
Universität Bamberg
Lehrstuhl Prof. Otto K. Ferstl
Lehrstuhl Prof. Elmar J. Sinz

Martin Schissler, Stephan Mantel,
Otto K. Ferstl, Elmar J. Sinz

Unterstützung von Kopplungsarchitekturen
durch SAP R/3

FORWIN-Bericht-Nr.: FWN-2001-008

© FORWIN - Bayerischer Forschungsverbund Wirtschaftsinformatik,
Bamberg, Bayreuth, Erlangen-Nürnberg, Regensburg, Würzburg 2001

Alle Rechte vorbehalten. Insbesondere ist die Überführung in maschinenlesbare Form sowie das Speichern in Informationssystemen, auch auszugsweise, nur mit schriftlicher Einwilligung von FORWIN gestattet.

Zusammenfassung

Die ganzheitliche Gestaltung von Wertschöpfungsketten erfordert eine überbetriebliche, die Grenzen einzelner Unternehmen übergreifende Sichtweise. Die Steuerung dieser überbetrieblichen Wertschöpfungsketten setzt flexible und umfassende Kopplungen zwischen den Anwendungssystemen der beteiligten Unternehmen voraus. In der vorliegenden Arbeit wird die Standardanwendungssoftware SAP R/3 Release 4.5B hinsichtlich der Unterstützung von Kopplungsarchitekturen untersucht. Es werden insgesamt vier Kopplungsmechanismen vorgestellt, die als Basis für die Realisierung von Kopplungsarchitekturen dienen können. Die Mechanismen werden detailliert beschrieben und hinsichtlich ihrer Eignung zur Umsetzung von verschiedenen Integrationskonzepten bewertet.

Stichworte

Kopplungsarchitektur, Kopplungsmechanismus, Integrationskonzept, SAP R/3, BAPI, IDoc, ALE, ITS, SAP Business Connector

Abstract

The holistic design of value chains requires an intercompany view, which goes beyond the scope of enterprise boundaries. The management of these intercompany value chains needs flexible and extensive coupling of the heterogeneous application systems of the involved companies. In this report we analyze the ERP Solution SAP R/3 Release 4.5B with regard to its support of coupling architectures. Four coupling mechanisms are presented, which can be used to implement coupling architectures. The mechanisms are described in detail and are evaluated with regard to their applicability to the implementation of integration concepts.

Keywords

Coupling architecture, coupling mechanism, integration concept, SAP R/3, BAPI, IDoc, ALE, ITS, SAP Business Connector

Inhalt

| | | |
|----------|---|----------|
| 1 | EINFÜHRUNG | 1 |
| 2 | KOPPLUNGSARCHITEKUR UND KOPPLUNGSMECHANISMUS | 2 |
| 3 | INTEGRATIONSMERKMALE UND -KONZEPTE | 4 |
| 3.1 | INTEGRATIONSMERKMALE..... | 4 |
| 3.2 | INTEGRATIONSKONZEPTE | 5 |
| 4 | DIE ARCHITEKTUR VON SAP R/3 | 7 |
| 5 | KOPPLUNGSMECHANISMEN VON SAP R/3 | 9 |
| 5.1 | BAPI (BUSINESS APPLICATION PROGRAMMING INTERFACE)..... | 10 |
| 5.1.1 | Ziele des Kopplungsmechanismus | 10 |
| 5.1.2 | Schnittstellensicht des Mechanismus | 10 |
| 5.1.3 | Innensicht des Mechanismus | 13 |
| 5.1.4 | Unterstütztes Integrationskonzept | 15 |
| 5.2 | IDOC (INTERMEDIATE DOCUMENTS) | 17 |
| 5.2.1 | Ziele des Kopplungsmechanismus | 18 |
| 5.2.2 | Schnittstellensicht des Mechanismus | 18 |
| 5.2.3 | Innensicht des Mechanismus | 21 |
| 5.2.4 | Unterstütztes Integrationskonzept | 24 |
| 5.3 | ALE (APPLICATION LINK ENABLING) | 26 |
| 5.3.1 | Ziele des Kopplungsmechanismus | 26 |
| 5.3.2 | Schnittstellensicht des Mechanismus | 26 |
| 5.3.3 | Innensicht des Mechanismus | 28 |
| 5.3.4 | Unterstütztes Integrationskonzept | 30 |

| | | |
|----------|---|-----------|
| 5.4 | ITS (INTERNET TRANSACTION SERVER)..... | 31 |
| 5.4.1 | Ziele des Kopplungsmechanismus | 31 |
| 5.4.2 | Schnittstellensicht des Mechanismus | 32 |
| 5.4.3 | Innensicht des Mechanismus | 33 |
| 5.4.4 | Unterstützes Integrationskonzept | 36 |
| 5.5 | SAP BUSINESS CONNECTOR..... | 37 |
| 6 | ZUSAMMENFASSUNG UND FAZIT..... | 40 |
| 7 | LITERATUR | 42 |

Abkürzungsverzeichnis

| | |
|---------|--|
| ABAP | Advanced Business Application Programming Language |
| ALE | Application Link Enabling |
| AwS | Anwendungssystem |
| BAPI | Business Application Programming Interface |
| BFA | Business Framework Architecture |
| CORBA | Common Object Request Broker Architecture |
| DCOM | Distributed Component Object Model |
| DIAG | Dynamisches Informations- und Aktions-Gateway |
| Dynpro | Dynamisches Programm |
| EDI | Electronic Data Interchange |
| EDIFACT | Electronic Data Interchange for Administration, Commerce and Transport |
| FB | Funktionsbaustein |
| HTML | Hypertext Markup Language |
| HTTP | Hypertext Transfer Protocol |
| HTTPS | Hypertext Transfer Protocol Secure |
| IAC | Internet Application Component |
| IfR | Interface Repository |
| IDoc | Intermediate Document |
| IP | Internet Protocol |
| ITS | Internet Transaction Server |
| RFC | Remote Function Call |
| RPC | Remote Procedure Call |
| TCP | Transmission Control Protocol |
| tRFC | transaktionaler Remote Function Call |

| | |
|-----|----------------------------|
| URL | Uniform Resource Locator |
| WAS | Web-Application-Server |
| WWW | World Wide Web |
| XML | Extensible Markup Language |

1 Einführung

Die ganzheitliche Gestaltung von Wertschöpfungsketten erfordert eine überbetriebliche, die Grenzen einzelner Unternehmen übergreifende Sichtweise. Die Steuerung dieser überbetrieblichen Wertschöpfungsketten setzt flexible und umfassende Kopplungen zwischen den Anwendungssystemen (AwSen) der beteiligten Unternehmen voraus. In diesem Beitrag werden Kopplungen unter dem Gesichtspunkt der Integration der beteiligten AwSe untersucht. Diese Integration kann auf unterschiedliche Weise erfolgen. Besondere Verwendung finden z.B. die Integrationskonzepte (vgl. [Fers92] und [FeSi01, 220ff])

- aufgabenträgerorientierte Funktionsintegration,
- datenflussorientierte Funktionsintegration,
- Datenintegration und
- Objektintegration.

Ein Integrationskonzept gibt eine Strategie für die Integration von AwSen vor. Die konkrete Umsetzung einer solchen Strategie erfolgt in Form einer Kopplungsarchitektur. In der vorliegenden Arbeit wird die Standardanwendungssoftware SAP R/3 Release 4.5B hinsichtlich der von ihr unterstützten Kopplungsarchitekturen untersucht. Die Betrachtung beschränkt sich auf die Integration eines R/3-Systems mit einem anderen AwS (Inter-AwS-Integration). Die Integration innerhalb einer R/3-Systems wird nicht betrachtet (Intra-AwS-Integration).

In Kapitel 2 werden zunächst die für diese Arbeit zentralen Begriffe Kopplungsarchitektur und Kopplungsmechanismus eingeführt. Anschließend werden in Kapitel 3 verschiedene Integrationskonzepte vorgestellt. In Kapitel 4 widmen wir uns dann zum ersten Mal SAP R/3 in Form einer knappen Beschreibung der Architektur eines R/3-Systems. Dieses Kapitel vermittelt die Grundlagen zum Verständnis von Kapitel 5, das den Kern der vorliegenden Arbeit bildet. Hier werden verschiedene von R/3 angebotene Kopplungsmechanismen vorgestellt und detailliert beschrieben. Zur Strukturierung der Beschreibung werden zwei Sichten, die Schnittstellensicht und die Innensicht, unterschieden. In der Schnittstellensicht liegt der Fokus auf der Schnittstelle zwischen den integrierten AwSen. Die beteiligten AwSe werden dabei nur in der Außensicht betrachtet. Es werden vor allem die Kopplungspartner und die ausgetauschten Objekte beschrieben. In der Innensicht-Betrachtung werden

anschließend die Mechanismen innerhalb der AwSe dargestellt. Für jeden der Kopplungsmechanismen wird untersucht, welches Integrationskonzept von diesem primär unterstützt wird. Kapitel 6 schließt die Arbeit mit einer Zusammenfassung und einem Fazit ab.

2 Kopplungsarchitektur und Kopplungsmechanismus

Gegenstand dieser Arbeit ist die Untersuchung verschiedener Kopplungsmechanismen, die durch die Standardanwendungssoftware SAP R/3 angeboten werden. In diesem Kapitel wird daher zunächst der Begriff des Kopplungsmechanismus eingeführt.

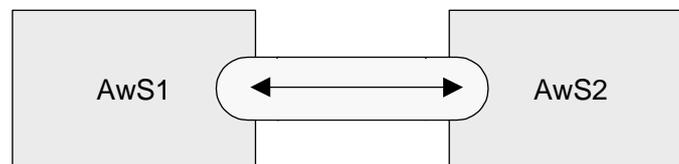


Abbildung 1: Kopplungsarchitektur zur Integration von zwei AwSen

Ein Kopplungsmechanismus ist ein Basismechanismus, der bei der Implementierung von Kopplungsarchitekturen genutzt wird. Eine **Kopplungsarchitektur** ist eine Projektion auf die AwS-Architekturen von zwei oder mehreren integrierten AwSen (vgl. Abbildung 1). Sie beschreibt in Form eines Bauplans alle für die Integration relevanten Elemente sowie die Beziehungen zwischen diesen Elementen.¹ Ausgangspunkt für die Erstellung einer Kopplungsarchitektur ist ein bestimmtes **Integrationskonzept**, das eine Strategie für die Integration von AwSen vorgibt. In [Fers92] und [FeSi01, 220ff] wird z.B. zwischen den Integrationskonzepten

- aufgabenträgerorientierte Funktionsintegration,
- datenflussorientierte Funktionsintegration,
- Datenintegration und
- Objektintegration

unterschieden.

¹ vgl. zum hier zu Grunde gelegten Architekturbegriff [Sin97]

| | | Phase | |
|------------|----------------------|---------------------------|---------------------------|
| | | Entwurf | Implementierung |
| Gegenstand | fachlich | Kopplungs- architektur | Kopplungs- mechanismus |
| | system- technisch | | |

Abbildung 2: Dimensionen der AwS-Entwicklung

In Abbildung 2 werden die beiden Begriffe Kopplungsarchitektur und Kopplungsmechanismus in die **Dimensionen der AwS-Entwicklung** eingeordnet. Die **Dimension Phase** nimmt Bezug auf das phasenorientierte Vorgehen bei der Entwicklung eines AwSs. Es werden in der Regel die sechs Phasen

- Planungsphase,
- Definitionsphase,
- Entwurfsphase,
- Implementierungsphase,
- Abnahme- und Einführungsphase
- sowie Wartungs- und Pflegephase

unterschieden (vgl. z.B. [Balz96]).

Wir beschränken uns in Abbildung 2 auf die Betrachtung der Entwurfs- und der Implementierungsphase. In der Entwurfsphase wird ausgehend von den Ergebnissen der Definitionsphase die Architektur eines AwSs festgelegt. Hierbei wird auch vom „Programmieren im Großen“ gesprochen (vgl. z.B. [Balz96, 632]). In der Implementierungsphase erfolgt anschließend das „Programmieren im Kleinen“. Dies umfasst vor allem die Konzeption von Datenstrukturen und Algorithmen sowie die Umsetzung in die Konstrukte der gewählten Programmiersprache (vgl. z.B. [Balz96, 926f]).

Die zweite in Abbildung 2 betrachtete **Dimension** bezieht sich auf den **Gegenstand** der AwS-Entwicklung und unterscheidet zwischen fachlichen und systemtechnischen Gegenständen. Die Entwurfsphase und somit auch die Implementierungsphase beschäftigt sich sowohl mit

technischen als auch mit fachlichen Aspekten eines AwSs. Ein fachlicher Aspekt kann z.B. ein Algorithmus zur Preisfindung sein. Ein Algorithmus, der den Zustand eines Objektes in einer relationalen Datenbank ablegt, ist hingegen vollständig der systemtechnischen Ebene zuzuordnen.

Die betrachteten **Kopplungsmechanismen** werden der Implementierungsphase zugeordnet. Sie bilden die Basis für die Implementierung einer konkreten Kopplungsarchitektur. Im Gegensatz zu einfacheren Basismechanismen, wie z.B. einem RPC (Remote Procedure Call), handelt es sich bei einem Kopplungsmechanismus um einen umfangreichen Mechanismus, der typischerweise sowohl systemtechnische als auch fachliche Aspekte abdeckt und einen engen Rahmen für die Entwicklung einer Kopplungsarchitektur vorgibt. Die Vorgabe eines Rahmens kann z.B. durch Definition von Platzhaltern erfolgen, die in der Entwurfsphase auszufüllen sind. Ein Kopplungsmechanismus implementiert bereits wesentliche Teile einer Kopplungsarchitektur. Im Idealfall deckt er bereits alle systemtechnischen Aspekte ab, so dass bei Entwurf und Implementierung einer Kopplungsarchitektur eine Konzentration auf fachliche Aspekte möglich ist.

3 Integrationsmerkmale und -konzepte

3.1 Integrationsmerkmale

Die Aufgaben eines AwSs bzw. im Falle einer anwendungssystemübergreifenden Betrachtung die Aufgaben der gekoppelten AwSe sind das Ergebnis einer Meta-Gestaltungsaufgabe, die als Automatisierungs- und Integrationsaufgabe bezeichnet wird (vgl. [Fers92]). Die Notwendigkeit zur überbetrieblichen Kopplung von AwSen resultiert z.B. aus Informationsbeziehungen zwischen automatisierten Aufgaben verschiedener Unternehmen.

Das Aufgabenobjekt der Gestaltungsaufgabe sind die zu integrierenden Aufgaben des Gesamtsystems. Sachziel der Gestaltungsaufgabe ist die Automatisierung der Aufgaben. Spezielle Formalziele dieser Meta-Aufgabe nehmen auf die Beziehungen zwischen den Aufgaben Bezug, sie werden als **Integrationsziele** bezeichnet [FeSi01, 215f]. Beziehungen zwischen Teilaufgaben eines AwSs sind Gegenstand der Intra-AwS-Integration des jeweiligen AwSs. Beziehungen zwischen Teilaufgaben mehrerer AwSe sind Gegenstand der Inter-AwS-Integration zwischen den beteiligten AwSen.

Die Integrationsziele nehmen auf ausgewählte Struktur- und Verhaltensmerkmale von AwSen, die **Integrationsmerkmale**, Bezug und legen Zielausprägungen der Integrationsmerkmale fest (vgl. Abbildung 3). Für eine detaillierte Beschreibung der verschiedenen Integrationsmerkmale soll hier auf [Fers92] und [FeSi01, 217ff] verwiesen werden.

| <i>Merkmalsgruppe</i> | | <i>Ziel: Einhaltung einer vorgegebenen Ausprägung des Merkmals...</i> |
|-----------------------|------------------|---|
| Struktur | Redundanz | Datenredundanz (DR) |
| | | Funktionsredundanz (FR) |
| | Verknüpfung | Kommunikationsstruktur (KS) |
| Verhalten | Konsistenz | Semantische Integrität (SI) |
| | | Operationale Integrität (OI) |
| | Zielorientierung | Vorgangssteuerung (VS) |

Abbildung 3: Integrationsmerkmale

3.2 Integrationskonzepte

Für die Durchführung der im vorherigen Abschnitt angesprochenen Meta-Gestaltungsaufgabe stehen unterschiedliche Methoden zur Verfügung. Ein Integrationskonzept gibt eine Strategie für die Umsetzung der Integrationsziele vor. Häufig genutzte Integrationskonzepte sind die Funktionsintegration, die Datenintegration und die Objektintegration. Sie erreichen bezüglich der Integrationsziele sehr unterschiedliche Zielerreichungsgrade und werden im Folgenden kurz vorgestellt. Eine detailliertere Betrachtung der verschiedenen Konzepte findet sich in [Fers92] und [FeSi01, 220ff].

