendung eines RFC-Adapters eine Nachricht an ein EAI-System gesendet. In Abhängigkeit vom Inhalt der Nachricht wird entweder die Nachricht per File-Adapter oder über einen ABAP-Proxy an ein weiteres ERP-System weitergeleitet.

Die Fallstudie *Bestellung über .NET Anbindung* von Herrn Kailer beschreibt einen Integrationsprozess, bei dem Bestelldaten aus einer .NET-Anwendung per Webservice Aufruf über einen SOAP-Adapter an ein EAI-System gesendet werden. Über einen RFC-Adapter werden dann die Bestelldaten in ein ERP-System weitergeleitet. Dort wird eine Bestellung angelegt und eine Antwort wird an die .NET-Anwendung zurückgesendet.

Die Fallstudie *Bestellung über Java-Proxy* von Herrn Awad stellt ein Szenario vor, bei dem Bestelldaten aus einem Java-Programm an ein EAI-System gesendet werden. Die Bestelldaten werden in ein ERP-System über einen SOAP-Adapter weitergeleitet, und nach Anlage einer Bestellung wird eine Antwort an das Java-Programm zurückgesendet.

Die Fallstudie von Herrn Gerg beschreibt eine internationale Bestellabwicklung. Zunächst erfolgt die Erfassung von Bestelldaten in einem ERP-System. Diese Bestelldaten sollen in ein ERP-System (Zielsystem) übertragen werden, um dort die Bestellabwicklung durchzuführen. Bei Abweichung der erfassten Bestellwährung von der Buchungskreiswährung € des Zielsystems wird ein externer Webservice für die Umrechnung zum aktuellen Wechselkurs über einen SOAP-Adapter aufgerufen. Über einen RFC-Adapter werden die Bestelldaten in das Zielsystem übertragen, und für die Durchführung der Bestellabwicklung wird dort eine Bestellung angelegt.

Die Fallstudie *Produktkatalog eines Shop-Systems* von Herrn Schmatzer stellt technische Möglichkeiten sowohl zur Erzeugung als auch zur Verwendung von Webservices in einer Java-Anwendung dar. Zunächst erfolgt die Beschreibung der Erzeu-

gung eines Webservices, der auf einem im ERP-System definierten Funktionsbaustein basiert. Danach kommuniziert eine Java-Anwendung als externer Web Service Consumer über SOAP-Nachrichten mit einem EAI-System unter Verwendung dieses Webservices. Das EAI-System sendet diese Anfragen über einen RFC-Adapter an ein ERP-System weiter und leitet die Antwort an die Java-Anwendung zurück.

Herr Bernecker beschreibt ein Praxisbeispiel einer B2B-Integration. Dabei wird ein *Supplier Finance Process* vorgestellt, der dazu genutzt werden kann, die Geschäftsabwicklung zwischen einem Kunden und seinem Lieferanten bezüglich der Zahlungsmodalitäten zu verbessern. Das Wesentliche dieses Prozesses besteht dabei aus einer Zahlungsvereinbarung zwischen einem Kunden und einem Lieferanten, bei der flexible Zahlungsziele vereinbart werden. Die Konfiguration erfolgt über einen Integrationsprozess unter Verwendung eines File-Adapters und eines Mail-Adapters sowie durch die Übertragung von Nachrichten über ein Third-Party Tool.

In einem weiteren Praxisbeispiel einer B2B-Integration beschreibt Herr Bernecker den automatisierten elektronischen Austausch von Bestellungen, Bestellbestätigungen und Rechnungen zwischen zwei Unternehmen hinsichtlich der Aspekte Security, Aufbau der Infrastruktur und Nutzung von SFTP als Übertragungsprotokoll. Dabei kommen ein IDoc-Adapter und ein File-Adapter zum Einsatz. Aufgrund eines nicht verfügbaren SFTP-Adapters wird als Alternative für den Datenaustausch zwischen einem EAI-System und einem SFTP-Server die Nutzung des Standard-File-Adapters und eines nativen SFTP-Clients auf Betriebssystemebene verwendet.

Herr Tajedini beschreibt, wie in der Praxis Testszenarien genutzt werden können, um den synchronen Aufruf von Nachrichten zwischen einem PI-System und mehreren ERP-Systemen zu testen. Dabei werden unter Verwendung eines auf Java Open Source basierten Tools eigene Test-Client-Szenarien implementiert.

Im letzten Abschnitt wird schließlich eine Übersicht der möglichen Umsetzung einer Prozessintegration mit den im Praxisteil vorgestellten Adapter-Typen und Proxy-Techniken dargestellt. Dabei soll veranschaulicht werden, wie der elektronische Nachrichtenaustausch der im Theorieteil vorgestellten logistischen Prozesse zu einer gegebenen Systemlandschaft implementiert werden kann. Es wird dazu die Umsetzung einer Prozessintegration am Beispiel der innerbetrieblichen Kommunikation eines SRM-Systems mit einem ERP-System sowie am Beispiel der innerbetrieblichen Kommunikation eines CRM-Systems mit einem ERP-System betrachtet. Weiterhin wird ein zwischenbetrieblicher logistischer Geschäftsprozess zwischen dem ERP-System eines Kunden und dem ERP-System eines Lieferanten über eine B2B-Integration unter Verwendung des XML-Standards von RosettaNet dargestellt.

2 SAP NetWeaver®

2.1 Einführung Informationssysteme

Ein Informationssystem besteht aus Menschen und Maschinen, die Informationen erzeugen und bzw. oder benutzen und die durch Kommunikationsbeziehungen miteinander verbunden sind (Hansen 2009). Ein *betriebliches Informationssystem* unterstützt die Leistungsprozesse und Austauschbeziehungen innerhalb eines Unternehmens sowie zwischen dem Unternehmen und seiner Umwelt (Hansen 2009). Über ein betriebliches Informationssystem kann der gesamte informations- und Daten verarbeitende Teilbereich eines Unternehmens abgebildet werden. Die Aufgaben von betrieblichen Informationssystemen sind.

