
Flexibilisierung von Organisations- und IT-Architekturen durch EAI

Stephan Aier und Dr. Marten Schönherr

Technische Universität Berlin
Competence Center EAI

Zusammenfassung. Enterprise Application Integration lässt sich nicht auf die Betrachtung technischer Komponenten beschränken. Solch weitreichende Integrationsstrategien bedürfen entsprechenden organisatorischen Veränderungen bei der Implementierung und müssen schon wegen der großen Komplexität, Bedeutung und Tragweite der Projekte in eine Gesamtunternehmensarchitektur eingebettet werden. Das Oberziel derartiger Infrastrukturprojekte ist die nachhaltige Flexibilisierung von Unternehmensarchitekturen. Der Beitrag schlägt, beginnend mit dem Paradigma der Flexibilität von Unternehmensarchitekturen, einen Bogen von der integrierten Modularisierung von Organisations- und IT-Strukturen, hin zu den derzeit bekanntesten Möglichkeiten des Architekturmanagements. Letztlich ergibt sich jedoch auch hier ein weites Feld offener Fragen am Ende des Beitrags.

Schlüsselwörter: Unternehmensarchitektur, Enterprise Application Integration, Systemintegration, Modularisierung

1	Problemstellung	5
2	Notwendigkeit flexibler Unternehmensarchitekturen	6
2.1	Sichten des Unternehmungswandels	7
2.2	Stärkung der Wandlungsfähigkeit	9
3	EAI im Spannungsfeld von Organisation und IT	10
3.1	Problemstellung der Systemintegration	11
3.2	Konzepte der EAI	13
3.2.1	Zentrale Integrationsplattform	13
3.2.2	Definition EAI	14
3.2.3	Integrationsziele und Nutzen von EAI	16
3.2.4	Kritische Punkte	18
3.3	Organisation und Systemintegration	18
3.3.1	Organisation der Systemintegration	19
3.3.2	Wechselwirkungen zwischen Organisation und Systemintegration	19
4	Modularisierung	25
4.1	Modularisierung der Organisation	27
4.1.1	Modularisierung auf Makroebene	28
4.1.2	Modularisierung auf Mikroebene	30
4.1.3	Realisierungsprinzipien	32
4.1.4	Probleme und Grenzen	34
4.2	Modularisierung der IT	34
4.2.1	User Interface Integration	35
4.2.2	Data Level Integration	36
4.2.3	Application Interface Integration	36
4.2.4	Method Integration	37
4.2.5	Service based Integration	39
4.2.6	Process Integration	40
5	Management modularisierter Architekturen	41

4	Stephan Aier, Marten Schönherr	
5.1	Definition Architektur	41
5.2	Holistische Architekturkonzepte	45
6	Fazit	49

1 Problemstellung

Eine der wichtigsten Erkenntnisse des EAI-Expertentages für Banken und Versicherungen im Februar 2003 war die Tatsache, dass unternehmensweite Systemintegration sich nicht auf die Betrachtung der technischen Komponenten beschränken lässt. So hat die HypoVereinsbank ein Konzept vorgestellt, in dem organisatorische Cluster eingeführt wurden, für die individuelle Integrationsstrategien entwickelt werden sollen. Die Credit Suisse nannte die Vorgehensweise anders, sprach aber über ähnliche Methoden und Ansätze.¹ Weitere Beiträge diskutierten die Etablierung serviceorientierter Architekturen, in denen sich die Granularität der Services mehr an die organisatorischen Anforderungen als die technischen Entwicklungsrichtlinien zu halten hatten. Natürlich geht es weiterhin um ein sehr technisch getriebenes Thema, allerdings erkennen immer mehr Unternehmen, dass die holistische Betrachtung von EAI im Gesamtunternehmenskontext einer der wichtigsten Erfolgsfaktoren für die nachhaltige Flexibilisierung von Unternehmensarchitekturen darstellt.

In diesem Beitrag werden die Kernprobleme und Diskussionspunkte des vorliegenden Buches – welches im Wesentlichen auf den Beiträgen zum EAI-Expertentag im Februar 2003 des Competence Centers für EAI in Banken und Versicherungen an der Technischen Universität Berlin basiert – im Zusammenhang dargestellt sowie verschiedene Lösungsansätze aufgezeigt.

Die Finanzdienstleistungsbranche unterliegt diesbezüglich einem besonderen Handlungsdruck, da hier praktisch alle Geschäftsprozesse durch eine starke IT-Unterstützung gekennzeichnet sind. Dies führte zu sehr komplexen und über lange Zeit gewachsenen Systemlandschaften mit zum Teil vergleichsweise alten, monolithischen aber auch sehr wichtigen Kernsystemen, die in einer komplexen Struktur miteinander verwoben sind.² MORGAN spricht in diesem Zusammenhang von der annähernden Gleichbedeutung von IT und Unternehmung.³

Wie sich zeigt, lässt sich diese technische Sichtweise jedoch nicht losgelöst von den Fragen der Unternehmungsorganisation – den Strukturen und Prozessen – behandeln. Zwischen beiden bestehen komplexe Wechselwirkungen und Abhängigkeiten.⁴ Um derart komplexe Gestaltungsaufgaben zu lösen, werden vermehrt Architekten zur integrierten Gestaltung von Organisation und IT zu einer nachhaltig flexiblen Unternehmungsarchitektur eingesetzt.

¹ Vgl. den Beitrag von HAGEN in diesem Buch S. 61 ff.

² Vgl. MOORMANN (1999b), S. 10 f.; HASSELBRING (2000), S. 34 ff.

³ Vgl. MORGAN (1997), S. 117.

⁴ Vgl. FRESE (2002), S. 200 ff.

Der vorliegende Beitrag wird die Frage nach der Entwicklung einer flexiblen Unternehmensarchitektur problematisieren und erste Lösungsansätze diskutieren. Dazu wird zuerst der Dauerzustand Wandel in Unternehmen beschrieben, danach werden EAI und Unternehmensorganisation in diesen Kontext eingeordnet. Darauf basierend werden Vorschläge zur Flexibilisierung von Unternehmen durch die Modularisierung ihrer Organisation und IT im Rahmen eines beide Bereiche umfassenden Architekturmanagements problematisiert. Letztendlich ergibt sich jedoch auch hier ein weites Feld von offenen Fragen, mit denen der Beitrag endet.

2 Notwendigkeit flexibler Unternehmensarchitekturen

Hier wird unter Unternehmensarchitektur das Zusammenwirken technologischer, organisatorischer und psychosozialer Aspekte bei der Entwicklung und Nutzung von betrieblichen soziotechnischen Informationssystemen verstanden.⁵ Das System Unternehmung ist den vielschichtigen Veränderungen seiner Umwelt ausgesetzt. Die drei wesentlichen Dimensionen der Umwelt von Organisationen sind *Komplexität*, *Dynamik* und *Abhängigkeit*.⁶

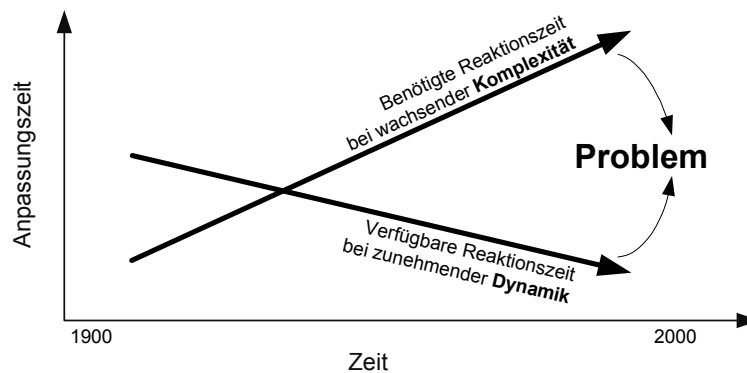


Abbildung 1. Zeitschere als Managementproblem, in Anlehnung an KRYSTEK (1999), S. 278

Insbesondere die Dimensionen Komplexität und Dynamik (Abbildung 1),⁷ verstärkt durch aktuelle Entwicklungen wie Globalisierung und Technisie-

⁵ Vgl. GRONAU (2003), S. 45; Aufgrund der besonderen Bedeutung des Begriffs *Architektur* wird dieser ausführlich im Kapitel 5 diskutiert.

⁶ Vgl. JURKOVICH (1974), S. 380 ff.

⁷ Vgl. HORVÁTH (1998), S. 3 ff.; HAHN/HUNGENBERG (2001), S. 109; KRYSTEK (1999), S. 266.

rung,⁸ stellen Unternehmungen vor Entscheidungs- und Handlungsprobleme zu deren Lösung sie grundsätzlich zwei Möglichkeiten der Reaktion haben:⁹

1. Sie können Maßnahmen zur Einwirkung auf die Umwelt mit dem Ziel ergreifen, Dynamik und/oder Komplexität durch Verringerung der Abhängigkeiten zu reduzieren.
2. Oder sie können die Anpassungsfähigkeit der Unternehmung erhöhen.¹⁰

Im vorliegenden Beitrag wird der zweite Punkt – die Erhöhung der Anpassungsfähigkeit der Unternehmung – weiter untersucht. Die zusammenfassende Erkenntnis, dass „Unternehmenswandel von einem in größeren zeitlichen Abständen zu organisierenden Ereignis zu einem Dauerzustand geworden ist bzw. werden muss“,¹¹ ist heute weithin akzeptiert. Die Aufgabe besteht nun darin, Unternehmungen für diesen Wandel zu flexibilisieren.

2.1 Sichten des Unternehmungswandels

KRÜGER differenziert den Wandel von Unternehmungen nach dem Wandlungsbedarf, der Wandlungsbereitschaft und der Wandlungsfähigkeit (Abbildung 2).¹²

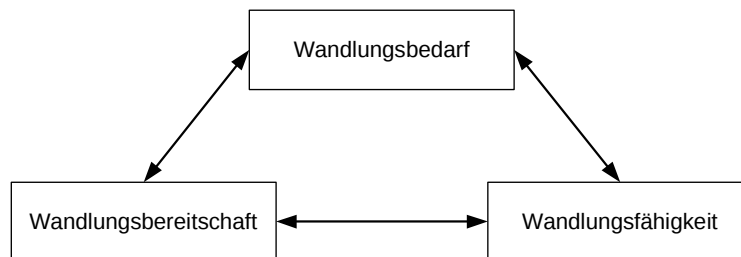


Abbildung 2. Elemente des Wandels, in Anlehnung an KRÜGER (1998), S. 228

⁸ Vgl. HAHN (1999a), S. 1038; MOORMANN (1999b), S. 5; FRESE (2000), S. 128.

⁹ Vgl. KIESER/KUBICEK (1992), S. 376 f.

¹⁰ Aus systemtheoretischer Sicht wird dies durch ASHBYS Gesetz der erforderlichen Varietät („Law of requisite variety“) ausgedrückt, wonach ein System die Varietät der Umwelt nur im Umfang der eigenen Varietät kompensieren kann; vgl. ASHBY (1970), S. 124 ff. „Only variety can destroy variety.“ ASHBY (1970), S. 207.

¹¹ KRÜGER (1998), S. 228.

¹² Vgl. dazu und zum Folgenden KRÜGER (1998), S. 228 ff.

Wandlungsbedarf ist danach das Ausmaß sachlich notwendiger Veränderungen der Unternehmung bzw. ihrer Subsysteme. Systeminterne oder -externe Impulse lösen Wandlungsbedarf aus, führen aber noch nicht notwendiger Weise auch zu Wandel. Werden Wandlungsprozesse nicht eingeleitet, so bauen sich Wandlungsbedarfe auf und werden oft erst bei erheblichem Problemdruck korrigiert.¹³ Externe Impulse ergeben sich, bspw. für die Bankenbranche, durch veränderte rechtliche Rahmenbedingungen mit Auswirkungen auf die Organisation und IT, wie dies durch Basel II oder die Mindestanforderungen an das Betreiben von Kreditgeschäften (MaK) der Fall ist,¹⁴ neue strategische Gestaltungsmöglichkeiten durch moderne Technologien oder notwendige Strukturänderungen durch den Konsolidierungsdruck in der Bankenbranche¹⁵.