Aufgabenträgerorientierte Funktionsintegration

Dieses Integrationskonzept ist auf teilautomatisierte Aufgaben ausgerichtet [FeSi01, 221]. Die Integration der Teilaufgaben erfolgt hierbei nicht durch Kopplung der zugehörigen AwSe sondern durch einen gemeinsamen personellen Aufgabenträger.

Datenflussorientierte Funktionsintegration

Bei der datenflussorientierten Funktionsintegration wird ein AwS als ein Netzwerk von Komponenten betrachtet, die Datenflüsse austauschen (vgl. [FeSi01, 222f]). Der Austausch der Datenflüsse erfolgt im Allgemeinen ohne Transaktionsbezug, d.h. die Datenpakete eines Flusses werden nicht notwendigerweise für jede einzelne Geschäftstransaktion ausgetauscht.

Bei der Inter-AwS-Integration wird dementsprechend auf die anwendungssystemübergreifende Vernetzung von Komponenten fokussiert. Das in der Praxis mittlerweile recht weit verbreitete EDI-Konzept, das einen automatisierten Austausch von Geschäftsdokumenten anstrebt, ist ein typisches Beispiel für eine datenflussorientierte Funktionsintegration (vgl. zu EDI z.B. [Deut95] und [Ball00]).

Die datenflussorientierte Funktionsintegration verfolgt primär das Integrationsziel bezüglich der Kommunikationsstruktur, also die Bildung eines Kommunikationssystems, das (1) die Nachrichten zwischen den beteiligten Systemkomponenten überträgt und (2) Informationen bezüglich der Kommunikationsstruktur bereitstellt. Zur Unterstützung des Integrationsziels Vorgangssteuerung liegen Teilansätze vor, z.B. in Form von Monitoren zur Überwachung von Vorgangsketten.

Datenintegration

Bei der Datenintegration operieren Funktionen über einer als externe Sicht (View) bezeichneten Teilmenge eines konzeptuellen Datenschemas. Das konzeptuelle Datenschema und die externen Sichten dienen dabei sowohl der Datenhaltung von Aufgabenobjekten als auch der Kommunikation zwischen Funktionen. Diese Kommunikation über externe Sichten stellt eine enge Kopplung der Funktionen dar [FeSi01, 223ff].

Das Integrationskonzept Datenintegration wird in SAP R/3 nur zur Intra-AwS-Integration nicht aber zur Inter-AwS-Integration verwendet. Aus diesem Grund wird die Datenintegration im Folgenden nicht weiter betrachtet.

Objektintegration

AwSe können objektorientiert aus konzeptuellen Objekten und Vorgangsobjekten zusammengesetzt werden. Konzeptuelle Objekte und Vorgangsobjekte sind untereinander lose gekoppelt und tauschen Nachrichten über Kommunikationskanäle aus. Konzeptuelle

Objekte erfassen und manipulieren in ihrem internen Speicher Zustände der realen Welt, d.h. sie bilden ein Modell des betrachteten Ausschnitts der realen Welt. Vorgangsobjekte führen Aufgaben durch und nutzen dazu die Dienste der konzeptuellen Objekte [FeSi01, 227].

Bezüglich der Datensicht korrespondiert der Verbund der konzeptuellen Objekte von AwSen weitgehend mit einem konzeptuellen Datenschema. Die Zielerreichungsgrade bezüglich der Integrationsziele Datenredundanz und Integrität sind daher vergleichbar. Zur Sicherung der semantischen Integrität wird z.B. die Kommunikation mit Hilfe von Nachrichten genutzt. Die Objektintegration berücksichtigt auch das Integrationsmerkmal Funktionsredundanz, das die mehrfache Realisierung von Funktionen erfasst. Ihrer Vermeidung dienen das Konzept der Generalisierung von Funktionen und die Nutzung von Vererbungsbeziehungen [FeSi01, 228].

Bei der Inter-AwS-Integration ist eine anwendungssystemübergreifende Vererbung im Allgemeinen wenig geeignet, da sie zu einer starken Abhängigkeit der integrierten AwSe in Bezug auf die Weiterentwicklung führt. Dies gilt insbesondere für den Fall der überbetrieblichen Integration. Sinnvoller ist hier die Delegation von Funktionen. Die Kommunikation aller Objekte erfolgt über ein eigenes Kommunikationssystem, das auch eine Kontrolle der Kommunikationsstruktur ermöglicht.

4 Die Architektur von SAP R/3

In diesem Kapitel wird ein knapper Überblick über die Architektur von SAP R/3 gegeben, der zum Verständnis des folgenden Kapitels hilfreich ist. Bei einem R/3-System handelt es sich um ein 3-stufiges Client/Server-System, bestehend aus einer Präsentationsschicht, einer Applikationsschicht und einer Datenbankschicht (vgl. [SAP99]).

Die **Präsentationsschicht** besteht aus einer beliebigen Anzahl an **SAPGUI-Komponenten**, die die Schnittstelle zum Benutzer bilden (vgl. [SAP99]). Den SAPGUI-Komponenten kommen zwei Aufgaben zu:

- Entgegennahme von Benutzerinteraktionen und Übermittlung dieser Interaktionen an die Applikationsschicht.
- Entgegennahme von Verarbeitungsergebnissen von der Applikationsschicht und Darstellung dieser Ergebnisse auf dem Bildschirm eines Benutzers.

Die Präsentationsschicht enthält keinerlei Anwendungslogik. Diese ist vollständig in der Applikationsschicht abgelegt.

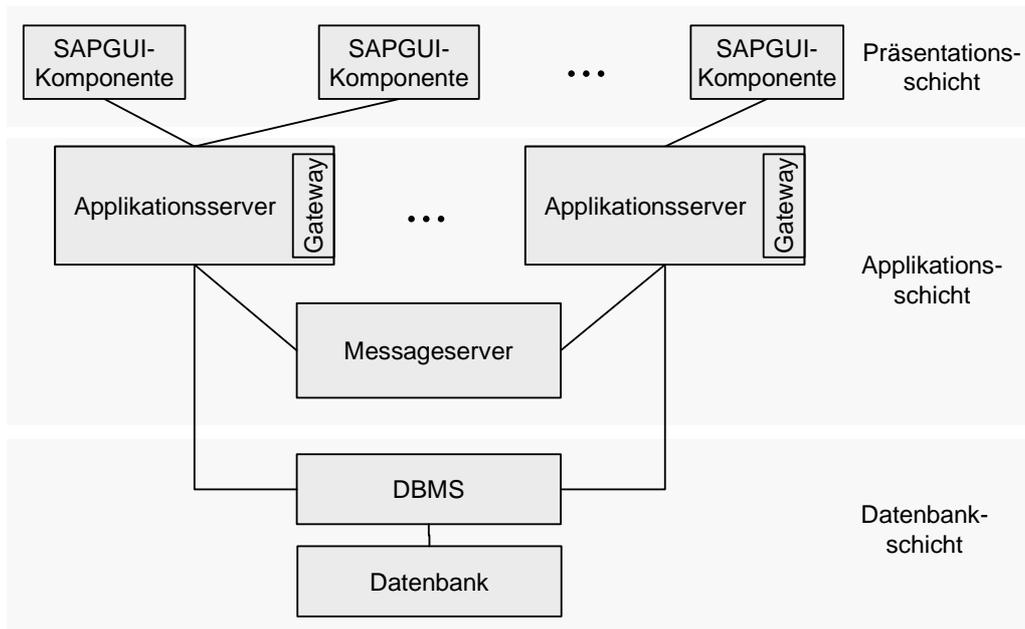


Abbildung 4: Architektur eines R/3-Systems (vgl. [SAP99])

Die **Applikationsschicht** umfasst einen oder mehrere Applikationsserver und einen Messageserver. Die **Applikationsserver** enthalten die Anwendungslogik in Form von Programmen, die in der von SAP entwickelten Programmiersprache ABAP/4 erstellt wurden (vgl. [SAP99]). Die SAPGUI-Komponenten kommunizieren direkt mit den Applikationsservern. Hinsichtlich der Zuordnung von fachlichen und systemtechnischen Anwendungsfunktionen zu Applikationsservern sind unterschiedliche Konfigurationen möglich. Hierdurch kann u.a. eine Lastverteilung unterstützt werden.

Jeder Applikationsserver enthält eine als **Gateway** bezeichnete Komponente. Dieser Komponente kommt im Kontext der vorliegenden Arbeit eine besondere Bedeutung zu, da sie als Vermittlungsinstanz für das von R/3 verwendete Kommunikationsprotokoll **RFC (Remote Function Call)** dient (vgl. zu RFC z.B. [Mend98, 305ff]). Der RFC ist die R/3-Variante eines RPC. Er ermöglicht den Aufruf von RFC-fähigen Funktionsbausteinen eines Applikationsservers über ein Netzwerk (vgl. zu Funktionsbausteinen z.B. [Mend98, 172ff]). Bei diesen Funktionsbausteinen handelt es sich um eine spezielle Form von ABAP/4-Programmen, auf die in Abschnitt 5.1.3 noch genauer eingegangen wird. Die Funktionsbausteine können von anderen R/3-Systemen aber auch von AwSen, bei denen es sich nicht um R/3-Systeme handelt, aufgerufen werden. Auch der Aufruf von Funktionen eines Nicht-R/3-Systems durch ein R/3-System wird durch den RFC unterstützt.

Neben den Applikationsservern enthält die Applikationsschicht noch einen **Messageserver** (vgl. [SAP99]). Dieser erfüllt vor allem zwei Funktionen:

- Zuordnung von Applikationsservern zu SAPGUI-Komponenten im Rahmen der Benutzeranmeldung unter Berücksichtigung der aktuellen Lastverteilung.
- Abwicklung der Kommunikation zwischen den Applikationsservern.

Die **Datenbankschicht** umfasst ein Datenbankmanagementsystem und eine zugehörige Datenbank. Diese Datenbank enthält die fachlichen Daten der Anwendungen. Aber auch die Customizing-Daten, die zur Konfiguration des R/3-Systems benötigt werden, und die in ABAP/4 realisierten Anwendungen sind in der Datenbank abgelegt (vgl. [Buck98, 180f]). Bei Ausführung einer Anwendung werden die zu der Anwendung gehörenden Programme aus der Datenbank geladen, in ein durch die Applikationsserver interpretierbares Format übersetzt und anschließend ausgeführt. Zur Erzielung einer ausreichenden Performance werden diese Abläufe durch einen Caching-Mechanismus unterstützt.

Ein R/3-System besteht nicht nur aus einer Menge von vorgefertigten Programmen, sondern enthält auch eine umfangreiche **Entwicklungsumgebung**. Mit dieser Entwicklungsumgebung können Kunden eigene ABAP/4-Programme erstellen und somit die Standardfunktionen ihren Anforderungen entsprechend erweitern. Ein R/3-System kann somit auch als Systemplattform für neu zu entwickelnde AwSe dienen. Der wesentliche Vorteil dieses Ansatzes liegt in der vollständigen Integration dieser Anwendungen in das R/3-System. Zum einen können die bereits vorhandenen Funktionen genutzt werden, zum anderen ist ein vollständiger Zugriff auf die in der R/3-Datenbank verwalteten Daten möglich.

5 Kopplungsmechanismen von SAP R/3

Dieses Kapitel bildet den Kern der vorliegenden Arbeit. Die folgenden Abschnitte widmen sich jeweils einem Kopplungsmechanismus von SAP R/3. Die Mechanismen werden vorgestellt, beschrieben und hinsichtlich des primär unterstützten Integrationskonzepts untersucht. Zur Strukturierung der Beschreibung werden die zwei Sichten Schnittstellensicht und Innensicht unterschieden. In der **Schnittstellensicht** liegt der Fokus auf der Schnittstelle zwischen den integrierten AwSen. Die AwSe werden nur in der Außensicht betrachtet. Es werden vor allem die Kopplungspartner und die ausgetauschten Objekte beschrieben. In der **Innensicht**-Betrachtung werden anschließend die Mechanismen innerhalb der AwSe dargestellt.

5.1 BAPI (Business Application Programming Interface)

Als ersten Kopplungsmechanismus möchten wir die Business Application Programming Interfaces (BAPIs) behandeln. Die ersten BAPIs wurden Ende 1996 auf den WWW-Seiten der SAP AG veröffentlicht. Mittlerweile haben sich die BAPIs zum zentralen Kopplungsmechanismus von R/3 entwickelt.

5.1.1 Ziele des Kopplungsmechanismus

Mit den BAPIs bietet R/3 eine objektorientierte Schnittstelle an, die externen Systemen und weiteren R/3-Komponenten der gleichen Installation den Zugriff

- auf die Funktionen einer R/3-Komponente und
- auf die durch die Komponente verwalteten Daten erlaubt.

Die externen AwSe, die mit einem R/3-System integriert werden sollen, werden in zunehmendem Maße mit Hilfe von objektorientierten Entwicklungsplattformen erstellt. Durch die Verwendung einer objektorientierten Schnittstelle wird ein nahtloser Zugriff aus solchen AwSen auf ein R/3-System unterstützt.

Die mit R/3 ausgelieferten BAPIs orientieren sich an der fachlichen Begriffswelt und bilden somit eine semantisch hohe Schnittstelle, die einen Zugriff auf ein R/3-System anhand von fachlichen Merkmalen ermöglicht. Die SAP garantiert eine langfristige Stabilität der BAPI-Schnittstellen. Sollten Änderungen an diesen Schnittstellen erforderlich sein, so wird auf Abwärtskompatibilität geachtet.

Als Informationsquellen bei der Erstellung dieses Abschnitts dienten vor allem das von SAP im WWW veröffentlichte BAPI-Benutzerhandbuch [SAP01e] und der ebenfalls im WWW verfügbare BAPI-Programmierleitfaden [SAP01d].

5.1.2 Schnittstellensicht des Mechanismus

1996 führte die SAP als Basis-Konzept für die Gesamtarchitektur von R/3 die **Business Framework Architecture** (BFA) ein (vgl. zur BFA [SAP01e, 4ff] und [ReEc99, 121ff]). Sie gibt einen Rahmen für die interne Strukturierung von R/3 und für die Anbindung von AwSen an ein R/3-System vor. Die BFA sieht die Aufteilung von R/3 in **Business Components** vor. Diese sollen getrennt installierbar und weiterentwickelbar sein. Als Beispiel für eine Business Component soll hier das Modul HR (Human Resources) genannt werden.

Eine Business Component besteht aus einer Menge von **Business Objects**. Die Business Objects repräsentieren fachliche Entitäten, wie z.B. Kundenaufträge. Sie kapseln die zur Beschreibung einer Entität benötigten Daten und die auf diesen Daten operierenden Funktionen. Die Funktionen eines Business Object werden als Methoden bezeichnet. Die Schnittstelle einer Business Component wird durch die in ihr enthaltenen Business Objects definiert (vgl. Abbildung 5). Methoden eines Business Objects, die von außerhalb der zugehörigen Business Component aufgerufen werden können, werden als BAPIs bezeichnet.