- Unterstützung der betrieblichen Abläufe eines Unternehmens
- Kommunikation mit Geschäftspartnern
- Entscheidungsunterstützung für das Management
- Durchführung der Finanzbuchhaltung und des Controlling
- Personalmanagement, Personalverwaltung und Personalakquisition
- Unterstützung von Bürotätigkeiten

Weiterhin werden zur Unterstützung der elektronischen Ausführung von betrieblichen Abläufen *Workflow-Management-Systeme* eingesetzt. Workflow-Management-Systeme sind Groupware-Systeme, die kooperative betriebliche Abläufe – sogenannte *Workflows* – auf einem Ablaufschema basierend aktiv steuern (Oberweis 1996). Ein betrieblicher Ablauf besteht aus manuellen, teilautomatisierten oder automatisierten Aktivitäten, die in einem Unternehmen nach bestimmten Regeln auf ein bestimmtes Ziel hin unter Verwendung von Ressourcen ausgeführt werden (Oberweis 1996). In der Literatur existieren unterschiedliche Bezeichnungen für betriebliche Abläufe, wie z.B. Geschäftsprozesse, Business Processes, Unternehmensprozesse, Kernprozesse, Geschäftsvorgänge oder auch nur Prozesse (im Zusammenhang mit einer Prozessorganisation (Gaitanides 1983)). Im Weiteren wird jedoch nur noch die Bezeichnung *Geschäftsprozess* verwendet.

ERP-Systeme (*Enterprise Resource Planning*) sind integrierte betriebswirtschaftliche Standardsoftwarepakete bzw. Komponenten. ERP-Systeme unterstützen informationstechnisch nahezu alle betrieblichen Funktionsbereiche und Prozesse eines Unternehmens, wie z.B. über die Komponenten *Finanz- und Rechnungswesen*, *Personalwirtschaft*, *Produktion*, *Vertrieb* und *Materialwirtschaft*. Technische Funktionsbereiche (CAx) werden dabei nicht unterstützt.

Bei einem integrierten betrieblichen Informationssystem können Geschäftsprozesse über mehrere Komponenten hinweg ausgeführt werden. Beispiel: Während des

Geschäftsprozesses der Anlage eines Kundenauftrags in der Komponente Vertrieb, kann eine Verfügbarkeitsprüfung in der Komponente Materialwirtschaft, die Anlage eines Planauftrags in der Komponente Produktion sowie eine Kreditlimitprüfung in der Komponente Debitorenbuchhaltung erfolgen. Die Integration erfolgt dabei über eine zentrale Datenbank mit gemeinsamen Stammdaten für alle Komponenten des Informationssystems. Beispielsweise wird ein Materialstammsatz nur einmal physisch im System gespeichert, jedoch enthält dieser Attributwerte für unterschiedliche Komponenten.

Ein Produkt, das verschiedene betriebliche Informationssysteme als Standardsoftware anbietet, ist die SAP Business Suite® des Unternehmens SAP AG (Business Suite 2011). Komponenten der SAP Business Suite® sind

- SAP Enterprise Resource Planning® (SAP® ERP®)
- SAP Supplier Relationship Management® (SAP® SRM®)
- SAP Customer Relationship Management® (SAP® CRM®)
- SAP Product Lifecycle Management® (SAP® PLM®)
- SAP Supply Chain Management® (SAP® SCM®)

Weiterhin werden durch die SAP® Business Suite auch branchenspezifische Ausprägungen der Anwendungen unterstützt (SAP 2011).

Das SAP® ERP-System ist ein Beispiel für ein integriertes betriebliches Informationssystem. Es unterstützt Geschäftsprozesse im Finanzwesen, in der Personalwirtschaft und in der Logistik. Aktuelle Releases sind die SAP® ERP Central Component (SAP® ECC) 5.0 bzw. 6.0 (SAP 2011).

2.2 SAP NetWeaver®-Komponenten

SAP NetWeaver® ist ein Produkt der SAP AG und stellt eine offene Integrations- und Anwendungsplattform zur Verfügung. Es verbindet heterogene Anwendungen zu durchgängigen Prozessen und integriert unternehmensweite Stamm- und Transaktionsdaten. SAP NetWeaver® ermöglicht die Umsetzung des Konzepts SAP® Enterprise SOA™. Es können Geschäftsprozesse über Technologiegrenzen hinweg vereinheitlicht, Anwendungen integriert sowie einfach und strukturiert auf Informationen zugegriffen werden. SAP NetWeaver® ist Grundlage für alle SAP®-Lösungen (NW 2011). So nutzt beispielsweise die SAP® Business Suite alle Schlüsselbereiche von SAP NetWeaver® (SAP 2011). Bedeutende Komponenten und Werkzeuge von SAP NetWeaver® sind

- SAP NetWeaver® Application Server
- SAP NetWeaver® Portal
- SAP NetWeaver® Developer Studio
- SAP NetWeaver® Master Data Management (SAP NetWeaver MDM)
- SAP NetWeaver® Composition Environment (SAP NetWeaver CE)
- SAP NetWeaver® Process Integration (SAP NetWeaver PI)

2.1.1 SAP NetWeaver® Application Server

Der SAP NetWeaver® Application Server ist die Plattform sowohl für andere SAP NetWeaver®-Komponenten als auch für ABAP- und Java-Anwendungen. Der SAP NetWeaver® Application Server ist die Weiterentwicklung des SAP® Web Application Servers und besteht aus dem *Application Server ABAP* (AS ABAP) und dem *Application Server Java* (AS Java). Der AS ABAP stellt die Technologiebasis und die Infrastruktur zur Ausführung von ABAP-Anwendungen zur Verfügung. Er bildet die Applikationsschicht der mehrstufigen Architektur eines ABAP-basierten SAP®-Systems und führt während der Ausführung die Kommunikation sowohl mit der Präsentations- als auch mit der Datenbankschicht durch (SAP 2011).