Wandlungsbereitschaft ist im Gegensatz zum objektiven Bedarf die subjektive Bereitschaft bzw. Akzeptanz des Wandels. Hier spielt die externe Verhaltensakzeptanz und die interne Einstellungsakzeptanz¹⁶ der beteiligten Personen eine entscheidende Rolle, welche diese zu Promotoren oder Opponenten des Wandels werden lässt. In der Praxis ist oft zu beobachten, dass Promotoren von Opponenten dominiert werden.¹⁷ Darum ist es notwendig, aktiv Promotoren für ein Wandlungsprojekt aufzubauen und Opponenten planvoll zu integrieren. Betont sei die Rolle der Unternehmungskultur zur Schaffung von Wandlungsbereitschaft in einer Unternehmung.¹⁸

Die *Wandlungsfähigkeit* stellt den Kern der Betrachtungen dieses Beitrags dar. Wandlungsfähigkeit lässt sich auf drei Ebenen, der Ebene des Individuums, der Ebene der organisatorischen Einheit und der Ebene der Unternehmung, betrachten. Hier werden vor allem die Ebene der Unternehmung, also des Gesamtsystems, sowie ihrer organisatorischen Einheiten, den Subsystemen, betrachtet.

Die Wandlungsfähigkeit höherer Systemebenen ergibt sich dabei aus den Fähigkeiten der niedrigeren Ebenen. Umgekehrt wird die Wandlungsfähigkeit

¹³ „Zugespitzt formuliert: ohne Krise kein Wandel („No breakthrough without breakdown“).“ KRÜGER (1998), S. 230.

¹⁴ Vgl. LAUPENMÜHLEN/MOSCHITZ (2002).

¹⁵ Vgl. GROUP OF TEN (2001).

¹⁶ Die *Verhaltensakzeptanz* ist durch das Verhalten der beteiligten Personen erkennbar, während die *Einstellungsakzeptanz* die tatsächliche Meinung über die Veränderungen widerspiegelt und oft im Verborgenen bleibt.

¹⁷ Vgl. KRÜGER (1998), S. 235.

¹⁸ Vgl. KIESER/KUBICEK (1992), S. 125; vgl. grundlegend zur Unternehmungskultur SCHEIN (1985), S. 3 ff.; BLEICHER (1999), S. 223 ff. sowie zu ihren Funktionen ULRICH (1984), S. 312 f.; FÖHR/LENZ (1992), S. 138 ff.

durch entsprechende Infrastrukturen und Maßnahmen der übergeordneten Ebenen beeinflusst. Bei gegebenem permanenten Wandlungsbedarf und vorausgesetzter Wandlungsbereitschaft gilt es, die Wandlungsfähigkeit strukturell zu erhöhen.

2.2 Stärkung der Wandlungsfähigkeit

Um mit intern und extern ausgelösten Wandlungsbedarfen effektiv umzugehen, sind in den Unternehmungen entsprechende Strukturen und organisatorische Instanzen zur Institutionalisierung von Wandel zu schaffen. Exemplarisch sei hier die mögliche Spannbreite an Maßnahmen von kontinuierlichen Prozessverbesserungen¹⁹ durch die Installation eines betrieblichen Vorschlagswesens²⁰ oder Kaizen²¹ bis zum Einsatz rein projektorientierter Task Forces angedeutet.

Neben institutionellen Maßnahmen werden in diesem Beitrag insbesondere strukturelle Maßnahmen zur Erhöhung der Wandlungsfähigkeit – im Folgenden Maßnahmen zur Flexibilisierung genannt – diskutiert.

Der Begriff der *Flexibilität* entstammt dem Lateinischen und bedeutet Veränderbarkeit, Beweglichkeit oder Biegsamkeit. Flexibilität ist die Eigenschaft eines Systems, bei gegebenem Wandlungsbedarf die Wandlungsfähigkeit bzw. das Wandlungspotenzial aktivieren zu können. Ein System ist dann flexibel, wenn dem Wandlungsbedarf ein in angemessener Zeit aktivierbares Wandlungspotenzial im System gegenübersteht.²²

Der Wandlungsbedarf enthält eine sachliche und eine zeitliche Dimension. Gefordert wird die Durchsetzung von Änderungen in einem bestimmten Zeitablauf. Daher ist das Wandlungspotenzial als Einflussfaktor auf die Flexibilität eines Systems um das Reaktionsvermögen zu ergänzen. Die Reaktionsfähigkeit bezieht sich jedoch nur auf die nachträgliche Anpassung an eine bestehende Anforderung. Gefordert wird, auch die antizipative Anpassung der Unternehmung, etwa durch Erkennen und Ausnutzen von Marktchancen, in die Definition des Flexibilitätsbegriffes einzubeziehen.²³ Flexibilität umfasst daher auch die Schaffung von Chancen für eine Unternehmung im Sinne von Innovationen.²⁴

¹⁹ Vgl. ÖSTERLE (1995), S. 22 f.

²⁰ Vgl. THOM (1996).

²¹ Vgl. IMAI (1993).

²² Vgl. KIESER/KUBICEK (1992), S. 378; KNOF (1992), S. 67; GRONAU (2000b), S. 127 f.

²³ Vgl. CORSTEN (1988).

²⁴ Vgl. KNOF (1992), S. 69.

Im Kontext dieses Beitrags meint Flexibilität vor allem Anpassungen der Geschäftsprozesse und sich daraus ergebende Auswirkungen auf die IT.

HILL/FEHLBAUM/ULRICH bezeichnen die Fähigkeit von Unternehmen, sich gut an wechselnde Umweltbedingungen anzupassen als Produktivität zweiter Ordnung. Bei dem Ziel, Unternehmungen und deren Organisation zu flexibilisieren um eine hohe Produktivität zweiter Ordnung zu erreichen, handelt es sich prinzipiell um ein konkurrierendes Gestaltungsziel zu dem Ziel, bei konstanten Anforderungen eine hohe Produktivität zu erreichen (Produktivität erster Ordnung).²⁵ In der Bankenbranche wird diesbezüglich von einer Zielveränderung, weg von den Kompetenzen ‚to run the bank‘ hin zu den Kompetenzen ‚to change the bank‘ gesprochen.

Insbesondere folgende Ausprägungen der Struktur von Unternehmungen sind geeignet die Flexibilität zu erhöhen:²⁶

- geringe Spezialisierung auf Stellen- und Abteilungsebene
- starke Dezentralisierung
- flache Hierarchien
- Minimierung der Stärke zentraler unterstützender Abteilungen (Stäbe)
- einfache, d. h. keine umfassenderen Matrixstrukturen
- verstärkter Einsatz von Selbstabstimmung und Organisationskultur zur Koordination innovativer Aktivitäten
- leichte Ergänzzbarkeit um temporäre Teams für größere innovative Vorhaben

Vor allem die erstgenannten Ausprägungen zielen auf eine Entkopplung der Strukturen und Prozesse durch eine Reduktion von Schnittstellen. Im Folgenden sollen diese Gedanken weiterentwickelt werden, um im Sinne einer Modularisierung auf verschiedenen Ebenen der Unternehmung, Entkopplung und flexible Rekonfiguration von Strukturen und Prozessen zu ermöglichen.²⁷

3 EAI im Spannungsfeld von Organisation und IT

Wie eingangs angedeutet, ist es heute nicht ausreichend ausschließlich formale organisatorische Gestaltungsaspekte bei der Flexibilisierung von Unter-

²⁵ Vgl. HILL/FEHLBAUM/ULRICH (1994), S. 164 ff.

²⁶ Vgl. KIESER/KUBICEK (1992), S. 382 ff. und die dort aufgeführten Studien.

²⁷ Vgl. Kapitel 4; PICOT/REICHWALD/WIEGAND (2001), S. 227 ff.

nehmungen zu betrachten. Vielmehr ist es notwendig auch die informations- und kommunikationstechnologischen Aspekte zu berücksichtigen und beide in einem integrierten Architekturmanagement-Ansatz zu verbinden.

Umgekehrt macht es wenig Sinn, Informationssysteme einzuführen, zu ändern oder deren Zusammenspiel anzupassen, ohne die Wechselwirkungen mit den Aufgaben und Prozessen, zu deren Unterstützung sie eingesetzt werden, zu berücksichtigen.²⁸

Die besondere Herausforderung liegt dabei in der Tatsache, dass es hier nicht um eine möglichst abgestimmte Einführung neuer Strukturen, Prozesse und IT-Systeme geht, sondern um das Management bestehender Strukturen, Prozesse und IT-Systeme, was die Komplexität des Vorhabens erheblich erhöht.

Im Folgenden wird EAI als ein umfassender Ansatz zur Anwendungsintegration dargestellt, der den Anspruch hat beide Bereiche – Organisation und IT – angemessen zu berücksichtigen.²⁹ Dazu werden zuerst kurz die Ausgangssituation der Systemintegration und die Ansätze der EAI skizziert, um dann zu zeigen, welche Berührungspunkte Systemintegration und Unternehmensorganisation haben und wie Fragestellungen der Organisation an EAI angebunden werden können.

3.1 Problemstellung der Systemintegration

Bedingt durch die zunächst zentral und später dezentral organisierte und vorangetriebene Systementwicklung und -verwaltung der Unternehmungen,³⁰ entstanden verschiedene, sehr komplexe Applikationen mit sich überschneidenden Funktionsumfängen. Die dezentrale, von Abteilungsinteressen und -anforderungen getriebene Systementwicklung führte zu heterogenen Strukturen bezogen auf Kriterien wie Hardwarebasis, Betriebssysteme, Entwicklungssysteme, Architekturprinzipien etc.³¹

Mitte der 1980er Jahre begannen speziell große Unternehmungen die Relevanz der Applikationsintegration zu erkennen. Getrieben wurde diese Ent-

²⁸ Vgl. KAIB (2002), S. 97 ff.; DERSZTELER (2000), S. 23 ff.; PICOT/REICHWALD/WIEGAND (2001), S. 195 ff.

²⁹ Vgl. KAIB (2002), S. 81.

³⁰ Die zentral bzw. dezentral organisierte Systementwicklung und -verwaltung wird in ihrer Gestaltung durch den zentralen bzw. dezentralen Einsatz von Rechnersystemen beeinflusst, welcher durch die Entwicklung von Großrechnern zu Kleinrechnern und PCs getrieben wird. Vgl. SOLARO (1998), S. 35 ff.; PICOT/REICHWALD/WIEGAND (2001), S. 147, 175 ff.; FRESE (2000), S. 129 ff.

³¹ Vgl. LINTHICUM (2000), S. 6 ff.; zu einem Praxisbeispiel vgl. KAIB (2002), S. 174 ff.

wicklung durch die zunehmend kostengünstige Verfügbarkeit von Netzwerktechnologie zur Bildung verteilter Rechnersysteme.³²

Als in den 1990er Jahren ERP-Systeme an Bedeutung gewannen, bestand in den Unternehmungen die Notwendigkeit, bereits vorhandene Applikationen mit ERP-Systemen zu verbinden. Es wurden nicht nur mehr Daten ausgetauscht, vielmehr sollten Geschäftsprozesse systemübergreifend unterstützt werden. Dies konnte nur durch die Entwicklung standardisierter Schnittstellen effizient realisiert werden. REN schreibt hierzu:

„Einst benutzten die Unternehmen die Client/Server-Technologien, um abteilungsbezogene Applikation aufzubauen, aber später bemerkten sie die enormen Vorteile, die dabei entstehen, wenn man die einzelnen Geschäftsprozesse und damit auch deren Applikationen miteinander verknüpft.“³³

Es entstand eine neue Herausforderung für IT-Abteilungen: die Integration immer komplexer werdender IT-Systemstrukturen. Umfangreiche Individualentwicklungen in Unternehmungen bzw. Unternehmungsteilen waren die Folge. Es entwickelte sich die in Abbildung 3 dargestellte Point to Point Integration.

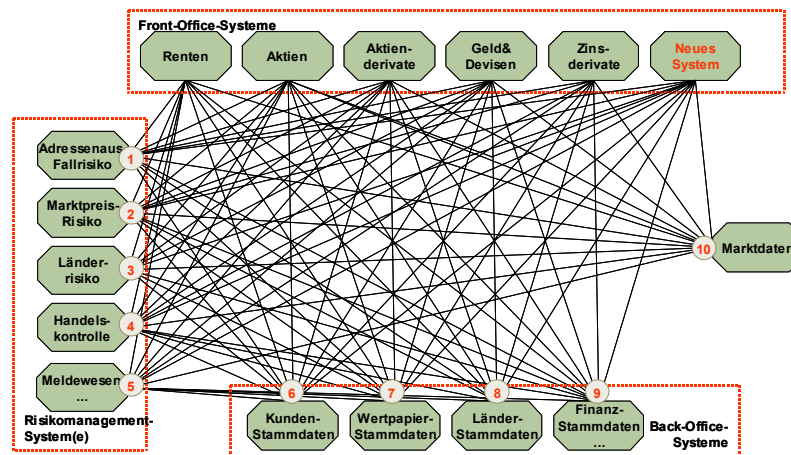


Abbildung 3. Point to Point Integration

³² Vgl. PICOT/REICHWALD/WIEGAND (2001), S. 147 ff.; FRESE (2000), S. 132 ff.