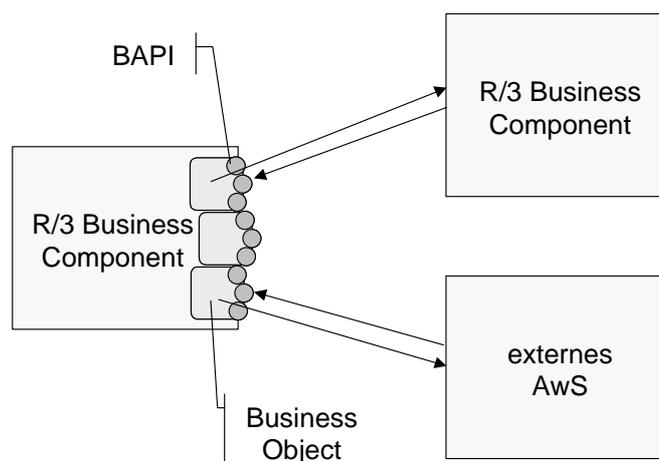


Abbildung 5: Elemente der Business Framework Architecture

Der BAPI-Kopplungsmechanismus unterstützt als **Kopplungspartner** nicht nur R/3-Systeme sondern auch AwSe, bei denen es sich nicht um R/3-Systeme handelt. Die Unterstützung von Nicht-SAP-Plattformen beschränkt sich i.d.R. allerdings auf das Anbieten von einfachen Kommunikationsmechanismen, die einen prozeduralen oder objektorientierten Aufruf von BAPIs ermöglichen (vgl. Abschnitt 5.1.3). Diese Mechanismen werden teilweise durch Fremdanbieter entwickelt und vertrieben.

Im Mittelpunkt des BAPI-Kopplungsmechanismus steht der Zugriff auf R/3-Komponenten. Ein Zugriff aus einer R/3-Komponente auf ein externes AwS oder eine andere R/3-Komponente der gleichen Installation war bis zu Release 4.5 nur durch direkte Verwendung des RFC möglich. Zu Release 4.5 wurde der BAPI-Kopplungsmechanismus jedoch dergestalt erweitert, dass auch ein solcher Zugriff unterstützt wird. Die weiteren Ausführungen konzentrieren sich auf den Zugriff auf eine R/3-Komponente.

Als **Übergabeobjekte** sind ausschließlich BAPI-Aufrufe und die zugehörigen Antworten vorgesehen (vgl. Abbildung 6). Es kann unterschieden werden zwischen

- standardisierten BAPIs (vgl. [SAP01d, 56-81]),
- Service-BAPIs (vgl. [SAP01e, 15-17]) und
- sonstigen BAPIs.

Bei den **standardisierten BAPIs** handelt es sich um BAPIs, die hinsichtlich ihrer Schnittstelle und Funktion durch SAP standardisiert wurden. Diese BAPIs sind für viele der mit R/3 ausgelieferten Business Objects bereits implementiert. Durch sie werden bestimmte Grundfunktionen bereitgestellt, wie z.B. Funktionen zum Lesen des Zustands eines Business Object.

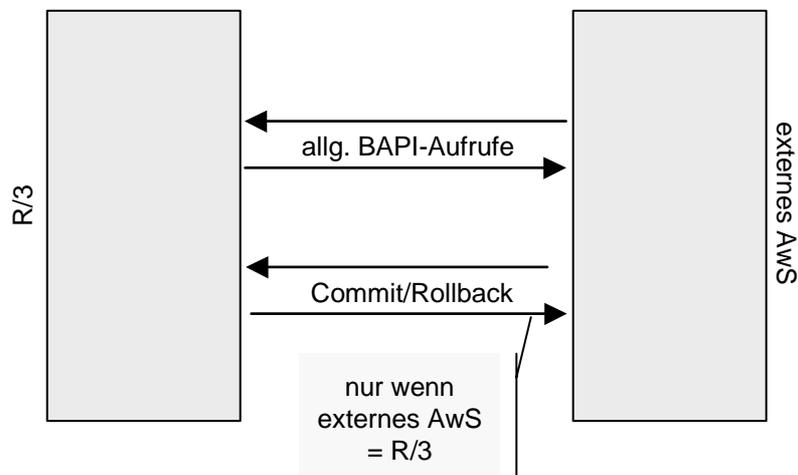


Abbildung 6: Schnittstellensicht BAPI-Kopplungsmechanismus

Mit Hilfe der **Service-BAPIs** werden verschiedene Hilfsfunktionen durch R/3 zur Verfügung gestellt. Hervorgehoben werden sollen hier die Service-BAPIs für die Transaktionssteuerung (vgl. [SAP01e, 28-32]). Vor Release 4.0 wurde davon ausgegangen, dass ein BAPI-Aufruf immer genau eine Transaktion umfasst. Die Commit/Rollback-Steuerung lag vollständig in der Verantwortung von R/3. Zu Release 4.0 wurde diese starke Einschränkung aufgehoben. Ein Commit oder Rollback kann nun durch ein externes System angestoßen werden. Hierzu wurden entsprechende Service-BAPIs implementiert. Ein 2-Phasen-Commit wird allerdings nicht unterstützt.

Die **Übergabe der BAPI-Aufrufe** erfolgt in der Regel direkt mit dem in Abschnitt 4 vorgestellten RFC. Aber auch eine Übertragung mittels der in Abschnitt 5.2 beschriebenen IDocs ist möglich. Diese Variante wird verwendet, wenn BAPIs im Rahmen von ALE-Szenarien zur asynchronen Kommunikation verwendet werden (vgl. Abschnitt 5.3).

5.1.3 Innensicht des Mechanismus

SAP R/3 wird bereits mit einer ganzen Reihe von Business Objects ausgeliefert. Diese Business Objects können durch die Kunden modifiziert und erweitert werden. Außerdem können mit Hilfe der R/3-Entwicklungsumgebung neue Business Objects implementiert werden. Die Definition eines Business Object in R/3 wird als **Objekttyp** bezeichnet (vgl. zu Objekttyp [SAP01e, 7f]). Eine der wesentlichen Herausforderungen bei der Entwicklung des BAPI-Kopplungsmechanismus lag sicherlich in der Umsetzung des objektorientierten Paradigmas auf Basis einer Entwicklungs- und Laufzeitumgebung, die durch die prozedurale Programmiersprache ABAP/4 und hinsichtlich der Datenbank durch das Relationenmodell geprägt ist.² SAP hat hier einen pragmatischen Weg gewählt, der im Folgenden vorgestellt wird.

Basis für die Realisierung der BAPIs sind die bereits erwähnten **Funktionsbausteine** (vgl. zu Funktionsbausteinen und Funktionsgruppen z.B. [Mend98, 172ff]). Inhaltlich zusammengehörende Funktionsbausteine werden in einer **Funktionsgruppe** zusammengefasst. Eine Funktionsgruppe besitzt einen Speicher, der von allen Funktionsbausteinen, die in der Funktionsgruppe enthalten sind, genutzt werden kann. Der Zugriff auf eine Funktionsgruppe kann nur über die Schnittstellen der enthaltenen Funktionsbausteine erfolgen.

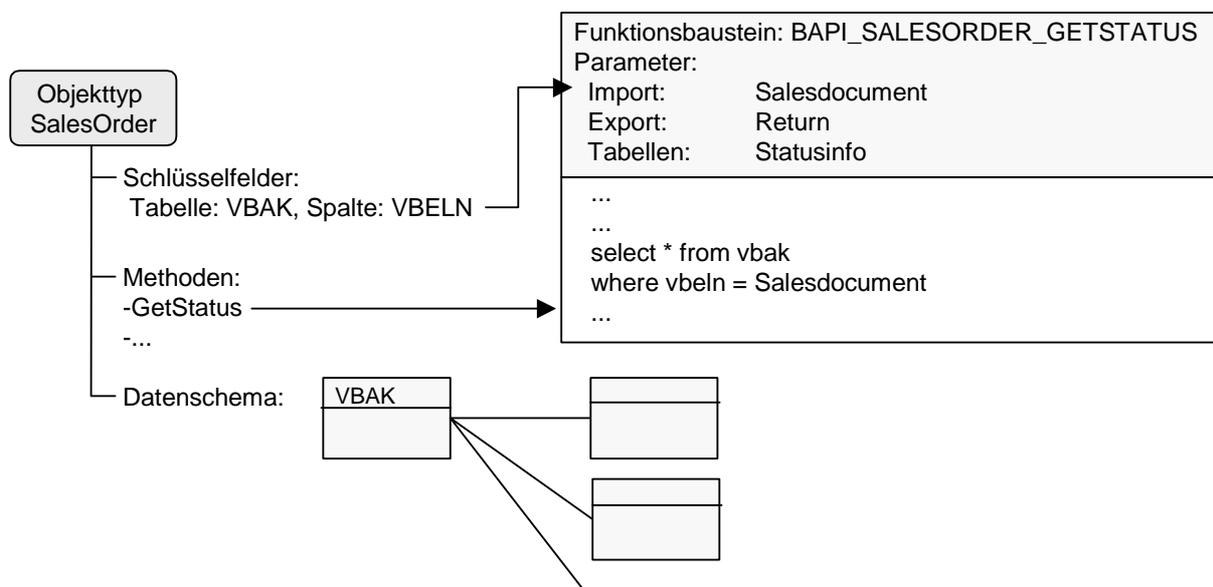


Abbildung 7: Beispiel für Realisierung eines Business Object

² ABAP/4 ist mittlerweile zu einer objektorientierten Sprache erweitert worden. Die Portierung des BAPI-Kopplungsmechanismus auf objektorientiertes ABAP ist zu Release 4.5 allerdings noch nicht erfolgt.

Die Funktionsgruppen können als der erste Schritt von ABAP/4 in Richtung Objektorientierung angesehen werden. Ein wesentliches Element von objektorientierten Programmiersprachen, die Unterscheidung zwischen Objekttyp und Objektinstanzen, fehlt hier allerdings noch völlig. Es steht immer nur eine Instanz einer Funktionsgruppe zur Verfügung. Die BAPIs der Objekttypen werden mit Hilfe der Funktionsbausteine realisiert (vgl. [SAP01d, 6f]). Alle Funktionsbausteine zu einem SAP-Business-Objekt sollen in einer gemeinsamen Funktionsgruppe zusammengefasst werden (vgl. [SAP01d, 26]).

Im Rahmen der Definition wird einem Objekttyp ein Teilausschnitt des R/3-Datenschemas zugeordnet. Dieser Ausschnitt umfasst alle Tabellen, die zur Beschreibung des Zustands der von dem Objekttyp repräsentierten fachlichen Entität erforderlich sind. Ein externes AWS kann auf die Inhalte dieser Tabellen nicht direkt, sondern nur über die BAPIs des Objekttyps zugreifen.

Eine Instanz eines Objekttyps wird von einer Menge von Tupeln aus den Tabellen des Objekttyps repräsentiert. Die Identifikation einer solchen Instanz erfolgt anhand einer oder mehrerer Spalten dieser Tabellen. Die zur Identifikation benötigten Spalten werden dem Objekttyp bei seiner Definition als Schlüsselfelder zugeordnet.

Die Methoden eines Objekttyps können in Instanz- und Klassenmethoden aufgeteilt werden. Eine Instanzmethode operiert auf einer Instanz des Objekttyps. Hier stellt sich das Problem, dass die Funktionsgruppen das Konzept einer Objektinstanz nicht direkt unterstützen. Dies wird gelöst, indem bei Funktionsbausteinen, die Instanzmethoden realisieren, die Schlüsselfelder des zugehörigen Objekttyps in die Schnittstelle aufgenommen werden (vgl. Abbildung 7).

Der **Zugriff auf die BAPIs einer R/3-Komponente** kann auf zwei unterschiedliche Weisen erfolgen:

- funktionsorientiert
- objektorientiert

Bei der funktionsorientierten Variante wird direkt der zugehörige Funktionsbaustein via RFC aufgerufen (vgl. [SAP01e, 25]). Der objektorientierte Zugriff kann auf unterschiedliche Weise realisiert sein (vgl. [SAP01e, 22f]). Eine typische Realisierungsform ist in Abbildung 8 dargestellt. Im aufrufenden System wird hierbei ein Proxy-Objekt verwendet, das die zur

Identifikation einer Instanz benötigten Schlüsselinformationen kapselt. Das Proxy-Objekt nimmt die Methodenaufrufe entgegen und ruft mittels RFC den zugehörigen Funktionsbaustein auf. Handelt es sich bei der aufgerufenen Methode um eine Instanzmethode, dann ergänzt das Proxy-Objekt den Aufruf noch mit den entsprechenden Schlüsselinformationen.

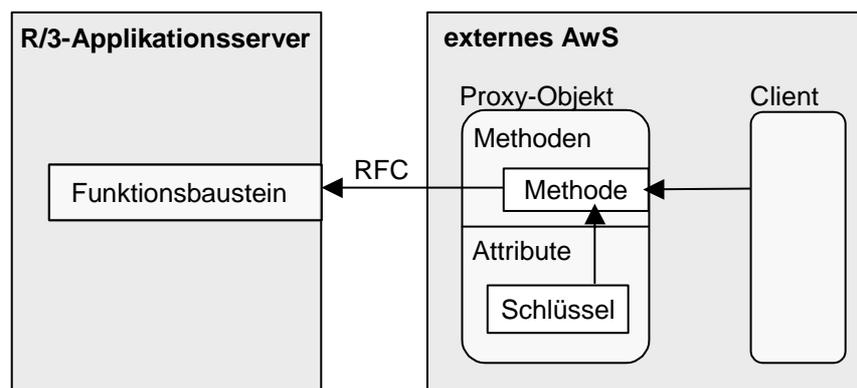


Abbildung 8: objektorientierter Aufruf von BAPIs

Einen Sonderfall stellt die in ALE-Szenarien verwendete Übertragung von BAPI-Aufrufen mittels IDocs dar (vgl. [SAP01d, 97f]).³ Der Aufruf wird hierbei mit Hilfe eines IDoc an das Empfänger-System übermittelt. Dort wird ein zugehöriger Funktionsbaustein zur Verarbeitung des IDoc aufgerufen. Der Funktionsbaustein extrahiert die benötigten Daten aus dem IDoc und ruft unter Verwendung dieser Daten das zugehörige BAPI auf. R/3 stellt Funktionen bereit mit denen die zur Übermittlung des Aufrufs benötigten Elemente, wie z.B. Eingangs-Funktionsbaustein und IDoc-Typ, generiert werden können. Handelt es sich beim Aufrufer um ein R/3-System, so kann auch ein Funktionsbaustein generiert werden, der das zu übermittelnde IDoc erstellt.

5.1.4 Unterstütztes Integrationskonzept

Bei der Beantwortung der Frage nach dem durch den BAPI-Kopplungsmechanismus primär unterstützten Integrationskonzept gehen wir vom Integrationskonzept Objektintegration aus. In den Tabellen 1 und 2 werden in der ersten Spalte die wesentlichen Konzepte der Objektintegration aufgeführt. In der zweiten Spalte wird für jedes Konzept die Unterstützung durch den BAPI-Kopplungsmechanismus beschrieben. Es wird zwischen einer Unterstützung auf systemtechnischer und auf fachlicher Ebene unterschieden.

Zusammenfassend lässt sich sagen, dass der BAPI-Kopplungsmechanismus die Kernkonzepte der Objektintegration

- Kapselung von Daten und Funktionen in Objekten und
- Kommunikation über Austausch von Nachrichten

sowohl auf systemtechnischer als auch auf fachlicher Ebene unterstützt. Bei Verwendung dieses Kopplungsmechanismus können Daten- und Funktionsredundanz minimiert werden.