Der Application Server Java stellt eine Java™ 2 Enterprise Edition (J2EE) derzeit mit Version 1.6 konforme Umgebung zur Entwicklung und Ausführung von J2EE - Programmen zur Verfügung. Damit können beispielsweise Web-Benutzeroberflächen entwickelt oder die standardbasierte Entwicklung von Java-Projekten, wie z.B. Webservices oder J2EE-Geschäftsanwendungen, unterstützt werden (SAP 2011).

2.1.2 SAP NetWeaver® Portal

Über SAP NetWeaver® Portal werden die wichtigsten Informationen und Anwendungen zentral in einem Unternehmensportal zusammengeführt und über alle Bereiche einheitlich und personalisiert bereitgestellt. Mit SAP NetWeaver® Portal sollen sowohl SAP®-Lösungen als auch Anwendungen von Drittanbietern, Legacy-Systeme, Datenbanken, unstrukturierte Dokumente, interne und externe Webinhalte sowie Collaboration-Werkzeuge schnell und effektiv integriert werden können. Offene Standards, Webservices und eine enge Integration mit anderen SAP NetWeaver®-Komponenten unterstützen diese Integration. Auch die Anbindung von Java- und .NET-Technologie wird ermöglicht. SAP NetWeaver® Portal bietet außerdem zentrale Dienste für das Wissensmanagement an. Einem Anwender sollen dadurch sowohl die Suche nach als auch die Verwaltung von strukturierten und unstrukturierten Informationen erleichtert werden. Für jeden Anwender kann eine personalisierte und rollenbasierte Konfiguration von Informationen und Anwendungen gemäß seinen fachlichen Anforderungen und Interessen über eine sogenannte Drag&Relate™ Funktionalität durchgeführt werden (NW Portal 2011). Ein Konzept zur Integration einer Anwendung in ein SAP NetWeaver® Portal beschreibt (Dickert 2010).

2.1.3 SAP NetWeaver® Developer Studio

Das SAP NetWeaver® Developer Studio kann wie der SAP NetWeaver® AS Java für die Entwicklung von J2EE-basierten Anwendungen eingesetzt werden. Das SAP NetWeaver® Developer Studio basiert auf Open-Source Eclipse¹, wobei Stan-

¹ <http://www.eclipse.org/>

dardfunktionen von Eclipse um eine Gruppe von Entwurfs-, Konstruktions- und Wartungswerkzeugen erweitert wurden. Alle diese Werkzeugsets sind in Eclipse als Perspektiven integriert. Eine Perspektive steuert in Eclipse die angezeigten Editoren und Sichten. Dies gibt Entwicklern die Möglichkeit, während der Arbeit an verschiedenen Aufgaben einfach zwischen den Perspektiven zu wechseln. Beispielsweise unterstützt die J2EE-Perspektive die Entwicklung und Anwendung der Technologien von J2EE, wie Java Server Pages (JSPs), Servlets und Enterprise Java Beans (EJBs), während die Webservice-Perspektive verschiedene Werkzeuge zur Definition, zur Suche und zum Testen von Webservices kombiniert (NW Dev 2011).

2.1.4 SAP NetWeaver® Master Data Management

Wie bereits in der Einleitung betont, gewinnt die Datenintegration neben der Prozessintegration in den Systemlandschaften der Unternehmen immer mehr an Bedeutung. Mit SAP NetWeaver® Master Data Management (SAP NetWeaver MDM) erfolgt eine zentrale und konsistente Stammdatenverwaltung und -verteilung. Dadurch können allen Anwendungen zur Ausführung von integrierten Geschäftsprozessen aktuelle und konsistente Stammdaten zu Lieferanten, Produkten oder Kunden zur Verfügung gestellt werden. Globale Attribute ermöglichen, dass alle beteiligten Systeme die gleichen Daten und Informationen erhalten. In den Zielsystemen können dann weitere lokale Attribute hinzugefügt werden, z.B. um individuelle regionale oder bereichsbezogene Anforderungen zu erfüllen (NW MDM 2011).

2.1.5 SAP NetWeaver® Composition Environment

SAP NetWeaver® Composition Environment (SAP NetWeaver® CE) stellt ein flexibles Entwicklungswerkzeug für zusammengesetzte Anwendungen (*Composite Applications*) zur Verfügung. Eine *Composite Application* ist eine Java-basierte Anwendung, die aus verschiedenen Webservices unterschiedlicher Quellen, z.B. unterschiedlicher Anwendungssysteme, aufgebaut ist. Durch diese Kombination von Webservices entsteht eine hohe Flexibilität bei der Modellierung und Ausführung von Geschäftsprozessen. Die lose gekoppelten Webservices heterogener Systeme können benutzerorientiert zusammengesetzt werden, um in einem Portal einem Anwender als eine Composite Application zur Verfügung gestellt zu werden. Die Entwicklung dieser zusammengesetzten Anwendungen erfolgt auf Basis des SAP NetWeaver® Developer Studios bzw. des SAP NetWeaver® Application Servers Java. Die Geschäftsprozess-Komposition erfolgt mit SAP NetWeaver® Business Process Management. Auch können Webservices des Enterprise Services Repository bzw. der Services Registry der SAP NetWeaver® Process Integration in die SAP NetWeaver® CE integriert werden (SAP 2011, NW CE 2011).