³³ REN (2002), S. 5.

Das sich im Folgenden ergebende Problem bestand und besteht in komplexen, zum Teil bereits klassisch integrierten Systemlandschaften, in welchen Änderungen, z. B. das Hinzufügen eines weiteren Systems, die Versionsumstellung bestehender Systeme, die Einbindung neuer Anforderungen aus der Änderung rechtlicher Rahmenbedingungen etc., eine Reihe weiterer, schwer zu beherrschender Änderungsnotwendigkeiten nach sich zogen und ziehen.

Weitere Relevanz erhält dieses Thema durch die Einführung zusätzlicher integrationsbedürftiger Systeme, z. B. im Rahmen von E-Business-Strategien, Supply-Chain-Management-Initiativen, ERP-Strategien sowie Mergers & Acquisitions.³⁴

Kritisch ist weiterhin die oft mangelnde Trennung der Fachebene sowie der IT-Ebene bei der Entwicklung von Point to Point Schnittstellen. Dadurch ist bei Änderungsanforderungen, ausgelöst durch technischen Änderungsbedarf oder fachliche Veränderungen, ein oft doppelter Aufwand notwendig, um Anpassungen durchzuführen. Insbesondere bei Fachanforderungen, deren Umsetzung durch rechtliche Restriktionen bestimmt sind, entstehen erhebliche Aufwendungen.³⁵ Schließlich behindert diese damals unter Umständen angemessene, da situativ günstige starre Integration, die heute essenzielle Möglichkeit der flexiblen Anpassung von Geschäftsprozessen und den sie unterstützenden IT-Systemen.

3.2 Konzepte der EAI

Nachdem das Umfeld beschrieben ist, in dem sich EAI einordnen lässt, adressiert das folgende Kapitel Definition, Ziele und Grenzen der unternehmensweiten Systemintegration.

3.2.1 Zentrale Integrationsplattform

Die Kernidee der Enterprise Application Integration (EAI) ist es, eine *zentrale Plattform* bereitzustellen, welche Applikationen über entsprechende, zum Teil *vorgefertigte Adapter* anbindet. In Abbildung 4 ist eine solche zentrale Integration dargestellt. Gegenüber der Point to Point Integration lässt sich eine er-

³⁴ Vgl. KELLER (2002), S. 9 ff.; GOSAIN/THILLAIRAJAH (2001), S. 4 f.; KAIB (2002), S. 43 ff.

³⁵ Laut der GARTNER GROUP geben Unternehmungen mehr als 35 % ihres IT-Budgets für Integrationszwecke aus. Vgl. ALEXANDER (2000), S. 9.

heblich geringere Anzahl an Schnittstellen realisieren.³⁶ Dieser Effekt verstärkt sich quadratisch bei steigender Anzahl der zu integrierenden Applikationen. Dem ist allerdings entgegenzuhalten, dass in der Realität selten jede Applikation mit jeder anderen kommunizieren muss³⁷ und die tatsächlichen Vorteile geringer, aber mit bis zu 43 % Kosteneinsparung gegenüber herkömmlicher Integration dennoch deutlich ausfallen.³⁸

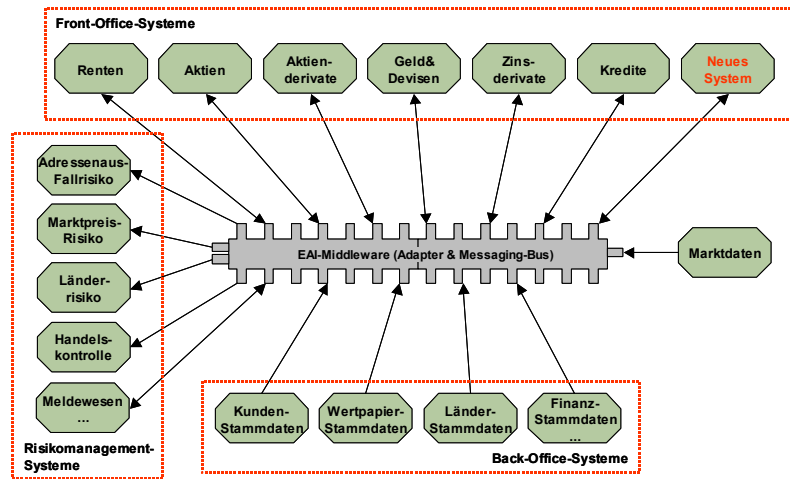


Abbildung 4. Integration über eine zentrale EAI-Plattform

3.2.2 Definition EAI

Basierend auf dem dargestellten Kerngedanken, der zentralen Integrationsplattform bietet die Literatur ein sehr uneinheitliches Bild.

Nach KELLER stellt EAI Softwaresysteme dar, die es erlauben, verschiedene Applikationen einer Unternehmung zu integrieren. Er grenzt EAI von B2B ab, da hier in der Regel mehrere Parteien beteiligt sind. Allerdings werden die Konzepte als strukturell und technisch gleich beschrieben. Der einzige

³⁶ Soll jede Applikation mit jeder anderen kommunizieren, so ergibt sich die Anzahl der Schnittstellen bei Point to Point Integration S_{PP} in Abhängigkeit zur Anzahl der Applikationen n zu $S_{PP} = n \cdot \frac{n-1}{2}$, während bei zentraler Integration lediglich $S_Z = n$ Schnittstellen erforderlich sind.

³⁷ Vgl. KELLER (2002), S. 23 f.

³⁸ RENK (2001), S. 62.

Unterschied scheint in der unternehmensübergreifenden Sichtweise des B2B-Ansatzes und des unternehmensinternen Fokus von EAI zu liegen.³⁹

Einer Unterscheidung nach Reichweite von Integrationsmaßnahmen setzt die GARTNER GROUP folgende allgemeine Beschreibung für Integrationsprojekte entgegen:

„a large [more than \$1 million], complex IS project that includes designing and/or building a customized architecture or application, as well as integrating it with new or existing hardware, packaged and costume software, and communication.“⁴⁰

Diesem pauschal über den Umfang/Preis und die Komplexität definierten Charakter liegt die Tatsache zugrunde, dass alle „großen“ IT-Projekte zumindest am Rande auch immer Integrationsaspekte mit sich bringen. Weitreichender und exakter definiert MISCHÉ den Begriff:

„Systems Integration involves a complete system of business processes, managerial practices, organizational interactions and structural alignments ... It is an all inclusive process designed to create relatively seamless and highly agile processes and organizational structures that are aligned with the strategic and financial objectives of the enterprise ... Systems integration represents a progressive and iterative cycle of melding technologies, human performance, knowledge and operational processes together.“⁴¹

Werden entgegen den abstrakten Definitionen die derzeit am Markt unter dem Label EAI angebotenen Produkte betrachtet, ergibt sich folgende Definition des Begriffes: EAI sind Werkzeuge und Services, mit denen die Aufgabe der Integration heterogener Systeme erleichtert wird. EAI ist jedoch im Wesentlichen eine technologieorientierte Methode, die sich auf verbesserte Point to Point Verbindungen konzentriert.⁴²

Allein an den vier Definitionen ist zu erkennen, dass von einer einheitlichen Sichtweise nicht gesprochen werden kann. Weder sind mögliche Integrations-

³⁹ Vgl. KELLER (2002), S. 5 ff. und auch KELLER in diesem Buch S. 94 f. Interessant ist in diesem Zusammenhang jedoch die Frage, wie die Funktionen in Anlehnung an eine Hub & Spoke-Architektur auf die einzelnen Partner verteilt werden, ohne wieder dem Problem der „Spaghetti-Integration“ gegenüber zu stehen.

⁴⁰ MISCHÉ (2002), S. 5.

⁴¹ MISCHÉ (2002), S. 5.

⁴² Vgl. GULP (2002).

szenarien gleich noch die Ziele und Methoden. Das folgende Kapitel fasst die wichtigsten Aspekte zusammen.

IT-Infrastrukturen können mit unterschiedlichen Methoden integriert werden. Sowohl Individual- als auch Standardsoftwaresysteme sollten dabei so miteinander verbunden sein, dass betriebswirtschaftlich relevante Daten prozessorientiert ausgetauscht werden. Die Syntax und Semantik der Daten in den operativen Systemen wird dabei nicht verändert. Zur Realisierung dieser Grundelemente werden neben den existierenden operativen Systemen zusätzliche Architekturen aufgebaut, die entweder individuell entwickelt oder als Produkt erworben und den individuellen Integrationsanforderungen angepasst werden. Neben diesen technischen Infrastrukturmaßnahmen gehen Integrationsprojekte immer mit Prozessänderungen einher. Die Betrachtung von Geschäftsprozessen bildet die Voraussetzung für EAI-Projekte.

3.2.3 Integrationsziele und Nutzen von EAI

Ziel von Integrationsprojekten ist vor allem die Steigerung des Nutzens beim Einsatz von IT. Diese Nutzenpotenziale von Integrationsprojekten werden in strategische und operative Potenziale unterschieden. Der operative Nutzen von Systemintegration kann zum einen in den Fachabteilungen und zum anderen in der IT-Abteilung betrachtet werden.⁴³ In Tabelle 1 werden diese Potenziale anhand der Kriterien Kosten, Zeit und Qualität gegliedert.

Neben den operativen Zielen steht vor allem die strategische Ebene bei der Systemintegration im Vordergrund. SCHILL beschreibt die folgenden Schwerpunkte:⁴⁴

- Steigerung der Integrationsfähigkeit für künftige Anwendungen und Prozesse
- Kosteneinsparungen in Merger&Aquisition- (M&A-) Projekten
- Schnellere Reaktion auf Marktveränderungen
- Standardisierung im IT- und Prozess-Bereich
- Investitionsschutz (IT-Investitionen)

Bei der Auswahl von Software wird die klassische Frage *Make or Buy* heute aus verschiedenen Gründen tendenziell meist mit Buy, d. h. dem Zukauf

⁴³ Vgl. KAIB (2002), S. 25 f.

⁴⁴ Vgl. SCHILL (2000), S. 18 ff.

Nutzen	Nutzenpotenzial in Fachbereichen	Nutzenpotenzial in IT-Abteilungen
Kosten	• Daten werden einmal eingegeben • Papierverbrauch wird reduziert • geringe Übertragungskosten	• Wiederverwendung von Komponenten • Point to Point Schnittstellen verringern • keine Veränderungen von Altsystemen
Zeit	• Beschleunigung von Abläufen • Verringerung von Medienbrüchen • Online-Verfügbarkeit von Daten	• Modifikation durch ex-post Integration von Komponenten • Administrative Vorgänge werden beschleunigt
Qualität	• Redundanz durch Mehrfacheingabe • Erhöhung der Datenqualität	• Validierte Komponenten von Altsystemen können genutzt werden • durch Prozessredesign werden Fehler aufgedeckt

Tabelle 1. Operative Nutzenpotenziale als Integrationsziele, vgl. SCHECKENBACH (1997), S. 7

von Standardsoftware beantwortet.⁴⁵ In einem solchen Szenario ist die Frage der Interoperabilität mit vorhandenen Systemen ein entscheidendes Auswahlkriterium. Ohne eine leistungsfähige Integrationsplattform wird oft auf Systeme des Herstellers der zu integrierenden Systeme zurückgegriffen, da hier zum Teil bereits Schnittstellen bestehen. Die so ausgewählten Systeme sind bezüglich der Erfüllung der restlichen Kriterien eines Anforderungskataloges oft nicht die beste Alternative. Im Gegensatz dazu lässt sich mit einer entsprechenden Integrationsplattform ein *best-of-breed*-Ansatz verfolgen, durch welchen die am besten zur Unternehmung passenden Applikationen eingeführt werden können.⁴⁶

Als globales Ziel kann die nachhaltige Flexibilisierung der IT-Infrastruktur und damit die nachhaltige Flexibilisierung der Unternehmungsprozesse angesehen werden.

⁴⁵ Vgl. FRANK/GRONAU (2002), S. 104; MOORMANN (1999b), S. 15 f.