Die weiterführenden Konzepte der Objektintegration werden jedoch nur unzureichend unterstützt. Die Implementierung von semantischen Integritätsbedingungen kann grundsätzlich in Form einfacher Methodenaufrufe erfolgen. Ein allgemeiner weiterführender Mechanismus wird allerdings nicht angeboten. Die objektorientierte Erweiterung der Sprache ABAP/4 bietet zwar einen Ereignismechanismus an, der grundsätzlich zur Realisierung von semantischen Integritätsbedingungen geeignet wäre, dieser kann aber zur Zeit nur innerhalb eines R/3-Systems und nicht über Systemgrenzen hinweg verwendet werden (vgl. hierzu [Weli00]).

| Konzepte der Objektintegration | Unterstützung durch den Kopplungsmechanismus |
|--|---|
| Kapselung von Daten und Funktionen in Objekten | fachl.: bereitgestellte fachliche Objekttypen syst.: Objekttyp-Mechanismus |
| Kommunikation über Nachrichten | fachl.: bereitgestellte fachliche BAPIs syst.: RFC, Proxy-Mechanismus, IDoc-Mechanismus |
| dediziertes Kommunikationssystem | fachl./syst.: abhängig davon wie BAPIs aufgerufen werden Bsp.: Softwarebus von CORBA (syst.) |

Tabelle 1: Vergleich Integrationskonzept / Kopplungsmechanismus

³ Der ALE-Kopplungsmechanismus wird detailliert in Abschnitt 5.3 behandelt.

| Konzepte der Objektintegration | Unterstützung durch den Kopplungsmechanismus |
|---|--|
| Delegation / Vererbung von Funktionen | Delegation: syst.: RFC, Proxy-Mechanismus, IDoc-Mechanismus Vererbung: - abhängig von Proxy-Objektyp - Bsp.: Vererbungsmechanismus von C++ (syst.) |
| Unterscheidung zwischen Vorgangsobjekten und konzeptuellen Objekten | keine Unterstützung |
| Transaktionsverwaltung | teilweise Unterstützung, kein 2-Phasen-Commit syst.: Commit- / Rollback-BAPIs |
| semantische Integritätsbedingungen | syst.: RFC, Proxy-Mechanismus, IDoc-Mechanismus |

Tabelle 2: Vergleich Integrationskonzept / Kopplungsmechanismus

Des weiteren ist zu beachten, dass der BAPI-Kopplungsmechanismus keine Trennung von Vorgangsobjekten und konzeptuellen Objekten vorsieht. Die verfügbare Dokumentation enthält eine solche Unterscheidung nicht einmal in Form einer allgemeinen Gestaltungsrichtlinie. Auch die fehlende Unterstützung eines 2-Phasen-Commit stellt eine erhebliche Beschränkung dar.

5.2 IDoc (Intermediate Documents)

In diesem Abschnitt werden wir einen Kopplungsmechanismus untersuchen, der schon sehr viel länger verfügbar ist als die im vorherigen Abschnitt besprochenen BAPIs. Der IDoc-Kopplungsmechanismus erlaubt einem R/3-System bereits seit Release 2.2 die Kommunikation mit anderen AwSen. Die Kommunikation erfolgt hierbei durch Austausch von Geschäftsdokumenten im Sinne des EDI-Konzepts (vgl. zu EDI z.B. [Deut95], [Ball00]).

5.2.1 Ziele des Kopplungsmechanismus

Der IDoc-Kopplungsmechanismus ermöglicht den Austausch von Geschäftsdokumenten zwischen einem R/3-System und einem anderen AwS. Durch ihn wurde das R/3-System gegenüber dem in der Praxis recht weit verbreiteten EDI-Konzept geöffnet. Das **EDI-Konzept** hat zum Ziel, den Austausch von Geschäftsdokumenten zwischen Unternehmen zu automatisieren. An die Stelle der Übertragung eines Dokuments mittels Brief, Fax oder Telefon tritt eine elektronische Übertragung, die direkt die AwSe der kommunizierenden Unternehmen verbindet. Hierdurch werden Transaktionen beschleunigt, Transaktionskosten verringert und Medienbrüche vermieden. Es wurden umfangreiche Bemühungen zur Entwicklung zugehöriger Standards unternommen. Hierbei konzentrierte man sich zunächst vor allem auf Standards, die den Aufbau der ausgetauschten Geschäftsdokumente auf technischer und fachlicher Ebene festlegen. Aus diesen Bemühungen ist unter anderem der branchenübergreifende EDIFACT-Standard hervorgegangen (vgl. [Iso88]). Neuere Entwicklungen, wie z.B. Open-EDI, streben auch eine Standardisierung der zugehörigen Prozesse an (vgl. zu Open-EDI [Iso97]).

In einer erweiterten Form kann der IDoc-Mechanismus auch zur Synchronisation von verteilt gehaltenen Datenbeständen verwendet werden. Diese Anwendung wird in der vorliegenden Arbeit in einem eigenen Abschnitt im Rahmen des ALE-Kopplungsmechanismus behandelt (vgl. Abschnitt 5.3).

Als Informationsquellen bei der Erstellung dieses Abschnitts dienten vor allem die mit dem R/3-System ausgelieferte Online Dokumentation [SAP99] sowie das Buch von Nagpal zu IDoc und ALE [Nagp99].

5.2.2 Schnittstellensicht des Mechanismus

Der IDoc-Mechanismus ist hinsichtlich der **Kopplungspartner** nicht beschränkt auf die Kommunikation zwischen zwei R/3-Systemen. Er erlaubt die bidirektionale Kommunikation zwischen einem R/3-System und einem beliebigen anderen AwS. Nicht-SAP-Plattformen werden von SAP allerdings nur rudimentär unterstützt. Die Unterstützung beschränkt sich hierbei auf das Anbieten eines einfachen Kommunikationsmechanismus. Die Kommunikation zwischen einem R/3- und einem R/2-System stellt für den Mechanismus einen Spezialfall dar, der sich in verschiedenen Punkten von der allgemeinen Vorgehensweise unterscheidet. Zur Vereinfachung der Darstellung wird dieser Sonderfall im Folgenden bei der Beschreibung des

IDoc-Kopplungsmechanismus und auch beim ALE-Kopplungsmechanismus nicht berücksichtigt.

Zentrales Element des IDoc-Kopplungsmechanismus sind die als **Übergabeobjekte** verwendeten Intermediate Documents (IDocs). Werden IDocs im Rahmen von EDI-Szenarien eingesetzt, so werden sie zur digitalen Repräsentation von Geschäftsdokumenten genutzt. Ein Kommunikationsschritt besteht hierbei im Austausch eines solchen IDoc. Neben dem Austausch von Geschäftsdokumenten sieht R/3 die Übermittlung von **Statusinformationen** zwischen den kommunizierenden AwSen vor (vgl. Abbildung 9). Mit Hilfe dieser Statusinformationen kann ein System, das ein Geschäftsdokument von einem R/3-System empfangen hat, dem R/3-System den aktuellen Bearbeitungsstatus mitteilen. Insbesondere ermöglicht dies die Rückmeldung von Verarbeitungsfehlern. Auch Statusinformationen können mittels IDocs übertragen werden.

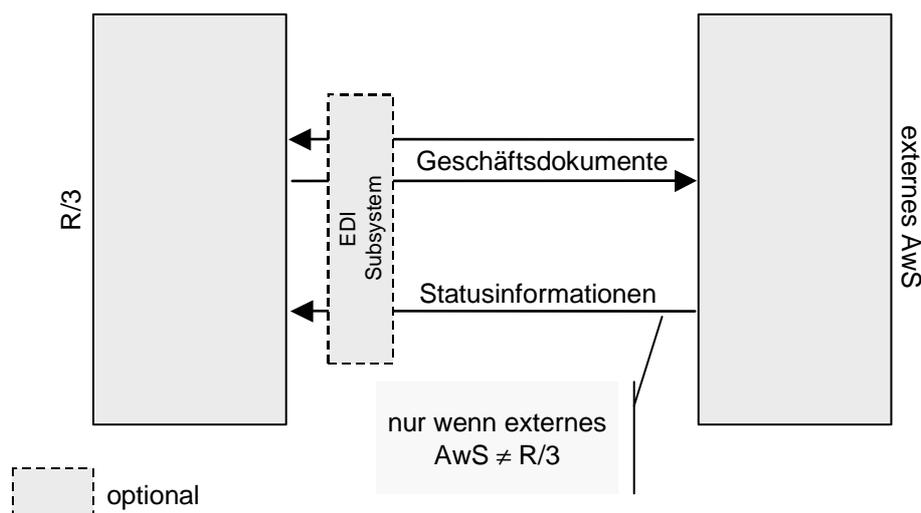


Abbildung 9: Schnittstellensicht IDoc-Kopplungsmechanismus

Bei Betrachtung der IDocs sind die drei Ebenen

- Meta-IDoc-Struktur,
- IDoc-Typ und
- IDoc

zu unterscheiden (vgl. [SAP99] und [Nagp99, 561ff]).

Die Meta-IDoc-Struktur ist durch R/3 fest vorgegeben und kann durch Kunden nicht verändert werden. Sie spannt einen Modellierungsraum auf für die Definition von IDoc-

Typen. Ein IDoc ist die Instanz eines solchen IDoc-Typs.⁴

Mit R/3 werden eine ganze Reihe von fertigen **IDoc-Typen** ausgeliefert, sowie EDI-Szenarien, die diese IDoc-Typen verwenden. Die ausgelieferten IDoc-Typen und Szenarien können durch die Kunden angepasst werden. Außerdem können neue IDoc-Typen und zugehörige Verarbeitungslogik zum Erzeugen und Verarbeiten der IDocs hinzugefügt werden. Bei der Festlegung der IDoc-Typen hat SAP eine proprietäre Lösung gewählt, die sich aber recht stark am EDIFACT-Standard orientiert. Zur Unterstützung von Standards wie z.B. EDIFACT oder ANSI X.12 muss ein **EDI-Subsystem** eingesetzt werden (vgl. zu EDI-Subsystemen z.B. [Nagp99, 64ff]). Dieses übernimmt die Konvertierung zwischen dem IDoc-Format und dem Fremd-Format. Durch diese Vorgehensweise wird eine Abkopplung von EDI-Standards erreicht. Die R/3-Anwendungslogik kann hierdurch von einem einzigen weitestgehend konstanten Format ausgehen. EDI-Subsysteme werden von einer ganzen Reihe von Drittanbietern angeboten. SAP selbst vertreibt keine EDI-Subsysteme.

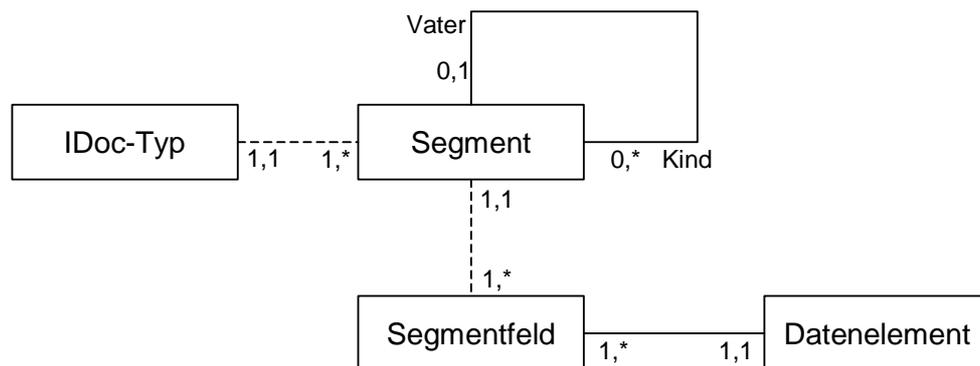


Abbildung 10: Meta-Modell der IDoc-Struktur

In Abbildung 10 ist die von R/3 vorgegebene **Meta-IDoc-Struktur** in Form eines Meta-Modells dargestellt.⁵ Jedes IDoc besteht aus einem Kontroll-Record, einem oder mehreren Status-Records und einem Datenteil. Der Kontroll-Record enthält Verwaltungsdaten, wie z.B. den Empfänger eines IDoc. Die Status-Records enthalten Informationen bzgl. des Verarbeitungsstatus eines IDoc. Die Struktur von Kontroll-Record und Status-Records ist fest von R/3 vorgegeben. Die Struktur des Datenbereichs, der die eigentlichen Nutzdaten enthält,

⁴ Zu beachten ist, dass ein IDoc-Typ keine vollständige typmäßige Beschreibung eines IDoc darstellt.

⁵ Bei der Erstellung des Meta-Modells wurde von dem in [FeSi01, 124] veröffentlichten Meta-Meta-Modell ausgegangen.

wird mit Hilfe der IDoc-Typen definiert. Ein IDoc-Typ besteht aus einem oder mehreren Segmenten, die in Hierarchien angeordnet werden können. Ein Segment enthält eine Reihe von Feldern. Der Typ eines solchen Feldes wird durch die Zuordnung eines Datenelements aus dem R/3-Dictionary festgelegt.

IDocs werden im ASCII-Format übertragen. Zu beachten ist, dass Segmentfelder in IDocs nicht durch Auszeichnungselemente (Tags) gekennzeichnet sind. Hierdurch ist die Lesbarkeit der übertragenen Dokumente stark eingeschränkt.

Da ein IDoc in Bezug auf seine Struktur nicht selbstbeschreibend ist, stellt sich außerdem das Problem der Synchronisation von Sender- und Empfängersystem hinsichtlich der verwendeten IDoc-Typen. R/3 unterstützt den Entwickler hierbei, indem es Funktionen anbietet, mit denen IDoc-Typen exportiert werden können. Diese exportierten IDoc-Typen können in Form von IDocs an die Kommunikationspartner übermittelt werden.

Die **Übertragung der IDocs** kann auf drei unterschiedlichen Wegen erfolgen:

- Dateitransfer
- Email
- tRFC (transaktionaler RFC)

5.2.3 Innensicht des Mechanismus

Im Folgenden wird nun die Architektur des Kopplungsmechanismus im Detail dargestellt. Dies erfolgt anhand einer Beschreibung der Abläufe in einem R/3-System bei Versand eines IDoc. Grundsätzlich können hierbei verschiedene Varianten unterschieden werden. Wir wollen uns aber nur auf eine dieser Varianten konzentrieren. Die nicht dargestellten Varianten stellen im Wesentlichen Vereinfachungen dieser Variante in dem Sinne dar, dass weniger Elemente des Mechanismus verwendet werden.

Abbildung 11 zeigt in kappen Zügen den **Ablauf**, der bei der betrachteten Variante **zum Versand eines IDoc** aus einem R/3-System führt (vgl. [SAP99] und [Nagp99, 41ff]):⁶

- 1 Ausgangspunkt ist typischerweise die Neuanlage oder Änderung eines Belegs, z.B. einer Bestellung. Die ändernde Anwendung informiert anschließend die

⁶ Die Darstellung ist in verschiedenen Punkten vereinfacht. Nicht eingegangen wird z.B. auf das Verhalten bei Fehlerfällen und das Anstoßen des empfangenden Systems bei Übertragung der IDocs mittels Dateitransfer.

Nachrichtensteuerung, die hier als Teil des IDoc-Kerns betrachtet werden soll. Die Nachrichtensteuerung ist eine allgemein verwendbare R/3-Komponente zur Automatisierung der Ausgabe und Folgeverarbeitung von partnerabhängigen Nachrichten.

- 2 Die Nachrichtensteuerung überprüft anhand sog. Konditionen, die im Rahmen des Customizing angelegt wurden, ob die Änderung die Erzeugung einer Nachricht zur Folge hat. Dieser Prozess wird als Nachrichtenfindung bezeichnet. Ergebnis ist ein Nachrichtenvorschlag, der vom Anwender noch über entsprechende Pflegemasken angepasst werden kann.