2.1.6 SAP NetWeaver® Process Integration

Einführung in SAP NetWeaver® Process Integration PI 7.1

Die *SAP NetWeaver® Process Integration (PI) 7.1* (im Weiteren als *PI-System* bezeichnet) ermöglicht prozessorientierte Integration zwischen SAP®- und Nicht-SAP®-Anwendungen innerhalb und über Unternehmensgrenzen hinweg (SAP 2011). Anwendungen sind dabei betriebliche Informationssysteme, z.B. ERP- oder SRM-Systeme. Nachrichten werden zwischen dem PI-System und den Anwendungen über Schnittstellen ausgetauscht. Eine Schnittstelle wird als *Inbound-* oder als *Outbound-Schnittstelle* kategorisiert. Eine Inbound-Schnittstelle liegt vor, wenn über diese eine Nachricht vom PI-System in eine Anwendung versendet werden soll. Eine Outbound-Schnittstelle wird definiert, falls von einer Anwendung eine Nachricht in das PI-System geschickt werden soll. Zwischenbetriebliche Prozesse können über einen standardbasierten Nachrichtenaustausch, z.B. über Verwendung des E-Business-Standards RosettaNet erfolgen. Zur Definition und zur Verwaltung von integrierten Geschäftsprozessen stehen in der PI-Umgebung (Auswahl über Web Einstiegsseite) folgende Komponenten bzw. Werkzeuge zur Verfügung.

- System Landscape Directory (SLD)
- Entwicklungsumgebung (Enterprise Services Repository)
- Laufzeitkomponente (Integration Directory)
- Konfiguration (Configuration and Monitoring)
- Process Integration Administration

Zur Ausführung von integrierten Geschäftsprozessen wird im PI-System der *Integration Server* verwendet. Der Integration Server dient als Host für die Laufzeit-Engines *Integration Engine* und zentrale *Advanced Adapter Engine*. Der Integration Server schließt auch die *Business Process Engine* für mittels BPEL modellierter Szenarien wie das *komponentenübergreifende Business Process Management (ccBPM)* ein. Bei Ausführung von integrierten Geschäftsprozessen liegen Nachrichten der Anwendungen in einem proprietären Format vor, z.B. in einer ABAP-Struktur. Zur Übertragung der Nachricht in ein PI-System muss dieses Format in ein PI-Format des PI-Systems konvertiert werden. Ebenso muss eine Nachricht in einem PI-Format in das proprietäre Format der Anwendung konvertiert werden können. Für die Übersetzung von einem Quellformat in ein PI-Format bzw. von einem PI-Format in ein Zielformat werden Adapter eingesetzt. In PI 7.1 stehen verschiedene Adapter in den Laufzeit-Engines zur Verfügung. Die Integration Engine unterstützt den Nachrichtenaustausch über IDoc-Adapter (ABAP-basiert), XI (Connectivity mit Proxy-Laufzeit), HTTP (ABAP-basiert) sowie die Connectivity mit Systemen oder Anwendungen, die auf Web Services Reliable Messaging (WS-Kanal) basieren. Die Advanced Adapter Engine stellt folgende Adapter bereit.

- RFC-Adapter
- SAP-Business-Connector-Adapter
- File/FTP-Adapter

- JDBC-Adapter
- JMS-Adapter
- SOAP-Adapter
- Mail-Adapter
- CDIX-Adapter
- RNIF-Adapter
- IDoc-Adapter (AAE)
- HTTP-Adapter (AAE)

Der RFC-Adapter dient der Kommunikation mit SAP®-Systemen und basiert auf SAP RFC-Funktionsbausteinen. Der BC-Adapter ermöglicht die Kommunikation mit dem SAP Business Connector, der File/FTP-Adapter den Austausch von Daten bzw. von Dateien mit externen Systemen. Der JDBC-Adapter wird für den Zugriff auf Datenbanken verwendet. Besonders bedeutend ist der SOAP-Adapter, der der Integration von Webservices dient, während der Mail-Adapter den Zugriff auf E-Mail-Server ermöglicht. Der CIDX-Adapter (Chemical Industry Data Exchange) unterstützt den branchenspezifischen Chem eStandard. Für die B2B-Integration wird der RNIF-Adapter (RosettaNet Implementation Framework) verwendet, der die Kommunikation mittels RosettaNet-Standard gewährleistet (Nicolescu 2009).