⁴⁶ Vgl. ROCH (2002), S. 35 f.; GOSAIN/THILLAIRAJAH (2001), S. 4 f.

3.2.4 Kritische Punkte

Neben den genannten Verbesserungen durch EAI gibt es weiterhin Integrationsprobleme, welche durch EAI-Systeme nicht gelöst werden. Es ist nicht davon auszugehen, dass sich verschiedene Applikationen ganz ohne Entwicklungs- bzw. Customizing-Aufwand integrieren lassen. Bei der Integration von individualentwickelter Software, wie z. B. Legacy-Anwendungen, reduzieren EAI-Systeme den Aufwand der Schnittstellenentwicklung nicht, da hier keine vorgefertigten Konnektoren zur Verfügung stehen.

Weiterhin darf nicht übersehen werden, dass die Einführung von EAI-Systemen mit erheblichem Aufwand und einer großen zu bewältigenden Komplexität verbunden ist. Neben den rein technischen Integrationskonzepten und -anforderungen, sind vor allem die organisatorischen Anforderungen der zu unterstützenden Geschäftsprozesse in angemessener Weise in die Planung einzubeziehen, die reine „Elektrifizierung“ von Geschäftsprozessen ohne deren angemessenen Redesigns ist zu vermeiden, da hierdurch keine effizienteren Abläufe geschaffen und Organisationsstrukturen zementiert werden.⁴⁷

3.3 Organisation und Systemintegration

Bei der Darstellung der technischen Grundideen der EAI wurde bereits die Bedeutung der angemessenen Beachtung der nichttechnischen organisatorischen Dimension betont. Im Folgenden wird nun die Fragestellung der Abhängigkeiten von Organisation und Systemintegration weiter thematisiert, um dann die Schnittpunkte zwischen EAI und Systemintegration zu systematisieren.

Die Analyse der Beziehungen zwischen Organisation und Systemintegration lässt zwei Diskursbereiche erkennen, die im Folgenden genauer umrissen werden:

1. Es ist die Frage zu beantworten, wie Systemintegration organisiert werden soll.
2. Es gilt die Wechselwirkungen zwischen Organisation und IT (und Systemintegration im Speziellen) zu analysieren und die gewonnenen Erkenntnisse bei der Gestaltung beider Bereiche anzuwenden.

⁴⁷ Vgl. KAIB (2002), S. 97 ff.; DERSZTELER (2000), S. 23 ff.; PICOT/REICHWALD/WIEGAND (2001), S. 195 ff.

3.3.1 Organisation der Systemintegration

Über die Beantwortung der ersten Frage herrscht in der Praxis weitgehend Einigkeit in Bezug auf die *organisatorische Verankerung* von Integrationsprojekten bzw. Unternehmungsarchitektur-Abteilungen. Da Informationssysteme entlang ablaufender Prozesse integriert werden sollen, welche in der Regel unterschiedliche Organisationseinheiten berühren, ist es einsichtig, dass entsprechende Kompetenzen je nach Größe der Unternehmung und Umfang der Integrationsaufgabe direkt unter der Unternehmungsführung bzw. der betroffenen Bereichsleitung verankert wird.⁴⁸

Über die Beantwortung der ersten Frage in Bezug auf eine *Vorgehensweise* werden derzeit aus klassischen Vorgehensmodellen der Softwareentwicklung und -einführung Vorgehensmodelle für die Einführung von EAI-Systemen entwickelt, wobei erste Ergebnisse vorliegen aber weiterer Forschungsbedarf in Zusammenarbeit mit der Praxis besteht.⁴⁹ Die Schwierigkeit, die derartige Projekte von klassischen Softwareentwicklungs- und -einführungsprojekten unterscheidet, ist vor allem die enorme Komplexität von EAI-Projekten. Die Komplexität ergibt sich hier durch die Zusammenführung von an sich schon sehr komplexen IT-Systemen, durch die Zusammenführung komplexer fachlicher Prozesse durch verschiedene Organisationseinheiten und letztlich durch die integrierte Betrachtung beider Bereiche.

3.3.2 Wechselwirkungen zwischen Organisation und Systemintegration

In der Wissenschaft hat die Diskussion um die gegenseitigen Abhängigkeiten und Beeinflussungen von Informationstechnologie und Organisation der Unternehmung eine lange Tradition, in der sowohl die Technologie (*technological imperative*), die Organisation (*organizational imperative*) als auch komplexe Wechselwirkungen zwischen beiden (*emergent perspective*)⁵⁰ als treibende Faktoren beschrieben wurden.⁵¹

Ebenso wie die Frage der Richtung der Wirkungen zwischen Organisation und IT, wurde die Frage nach den Wirkungen selbst in empirischen Untersuchungen verschieden und zum Teil widersprüchlich beantwortet. Gründe

⁴⁸ Vgl. HAGEN, S. 67, 80 sowie KELLER, S. 110 in diesem Buch.

⁴⁹ Vgl. KAIB (2002), S. 123 ff.

⁵⁰ Vgl. MARKUS/ROBEY (1988), S. 585 ff.; zur zeitlichen Folge und Erklärung der Sichtweisen vgl. KIESER/KUBICEK (1992), S. 349 ff.

⁵¹ Vgl. LEAVITT/WHISLER (1958); APPLEGATE/CASH/MILES (1988); LIEFER (1988); ROCKART/SHORT (1989); BURGFELD (1998); TEUBNER (1999). Eine gute Übersicht über Studien sowie die Analyse von Ergebnissen finden sich bei LEWIN/HUNTER (1998), S. 252 ff.

dafür liegen in der Abhängigkeit der Wirkungen von der betrachteten Technologie, in unterschiedlichen methodischen Vorgehensweisen sowie in unterschiedlichen zu Grunde liegenden Organisationsparadigmen.⁵²

Einen Schwerpunkt der aktuellen Diskussion bildet die Analyse von Abhängigkeiten zwischen Internettechnologien und Organisation.⁵³ Diese Darstellungen schreiben den Internettechnologien jedoch zum Teil undifferenziert die fundamentale Veränderung und Unterstützung der Prozesse der Unternehmung zu. Insbesondere erfolgt eine pauschal positive Bewertung des Einflusses von Internettechnologien auf die Informationsversorgung sowie die Flexibilität der Unternehmung.⁵⁴ Diese stellen zwar prinzipiell neue, vorteilhafte und technisch standardisierte Möglichkeiten der Kommunikation dar, jedoch wird dadurch das vorliegende Engpassproblem nicht gelöst. Engpässe sind meist nicht die Verarbeitungskapazitäten der IT-Infrastruktur, sondern die begrenzten und teuren Kapazitäten menschlicher Entscheidungsträger.

FRESE folgend, wird hier nicht davon ausgegangen, dass Informationstechnologie und Unternehmungsorganisation in einem deterministischen Ursache-Wirkungs-Zusammenhang stehen.⁵⁵ Vielmehr wird – speziell für Integrations-technologien – angenommen, dass Informationstechnologie eine den Gestaltungsspielraum des Organisators erweiternde Option darstellt.⁵⁶ Insbesondere im Fall der Systemintegration durch EAI dürfte das der Fall sein, da hier das Ziel gerade in der Flexibilisierung der Unternehmung (und auch ihrer IT) besteht. LANG/UTIKAL unterscheiden die auftretenden Effekte in *direkte* und *indirekte Effekte*.⁵⁷

- *Direkte Effekte* wirken sich unmittelbar auf die grundlegenden Elemente der organisatorischen Gestaltung – Information und Kommunikation – aus. Information und Kommunikation werden dabei als wesentliche Ein-

⁵² Vgl. LEWIN/HUNTER (1998), S. 271 f.; vgl. zu den Paradigmen der Organisation KIESER (2002); HILL/FEHLBAUM/ULRICH (1998), S. 405 ff. sowie zum Problem der interparadigmatischen Vergleichbarkeit, dem *Inkommensurabilitätsproblem*, SCHERER (2002), S. 19 ff.

⁵³ Vgl. FRESE (2002); KRÜGER (2002); LANG/UTIKAL (2002); PICOT/NEUBURGER (2002).

⁵⁴ Vgl. LANG/UTIKAL (2002), S. 167; allgemein auf Informationstechnologie bezogen findet sich diese undifferenzierte Betrachtung ebenfalls oft. Hier werden der integrierten Informationstechnologie umfassende Problemlösungspotenziale zugeschrieben ohne den dabei zwangsläufig auftretenden Detailproblemen die entsprechende Beachtung zuteil werden zu lassen; vgl. exemplarisch BURGFELD (1998), S. 44 ff.

⁵⁵ Vgl. FRESE (2000), S. 142 f.

⁵⁶ Vgl. FRESE (2002), S. 201; LANG/UTIKAL (2002), S. 158 f.

⁵⁷ Vgl. LANG/UTIKAL (2002), S. 159 f.

flussfaktoren auf die Leistungsfähigkeit von Organisationsstrukturen als Infrastrukturen zur arbeitsteiligen Aufgabenbearbeitung angesehen.⁵⁸

- *Indirekte Effekte* wirken auf die Rahmenbedingungen organisatorischer Gestaltung – die Unternehmungs- und Wettbewerbsstrategie. Kommt der Organisation die Aufgabe der Umsetzung strategischer Ziele der Unternehmung zu, so wirken die Strategie beeinflussende Technologien indirekt auf die organisatorische Gestaltung der Unternehmung.

Bislang sind keine umfassenden Theorien über Wirkungszusammenhänge zwischen EAI und Organisation verfügbar, ebenso keine empirischen Studien. Jedoch werden von Praktikern regelmäßig verschiedene organisatorische Maßnahmen im Zusammenhang mit EAI angeführt wovon hier zwei genannt werden sollen:

1. Derzeit wird in der Bankenbranche intensiv die *Verringerung der Fertigungstiefe*⁵⁹ unter Zuhilfenahme von EAI-Technologien diskutiert und betrieben. Dies geschieht in Analogie zur Senkung der Fertigungstiefe in der Automobilindustrie in den 1980er Jahren.
2. *Modularisierung* als Mittel zur Flexibilisierung von Organisation und IT.

Ausgehend von dieser allgemeinen Betrachtungsweise wird im Folgenden dargestellt, wie die Anforderungen der Prozessorientierung in EAI-Systemen zu berücksichtigen sind, um eine flexible Prozessintegration der Unternehmung zu unterstützen.

Flexible Prozessintegration

Zentrale Einflussfaktoren auf die strategische Ausrichtung der Unternehmung, ihr Geschäftsmodell und der damit in Verbindung stehenden Unternehmungsorganisation⁶⁰ sind die Anforderungen des relevanten Marktes. Darauf aufbauend sind speziell zur Unterstützung arbeitsteiliger Problemlösungsstrategien entsprechende Informationssysteme bereitzustellen. Ziele der Nutzung solcher rechnergestützten Informationssysteme ist die schnelle und qualitativ angemessene Bearbeitung von Aufgaben, z. B. durch die Bereitstellung entscheidungsrelevanter Informationen⁶¹ oder durch die Automatisierung von

⁵⁸ Vgl. PICOT/REICHWALD/WIEGAND (2001), S. 23 ff.

⁵⁹ Vgl. MOORMANN (1999b), S. 14 f.

⁶⁰ Vgl. CHANDLER (1966), S. 314 ff.

⁶¹ Vgl. PICOT/REICHWALD/WIEGAND (2001), S. 69.

Problemlösungsschritten⁶². EAI-Systeme sollen die effiziente Umsetzung dieses Anspruchs unterstützen. Es ist festzuhalten, dass das primäre Ziel von EAI die effiziente Unterstützung der Zielerreichung der Unternehmung und nicht die bloße technische Verbindung von Applikationen ist (Abbildung 5).