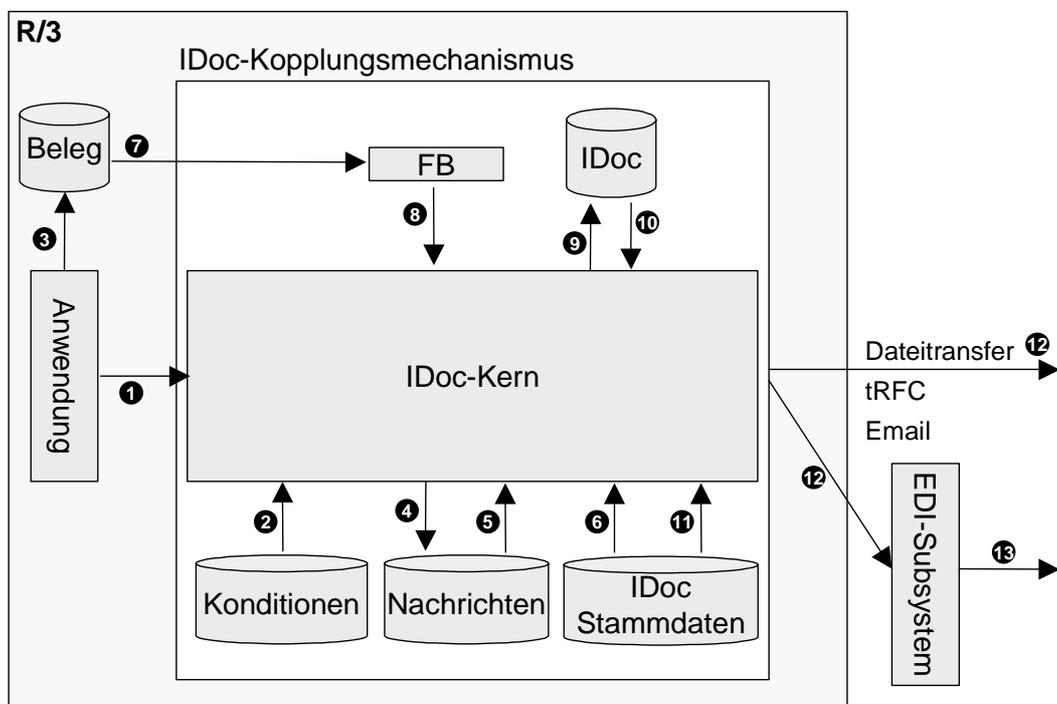


Abbildung 11: Ausgangsverarbeitung

- 3/4 Anschließend schreibt die Anwendung ihre Belegänderungen auf die Datenbank. Gleichzeitig werden die vorgeschlagenen Nachrichten auf der Datenbank abgelegt.
- 5/6 Die im Rahmen der Nachrichtenfindung ermittelten Nachrichten können sofort oder zu einem späteren Zeitpunkt verarbeitet werden. Anhand der IDoc-Stammdaten wird hierbei zunächst ermittelt
 - welcher IDoc-Typ für die Nachricht zu versenden ist

- und welcher Funktionsbaustein ein IDoc zu dem ermittelten IDoc-Typ erstellen kann.
- 7 Der ermittelte Funktionsbaustein wird durch den IDoc-Kern aufgerufen. Der Funktionsbaustein extrahiert die zur Erstellung des IDoc notwendigen Daten aus der R/3-Datenbank und erzeugt das IDoc.
- 8/9 Das erzeugte IDoc wird an den IDoc-Kern übergeben. Dieser legt es in verschiedenen Protokoll-Tabellen ab.
- 10 Nun erfolgt der letzte Schritt im Rahmen der Ausgangsverarbeitung, das fertige IDoc wird versendet. Dies kann wiederum direkt im Anschluss an die Erstellung des IDoc oder zeitlich entkoppelt erfolgen. Der IDoc-Kern liest hierfür das erstellte IDoc aus der Datenbank.
- 11 Mit Hilfe der IDoc-Stammdaten wird der für die Übermittlung zu verwendende Port ermittelt. Dieser beschreibt insbesondere auf welchem Weg das IDoc an das Zielsystem übertragen werden soll. Ein möglicher Kommunikationsmechanismus ist hierbei der tRFC, bei dem es sich um eine Variante des in Kapitel 4 vorgestellten RFC handelt. Er erweitert diesen um transaktionale Eigenschaften. Ein tRFC-Aufruf wird erst an das Zielsystem übertragen, nachdem die zugehörige Transaktion im Quellsystem mit Commit beendet wurde. Alle tRFC-Aufrufe einer Transaktion des Quellsystems werden im Zielsystem in einer gemeinsamen Transaktion abgearbeitet.
- 12/13 Anschließend erfolgt die eigentliche Übertragung des IDoc. Wird ein EDI-Subsystem eingesetzt, so wird das IDoc an das Subsystem übermittelt. Dieses übernimmt dann den weiteren Versand.

Die **Abläufe beim Empfang eines IDoc** in einem R/3-System sollen hier nur in stark verkürzter Form dargestellt werden. Eine detaillierte Behandlung findet sich in [SAP99] und [Nagp99, 52ff]. Auch ein eingehendes IDoc wird durch das R/3-System in Protokoll-Tabellen abgelegt. Anschließend wird anhand der IDoc-Stammdaten der für die Verarbeitung des IDoc zuständige Funktionsbaustein ermittelt und schließlich aufgerufen. Der Funktionsbaustein extrahiert die benötigten Informationen aus dem IDoc und aktualisiert mit diesen Informationen die Datenbank des R/3-Systems.

Alternativ zu dem beschriebenen Vorgehen kann zur Verarbeitung eines eingehenden IDoc auch das mit dem R/3-System ausgelieferte Workflow-Management-System genutzt werden.

An die Stelle des Aufrufs eines Funktionsbausteins zur Verarbeitung des IDoc tritt dann die Erzeugung eines Workitems, das dem zuständigen Bearbeiter zugestellt wird.

5.2.4 Unterstütztes Integrationskonzept

Bei der Untersuchung des durch den IDoc-Kopplungsmechanismus unterstützten Integrationskonzepts gehen wir vom Integrationskonzept datenflussorientierte Funktionsintegration aus. In Tabelle 3 werden die Konzepte des Integrationskonzepts aufgeführt und hinsichtlich ihrer Unterstützung auf systemtechnischer und fachlicher Ebene durch den Mechanismus überprüft.

| Konzepte der datenflussorientierten Funktionsintegration | Unterstützung durch den Kopplungsmechanismus |
|---|--|
| Austausch von Daten zwischen Komponenten | fachl.: bereitgestellte fachliche IDoc-Typen, EDI-Szenarien, IDoc-Kern syst.: tRFC, ... |
| Austausch erfolgt im Allgemeinen ohne Transaktionsbezug | syst.: Hintergrundjob |
| Kontrolle Kommunikationsstruktur und Kommunikationsablauf | fachl.: Partnervereinbarungen in IDoc-Stammdaten syst.: passives Monitoring, aktives Monitoring, Ausnahmebehandlung |
| Überwachung von Vorgangsketten | keine Unterstützung |

Tabelle 3: Vergleich Integrationskonzept / Kopplungsmechanismus

Zusammenfassend lässt sich sagen, dass fast alle Teilkonzepte der datenflussorientierten Funktionsintegration durch den IDoc-Kopplungsmechanismus unterstützt werden. Die einzige Ausnahme bildet hierbei die durch das Integrationskonzept vorgesehene Möglichkeit zur Überwachung von Vorgangsketten. Ein Konstrukt Vorgangskette, das eine Klammer um eine

Folge von zusammengehörigen Nachrichten bildet, ist im IDoc-Kopplungsmechanismus nicht vorgesehen.

Der Austausch von Daten zwischen Komponenten kann als das zentrale Konzept der datenflussorientierten Funktionsintegration angesehen werden. Dieses Kernkonzept wird durch den IDoc-Kopplungsmechanismus sowohl auf systemtechnischer als auch auf fachlicher Ebene unterstützt.

Das Absenden eines IDoc aus einem R/3-System kann aufgrund der beschriebenen Möglichkeiten zur zeitlichen Entkopplung von Nachrichtenfindung, Nachrichtenverarbeitung und IDoc-Versand ohne Transaktionsbezug erfolgen. Der Zeitpunkt des Absenden wird dabei bestimmt durch Definition eines zeitlichen Rhythmus oder eines konkreten Zeitpunkts. Zusätzlich wird ein Datenaustausch mit Transaktionsbezug unterstützt. Der Versand erfolgt hierbei direkt im Anschluss an die Erzeugung / Änderung des Belegs in der R/3-Datenbank.

Das Integrationsziel der **Überwachung der Kommunikationsstruktur** wird durch den Mechanismus auf sehr vielfältige Weise unterstützt. Unterschieden wird hierbei zwischen einem passiven Monitoring, einem aktiven Monitoring und der Ausnahmebehandlung (vgl. [SAP99] und [Nagp99, 214ff]). Wie bereits erwähnt werden alle aus- und eingehenden IDocs durch R/3 protokolliert. Dieses Protokoll umfasst neben dem Inhalt der IDocs noch eine Reihe von zusätzlichen Informationen, wie z.B. Bearbeitungsstatus. Unter dem Begriff passives Monitoring werden eine Reihe von Reports zur Analyse dieser Protokolle zusammengefasst. Neben der Selektion von einzelnen IDocs werden auch verschiedene aggregierte Sichten unterstützt.

Mit Hilfe des aktiven Monitoring und der Ausnahmebehandlung kann ein zuständiger Sachbearbeiter automatisch benachrichtigt werden, wenn bei der Verarbeitung von IDocs Fehler aufgetreten sind. Die Benachrichtigung erfolgt hierbei mit Hilfe des R/3 Workflow-Management-Systems.

Zusätzlich zu den beschriebenen Mechanismen, die eine Überwachung der tatsächlich stattfindenden Kommunikation erlauben, unterstützt R/3 in Form der IDoc-Stammdaten eine Kontrolle der Kommunikationsstruktur. Diese Stammdaten beschreiben unter anderem welche Kommunikationspartner existieren und welche IDoc-Typen mit diesen Partnern ausgetauscht werden.

5.3 ALE (Application Link Enabling)

Mit dem ALE-Kopplungsmechanismus bietet R/3 seit Release 3.0 einen Mechanismus zur Synchronisation von verteilt gehaltenen Datenbeständen an. Ursprünglich handelte es sich beim ALE-Mechanismus um eine Erweiterung des in Abschnitt 5.2 beschriebenen IDoc-Kopplungsmechanismus. Neuere Entwicklungen gehen allerdings dahin den IDoc-Kopplungsmechanismus als Basis durch die in Abschnitt 5.1 beschriebenen BAPIs abzulösen.

5.3.1 Ziele des Kopplungsmechanismus

Der ALE-Kopplungsmechanismus unterstützt die redundante Haltung von Konfigurationsdaten, Stammdaten und Bewegungsdaten und schafft somit die Voraussetzungen zur Aufteilung einer R/3-Installation in mehrere Teil-Installationen, die auf getrennten Rechnern betrieben werden können. Neben der Synchronisation von Datenbeständen innerhalb einer R/3-Installation wird auch eine Synchronisation mit externen AwSen unterstützt.

Als Informationsquellen bei der Erstellung dieses Abschnitts dienten vor allem die mit dem R/3-System ausgelieferte Online Dokumentation [SAP99] sowie [Nagp99].

5.3.2 Schnittstellensicht des Mechanismus

Der ALE-Mechanismus ist in zahlreichen Varianten verfügbar. Insbesondere ist, wie bereits erwähnt, eine Unterscheidung nach dem verwendeten Basis-Mechanismus möglich. Wir wollen uns hier auf die Darstellung einer der Varianten des Mechanismus beschränken. In den folgenden Abschnitten werden wir beschreiben, wie mit Hilfe von ALE Stammdaten unter Verwendung von IDocs (vgl. zu IDocs Abschnitt 5.2.2) ausgetauscht und synchron gehalten werden können.

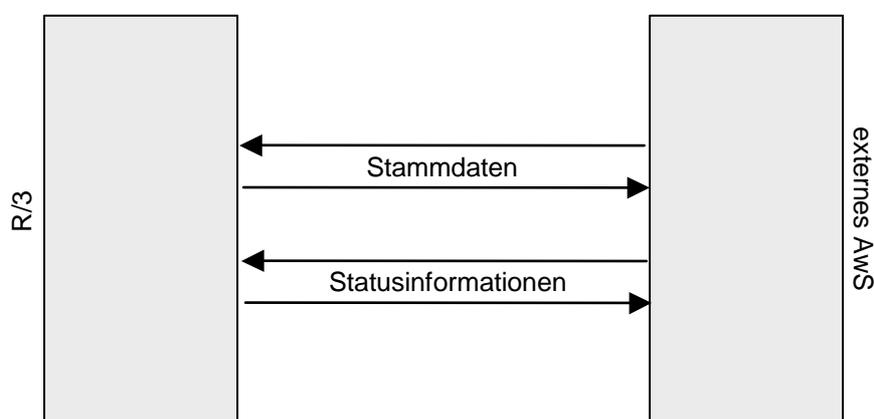


Abbildung 12: Schnittstellensicht ALE-Kopplungsmechanismus

ALE erlaubt hinsichtlich der **Kopplungspartner** die automatisierte Übertragung von Daten aus einem R/3-System an ein beliebiges anderes AwS und den Import von Daten in ein R/3-System. Diese Daten können, solange sie im erwarteten Format geliefert werden, von einem beliebigen AwS stammen. Die Unterstützung von Nicht-SAP-Plattformen durch SAP ist allerdings rudimentär. Sie beschränkt sich wie schon beim IDoc-Kopplungsmechanismus auf das Anbieten eines einfachen Kommunikationsmechanismus.

Wie schon in Abschnitt 5.2 werden auch hier IDocs als **Übergabeobjekte** verwendet. Allerdings werden hier mit Hilfe der IDocs keine Geschäftsdokumente sondern Inhalte von Datenbanktabellen übertragen. R/3 enthält bereits eine ganze Reihe von fertigen IDoc-Typen und ALE-Szenarien, die diese IDoc-Typen verwenden. Diese IDoc-Typen und Szenarien können durch Kunden angepasst werden. Außerdem können neue IDoc-Typen und zugehörige Verarbeitungslogik zum Erzeugen und Verarbeiten der IDocs hinzugefügt werden.

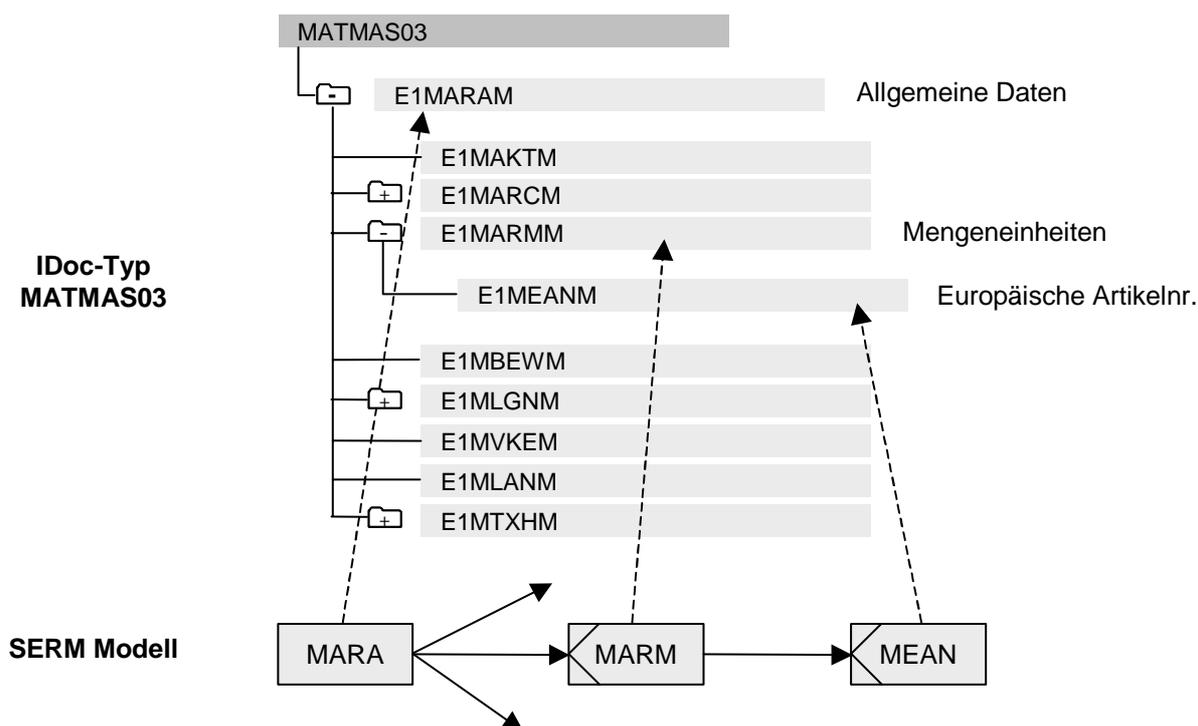


Abbildung 13: Ausschnitt aus IDoc-Typ MATMAS03 mit zugehörigem Datenmodell⁷

Die Struktur der verwendeten IDoc-Typen orientiert sich eng an den zugehörigen Datenbanktabellen (vgl. [SAP99]). Ein IDoc-Segment repräsentiert typischerweise eine Datenbanktabelle. Beziehungen zwischen Tabellen können durch Hierarchiebeziehungen

⁷ Zur Darstellung des Datenmodells wurde der Modellierungsansatz SERM (Strukturiertes Entity-Relationship-Modell) verwendet (vgl. zu SERM [FeSi01, 143ff]).

zwischen Segmenten abgebildet werden. In Abbildung 13 ist beispielhaft für einen Teil der Segmente des IDoc-Typs MATMAS03 die Zuordnung zu den zugehörigen Entitäten des Datenmodells dargestellt. Der IDoc-Typ MATMAS03 wird zur Übermittlung von Materialstammdaten eingesetzt.