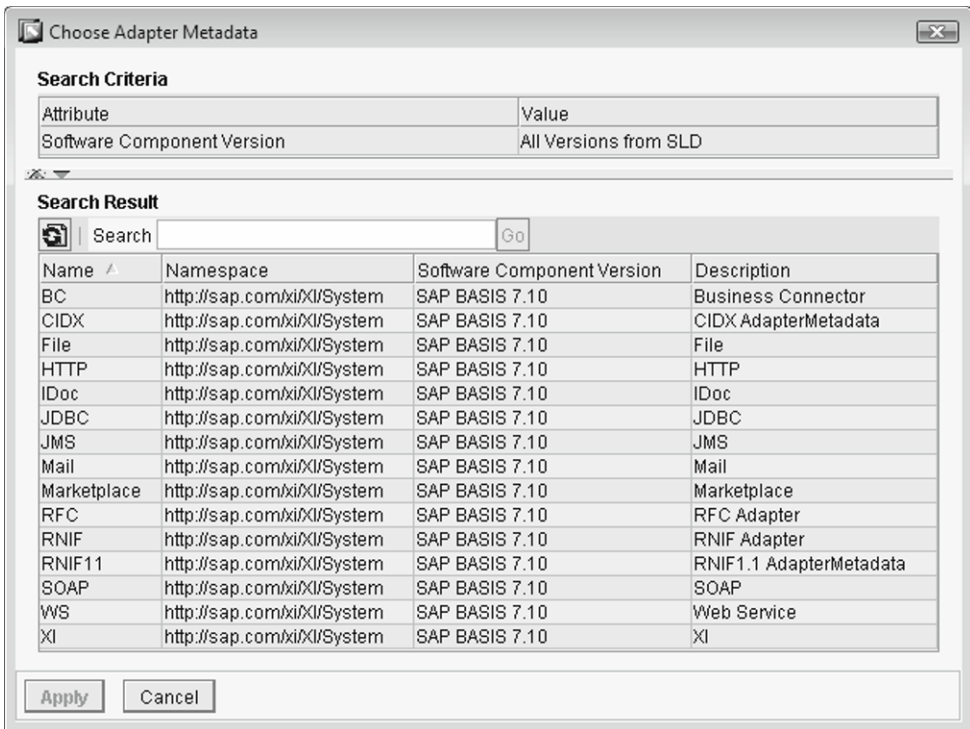


Abbildung 2-1: Suchhilfe für Adapter im Integration Directory

Abbildung 2-1 zeigt verfügbare Adapter im PI-System 7.1. Für Anwendungen, zu denen ein Adapter im Integration Server nicht verfügbar ist, können eigene Adapter entwickelt bzw. erworben werden. Da es sich bei den Adaptern um Standard-J2EE-Komponenten handelt, kann für die Entwicklung auf eine klassische Java-Entwicklungsumgebung zurückgegriffen werden (Banner 2007). Eine Adapter-Entwicklung wird in (Awad 2010) sowie in Abschnitt 12.3 beschrieben. Bei komplexen systemübergreifenden Geschäftsprozessen ist die zustandslose Verarbeitung von Nachrichten über den Integration Server nicht mehr ausreichend. Um dann Nachrichten zueinander in Beziehung zu setzen und komplexere Abläufe über Schleifen abwickeln zu können, werden in der Entwicklungsumgebung sogenannte *Integrationsprozesse* angelegt (Nicolescu 2009). Integrationsprozesse können dann über die *Business Process Engine* systemübergreifend ausgeführt werden.

System Landscape Directory

Die Ermittlung von physischen Adressen von Sender- und Empfängersystemen wird bei einer Middleware als *Routing* bezeichnet. Während bei einer *Point-to-Point Topologie* die Anwendungen die Adressen von Sender- und Empfängersystemen selbst ermitteln müssen, führt dies das PI-System zentral durch. Basis zur Durchführung des Routing in einem PI-System ist das *System Landscape Directory (SLD)*. Im SLD wird die interne Systemlandschaft abgebildet. Dazu gehören einerseits die beteiligten Systeme und ihre Beziehung zueinander (z.B. Konfiguration des Integration Servers), andererseits die in der Systemlandschaft installierten Softwareprodukte (Nicolescu 2009).

Beteiligte Systeme werden in Technische Systeme und (logische) Business-Systeme unterschieden. Ein Technisches System beschreibt ein verwendetes Backend-System, z.B. ein SAP® ERP-System basierend auf einer Datenbank SAP® MaxDB™. Neben SAP®-Systemen können aber auch beliebige andere Systeme im SLD erfasst werden (Nicolescu 2009). Ein Business-System hat eine Referenz auf ein Technisches System, z.B. kann zu einem Mandanten eines SAP®-Systems, das als Technisches System definiert ist, ein Business-System eingerichtet werden. Business-Systeme repräsentieren Anwendungen und werden für die Konfigurationsobjekte des PI-Systems im Integration Directory verwendet. Sie werden dort zur Übertragung von Nachrichten und zur Durchführung des Routing als *Sender- und Empfängersysteme* definiert. Die Trennung in Technische und in Business-Systeme hat den Vorteil, dass Technische Systeme auf einer neuen Hardware installiert werden können, ohne dass sich diese Änderung auf die zugeordneten Business-Systeme auswirkt. Eine Änderung der Konfigurationsobjekte ist daher nicht notwendig (Nicolescu 2009).

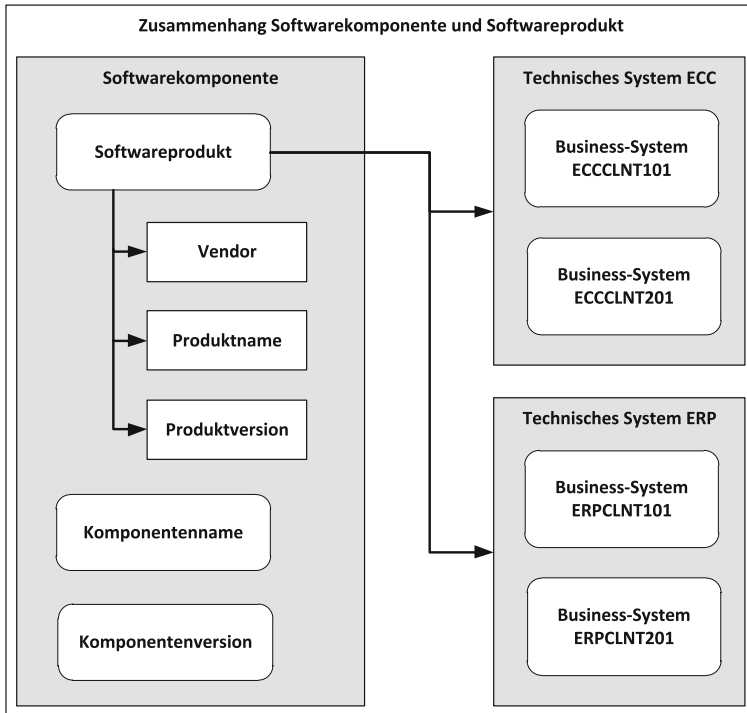


Abbildung 2-2: Softwarekomponente und Softwareprodukt (Kienegger 2010)