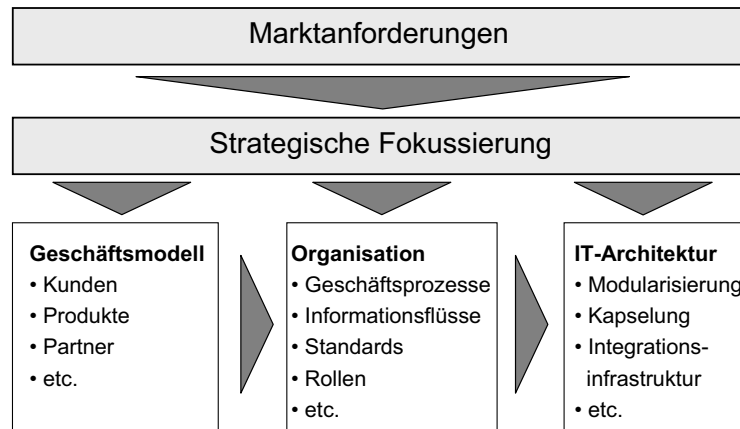


Abbildung 5. Gestaltungsrichtungen von der Strategie zur IT

Aktuell anhaltende Tendenzen der Gestaltung der Unternehmungsorganisation sind zum einen die Prozessorientierung⁶³ und zum anderen die Flexibilisierung von Prozessen durch deren Modularisierung⁶⁴ und der Implementierung serviceorientierter Architekturen (SOA).⁶⁵

Die Prozessorientierung zielt auf die Ausrichtung der Organisationseinheiten an Prozessen als Ketten zusammenhängender Aktivitäten zur Erstellung einer Leistung. Angestrebt wird die Reduktion organisatorischer Schnittstellen im Leistungserstellungsprozess durch die Reintegration zusammenhängender Aufgaben,⁶⁶ um Verbesserungen „in critical, contemporary measures of performance, such as cost, quality, service, and speed“⁶⁷ zu erreichen.

Um den treibenden Faktor – die Leistungserstellungsprozesse – bei der Gestaltung von Informationssystem-Architekturen entsprechend zu berücksichtigen, ist die Verankerung der Prozesssicht im EAI-Ansatz notwendig, um somit den

⁶² Vgl. DERSZTELER (2000), S. 138 ff.

⁶³ Vgl. HAMMER/CHAMPY (2001), S. 32 ff.

⁶⁴ Vgl. Kapitel 4.

⁶⁵ Vgl. zu SOA den Beitrag von GALLAS, S. 175 ff. in diesem Buch.

⁶⁶ Vgl. HAMMER/CHAMPY (2001), S. 53 ff.

⁶⁷ Vgl. HAMMER/CHAMPY (2001), S. 32.

technologiegetriebenen Integrationsansatz der EAI, in einen markt-, produkt- und prozessgetriebenen Ansatz zu transformieren.⁶⁸ Durch die methodische Bindung der fachlich dominierten Prozessebene an die IT-Ebene soll eine rein technisch getriebene Entwicklung der IT-Systeme und damit eine Zementierung oder unzureichende Unterstützung der Prozesse – wie dies bei der klassischen Point to Point Integration oft der Fall ist – vermieden werden. Weiterhin soll durch die explizite Berücksichtigung der Prozessebene eine Schicht geschaffen werden, die die fachlich getriebenen Abläufe abbildet und Schnittstellen zur sie durch Services unterstützenden IT-Ebene bereitstellt. Damit wird vermieden, die Geschäftslogik fest in der IT-Implementierung zu verankern, was zu unflexiblen und nicht transparenten Strukturen führt.

Ein prozessorientiertes EAI-Architekturmodell

Im Folgenden wird basierend auf der Kritik bestehender EAI-Architekturmodelle ein EAI-Ordnungsmodell vorgestellt, wobei vor allem auf den Aspekt der Modellierung fokussiert wird.

Grundsätzlich besteht in der wissenschaftlichen Diskussion, wie auch in der Praxis häufig das Problem von Übertragungsverlusten betriebswirtschaftlicher Fachkonzepte in IT-Konzepte und -Implementierungen. Dies wird zum einen durch unterschiedliche Modellierungsansätze, z. B. Ereignisgesteuerten Prozessketten (EPK) auf Fachkonzeptebene und der Unified Modeling Language (UML) auf IT-Konzeptebene, als auch durch unterschiedliche Denkwelten der jeweiligen Spezialisten verursacht. Auch im vorliegenden Fall tritt dieses Problem prinzipiell bei jeder Änderung bzw. Anpassung von Prozessen und der darauf folgenden Anpassung von IT-Systemen auf. So ist die Dimension der Unternehmungsorganisation in angemessener Weise in Implementierungen zu berücksichtigen.⁶⁹

In der Literatur sind eine Reihe unterschiedlicher Architekturmodelle vorgestellt worden, die diesem Umstand durch die Einbeziehung einer Geschäftsprozessebene Rechnung tragen.⁷⁰ Weiteres Charakteristikum dieser Modelle ist die Betrachtung der Datenebene als mögliche Integrationsebene. Zusammenfassend lassen sich folgende kritische Punkte der Modelle nennen:

⁶⁸ Vgl. GRUDÉN/STRANNEGARD (2003), S. 8 ff.

⁶⁹ Vgl. TAYLOR (1995), S. 9 ff.; TEUBNER (1999), S. 159.

⁷⁰ Vgl. LINTHICUM (2000), S. 16; BUHL/CHRIST/PAPE (2001), S. 20 ff.; RING (2000), S. 26; POLLOCK (2001), S. 49; KELLER (2002), S. 43; MAY (2001), S. 57; RUH/MAGINNIS/BROWN (2001), S. 17 ff.; WILKENS (1999), S. 13 ff.

1. Es wird nicht hinreichend zwischen der Daten-, Funktions- und Objektebene differenziert.
2. Die zu überbrückende Lücke zwischen den letztgenannten Ebenen und der Prozessebene ist oft zu groß, da in der Realität oftmals keine bzw. nur unvollständig modularisierte Applikationen und Managementsysteme, z. B. als verteilte Objektinfrastruktur (wie CORBA), vorhanden sind.

Das folgende Architekturmodell soll die genannten Kritikpunkte angemessen berücksichtigen (Abbildung 6). Ziel der Modellgestaltung war es, logisch eigenständige Ebenen zu etablieren, welche die Services unterliegender Ebenen nutzen.

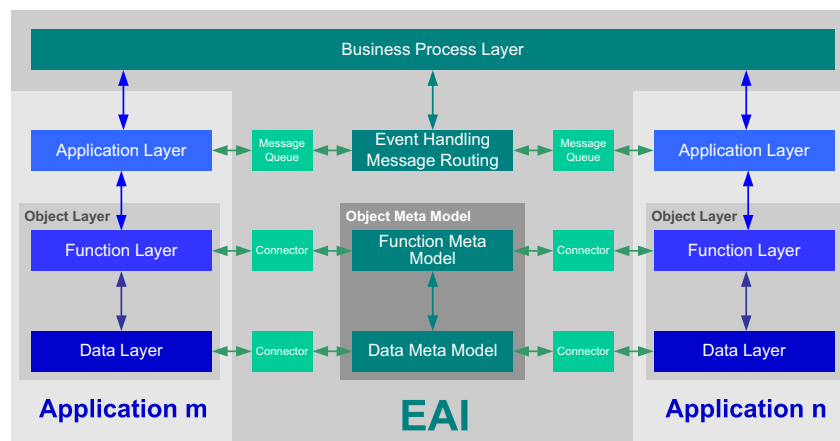


Abbildung 6. Vier-Ebenen EAI-Architekturmodell

Auf dem *Process Layer* des Modells werden die Geschäftsprozesse der Unternehmung abgebildet. Die Modellierung des Process Layer soll frei von technischen Details wie den verwendeten Applikationen sein. Ziel ist es, modularisierte schlanke Prozesse zu definieren, die die betriebswirtschaftlichen Zusammenhänge und Abläufe beschreiben.

Auf dem *Application Layer* erfolgt die prozessspezifische Abbildung des IT-Systems. Zum einen sind den einzelnen Teilprozessen Applikationen zuzuordnen, zum anderen ist die Interaktion der Applikationen, d. h. der Austausch von Informationen, z. B. durch Nachrichten (*Messages*), sowie die jeweiligen auslösenden Ereignisse (*Events*) und Regeln des Zusammenspiels (*Rules, Routing*), zu definieren.

Auf dem *Object Layer* erfolgt das Mapping mit Hilfe weitgehend vorbereiteter Konnektoren auf das Meta-Objektmodell und zurück zur Zielapplikation. Bei nicht objektorientierter Betrachtungsweise lässt sich der Object Layer entsprechend in einen *Function Layer* und einen *Data Layer* aufspalten.

Ziel der Modellierung klar abgegrenzter Ebenen ist die Entkoppelung der Ebenen, um z. B. rein technische Veränderungen auf dem Object Layer vor den ablaufenden Prozessen zu verbergen. Gleichzeitig muss es eine konsistente Verbindung zwischen den Layern geben, um geänderte Geschäftsprozesse entsprechend angepasst durch IT-Systeme zu unterstützen. Dazu sind konsistent verknüpfte Modellierungssprachen und -methoden zu entwickeln und einzusetzen. Die Basis dafür können bestehende Sprachen der Layer, wie EPK,⁷¹ und UML⁷² sein. Zu prüfen ist die Eignung aktueller Ansätze der Model Driven Architecture (MDA).⁷³

In diesem Kapitel wurde gezeigt, mit welchen Ansätzen EAI die Probleme der klassischen Systemintegration versucht zu lösen und welche Berührungspunkte sich dabei mit der Unternehmungsorganisation ergeben. Im folgenden Kapitel wird nun die Modularisierung der Organisation und der zu integrierenden IT als wesentliche Voraussetzung für die Flexibilisierung von Unternehmungen und damit als Voraussetzung für den effizienten Einsatz von EAI untersucht.

4 Modularisierung

Ein aktuell diskutiertes Mittel zur Komplexitätsreduktion innerhalb von IT- und Organisationsarchitekturen und zur Flexibilisierung⁷⁴ dieser ist die Modularisierung innerhalb der Architekturen. Nachfolgend werden die Grundgedanken der Modularisierung und die verfolgten Ziele dargestellt. Hierbei wird vorerst nicht nach IT- und Organisationssicht unterschieden, sondern es werden unter Zuhilfenahme der Systemperspektive allgemeine Charakteristika von Modulen beschrieben, welche zum Teil der Organisationsliteratur, zum Teil der IT-Literatur entnommen sind und dennoch für beide Bereiche gleichrangig gelten sollen.

Ein *Modul* besteht aus zwei Teilen, der *Modulschnittstelle* und dem *Modulrumpf*. Die Modulschnittstelle enthält dabei die Spezifikation der Leistungen des Mo-

⁷¹ Vgl. SCHEER (1995), S. 307 ff.

⁷² Vgl. OESTERREICH (1998).

⁷³ Wie in den Beiträgen von GALLAS, S. 175 ff. und PETKOFF, S. 227 ff. in diesem Buch.

⁷⁴ Vgl. zur Definition Kapitel 2.

duls und alle Annahmen, die für seine Umwelt zur „Benutzung“ notwendig sind. Der Modulrumpf implementiert die in der Schnittstelle spezifizierten Leistungen des Moduls.

Modularisierung bedeutet allgemein die Strukturierung eines Systems in relativ kleine, teilautonome und überschaubare Subsysteme (Module, Segmente, Fraktale, Handlungseinheiten).

Die Komplexitätsreduktion⁷⁵ ergibt sich dabei also aus der Subsystembildung innerhalb des Systems Unternehmung.⁷⁶ Die Subsystembildung wirkt deswegen komplexitätsreduzierend, da sie zum einen die subsysteminterne Komplexität im Sinne der Kapselung vor der Subsystemumwelt verbirgt, zum anderen durch die Reduktion auf wenige bekannte Schnittstellen eine Entkopplung der Subsysteme im Sinne einer Reduktion von Abhängigkeiten bewirkt. Mit anderen Worten: Komplexität ist systemtheoretisch ein Selektionszwang und Systemdifferenzierung durch Subsystembildung eine Möglichkeit der Selektion zur Reduktion der Komplexität.⁷⁷ Im Zusammenhang damit steht die Fähigkeit der einzelnen Subsysteme, selbständig zu agieren und sich selbst organisieren zu können.⁷⁸ Dies entspricht der Forderung nach Dezentralisierung und Selbststeuerung. Die Unternehmensziele sollen dabei das Handeln der einzelnen Einheiten bestimmen.

Die Flexibilisierung der Strukturen und damit der Abläufe ergibt sich durch die nun leichtere Rekonfigurationsmöglichkeit der entkoppelten Module über ihre definierten Schnittstellen. Komplexitätsreduktion kann somit als Voraussetzung für die Flexibilisierung angesehen werden.