Da zur Übermittlung der Stammdaten IDocs verwendet werden, steht der schon in Abschnitt 5.2 beschriebene Mechanismus zur Übertragung von Informationen bzgl. des **Verarbeitungsstatus** eines IDoc auch hier zur Verfügung. Zusätzlich wird noch der ALE-spezifische Audit-Mechanismus zur Übertragung von Statusinformationen angeboten. Auch diese Informationen werden mittels IDocs übermittelt.

5.3.3 Innensicht des Mechanismus

Bei der hier beschriebenen Variante des ALE-Mechanismus zur Verteilung von Stammdaten handelt es sich wie bereits erwähnt um eine **Erweiterung des IDoc-Kopplungsmechanismus**. Die Erweiterung umfasst im Kern

- Funktionen zur Erfassung von relevanten Datenbankänderungen (Änderungszeiger) und
- Funktionen zur Verwaltung von Informationen, die beschreiben welche Systeme an welchen Datenbankänderungen interessiert sind (Verteilungsmodell).

Im Folgenden werden die **Vorgänge** in einem R/3-System **beim Export von Stammdaten** beschrieben (vgl. [SAP99], [Nagp99, 376ff] und [Nagp99, 388ff]):⁸

- 1/2 Ausgangspunkt ist eine Stammdatenänderung durch eine R/3-Anwendung. Die Datenbankänderung wird von der Änderungsbelegkomponente erfasst. Diese Komponente wird nicht nur im Kontext von ALE verwendet. Sie dient allgemein der Protokollierung von Datenbankänderungen in Form von Änderungsbelegen.
- 3/4 Der ALE-Kern setzt auf diesen Änderungsbelegen auf. Auf Basis der Stammdaten des Änderungszeigermechanismus entscheidet er, ob die durchgeführten Änderungen repliziert werden müssen.

⁸ Die Darstellung ist in verschiedenen Punkten vereinfacht. Nicht eingegangen wird z.B. auf das Verhalten bei Fehlerfällen.

- 5 War die Änderung für ALE relevant, so legt der ALE-Kern entsprechende Merker in Form von Änderungszeigern an. Aus diesen geht hervor welche Tupel/Feld/Tabellen-Kombination geändert wurde.

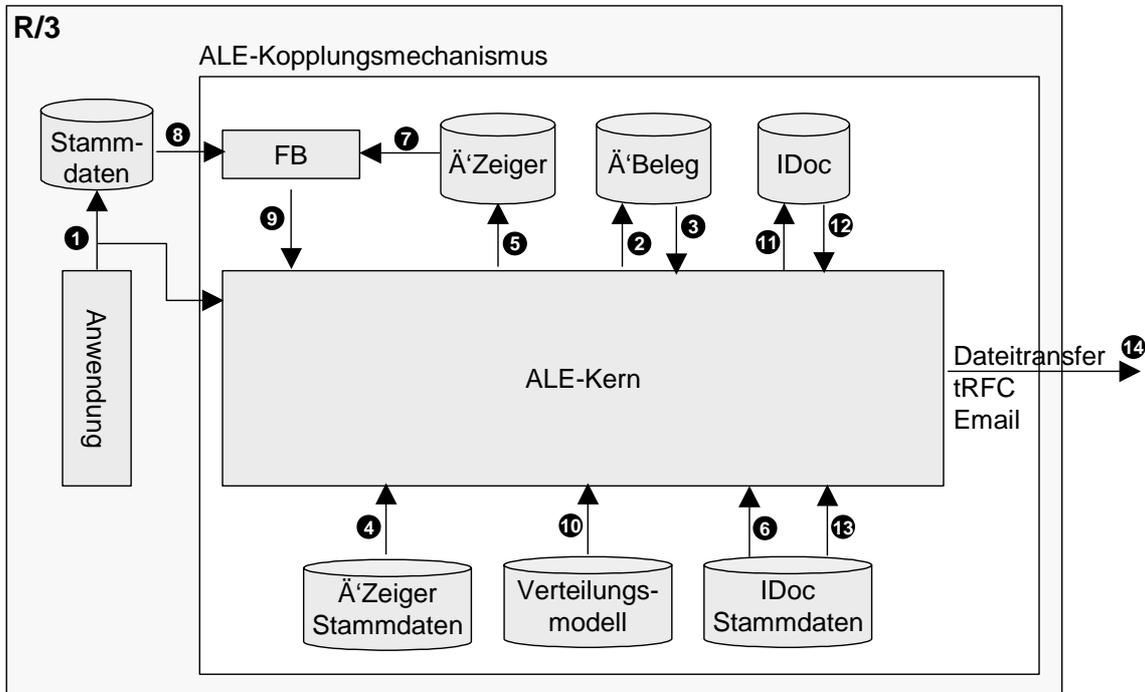


Abbildung 14: Ausgangsverarbeitung

- 6/7 Diese Änderungszeiger werden anschließend mit Hilfe von Funktionsbausteinen verarbeitet. Diese Verarbeitung kann manuell angestoßen oder in Form eines Hintergrundjobs eingeplant werden. Der für die Verarbeitung eines Änderungszeigers zuständige Funktionsbaustein wird anhand der IDoc-Stammdaten ermittelt.
- 8/9 Der Funktionsbaustein extrahiert die geänderten Stammdaten und erstellt ein als Master-IDoc bezeichnetes IDoc. Das Master-IDoc wird an den ALE-Kern übergeben.
- 10/11 Der ALE-Kern ermittelt nun anhand des Verteilungsmodells an welche AwSe das IDoc übertragen werden muss. Für jeden Empfänger wird ein Kommunikations-IDoc erstellt und in Protokoll-Tabellen abgelegt. Bei der Erstellung der Kommunikations-IDocs können empfängerspezifische Anforderungen berücksichtigt werden. Mit Hilfe von Filtern können z.B. die replizierten Daten reduziert werden. Außerdem können Felder in ein vom Empfängersystem benötigtes Format konvertiert werden (Feldumsetzung).

12/13/14 Zeitlich entkoppelt oder direkt im Anschluss an die Erstellung der Kommunikations-IDocs erfolgt der IDoc-Versand. Der zu verwendende Kommunikationsmechanismus wird hierfür für jedes Empfängersystem aus den Partnervereinbarungen der IDoc-Stammdaten ermittelt.

Mit den beschriebenen Mechanismen können Änderungen an Stammdaten an ein externes AwS übertragen werden. Vor Beginn einer solchen Delta-Replikation muss das Empfängersystem typischerweise einmalig mit den vollständigen Stammdaten versorgt werden. R/3 bietet für die ausgelieferten IDoc-Typen verschiedene ABAP/4-Programme an, mit denen eine solche Initialversorgung durchgeführt werden kann. Das empfangende System wird bei Verwendung dieser Programme durch den Anwender explizit vorgegeben. Die zu replizierenden Daten können anhand von Selektionsmasken eingegrenzt werden.

Die **Eingangsverarbeitung** des ALE-Kopplungsmechanismus soll hier nicht explizit dargestellt werden. Eine umfangreiche Behandlung findet sich in [SAP99] und [Nagp99, 402ff]. Sie entspricht weitestgehend der normalen IDoc-Eingangsverarbeitung. Zusätzlich wurden, wie schon bei der Ausgangsverarbeitung, Funktionen zur Filterung von IDoc-Daten und zur Feldumsetzung berücksichtigt.

5.3.4 Unterstütztes Integrationskonzept

In Tabelle 4 wird der ALE-Kopplungsmechanismus dem Integrationskonzept datenflussorientierte Funktionsintegration gegenübergestellt. Da es sich bei ALE weitestgehend um eine Erweiterung des IDoc-Kopplungsmechanismus handelt, kommen wir hier zu einem ähnlichen Ergebnis wie in Abschnitt 5.2.4.

Die beim IDoc-Mechanismus bemängelte fehlende Unterstützung von Vorgangsketten spielt bei ALE keine Rolle, da hier Vorgangsketten nicht benötigt werden. Die wesentlichen Erweiterungen des ALE-Mechanismus sind das Verteilungsmodell und das Konzept der Änderungszeiger. Vor allem durch das Verteilungsmodell können die Replikationsbeziehungen auf fachlicher Ebene modelliert und dokumentiert werden. Die bei der Diskussion des IDoc-Kopplungsmechanismus bereits angesprochenen Funktionen zur Kontrolle der IDoc-Kommunikation werden durch den Audit-Mechanismus ergänzt, der zusätzliche Möglichkeiten zur Überwachung des Verarbeitungsstatus von IDocs anbietet.

| Konzepte der datenflussorientierten Funktionsintegration | Unterstützung durch den Kopplungsmechanismus |
|---|--|
| Austausch von Daten zwischen Komponenten | fachl.: bereitgestellte IDoc-Typen, ALE-Szenarien, ALE-Kern syst.: tRFC, ... |
| Austausch erfolgt im Allgemeinen ohne Transaktionsbezug | syst.: Hintergrundjob |
| Kontrolle Kommunikationsstruktur und Kommunikationsablauf | fachl.: Partnervereinbarungen in IDoc-Stammdaten, Verteilungsmodell syst.: passives Monitoring, aktives Monitoring, Ausnahmebehandlung, ALE-Audit, Änderungszeiger Stammdaten |
| Überwachung von Vorgangsketten | keine Unterstützung |

Tabelle 4: Vergleich Integrationskonzept / Kopplungsmechanismus

5.4 ITS (Internet Transaction Server)

Der im Folgenden untersuchte Internet Transaction Server (ITS) kann nur als Kopplungsmechanismus im erweiterten Sinne verstanden werden. Er unterscheidet sich von den anderen beschriebenen Mechanismen dahingehend, dass bei ihm die Integration der beteiligten AwSe nicht durch direkte Kopplung der AwSe erfolgt, sondern über einen gemeinsamen personellen Aufgabenträger. Dieser Punkt wird in Abschnitt 5.4.4 noch weiter ausgeführt.

5.4.1 Ziele des Kopplungsmechanismus

Mit Hilfe des Internet Transaction Server soll das R/3-System für das World Wide Web geöffnet werden. Der ITS ermöglicht die Entwicklung von WWW-Anwendungen in ABAP/4 unter Nutzung der existierenden R/3 Laufzeit- und Entwicklungsumgebung. Diese Anwendungen können mit Hilfe eines konventionellen Internet Browser, wie z.B. dem

Internet Explorer, genutzt werden und sind vollkommen in das R/3-System integriert. Sie haben Zugriff auf die Funktionen von R/3 und auf die in der R/3-Datenbank verwalteten Daten.

Als Informationsquelle bei der Erstellung dieses Kapitels diente vor allem [PHMZ98].

5.4.2 Schnittstellensicht des Mechanismus

Wie in Kapitel 4 beschrieben greift ein Benutzer auf ein R/3-System typischerweise mittels der SAPGUI-Komponente zu. Seit Release 4.0 ist eine **Java-Version des SAPGUI** im Lieferumfang von R/3 enthalten. Mit der Java-Version werden zwei Ziele verfolgt (vgl. [PHMZ98, 155f]):

- Reduktion der Kosten für Wartung und Installation des SAPGUI. Das Java-SAPGUI ist als Applet realisiert und kann somit einfach von einem HTTP-Server heruntergeladen werden.
- Plattformunabhängigkeit des SAPGUI durch Nutzung einer Java Virtual Machine als Basismaschine.

Die mit dem Java-SAPGUI realisierte Benutzeroberfläche unterscheidet sich nicht von der Oberfläche des konventionellen SAPGUI. Beide Varianten erlauben durch die Unterstützung des TCP/IP-Protokolls grundsätzlich auch einen Zugriff auf die R/3-Applikationsserver via Internet/Intranet.

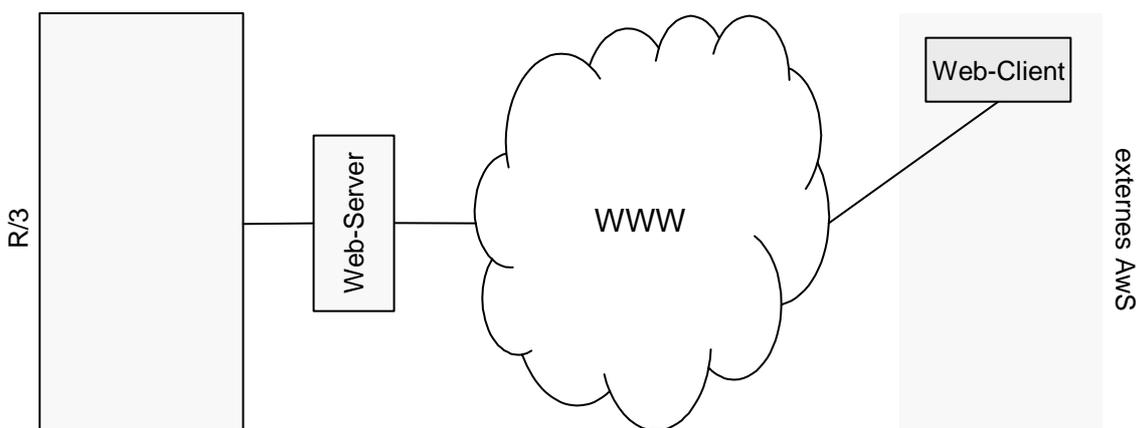


Abbildung 15: Schnittstellensicht ITS-Kopplungsmechanismus

Die durch den ITS vorgegebene Architektur für WWW-Anwendungen unterscheidet sich in der Schnittstellensicht nur in einem Punkt von den beiden genannten GUI-Varianten.

Basismaschine für die Benutzeroberfläche ist hier nicht ein SAPGUI sondern ein konventioneller Internet Browser (vgl. Abbildung 15). Die auf diesem Browser realisierte Benutzeroberfläche orientiert sich nicht an der konventionellen SAP Oberfläche. Die Realisierung erfolgt in Form von HTML-Seiten, wodurch auf den speziellen Bedarf zugeschnittene Oberflächen erstellt werden können, die auch ohne R/3-Vorkenntnisse bedient werden können. Verwendet das mit dem R/3-System zu integrierende externe AwS auch HTML-Seiten zur Realisierung der Benutzeroberflächen, so ist sogar eine Integration der ITS-Seiten in die Oberfläche des externen AwS möglich.

5.4.3 Innensicht des Mechanismus

Der ITS definiert im weiteren Sinne eine Architektur für auf R/3 basierende WWW-Anwendungen. Im engeren Sinne versteht man unter dem ITS einen Mechanismus zur Abwicklung der Kommunikation zwischen einem Web-Server, der einen Web-Client bedient, und einem R/3- System.