Ein *Softwareprodukt* stellt den eigenen Bereich für ein Entwicklungsprojekt dar und muss im Softwarekatalog des SLD angelegt werden. Zu einem Softwareprodukt werden ein sogenannter *Vendor*, der z.B. die Abteilung oder das Projekt repräsentiert, ein Produktname sowie eine Produktversion angegeben. Um im Enterprise Services Repository auf das Softwareprodukt zugreifen zu können, werden eine oder mehrere *Softwarekomponenten* dem Softwareprodukt zugeordnet. Jede dieser Komponente kann wiederum mehrere *Softwarekomponentenversionen* enthalten (siehe Abbildung 2-2). Um eine Verbindung zwischen einem Softwareprodukt und einer Softwarekomponente herstellen zu können, ist die Angabe einer *Software Unit* notwendig. Zu einer Softwarekomponente werden ebenfalls der Vendor und die Versionsnummer eingetragen (Kienegger 2010). Um mit Systemen des SLD kommunizieren zu können, muss ein Softwareprodukt den benötigten Technischen Systemen und Business-Systemen zugeordnet werden. Durch diese Zuordnung im SLD können dann die Softwarekomponenten dieses Softwareprodukts Objekte aus Business-Systemen importieren, Nachrichten in Business-Systeme verschicken oder Nachrichten aus Business-Systemen empfangen.

Entwicklungsumgebung im Enterprise Services Repository

Im Enterprise Services Repository (ES Repository) stehen die beiden Werkzeuge Enterprise Services Builder (ES Builder) und Services Registry zur Verfügung. Im Enterprise Services Builder werden logische Entwicklungsobjekte erzeugt, wie z.B. die Schnittstellen zum Austausch von Nachrichten oder die Regeln für die Durchführung des Mapping von Datenstrukturen. Weiterhin kann die Definition von Integrationsszenarien komplexer und verteilter sowohl innerbetrieblicher als auch zwischenbetrieblicher Geschäftsprozesse in einem Prozessmodell über einen grafischen Prozess-Editor (BPEL-Prozessdesigner) erfolgen. Entwicklungsobjekte werden im Enterprise Services Builder in einem eigenen Entwicklungsbereich angelegt. Dieser Bereich wird als *Softwarekomponente* bezeichnet und wurde im Abschnitt *System Landscape Directory* bereits erläutert. Die verschiedenen *Softwarekomponentenversionen* der Softwarekomponente können aus dem SLD in den ES Builder über den Bereich *Work Areas* importiert werden. Dazu sollte zusätzlich die Angabe eines Namensraums in Form einer http-Adresse erfolgen, um aus Business-Systemen importierte Entwicklungsobjekte zwischen Softwarekomponentenversionen unterscheiden zu können. Nach erfolgreichem Import der Softwarekomponentenversion können Entwicklungsobjekte folgender Kategorien erzeugt werden.

- Interface-Objekte
- Mapping-Objekte
- Adapter-Objekte
- Integrationsprozesse und Prozessintegrationsszenarien
- Modellierungen

Datentypen, Message-Typen und Service-Interfaces werden den Interface-Objekten zugeordnet. Mapping-Objekte werden unterteilt in Message-Mapping und Operation-Mapping. Ein Integrationsprozess ist in ein Prozessintegrationsszenario eingebettet.

Datentyp und Message-Typ

Basis einer Schnittstelle ist ein Datentyp, der die Struktur einer Nachricht in Form eines XSD-Dokuments beschreibt. Dazu werden Elementtypen des XML-Schemas wie *String*, *Integer*, *Boolean*, usw. für die Definition von Elementen eines Datentyps verwendet. Es können sowohl einfache als auch komplexe und geschachtelte Datentypen definiert sowie eine Kardinalität zu jedem Element angegeben werden. Ein Beispiel für einen Datentyp im ES-Repository zeigt Abbildung 2-3.