Bei der Modularisierung sollen folgende Gestaltungsziele beachtet werden:⁷⁹

- Abstraktion von der Implementierung
- Kapselung im Sinne des Verbergens der internen Funktionsweise
- Austauschbarkeit

⁷⁵ Aus Sicht der Systemtheorie meint *Komplexität* allgemein die Gesamtheit der möglichen Ereignisse und Zustände. Mit der Zahl der Ereignisse und Zustände steigt die Zahl der zwischen ihnen möglichen Relationen und somit auch die Komplexität. Das menschliche Vermögen Komplexität aufzunehmen ist angesichts der Komplexität der „Welt“ überfordert. Systeme tragen durch den Ausschluss von Möglichkeiten zur *Komplexitätsreduktion* bei. Sie bilden Inseln geringerer Komplexität im Vergleich zur Umwelt. Vgl. KNEER/NASSEHI (2000), S. 40 f.

⁷⁶ Vgl. KRCAL (2003), S. 13.

⁷⁷ Vgl. STÜNZER (1996), S. 73.

⁷⁸ Solche Systeme sind Gegenstand der aktuelleren Systemtheorie und werden dort *autopoietische Systeme* genannt; vgl. dazu STÜNZER (1996), S. 45 ff.

⁷⁹ Vgl. ROMBACH (2003), S. 30 f.; LANG (1997), S. 10, 89.

- Wiederverwendbarkeit
- Zeitliche Gültigkeit
- Orthogonalität (im Sinne von „sich gegenseitig nicht beeinflussend“)
- Überschneidungsfreiheit
- Vollständigkeit (Abgeschlossenheit)
- Allgemeingültigkeit
- Interoperabilität
- Wohldefinierte Schnittstellen
- Schnittstellenminimalität⁸⁰
- Generizität⁸¹

Durch die Anwendung dieser Prinzipien werden Module geschaffen, die kombinierbar, wiederverwendbar und gut änderbar sind.⁸²

Im Folgenden wird nun dargestellt, welche Ansätze sich bei der Modularisierung der Organisation und der Informationssysteme ergeben.

4.1 Modularisierung der Organisation

Für die Modularisierung der Organisation, d. h. der Strukturen und Prozesse, unterscheiden PICOT/REICHWALD/WIEGAND drei relevante Ebenen: Die Modularisierung auf *Unternehmungsebene*, auf *Prozessebene* und auf der *Ebene der Arbeitsorganisation*. Da in diesem Beitrag Systemintegration und EAI vor allem als Backend-Integration beschrieben wird und keine arbeitswissenschaftliche Analyse der Unternehmungsarchitektur erfolgen soll, werden nachfolgend ausschließlich die beiden erstgenannten Ebenen diskutiert. Im Folgenden wird von der Ebene der Gesamtunternehmung als Makroebene und von der Ebene der Geschäftsprozesse als Mikroebene gesprochen (Abbildung 7).⁸³

Mit der Definition einer Makro- und einer Mikroebene werden zwei Ziele verfolgt:

⁸⁰ In Bezug zur Systemtheorie bedeutet Schnittstellenminimalität die Reduktion der Relationen des Subsystems zu seiner Umwelt und somit Maximierung – im Sinne von Verschiebung – der Relationen in den Subsystemen.

⁸¹ Generizität bedeutet hier die Separierung varianter von invarianten Bestandteilen eines Bausteins, die Parametrisierung der varianten Teile durch generische Parameter.

⁸² Vgl. ROMBACH (2003), S. 30.

⁸³ Vgl. zu anderen Begriffsbildungen der Makro-, Meso- und Mikroebene PICOT/REICHWALD/WIEGAND (2001), S. 241.

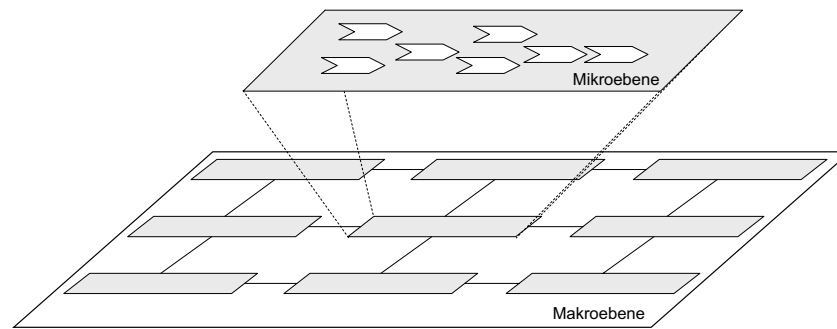


Abbildung 7. Modulebenen der Organisation

1. Festigung und zugleich Flexibilisierung der Strukturen
2. Definition überschaubarer und managebarer Module

Die Makroebene hat eine stark strukturierende Wirkung auf die Organisation. Die Module der Makroebene, die so genannten Makromodule, sollen über einen längeren Zeitraum stabil sein. Die Makroebene bildet damit ein Bezugssystem als Grundlage für die Gestaltung der Module der Mikroebene innerhalb der jeweiligen Makromodule. Die Module der Mikroebene hingegen sollen die Dynamik der wechselnden Prozessanforderungen durch entsprechende Rekonfiguration widerspiegeln. Ziel ist es also, eine bezuggebende, stabile, organisationsinvariante Ebene – die Makroebene – und eine dynamische, flexibel konfigurierbare Ebene – die Mikroebene – zu bilden.

Nachfolgend werden die Charakteristika der Makro- und der Mikroebene dargestellt.

4.1.1 Modularisierung auf Makroebene

Um eine direkte Abbildung strategieorientierter Organisationsstrukturen auf die IT-Infrastruktur zu erreichen, werden mehr als die klassischen ablauf- bzw. aufbauorganisatorischen Sichtweisen benötigt. Das Ziel der Definition der Makro-Module besteht in der Betrachtung der Gesamtorganisation mit Hilfe geeigneter Kriterien, um genau diesen Schritt ableiten zu können. Im Folgenden werden einige Kriterien zur Bildung von Makromodulen aufgezählt:

- Ähnlichkeit der relevanten Prozesse innerhalb eines Moduls
- Ähnlichkeit des notwendigen Prozess-Know-hows

- Minimierung der externen Abhängigkeiten (mit anderen Makromodulen)
- Maximierung der internen Kohärenz
- Unabhängigkeit von operativen Prozessanpassungen
- Branchenspezifische Best Practices für die Abgrenzung der Module

Die aufgezählten Kriterien werden z.T. im Bereich der Componentware verwendet, um die Granularität von Softwarekomponenten zu bestimmen. Die analoge Verwendung auf Makroebene führt zumindest zu ähnlichen Methoden, mit denen später ein Match zwischen Makro-, Mikro- und Softwarearchitekturebene erreicht werden kann.

Die definierten Makro-Module stellen keine Handlungsanweisung für eine Reorganisation dar. Sie verfolgen vielmehr das Ziel, eine besondere Sichtweise auf die Organisation zu gewähren. Diese Sichtweise unterstützt die Modularisierung der jeweiligen IT-Systeme, die die Prozesse eines Makro-Moduls unterstützen. In Abbildung 8 wird anhand einer klassischen Matrixorganisation die sich nach Sparten und Funktionsbereichen aufstellt gezeigt, wie Makro-Module durch die oben aufgeführten Kriterien gebildet werden. Dabei stellen die farbig gleich hervorgehobenen Bereiche jeweils ein Makro-Modul dar. Beispielsweise wurden innerhalb der verschiedenen Vertriebsorganisationen zwei Bereiche als Makro-Modul definiert. Grund dafür könnte sein, dass sich die Prozesse in beiden Bereichen stark ähneln und ähnlicher Informationen zur Ausführung bedürfen.

	Produkt I	Produkt II	Produkt III	Produkt n
Vertrieb				
Produktion				
Absatz				
Verwaltung				

Abbildung 8. Definition von Makro-Modulen in einer Matrixorganisation

Auf der Ebene der Makromodule sollen Strukturähnlichkeiten die Grundlage für die Definition bilden. Es können situativ weitere Kriterien zu den oben genannten bestimmt werden, die zur Abgrenzung der Module beitragen. Prozesse laufen i. d. R. modulübergreifend, d. h. es wird keine Abgrenzung nach Anfang bzw. Ende von Prozessketten vorgenommen.

Ziel der Gestaltung ist eine weitgehend stabile, *organisationsinvariante* Makroebene, d. h. bei Produkt- oder Prozessänderungen sollten die Grenzen und Schnittstellen der Module unverändert bleiben.

Es stellt sich die Frage, ob ein Architekt in der Lage sein wird, diese Anforderung tatsächlich nachhaltig zu erfüllen. Die Unsicherheit über die zukünftigen Veränderungen steht diesem Ziel kontraproduktiv gegenüber.

4.1.2 Modularisierung auf Mikroebene

Innerhalb der definierten Makromodule sollen jetzt die Strukturähnlichkeiten genauer betrachtet werden. Für die IT-Systemwelt sind vor allem die innerhalb der Module vorhandenen Geschäftsprozesse relevant. Hier müssen alle Prozessschritte (Aktivitäten) verglichen und die strukturgleichen identifiziert werden. Ein Geschäftsprozess setzt sich prinzipiell aus verschiedenen Aktivitäten zusammen. Die Wahrscheinlichkeit, dass sich in einer Menge ähnlicher Prozessketten identische Aktivitäten befinden, ist sehr hoch. Selbst wenn sich die Teilprozesse nur sehr leicht unterscheiden, kann betrachtet werden, ob es im Sinne einer Komplexitätsreduktion möglich ist, die tatsächlichen Geschäftsprozesse so zu modifizieren, dass sie möglichst viele identische Teilprozesse verwenden.

Bevor aus den so konsolidierten Prozessen wiederverwendbare Module gebildet und diese in einer Art Prozess-Repository organisiert werden, um bei Rekonfigurationsbedarfen darauf zurückzugreifen, ist jedoch zu prüfen, ob sich die betrachteten Prozesse dazu eignen.

GERYBADZE leitet dazu aus den Methoden der modularen Produktgestaltung Kriterien für die Bestimmung des Grades der Modularisierbarkeit von Prozessen ab. Eine gute Modularisierbarkeit ergibt sich nach GERYBADZE, wenn⁸⁴

- sich für jede Aktivität eine genau definierte Funktion des Gesamtsystems definieren lässt,
- die Qualität des Ergebnisses dieser Aktivität genau bestimmt werden kann,

⁸⁴ Vgl. GERYBADZE (2003), S. 91 f. und die dort angegebene Literatur.

- für den jeweiligen Output Preise bzw. Verrechnungspreise bestimmt werden können und
- die Schnittstellen sehr genau definiert werden können.

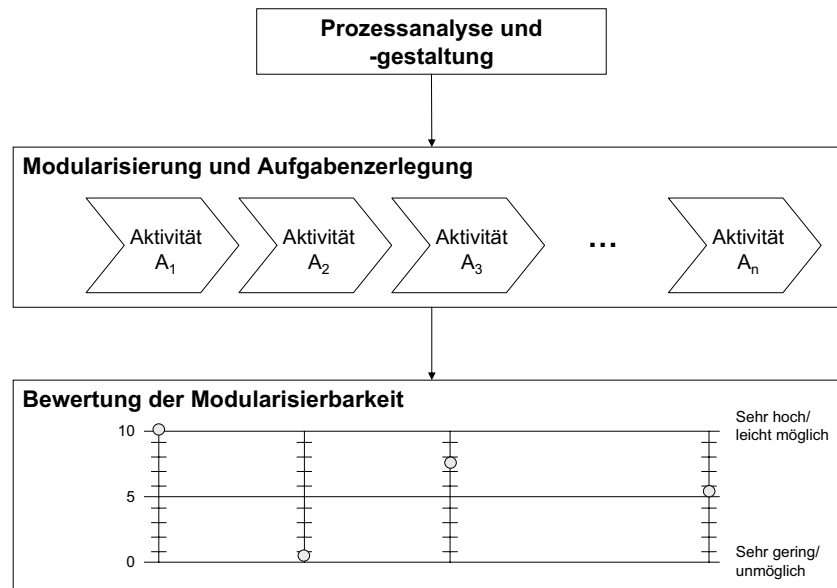


Abbildung 9. Modularisierung auf Prozessebene, in Anlehnung an GERYBADZE (2003), S. 91

In Abbildung 9 wird eine Metrik dargestellt, in der die Modularisierbarkeit für jede Aktivität bewertet wird. Aktivitäten mit Werten zwischen 6–10 eignen sich gut um in modularen Strukturen eingesetzt zu werden, Aktivitäten mit Werten zwischen 1–4 sind nur schwer in modulare Strukturen zu integrieren.