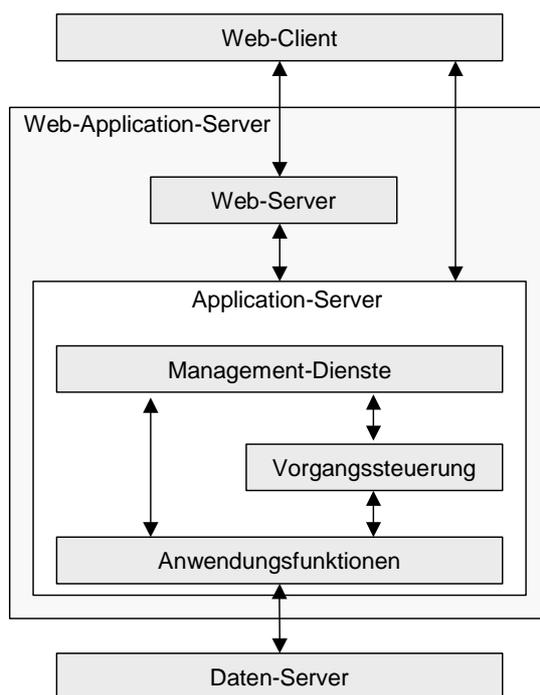


Abbildung 16: Architektur eines WAS-basierten AwSs [SiKM00]⁹

⁹ Die in [SiKM00] vorgesehene Anbindung von Legacy-Anwendungen ist hier nicht relevant und wird aus diesem Grund nicht betrachtet.

Die durch den ITS vorgegebene Architektur entspricht weitestgehend der in [SiKM00] vorgestellten allgemeinen Architektur von auf **Web-Application-Servern** (WAS) basierenden AwSen. In dieser Architektur wird zwischen den vier Komponenten Web-Client, Web-Server, Application-Server und Daten-Server unterschieden (vgl. Abbildung 16). Web-Server und Application-Server werden unter dem Begriff Web-Application-Server zusammengefasst. Dem Application-Server obliegt die Verwaltung und Ausführung der vom Web-Client aufgerufenen Anwendungsfunktionen sowie die Steuerung des Zusammenwirkens der Anwendungsfunktionen. Die Kommunikation zwischen Web-Client und Application-Server erfolgt im Regelfall nicht direkt sondern über Zwischenschaltung eines Web-Server. Ein direkter Zugriff wird allerdings nicht ausgeschlossen.

Durch den ITS wird die beschriebene Architektur in der in Abbildung 17 dargestellten Form konkretisiert. Eine ITS-Anwendung wird hierbei als Internet Application Component (IAC) bezeichnet (vgl. [PHMZ98, 183ff]). Sie besteht aus einem ABAP/4-Programm und einer Menge von HTML-Templates. Das ABAP/4-Programm wird unter Kontrolle eines konventionellen R/3-Applikationsservers ausgeführt.

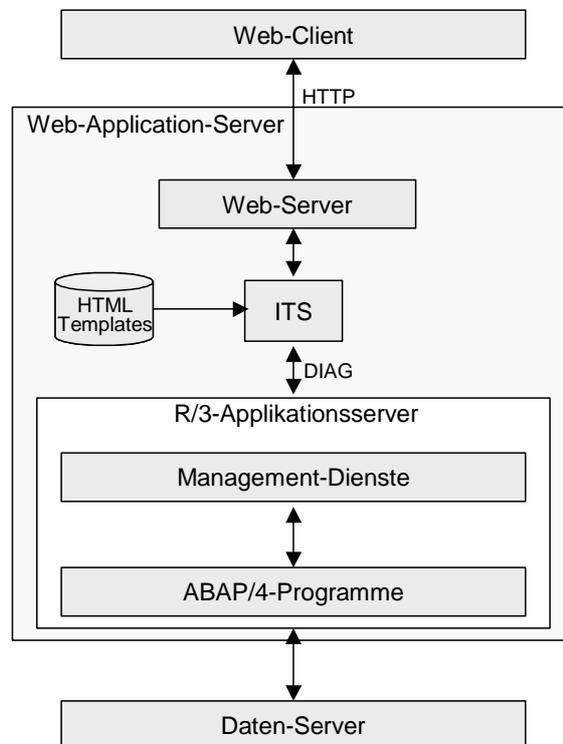


Abbildung 17: Architektur eines ITS-basierten AwSs

Benutzerinteraktionen werden bei einem konventionellen ABAP/4-Programm über sogenannte **Dynpros** (dynamische Programme) abgewickelt (vgl. zu Dynpros z.B. [Mend98, 189ff]). Ein Dynpro beschreibt hierbei

- den Aufbau der anzuzeigenden Bildschirmmaske in einer Form, die durch ein SAPGUI interpretiert werden kann, und
- die Beziehungen zwischen der Bildschirmmaske und dem zugehörigen ABAP/4-Programm.

Letzteres umfasst unter anderem die Information welche Programmfunktion zur Behandlung einer Benutzerinteraktion aufgerufen werden muss. Die Kommunikation zwischen Applikationsserver und SAPGUI wird über das DIAG-Protokoll abgewickelt.

Auch für eine ITS-Anwendung sind Dynpros zu erstellen. Ihnen kommt hier jedoch eine leicht veränderte Rolle zu (vgl. [PHMZ98, 161ff] und [PHMZ98, 173ff]). Sie beschreiben auch hier aus welchen Elementen eine Bildschirmmaske besteht. Das eigentliche Aussehen der Bildschirmmasken, insbesondere das Layout, wird allerdings in Form von **HTML-Seiten** festgelegt. Die Beziehungen zwischen einem Dynpro und der zugehörigen HTML-Seite werden in der HTML-Seite mit HTMLBusiness beschrieben, einer SAP-eigenen Erweiterung von HTML. HTMLBusiness ermöglicht z.B. die Zuordnung eines Dynpro-Eingabefelds zu einem HTML-Eingabefeld. Eine um HTMLBusiness-Elemente angereicherte HTML-Seite wird als HTML-Template bezeichnet.

Der ITS bildet das Bindeglied zwischen den R/3-Applikationsservern und einem Web-Server, der die Web-Clients bedient. Die Kommunikation zwischen ITS und Applikationsservern wird ebenfalls über das DIAG-Kommunikationsprotokoll abgewickelt.

Der **ITS** übernimmt in dieser Architektur vor allem die folgenden **Aufgaben**:

- Übertragen der in den Dynpros enthaltenen Daten in die vorbereiteten HTML-Templates.
- Übertragen der vom Web-Client gelieferten Daten in das am DIAG-Kommunikationskanal erwartete Dynpro-Format.
- Benutzer-Authentifizierung
- Synchronisation des Zustands von Web-Client und R/3-Anwendung. Hierfür muss vor allem die Zustandslosigkeit des HTTP-Protokolls überwunden werden.

Bei der **Darstellung einer Bildschirmmaske** in einem Web-Client ergibt sich folgender Ablauf:

1. Der Applikationsserver erstellt zunächst ein Dynpro mit den vom ABAP/4-Programm gelieferten Daten.
2. Dieses Dynpro wird über den DIAG-Kommunikationskanal an den ITS übermittelt.
3. Der ITS liest das zu dem Dynpro gehörende HTML-Template.
4. Der ITS überträgt die im Dynpro abgelegten Informationen in das HTML-Template.
5. Der ITS übergibt die fertige HTML-Seite an den Web-Server.
6. Der Web-Server übermittelt die HTML-Seite an den Web-Client.

5.4.4 Unterstütztes Integrationskonzept

Der ITS unterscheidet sich, wie schon angedeutet, von den anderen hier beschriebenen Kopplungsmechanismen dadurch, dass er keine direkte Kopplung von AwSen ermöglicht. Trotzdem kann der ITS zur Integration von AwSen genutzt werden und zwar im Sinne der in Abschnitt 3.2 beschriebenen aufgabenträgerorientierten Funktionsintegration. Zentrales Konzept der aufgabenträgerorientierten Funktionsintegration ist die Integration von AwSen durch gemeinsame personelle Aufgabenträger. Im Folgenden wird anhand eines Beispiels erläutert, wie der ITS zur Realisierung eines solchen Integrationskonzepts verwendet werden kann.

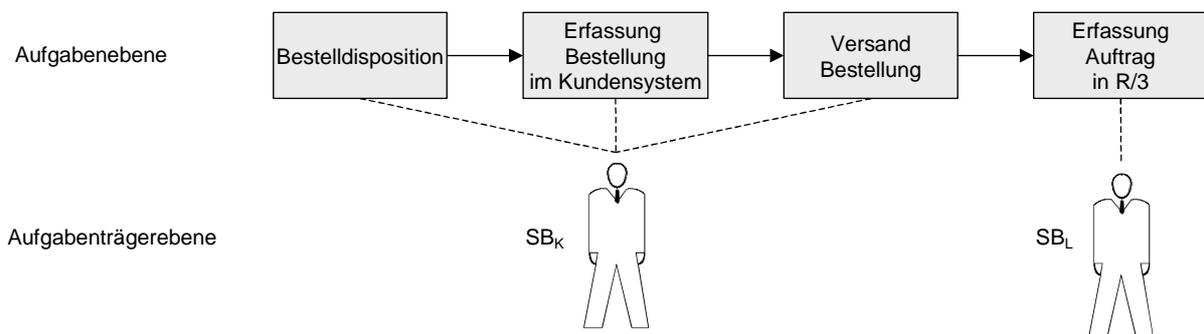


Abbildung 18: Bestellszenario ohne ITS-Anwendung

In Abbildung 18 ist ein typischer Bestellprozess dargestellt. Es wird unterstellt, dass der Lieferant seine Auftragsabwicklung mit Hilfe eines R/3-Systems durchführt. Der Sachbearbeiter des Kunden hat in diesem Szenario keinen Zugriff auf das R/3-System des

Lieferanten. Die Bestellung muss also nach Erfassung im Kundensystem an den Lieferanten übermittelt werden. Dort erfolgt anschließend die Erfassung in R/3.

Eine typische ITS-Anwendung wäre eine einfache Bestellmaske, die es dem Sachbearbeiter SB_K erlaubt, Aufträge direkt in das R/3-System des Lieferanten einzupflegen. Zur Bedienung dieser Maske sind keine R/3-Kenntnisse erforderlich, da sich die Benutzeroberflächen der IAC-Anwendungen nicht an der konventionellen SAP Oberfläche orientieren.

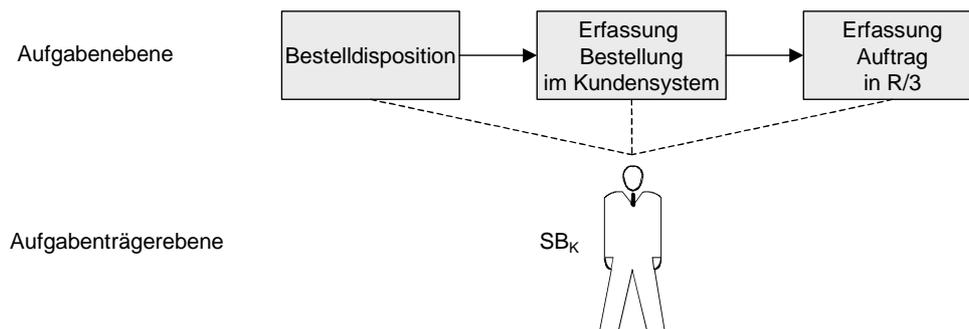


Abbildung 19: Bestellszenario mit ITS-Anwendung

Durch diese Bestellmaske kann der Sachbearbeiter des Kunden den gesamten Bestellprozess alleine durchführen (vgl. Abbildung 19). Ein Sachbearbeiter auf Seite des Lieferanten wird nicht benötigt. Die Transaktionskosten und die Transaktionszeit können reduziert werden.

Neben der reinen Auftragserfassung könnte die Bestellmaske z.B. noch zusätzliche Auskunftsfunktionen zur Unterstützung der Bestelldisposition anbieten. Der Sachbearbeiter des Kunden könnte hierdurch direkten Zugriff auf Informationen, wie z.B. Lieferzeiten und Preise, bekommen.

5.5 SAP Business Connector

Der SAP Business Connector kann nicht als eigenständiger Kopplungsmechanismus bezeichnet werden, da er nur einen kleinen Teilaspekt einer Kopplungsarchitektur abdeckt. Er soll trotzdem hier vorgestellt werden, da er ein wesentliches Teilelement der E-Commerce Strategie von SAP darstellt und beliebig mit den in den Abschnitten 5.1, 5.2 und 5.3 beschriebenen Kopplungsmechanismen kombiniert werden kann.

In der Fachwelt wird seit längerem **XML (Extensible Markup Language)** als wichtige Basistechnologie bei der Integration von AwSen diskutiert (vgl. zu XML z.B. [W3C98]). Als zentraler Einsatzbereich von XML wird dabei die Strukturierung der zwischen den AwSen

ausgetauschten Nachrichten gesehen. Man verspricht sich hiervon unter anderem folgende Vorteile (vgl. z.B. [Merz99, 347]):

- preiswerte und allgemein verfügbare Infrastruktur
- Unterstützung bei der Entwicklung von Nachrichtenstandards, z.B. durch das Konzept der XML-Namensräume und durch XML-Schemata

Dieses Paradigma wird mittlerweile auf breiter Basis unterstützt. Stellvertretend sollen hier nur OASIS (Organization for the Advancement of Structured Information Standards) und das BizTalk-Framework von Microsoft genannt werden (vgl. [Micr00]).

Mit dem **SAP Business Connector** bietet auch SAP eine XML-Schnittstelle zum System R/3 an (vgl. z.B. [SAP01b] und [SAP00]). Es handelt sich allerdings nicht um eine vollständig neue Schnittstelle. Der SAP Business Connector erlaubt nur einen in gewissem Maße vereinheitlichten XML-basierten Zugriff auf die bereits existierenden Schnittstellen.

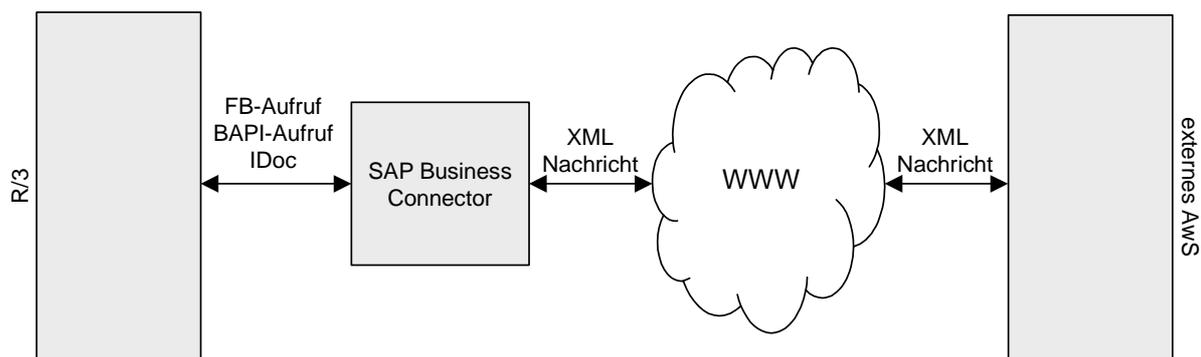


Abbildung 20: SAP Business Connector

Der hier beschriebene Stand des SAP Business Connector entspricht der Version 3.1 in Kombination mit der IfR-Erweiterung (Interface Repository). Diese Version unterstützt die Schnittstellentypen

- RFC-fähige Funktionsbausteine (vgl. [SAP01b, 19ff]),
- BAPIs (vgl. [SAP01b, 9ff]) und
- IDocs (vgl. [SAP01b, 17f]).

Der SAP Business Connector ist, wie in Abbildung 20 dargestellt, als eigenständige Komponente außerhalb des R/3-Systems realisiert. Er erfüllt zwei zentrale **Aufgaben**:

- **Entgegennahme von XML-Aufruf-Nachrichten** von externen AwSen und Übermittlung dieser Nachrichten in Form von BAPI-Aufrufen, Funktionsbaustein-Aufrufen oder IDocs an das Ziel-R/3-System. Entgegennahme der Ergebnisse der Aufrufe, Umsetzung der Ergebnisse in XML-Antwort-Nachrichten und Übermittlung der Nachrichten an das aufrufende AwS.
- **Entgegennahme von BAPI-Aufrufen, Funktionsbaustein-Aufrufen oder IDocs** aus einem R/3-System, Umsetzung dieser Aufrufe / IDocs in zugehörige XML-Aufruf-Nachrichten und Übermittlung dieser Nachrichten an die Ziel-AwSe. Entgegennahme der zugehörigen XML-Antwort-Nachrichten und Übergabe dieser Nachrichten im entsprechenden Format an das R/3-System, das den Aufruf durchgeführt hat.