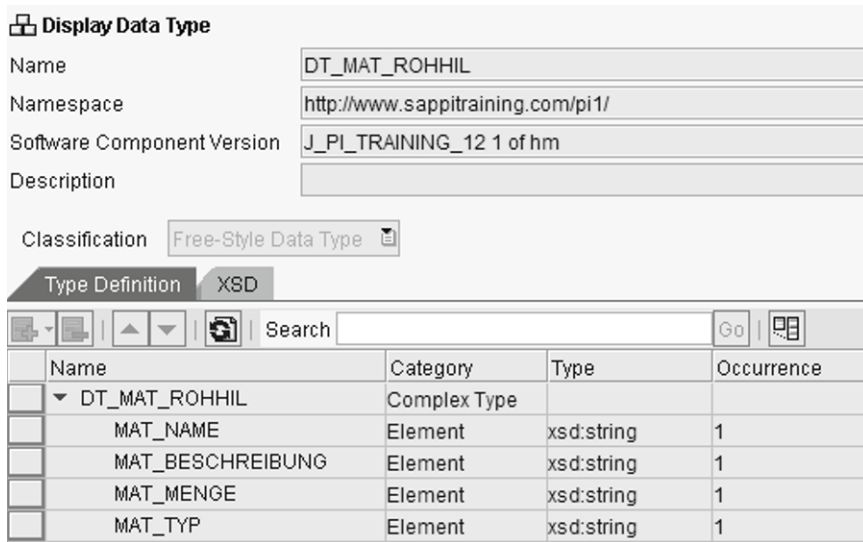


Abbildung 2-3: Beispiel für einen Datentyp

Datentypen im PI-Umfeld werden nur zur Modularisierung von Datenformaten verwendet und können nicht in einer Schnittstelle bzw. in einem Mapping verwendet werden. Daher werden Datentypen in Message-Typen eingebettet (Nicolescu 2009). Ein Message-Typ beschreibt das genaue Datenformat einer Schnittstelle durch Referenz auf genau einen Datentyp. Message-Typen können bei einer Schnittstelle als *Input-Message-Typ* für den Eingang einer Nachricht in eine Anwendung oder als *Output-Message-Typ* für den Ausgang einer Nachricht aus einer Anwendung verwendet werden. Ein Message-Typ zum Datentyp aus Abbildung 2-3 ist in Abbildung 2-4 dargestellt.

Als Alternative zur manuellen Anlage eines Message-Typs kann ein Message-Schema von einer externen Quelle importiert werden. Eine externe Quelle kann z.B. folgende URL des SAP® Webservice Repository zu einem SAP®-System sein.

<http://host:port/sap/bc/bsp/sap/webservicebrowser/search.html?sap-client=mandant>

Beim Import wird dann eine *externe Definition* in Form einer WSDL-Datei als Entwicklungsobjekt automatisch angelegt. Diese referenziert dann auf ein Message-Schema, z.B. auf einen BAPI®. Eine externe Quelle kann aber auch eine UDDI Registry² sein, die Webservices in Form von WSDL-Dateien zur Verfügung stellt. Bei Webservices erfolgt ebenfalls ein Import von Message-Schemata, die z.B. als Request-, als Response- oder als Fault-Message in Operationen von Service-Interfaces

² Siehe z.B.: <http://www.webservices.net> oder <http://www.cdyne.com>

verwendet werden können (SAP 2011). Ab SAP® ECC 6.0 steht auch eine Entwicklungsmöglichkeit für Webservices basierend auf einem RFC-Funktionsbaustein in der ABAP Workbench zur Verfügung. Alle diese genannten Webservices können in der Services Registry des PI-Systems zentral registriert werden.

Edit Message Type

Name: MT_MAT_ROHHIL

Namespace: http://www.sappitraining.com/pi1/

Software Component Version: J_PI_TRAINING_12 1 of hm

Description:

Data Type Used: DT_MAT_ROHHIL

XML Namespace: http://www.sappitraining.com/pi1/

Structure XSD

Name	Category	Type	Occur
MT_MAT_ROHHIL	Element	DT_MAT_ROHHIL	
MAT_NAME	Element	xsd:string	1
MAT_BESCHREIBUNG	Element	xsd:string	1
MAT_MENGE	Element	xsd:string	1
MAT_TYP	Element	xsd:string	1

Abbildung 2-4: Beispiel für einen Message-Typ

Service-Interface

Eine Schnittstelle in PI 7.1 wird als *Service-Interface* bezeichnet. Zu einem *Service-Interface* werden ein Name und ein Namensraum angegeben. Zu einer Schnittstelle erfolgt die Angabe der Kategorie durch Auswahl von *Inbound*, *Outbound* oder *Abstract*. Während Outbound- oder Inbound-Schnittstellen über eine implementierte Schnittstelle in einer Anwendung als Gegenstück verfügen, werden abstrakte Service-Interfaces nur von Integrationsprozessen zum Senden oder Empfangen von Nachrichten benutzt. Ein abstraktes Service-Interface hat keine von vornherein definierte Richtung und kann sowohl für das Senden als auch für das Empfangen von Nachrichten verwendet werden (SAP 2011). Einem Service-Interface können eine oder mehrere Operationen jeweils mit Angabe eines Modus zugeordnet werden. Der Modus einer Operation gibt an, ob zu einer Nachricht eine Antwort erwartet wird (*synchron*) oder nicht (*asynchron*). Über die verwendete Operation wird dadurch der Kommunikationsmodus der Schnittstelle definiert. Eine Operation referenziert auf einen oder auf mehrere Message-Typen bzw. Message-Schemata. Zu einer asynchronen Inbound-Schnittstelle wird der Operation ein Input-Message-Typ als Request und zu einer asynchronen Outbound-Schnittstelle ein

Output-Message-Typ als Request zugeordnet. Bei einer synchronen Schnittstelle werden zur Operation ein Input-Message-Typ und ein Output-Message-Typ angegeben. Dabei werden anhand der Kategorie der Schnittstelle die Message-Typen des Request und der Response bestimmt. Bei einer Outbound-Schnittstelle wird dem Request der Output-Message-Typ, bei einer Inbound-Schnittstelle der Input-Message-Typ zugeordnet.

Außerdem kann zur Operation einer synchronen Schnittstelle oder einer asynchronen Inbound-Schnittstelle ein Fault-Message-Typ definiert werden. Über den Fault-Message-Typ können Verarbeitungsinformationen, z.B. eine fehlerhafte Bearbeitung, aus dem PI-System gesendet werden (Nicolescu 2009). Ein Beispiel für ein abstraktes asynchrones Service-Interface zum Message-Typ aus Abbildung 2-4 zeigt Abbildung 2-5.