Neben der generellen Frage der Modularisierbarkeit ist die Bestimmung der *optimalen Modulgröße* entscheidend für die erfolgreiche Umsetzung der Modularisierung auf Prozessebene. Zu große Module können die erhoffte Flexibilität der Organisation verhindern, da die Modularisierung ja gerade kleine, flexibel integrierbare Fragmente schaffen soll, die beweglicher sind als eine gesamte monolithische Struktur. Zu große Module können durch ihre Komplexität die Beherrschbarkeit der Prozesse innerhalb des Moduls einschränken.

Andererseits bedeuten zu viele kleine Module eine stärkere Arbeitsteilung und Zergliederung. Je kleiner die Module, desto spezialisierter müssen naturgemäß

die Prozesse innerhalb eines Moduls sein. Viele Kritikpunkte, die aus klassischen Organisationsformen bekannt sind, wie z. B. hohe Anzahl an Schnittstellen, Zentralisierung der Entscheidungskompetenzen und die damit verbundenen Koordinationskosten sowie mangelnde Motivation der Mitarbeiter treffen auch bei Bildung zu vieler kleiner Module zu.

Um allgemeine Kriterien zur Schaffung einer optimalen Modulgröße zu definieren, müssen die Charakteristika der Modularisierung betrachtet werden.

Die zu beachtende *Mindestgröße eines Moduls* ergibt sich aus den Aktivitäten für ein klar definierbares (Zwischen-)Produkt. Die *maximale Modulgröße* wird durch die Beherrschbarkeit der Komplexität innerhalb eines Moduls bestimmt. Die Komplexität darf nicht so groß sein, dass die Verantwortlichen das Modul nicht mehr steuern können.

Diese Komplexität ist abhängig von der Anzahl der Teilprozesse und ihrer Beziehungen in einem Modul. Denn je mehr Relationen innerhalb eines Moduls existieren, desto komplexer wird das System. Auch kann die Komplexität davon abhängig sein, wie hoch die Dynamik innerhalb des Moduls ist, d. h. in welchen zeitlichen Abständen es zu Restrukturierungen innerhalb Moduls selbst kommt. Wenn Restrukturierungen innerhalb eines Moduls häufig vorkommen, dann steigt die dynamische Komplexität des Moduls.

Die Komplexitätsreduktion durch die Zusammenfassung von Aufgaben einerseits und die Bildung möglichst kleiner flexibler Einheiten andererseits können demnach durchaus im Widerspruch zueinander stehen. Dies ist der Fall, wenn die minimale Modulgröße aus Prozesssicht die Grenzen der Beherrschbarkeit einer kleinen Einheit überschreitet⁸⁵ oder wenn durch die Modulgröße keine ausreichende Varietät für flexible Rekonfigurationen gegeben ist. Die Prozessintegration unterstützende IT-Systeme sollen diesen Zielkonflikt bis zu einem bestimmten Grad abschwächen oder auflösen.

4.1.3 Realisierungsprinzipien

Die Realisierung einer modularen Organisation ist stark von den situativen Rahmenbedingungen abhängig. Als wesentliche Einflussfaktoren können die *Größe* der Organisation, ihr *Angebotsprogramm*, ihre *Internationalisierung* und *Kultur*, die bestehende *Strukturierung* sowie die *Organisationsumwelt* genannt werden.⁸⁶

⁸⁵ Vgl. PICOT/REICHWALD/WIEGAND (2001), S. 233.

⁸⁶ Vgl. KIESER/KUBICEK (1992), S. 199 ff.

Trotz dieser situativen Abhängigkeit lassen sich einige grundsätzliche Realisierungsprinzipien finden die im Folgenden beschrieben werden.

- Das Hauptprinzip ist, *kleine Organisationseinheiten* zu bilden. Sie müssen groß genug sein, zusammengehörige Prozesse zu einem Objekt (z. B. Produkt, Produktgruppe etc.) zu umfassen. Sie dürfen aber den Umfang und die Komplexität betreffend die Aufnahmegrenzen und Problemlösungskapazitäten des Menschen nicht überschreiten.
- Die *Prozessorientierung* kommt durch die Forderung zum Ausdruck, die Module durchgängig an den Prozessen zur Erstellung von Leistungen zu orientieren. Die Prozessorientierung zeigt, dass die Modularisierung nicht auf eine funktionsorientierte, sondern auf eine objektorientierte Strukturierung abzielt. Hierdurch soll die Flexibilität der modularen Struktur erhalten bleiben, ohne dass es zu einer großen Anzahl von Schnittstellen und Brüchen zwischen den einzelnen Modulen kommt. In diesem Sinne handelt es sich bei der Modularisierung nicht um ein neues Organisationskonzept, sondern um eine sinnvolle Erweiterung der Prozessorientierung.
- In engem Zusammenhang mit der Prozessorientierung steht die *Kunden- bzw. Marktorientierung*. Hiernach sollen sich alle Wertschöpfungsaktivitäten der Module an den externen und/oder internen Kundenanforderungen orientieren. Diese Forderung resultiert aus dem Hauptziel, der flexiblen Anpassung an die Erfordernisse des Marktes.
- *Integriertheit der Aufgaben*: Die Prozesse in einem Modul sollen weitestgehend ihrer Art nach zusammengehören, um die Abgeschlossenheit der in einem Modul konzentrierten Prozesse zu gewährleisten.
- *Dezentralisierung von Entscheidungskompetenz und Ergebnisverantwortung*: Entscheidungen sollen dort gefällt werden, wo die Probleme entstehen. Zugleich wird den Entscheidungsträgern auch die Verantwortung für ihr Handeln übertragen. Mit diesem Prinzip sollen die aus den traditionellen Organisationsformen bekannten Probleme wie z. B. lange, fehlerhafte Entscheidungswege und geringe Motivation der Mitarbeiter vermieden werden. Notwendig hierfür sind insbesondere flache Hierarchien. Ein Vorteil, der sich aus der Dezentralisierung ergibt, ist die Reduktion der Fremdkontrollkosten.
- *Nicht-hierarchische Koordinationsformen*: Die Koordination autonomer Handlungseinheiten, die nicht mehr in einem hierarchischem Verhältnis zueinander stehen, erfordert neue Formen der Zusammenarbeit, welche nicht mehr auf Fremdsteuerung, sondern auf Selbststeuerung basieren.

4.1.4 Probleme und Grenzen

Die größten Probleme des beschriebenen Spannungsfeldes liegen in der Abgrenzung der Module. Das gilt sowohl für die Makro- als auch für die Mikroebene, die durch die vorhergehende Definition der Makroebene bereits stark vorgegeben wird. Eine organisationsinvariante Makroebene birgt die Gefahr in sich, dass notwendige Veränderungen auf Mikroebene nicht greifen. Weiterhin stehen sich die Konzepte der Modularisierung mit dem Ziel der Flexibilisierung und die Prozessorientierung, mit dem Ziel der durchgängigen Abbildung von Geschäftsprozessen gegenüber.

4.2 Modularisierung der IT

Das Abhängigkeitsverhältnis zwischen IT und Organisation steht soweit unbestritten fest. Über Systemintegration sprechend, muss darauf hingewiesen werden, dass alle dargestellten Konzepte und Theorien auf der Annahme beruhen, über Implementierung und Neueinführung von IT zu sprechen. Um allgemeine Regeln über die Flexibilisierung von Organisationen im Zusammenhang mit Systemintegrationskonzepten zu erarbeiten, sollen im Folgenden kurz die wichtigsten Integrationsmethoden beschrieben werden, die in der Literatur besprochen sind.

Die Literatur bietet wiederum vielfältige Ansätze, nach denen Integrationsmethoden klassifiziert werden können. DANGELMAIER spricht von einer traditionellen Klassifizierung, die sich an der technischen Realisierung orientiert.⁸⁷ Innerhalb dieser Klassifizierung lassen sich am besten die Ebenen nach LINTHICUM aufgreifen und beschreiben. Dieser unterscheidet in Integration auf User Interface Level, Data Level, Application Interface Level und Method Level.⁸⁸

Es existieren daneben noch zahlreiche weitere Einteilungen, die z. T. widersprüchlich sind. Ihnen liegen unterschiedliche Sichtweisen und Schwerpunkte zugrunde. RUH nimmt eine Einteilung in Presentation Integration Model, Data Integration Model und Functional Integration Model vor.⁸⁹ Die Präsentationsintegration entspricht im Wesentlichen dem oben genannten Ansatz der Integration auf User Interface Level von LINTHICUM, ebenso ist das Data Integration Model mit dem Data Level vergleichbar. Die Funktionsintegration kann als Kombination zwischen Application Interface Level und Method Level verstanden werden.

⁸⁷ Vgl. DANGELMAIER et al. (2002).

⁸⁸ Vgl. LINTHICUM (2000), S. 18 ff.

⁸⁹ Vgl. RUH/MAGINNIS/BROWN (2001), S. 18 ff.; KAIB (2002), S. 60 ff.

4.2.1 User Interface Integration

User Interface Integration nutzt die Benutzerschnittstelle, um auf Daten und Prozesse in Applikationen zuzugreifen. Dieser Ansatz wird insbesondere bei Legacy-Systemen wie Mainframe-Anwendungen ohne Application Programming Interface (API) oder Datenbankschnittstelle genutzt. Er stellt in diesem Zusammenhang oft die einzige Kommunikationsmöglichkeit dar, ohne direkten Eingriff in den Quellcode der Applikation. Es gibt neben der Kostenersparnis dieser Variante weitere Gründe eine User Interface Integration zu wählen. Oft fehlt in Unternehmen bereits vollständig die Kompetenz der Beherrschung der Entwicklungsumgebungen, in den z. T. 25 Jahre alte Host-Applikationen geschrieben wurden. Die Anwendungen müssen weiter betrieben werden, bleiben technisch aber unverändert, da das Risiko von nicht vorhersehbaren Nebeneffekten zu hoch ist. Der Integrator muss die Anwendung hinsichtlich des internen Datenspeichermechanismus, der Präsentation der Daten an der Benutzerschnittstelle und der Anwendungslogik gut verstehen. Die Applikation an sich wird technisch nicht verändert.⁹⁰ Technisch bedeutet das den programmatischen, mechanischen Zugriff einer Middleware auf die Benutzerschnittstelle. Dazu werden Emulatoren verwendet, die für verschiedene Entwicklungswerkzeuge existieren und die Interaktionen auf User Interface Level automatisieren.⁹¹

Trotz langjährigem Einsatz dieser Methode aus Mangel an Alternativen bestehen Probleme bezüglich der Leistungsfähigkeit, der Stabilität und der Skalierbarkeit. Im Allgemeinen wird Integration auf dieser Ebene als Notlösung betrachtet.⁹²

Als modernere Verfahren werden oft Möglichkeiten aufgezählt, die in Browser-Clients mehrere Applikationen für den Benutzer zusammenführen. Dabei geht es allerdings nur um eine Integration auf Frontend-Ebene, die hier nicht beschrieben werden soll. Die eigentlichen Systeme im Hintergrund werden inhaltlich in keiner Weise synchronisiert. Einige Anbieter realisieren zumindest diverse Zusatz-Features wie Single-Sign-On (einheitliches und einmaliges Login) und Personalisierung.

⁹⁰ Vgl. LINTHICUM (2000), S. 82.

⁹¹ Dies wird oft auch „Screen Scraping“ genannt. Siehe vertiefend KELLER (2002), S. 63 f.

⁹² Vgl. RIEHM/VOGLER (1996), S. 47 f.

4.2.2 Data Level Integration

Das zu integrierende System greift bei der Data Level Integration direkt auf die Daten einer Datenbank zu. Die Business-Logik anderer Applikationen, die ebenfalls die betroffene Datenquelle nutzen, wird nicht verwendet. Die Daten werden gelesen, nach individuellen Regeln transformiert oder durch eigene Logik interpretiert um in die gleiche Quelle zurück geschrieben bzw. auf andere Art und Weise weiterverwendet zu werden.⁹³

Nach KELLER lassen sich für diese Integration über Datenbanken zwei Szenarien unterscheiden:⁹⁴

1. Eine Applikation nutzt mehrere Datenquellen, die über diverse Technologien⁹⁵ so vorgehalten werden, dass der Entwickler virtuell nur von einer Datenbank ausgeht.
2. Mehrere Applikationen nutzen eine gemeinsame Datenquelle und kommunizieren über diese.

Durch die Verwendung einer Datenbank von mehreren Applikationen besteht die Gefahr, dass verschiedene Interpretationen der Geschäftslogik, die in den Systemen durchaus gegeben sein können, inkonsistente Datenbestände aufbauen.