Die Übermittlung der XML-Nachrichten erfolgt hierbei mittels HTTP oder HTTPS.

```
<doc:SalesOrder.ChangeFromData xmlns:doc="urn:sap-com:document:sap:business"
SalesDocument="1234">
  <OrderHeaderIn>
    <COLLECT_NO>...</COLLECT_NO>
    <SALES_ORG>...</SALES_ORG>
    ...
  </OrderHeaderIn>
  ...
  <OrderItemsIn>
    <item>
      <ITM_NUMBER>...</ITM_NUMBER>
      <HG_LV_ITEM>...</HG_LV_ITEM>
      <PO_ITM_NO>...</PO_ITM_NO>
      ...
    </item>
    ...
  </OrderItemsIn>
  ...
</doc:SalesOrder.ChangeFromData>
```

Abbildung 21: Business Document [SAP01b, 11]

Der Business Connector unterstützt verschiedene **Formate zur Repräsentation der Aufrufe, Antworten und IDocs**. Wir wollen uns hier auf die in [SAP01b] dargestellten Formate beschränken. Eine XML-Nachricht besteht hierbei immer aus den zwei Elementen

- Umschlag und
- Business Document.

Das **Business Document** enthält die eigentlichen Informationen, die benötigt werden, um den Aufruf, das Ergebnis des Aufrufs oder das zu übermittelnde IDoc zu beschreiben. In **Fehler! Verweisquelle konnte nicht gefunden werden.** ist ausschnittsweise ein Business Document dargestellt, das zum Aufruf des BAPI ChangeFromData des Objekttyps SalesOrder verwendet werden kann. Mit Hilfe dieses BAPI kann ein Kundenauftrag verändert werden. Das Business Document beschreibt

- auf welches BAPI sich die Nachricht bezieht („SalesOrder.ChangeFromData“),
- auf welche Objektinstanz sich die Nachricht bezieht („SalesDocument=1234“) und
- welche Daten dem BAPI beim Aufruf in seiner Schnittstelle übergeben werden sollen („<COLLECT_NO>...</COLLECT_NO >...“).

Zur Serialisierung der Schnittstellenparameter verwendet der SAP Business Connector die in [SAP01c] festgelegten Umsetzungsregeln.

Der **Umschlag** einer XML-Nachricht dient zum einen als Transportmittel für das Business Document, zum anderen enthält er zusätzliche technische Informationen, die bei der Nachrichtenübermittlung benötigt werden, wie z.B. den Empfänger der Nachricht und eine eindeutige ID für die Nachricht. Der Business Connector unterstützt den in [Micr00] beschriebenen BizTalk-Umschlag. Die XML-Nachricht besteht hierbei aus einem vollständigen XML-Dokument, wobei die Body-Sektion das eigentliche Business Document und die Header-Sektion die technischen Informationen entsprechend der BizTalk-Spezifikation enthält.

6 Zusammenfassung und Fazit

In der vorliegenden Arbeit wurden eine Reihe von Kopplungsmechanismen der Standardanwendungssoftware SAP R/3 hinsichtlich ihrer Eignung zur Realisierung von Integrationskonzepten untersucht. In die Betrachtung einbezogen wurden die Kopplungsmechanismen

- Business Application Programming Interfaces,
- Intermediate Documents,
- Application Link Enabling
- und Internet Transaction Server,

wobei der Internet Transaction Server nur als Kopplungsmechanismus im erweiterten Sinne verstanden werden kann. Wir konnten feststellen, dass jedes der in [FeSi01, 220ff] beschriebenen Integrationskonzepte durch R/3 grundsätzlich unterstützt wird. Die grundlegenden Teilkonzepte der Integrationskonzepte wurden in jedem Fall vollständig unterstützt, die weiterführenden Teilkonzepte allerdings nur teilweise. Das Integrationskonzept Datenintegration wird von R/3 nur im Rahmen der Intra-AwS-Integration also zur Integration innerhalb eines R/3-Systems eingesetzt.

Erfreulich ist, dass SAP sich nicht darauf beschränkt reine Kommunikationsmechanismen anzubieten. Die beschriebenen Kopplungsmechanismen decken zumindest innerhalb von R/3 die systemtechnischen Aspekte einer Kopplungsarchitektur weitestgehend ab. Auch Teile der fachlichen Architektur werden bereits mitgeliefert, wobei hier sicherlich noch Verbesserungspotential besteht. Hinsichtlich der durch die Kopplungsmechanismen angebotenen Funktionen ist allerdings eine erhebliche Asymmetrie festzustellen. Nicht-SAP-Plattformen werden durch SAP häufig nur in Form von einfachen Kommunikationsmechanismen, wie z.B. dem Remote Function Call, unterstützt.

Unter den untersuchten Kopplungsmechanismen hat sich der BAPI-Kopplungsmechanismus als der am weitesten entwickelte erwiesen. Er ermöglicht auf Basis der zahlreichen mit R/3 ausgelieferten Business Objects und der zugehörigen BAPIs einen an fachlichen Merkmalen orientierten Zugriff auf die Daten und Funktionen eines R/3-Systems. Auf systemtechnischer Ebene erlaubt er einen Zugriff auf R/3 über weit verbreitete moderne Middleware-Technologien, wie z.B. CORBA und DCOM.

Beim Einsatz der Kopplungsmechanismen stellt sich das Problem der Ermittlung einer für einen konkreten Anwendungsfall geeigneten Schnittstelle in der großen Menge der verfügbaren Schnittstellen. Erschwerend kommt hierbei hinzu, dass häufig für einen bestimmten Zweck mehrere Schnittstellen zur Verfügung stehen. SAP löst dieses Problem mittlerweile zumindest teilweise mit dem unter <http://ifr.sap.com> öffentlich verfügbaren Interface Repository (IfR). In diesem Repository sind zur Zeit ca. 2000 BAPIs, 250 IDocs und 250 RFC-fähige Funktionsbausteine in Form von XML-Schemata dokumentiert.

Als besonders problematisch erwies sich bei der Erstellung des Arbeitsberichts die teilweise recht hohe Komplexität der betrachteten Mechanismen in Verbindung mit der geringen Qualität der Dokumentation. Eine wesentliche Ursache für die hohe Komplexität ist die Tatsache, dass es sich bei SAP R/3 um ein über viele Jahre gewachsenes AwS handelt. Dieses

Problem tritt besonders an den Schnittstellen zu Tage, da hier bei der Weiterentwicklung des Systems auf Abwärtskompatibilität geachtet werden muss.

Zu einem ähnlichen Ergebnis bzgl. der von R/3 angebotenen Kopplungsmechanismen kommt Linthicum in seinem Buch zu B2B Application Integration:

“The upside is that unlike many packaged applications, SAP has well-defined and well-tested built-in interfaces that adequately provide for the sharing of data and processes with the outside world. The downside is that these same interfaces are complex, their architecture is confusing, information is lacking about how best to leverage them, and various other technical issues still require resolution.” [Lint00, 349]

7 Literatur

- [Ball00] Ballnus, R.: Erfolg mit EDI und E-Commerce. Tectum, Marburg 2000.
- [Balz96] Balzert, H.: Lehrbuch der Softwaretechnik – Software-Entwicklung. Spektrum, Heidelberg 1996.
- [Buck98] Buck-Emden, R.: Die Technologie des SAP R/3-Systems. 4. Auflage, Addison-Wesley, Bonn 1998.
- [BuKö00] Buxmann, P.; König, W.: Zwischenbetriebliche Kooperationen auf Basis von SAP-Systemen. Springer, Berlin 2000.
- [Deut95] Deutsch, M.: Unternehmenserfolg mit EDI - Strategie und Realisierung des elektronischen Datenaustausches. Vieweg, Braunschweig 1995.
- [Fers92] Ferstl, O. K.: Integrationskonzepte Betrieblicher Anwendungssysteme. Fachberichte Informatik der Universität Koblenz-Landau Nr.1/1992.
- [FeSi01] Ferstl, O. K.; Sinz, E. J.: Grundlagen der Wirtschaftsinformatik. 4. Auflage, Oldenbourg, München 2001.
- [FSH+97] Ferstl, O. K.; Sinz, E. J.; Hammel, Ch.; Schlitt, M.; Wolf, S.: Bausteine für komponentenbasierte Anwendungssysteme. In: HMD 197/1997, S. 24 – 46.
- [Iso88] ISO: Electronic data interchange for administration, commerce and transport (EDIFACT) - Application level syntax rules (ISO 9735). ISO, 1988.
- [Iso97] ISO: The Open-EDI Reference Model ISO/IEC 14662. ISO, 1997.

- [Lint00] Linthicum, D. S.: B2B Application Integration - e-Business-Enable Your Enterprise. Addison-Wesley, Boston 2000.
- [Mend98] Mende, U.: Softwareentwicklung für R/3 - Data Dictionary, ABAP/4, Schnittstellen. Springer, Berlin 1998.
- [Merz99] Merz, M.: Electronic Commerce – Marktmodelle, Anwendungen und Technologien. dpunkt, Heidelberg 1999.
- [Micr00] Microsoft AG: BizTalk Framework 2.0: Document and Message Spezifikation. <http://www.microsoft.com/biztalk/techinfo/BizTalkFramework20.doc>, 2000, Abruf am 2001-02-05.
- [MKR+01] Mantel, S.; Knobloch, K.; Ruffer, T.; Schissler, M.; Schmitz, K.; Ferstl, O. K.; Sinz, E. J.: Analyse der Integrationspotenziale von Kommunikationsplattformen für verteilte Anwendungssysteme. FORWIN-Bericht FWN-2000-009, 2001.
- [Mose99] Moser, G.: SAP R/3 Interfacing using BAPIs - A practical guide to working within the SAP Business Framework. Vieweg, Wiesbaden 1999.
- [Nagp99] Nagpal, A.: ALE, EDI, & IDoc Technologies for SAP. Prima, Rocklin California 1999.
- [PHMZ98] Perez, M.; Hildenbrand, A.; Matzke, B.; Zencke, P.: Geschäftsprozesse im Internet mit SAP R/3. Addison-Wesley, Bonn 1998.
- [ReEc99] Rensmann, J.; Eckardt, F.: SAP R/3 - Programmierung mit Delphi/Connect. Addison-Wesley, Bonn 1999.
- [SAP00] SAP AG: SAP Interface Repository Add-On for SAP Business Connector. <http://ifr.sap.com/home/documents/IFRAddOnGuide.pdf>, Abruf am 2001-02-05.
- [SAP01a] SAP AG: SAP Business Connector – Comprehensive XML collaboration over the Internet. <http://www.sap-ag.de/solutions/technology/pdf/50033842.pdf>, 1999, Abruf am 2001-02-01.
- [SAP01b] SAP AG: Representation of XML Interfaces in the SAP Interface Repository. http://ifr.sap.com/home/documents/XML_E.pdf, Abruf am 2001-02-02.
- [SAP01c] SAP AG: Serialization of ABAP Data in XML. http://ifr.sap.com/home/documents/ABAP_Serialization.htm, Abruf am 2001-02-12.

- [SAP01d] SAP AG: BAPI-Programmierleitfaden. <http://www.sap-ag.de/solutions/technology/bapis/edu/docu/45a/prog45ad.doc>, Abruf am 2001-02-12.
- [SAP01e] SAP AG: BAPI-Benutzerhandbuch. <http://www.sap-ag.de/solutions/technology/bapis/edu/docu/45a/useg45ad.doc>, Abruf am 2001-02-12.
- [SAP99] SAP AG: SAP Online Help Release 4.5B. 1999.
- [Schü98] Schüppler, D.: Informationsmodelle für überbetriebliche Prozesse. Lang, Frankfurt 1998.
- [SiKM00] Sinz, E. J.; Knobloch, B.; Mantel, S.: Web-Application-Server. In: Wirtschaftsinformatik, 42. Jg. (2000), Heft 6, S. 550-552.
- [Sinz97] Sinz, E. J.: Architektur betrieblicher Informationssysteme. Bamberger Beiträge zur Wirtschaftsinformatik der Universität Bamberg Nr. 40.
- [Weli00] Welingkar, B.: Publishing Business Events out of R/3. <http://www.intelligenterp.com/columns/welingkarSep.shtml>, 2000, Abruf am 26.4.2001.
- [W3C98] World Wide Web Consortium: Extensible Markup Language (XML) 1.0 - W3C Recommendation. <http://www.w3.org/TR/REC-xml>, 1998, Abruf am 26.4.2001.

Folgende FORWIN-Berichte sind bisher erschienen:

FWN-2000-001

Mertens, P.

FORWIN – Idee und Mission

E-Business * Supply Chain Management * Betriebliche Software-Bausteine

FWN-2000-002

Sinz, E. J.

Die Projekte im Bayerischen Forschungsverbund Wirtschaftsinformatik (FORWIN)

FWN-2000-003

Kaufmann, Th.

Marktplatz für Bausteine heterogener betrieblicher Anwendungssysteme

FWN-2000-004

Schaub, A.; Zeier, A.

Eignung von Supply-Chain-Management-Software für unterschiedliche Betriebstypen und Branchen – untersucht am Beispiel des Produktions-Prozessmodells zum System SAP APO

FWN-2000-005

Friedrich, M.

Konzeption eines Componentware-basierten Supply-Chain-Management-Systems für kleine und mittlere Unternehmen

FWN-2000-006

Schmitzer, B.

Klassifikationsaspekte betriebswirtschaftlich orientierter Frameworks

FWN-2000-007

Zeier, A.; Hauptmann, S.

Ein Beitrag zu einer Kern-Schalen-Architektur für Supply-Chain-Management (SCM)-Software, Teil I: Anforderungen an den Kern einer SCM-Software und deren Abdeckung in SAP APO 2.0/3.0

FWN-2000-008

Maier, M.

Bestandsaufnahme zu Jobbörsen im WWW

FWN-2000-009

Mantel, S.; Knobloch, B.; Ruffer, T.; Schissler, M.; Schmitz, K.; Ferstl, O. K.; Sinz, E. J.

Analyse der Integrationspotenziale von Kommunikationsplattformen für verteilte Anwendungssysteme

FWN-2000-010

Franke, Th.; Barbian, D.

Platform for Privacy Preferences Project (P3P) - Grundsätze, Struktur und Einsatzmöglichkeiten im Umfeld des "Franken-Mall"-Projekts

FWN-2000-011

Thome, R.; Hennig, A.; Ollmert, C.

Kategorisierung von eC-Geschäftsprozessen zur Identifikation geeigneter eC-Komponenten für die organisierte Integration

FWN-2001-001

Zeier, A.; Hauptmann, S.

Ein Beitrag zu einer Kern-Schalen-Architektur für Supply-Chain-Management (SCM)-Software, Teil II: Anforderungen an die Schalen einer SCM-Software und deren Abdeckung in SAP APO 2.0/3.0

FWN-2001-002

Lohmann, M.

Die Informationsbank ICF – eine wissensbasierte Werkzeugsammlung für die Software-Anforderungsanalyse

FWN-2001-003

Hau, M.

Das DATEV-Komponenten-Repository - Ein Beitrag zu Marktplätzen für betriebswirtschaftliche Software-Bausteine

FWN-2001-004

Schoberth, Th.

Virtual Communities zur Unterstützung von Infomediären

FWN-2001-005

Kronewald, K.; Menzel, G.; Taumann, W.; Maier, M.

Portal für bürgergerechte Dienstleistungen in der Sozialen Sicherheit

FWN-2001-006

Maier, M.

Strukturen und Prozesse im "Netzwerk für Arbeit"

FWN-2001-007

Maier, M.; Gollitscher, M.

Überlegungen zum Skill-Matching-Modul eines Leitstands für den regionalen, zwischenbetrieblichen Personalaustausch

FWN-2001-008

Schissler, M.; Mantel, S.; Ferstl, O. K.; Sinz, E. J.

Unterstützung von Kopplungsarchitekturen durch SAP R/3