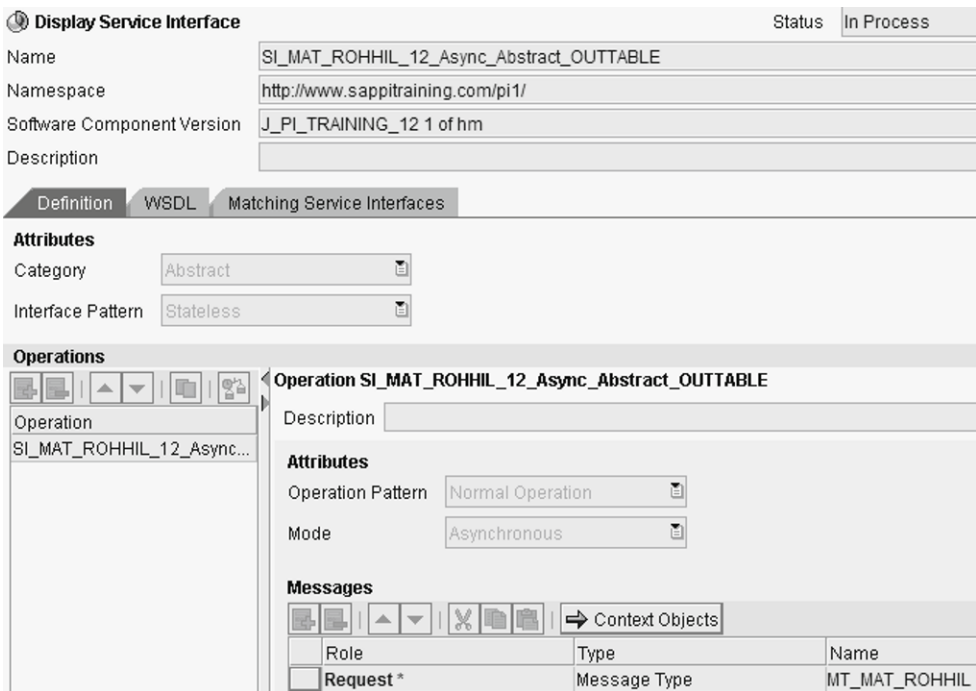


Abbildung 2-5: Abstraktes asynchrones Service-Interface

Alternativ zur manuellen Anlage eines Service-Interface können auch RFC- und IDoc-Schnittstellen, die auf RFC-Funktionsbausteinen bzw. SAP® IDocs³ basieren, aus einem SAP®-System, das mindestens auf einem Application Server ABAP 6.20

³ Abkürzung für *Intermediate Document*.

basiert, in das ES Repository importiert werden. Importierte Schnittstellen sind dann auf der gleichen funktionalen Ebene wie Operationen, daher können diese Typen von Schnittstellen im ES Builder oder im Integration Builder an Stellen referenziert werden, an denen auch eine Operation eines Service-Interface referenziert werden kann (SAP 2011) (siehe Abschnitt *Operation-Mapping*).

Für die in Abschnitt *Datentyp und Message-Typ* beschriebenen externen Definitionen, die in Form einer WSDL-Datei importiert wurden, muss explizit ein synchrones Service-Interface mit Angabe der Kategorie angelegt werden. Importierte Schnittstellen und externe Definitionen referenzieren nicht auf einen Message-Typ, sondern auf ein Message-Schema. In Abbildung 2-6 ist der Bezug von Service-Interfaces, externen Definitionen und importierten Schnittstellen zu Operations dargestellt (modifizierte Form aus (SAP 2011)).

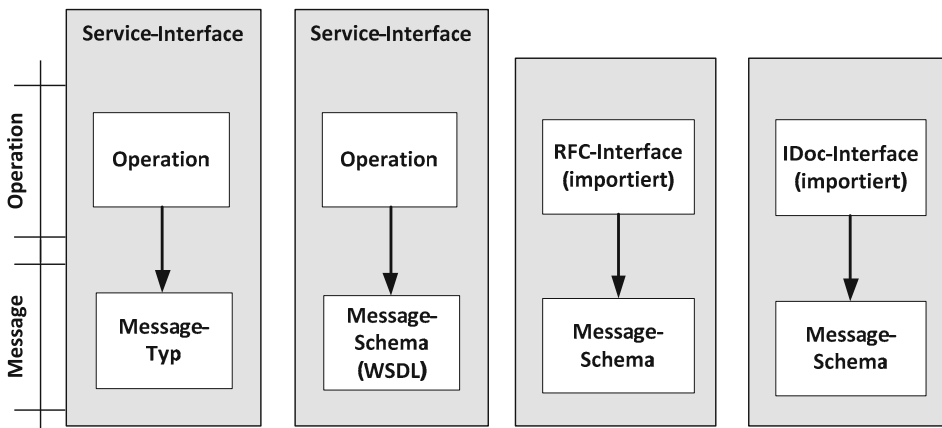


Abbildung 2-6: Bezug Service-Interfaces und importierte Schnittstellen zu Operations

Message-Mapping

Da die verschiedenen Schnittstellen unterschiedliche Datentypen enthalten, müssen die einzelnen Elemente des Quellformats einer Senderschnittstelle den Elementen des Quellformats der Empfängerschnittstelle zugeordnet werden. Diese Definition der Zuordnung von Elementen wird *Mapping* genannt und stellt eine der wesentlichen Entwicklungsaufgaben innerhalb der Prozessintegration dar. Durch Transformationswerkzeuge erfolgt ein Mapping unter Beachtung der jeweiligen Syntax und Semantik, der definierten Transformationsregeln und gegebenenfalls durch Verwendung standardisierter Austauschformate. Beispielsweise erfolgt ein Mapping vom proprietären SAP® ABAP-Format auf ein SAP® XML-Format bei einer A2A-Integration oder vom proprietären SAP® ABAP-Format auf das Standardformat von RosettaNet bei einer B2B-Integration. Für die Durchführung des Mapping zwischen unterschiedlichen XML-Formaten bzw. XML-Datentypen kann XSLT (*Extensible Stylesheet Language Transformation*) verwendet werden. Beim