4.2.3 Application Interface Integration

Bei der Integration auf Ebene der Application-Interfaces greift die integrierende Software auf die Daten und Businessprozesse einer Anwendung mit Hilfe von Application Programming Interfaces (API) unter Verwendung der Applikationslogik des Systems zu. Die Daten werden dann in ein für die Zielanwendung oder die zentrale EAI-Instanz verständliches Format umgewandelt und übertragen. Ein Vorteil dieser Methode im Vergleich zur Data Level Integration besteht darin, dass die Geschäftslogik für die Konsistenz und die Einhaltung der Datenformate garantiert. Als Nachteil werden in der Regel die API-Restriktionen bezüglich des Datenzugriffs und Funktionsumfangs sowie

⁹³ Vgl. LINTHICUM (2000), S. 80.

⁹⁴ Vgl. KELLER (2002), S. 67.

⁹⁵ Es stehen zahlreiche datenzugriffsorientierte Middleware-Tools zur Verfügung, um Informationen auch zwischen verschiedenen Datenbanktypen, z. B. objektorientierten, relationalen, dateiorientierten und multidimensionalen Datenbanken, auszutauschen. [Anm. d. A.]

die Komplexität der systemübergreifenden Funktionsintegration im Falle fehlender Standard-APIs betrachtet.⁹⁶

Besonders geeignet ist dieser Typ der EAI natürlich für Systeme, die mit API ausgeliefert werden. Tatsächlich werden immer mehr Softwareprodukte nicht wie früher als abgeschlossene Systeme verkauft, sondern explizit für den Austausch mit anderen Systemen konzipiert. Standardmechanismen wie Javas Remote Method Invocation (RMI), Common Object Request Broker Architecture (CORBA), Internet Inter-ORB Protocol (IIOP) und Microsofts Distributed Component Object Model (DCOM) bieten Grundlagen für die Funktions- bzw. Objektintegration.⁹⁷ Die Objektorientierung wird im folgenden Kapitel wieder aufgegriffen.

4.2.4 Method Integration

Bei der Integration auf Methodenebene wird die Geschäftsprozesslogik systemübergreifend verteilt und genutzt. Anwendungen können gegenseitig ihre Methoden aufrufen und auf bereits bestehende Anwendungen zurückgreifen. Es gibt zwei technische Möglichkeiten, dies zu realisieren. Einerseits kann die Geschäftsprozesslogik im Rahmen einer Client/Server-Architektur auf einem gemeinsamen Server (Application Server) abgelegt und dort von den Anwendungen aufgerufen werden. Andererseits können die Anwendungen so umgestaltet werden, dass sie weiteren Systemen ihre Methoden über Schnittstellen anbieten. Im zweiten Fall müssen die Applikationen mittels einer der oben beschriebenen Komponentenstandards „umorganisiert“ werden.⁹⁸

Vor allem ältere Programme und Individualentwicklungen besitzen oft keine standardisierten Schnittstellen. Hier müssen durch Eingriff in den Quellcode mit erheblichem Aufwand APIs hinzuprogrammiert werden. Das so genannte Wrapping ist in diesem Zusammenhang weit verbreitet. Hier werden die in der Anwendung enthaltenen Geschäftsprozesse als Methoden in Objekten gemäß einem verteilten Objektstandard (CORBA, COM etc.) zugänglich gemacht. Die Anwendung erscheint dann als verteiltes Objekt.⁹⁹ Wrapping wird vor allem bei Legacy-Systemen¹⁰⁰ angewandt. Die Integration von nicht

⁹⁶ Vgl. KAIB (2002), S. 67.

⁹⁷ Vgl. LINTHICUM (2000), S. 37 f.

⁹⁸ Vgl. LINTHICUM (2000), S. 19 f., 61 ff.

⁹⁹ Vgl. LINTHICUM (2000), S. 57 ff.

¹⁰⁰ Legacy-Systeme sind in mehrjähriger Arbeit entstandene individuelle und komplexe Großrechnersysteme, deren Funktionalitäten und Datenstrukturen große Teile der Kerngeschäftsprozesse abbilden und die nur mit sehr großem Aufwand neu konzipiert und implementiert werden könnten. Vgl. dazu u. a. GRUHN/SCHÖPE (2002).

komponentenfähigen Infrastrukturen kann entweder durch Implementierung kompletter Neusysteme oder durch Legacy-Wrapping realisiert werden.

Bei der Neuimplementierung wird der fachliche Inhalt des Legacy-Systems durch Software Reengineering bzw. Reverse Engineering in eine neue Softwaretechnologie konvertiert, d.h. neue Software wird unter Beibehaltung der Geschäftsregeln, Aufbau-logik und Datenstrukturen implementiert.¹⁰¹ Eine Neuentwicklung der Funktionen der Legacy-Systeme oder eine Neuorganisation der Datenhaltung würde aufgrund der gewachsenen Komplexität dieser Systeme sehr hohe Kosten verursachen und viel Zeit in Anspruch nehmen.

Aus den genannten Gründen verlagert man die Bemühungen von der Migration durch Reimplementierung zur Integration durch Wrapper. Dort versucht man, bestehende Funktionen oder existierende Legacy-Datenbestände ohne gravierende Anpassungen zu nutzen und so nicht auf eine Neuentwicklung der Funktionen oder eine Reorganisation der Datenhaltung des Legacy-Systems angewiesen zu sein.¹⁰²

Die primäre Aufgabe eines Wrappers ist die Kapselung der in der Datenquelle gespeicherten Daten und ihrer Schnittstelle nach außen. Alle verwendeten Datenquellen behalten ihre logische und physikalische Unabhängigkeit bei. Ein Wrapper ist immer genau einer Datenquelle zugeordnet. Gibt es in einem System, das ausschließlich Wrapping verwendet, mehrere Datenquellen, so existieren ebenso viele Wrapper.¹⁰³

Hier offenbart sich der große Nachteil der Integration auf Methodenebene. Nahezu alle beteiligten Applikationen müssen hinsichtlich der gewählten Technologie teilweise massiv umgestaltet werden, was im Gegensatz zum Ziel von EAI steht, möglichst nicht-invasiv und kostengünstig zu sein.

Allerdings wird seit vielen Jahren in der Diskussion um effiziente Architekturmodelle die Komponentenorientierung als strategisch wichtigster Ansatz gehandelt. Die derzeitigen Bemühungen um effiziente Lösungen der Systemintegration sind teilweise Ergebnis der inkonsequenten Umsetzung von Architekturen ohne Beachtung der bereits lange bekannten Komponentenstandards in der jüngeren Vergangenheit.

¹⁰¹ Vgl. HAMPEL (2002), S. 3.

¹⁰² Vgl. GRUHN/SCHÖPE (2002), S. 2.

¹⁰³ Vgl. LEGNER (2000b), S. 5 f.

4.2.5 Service based Integration

Serviceorientierung und Systemintegration wurden viele Jahre unabhängig voneinander diskutiert. So basieren Aussagen zur Systemintegration vor allem auf den bisher beschriebenen Methoden. Je stärker die Integrationsprojekte allerdings mit objektorientierten Verfahren implementiert werden desto größer erscheint die Nähe zur Serviceorientierung. Die Abgrenzung von Begrifflichkeiten wie SOA (Service Oriented Architecture), Componentware und Web Services fällt insofern schwer, als dass viele Publikationen nicht sauber zwischen diesen Themen differenzieren.

Die dienstorientierte Anwendungsintegration ist mit der Integration auf Interface- und Methodenlevel vergleichbar. Sie stellt eine alternative Sichtweise dar, die versucht, „gewrappte“ Module aus Altanwendungen und bereits serviceorientiert implementierte Neuanwendungen innerhalb einer SOA zu integrieren. Voraussetzungen dafür sind Rahmenbedingungen der Architektur. Angelehnt an die Architektur der Verwendung von Web Services in verteilten Umgebungen müssen die Eigenschaften der Services an zentraler Stelle einheitlich abgebildet, die Kommunikation zwischen den Services geregelt und die Auswahl der richtigen Services gewährleistet werden.¹⁰⁴

Zwischen der traditionellen Umsetzung von EAI und der serviceorientierten Integration besteht ein entscheidender Unterschied. Da die meisten Entwicklungswerkzeuge inzwischen eine Serviceorientierung unterstützen, werden Anwendungen in Zukunft sehr viel einfacher als Services entwickelbar sein. Altanwendungen sind jedoch oft schwer in als Service abgrenzbare Module aufzuspalten. Neben Entwicklungsumgebungen die eine Neuimplementierung mit Servicecharakter aktiv unterstützen, bieten einige Integrations-Suiten inzwischen ebenfalls Tools, die das Wrappen von Altanwendungen in Services erleichtern. Die Kosten dieser initialen Umwandlung können sehr hoch sein und oft widerspricht der Eingriff in den Quellcode den strategischen Richtlinien der IT-Verantwortlichen, die vor allem monolithische Legacies unberührt lassen wollen.¹⁰⁵

In der serviceorientierten Integration geht es um den Zugang zu Anwendungsdiensten über gut definierte Schnittstellen. Im Vordergrund stehen die Wiederverwendung bereits existierender Anwendungsbausteine und die vereinfachte Pflege der Systeme. Mit zahlreichen Technologien und Mechanismen, z. B. Frameworks, Transaktionen und verteilten Objekten, wird seit

¹⁰⁴ Vgl. u. a. LUBLINSKY/FARRELL (2003), S. 30 ff.

¹⁰⁵ Vgl. APICELLA (2002).

geraumer Zeit versucht, solch eine Service Oriented Architecture aufzubauen. Im Kontext der Systemintegration muss allerdings beachtet werden, dass eines der primären Ziele der EAI, die Reduktion der Punkt-zu-Punkt-Integrationsszenarien nicht durch eine ähnlich komplexe SOA ersetzt wird.

Zwei Hauptprobleme bei der Anbindung von Altanwendungen sind standardisierte Schnittstellen und die Granularität der Services. Die „Wrapping“-Problematik von Legacy-Systemen wurde bereits beschrieben. Bei der serviceorientierten Integration bestehender Anwendungen kommt es auf die Granularität der als Services verpackten Funktionen an. So macht es Sinn, diejenigen Funktionen als Services zugänglich zu machen, die allgemein benötigt und somit wieder verwendet werden sollten. Sie sollten eine komplette Arbeitseinheit ausführen sowie in ihrer Funktion und im Ergebnis gut beschreibbar sein.¹⁰⁶

Mit größter Wahrscheinlichkeit liegen die Altanwendungen jedoch monolithisch bzw. in falscher Granularität vor.¹⁰⁷ Das Problem der Granularität stellt sich auch für Anwendungen, die in Programmiersprachen geschrieben wurden, die mehr oder weniger direkt unter Anwendung spezieller Tools in Web Services umgewandelt werden können. Obwohl sich die Verzeichnisse, die Services beschreiben aus Definitionen von Schnittstellen bereits bestehender Technologien verteilter Softwaresysteme (Java-Klassen, JavaBeans, CORBA-Objekte, Visual Basic-Klassen, C#) erzeugen lassen, sind diese zumeist ebenfalls nicht in der für Services geeigneten Granularität definiert.¹⁰⁸

4.2.6 Process Integration

Die geschäftsprozessorientierte Anwendungsintegration hat ihren Fokus auf einer höheren Abstraktionsebene als die informationsorientierte. Sie richtet ihren Blick auf das Geschäftsmodell bzw. die Geschäftsprozesse mit der Geschäftslogik, den Prozessabläufen und der Hierarchisierung von Prozessen in Subprozesse. Der Ansatz ist eng mit dem der Geschäftsprozessautomatisierung verbunden. Die Prozesse werden einheitlich zentral gestaltet und definiert. Die Anwendungen werden letztendlich durch die Anbindung an die Prozesse integriert. Voraussetzung dafür sind Schnittstellen zu den Funktionen und Daten der Anwendungen, die in einer geeigneten Granularität vorhanden sein müssen. Die dienstorientierte und informationsorientierte Anwendungsintegration spielt bei der Bereitstellung der Schnittstellen eine Rolle.

¹⁰⁶ Vgl. NARSU/MURPHY (2003), S. 5.

¹⁰⁷ Vgl. ERLIKH (2003), S. 2.

¹⁰⁸ Vgl. NEWCOMER (2002), S. 85, 110.

„Durch seine Prozessmanagementfunktionalität hebt sich der EAI-Lösungsansatz wesentlich von dem traditionellen Middleware-basierten Integrationsansatz ab.“¹⁰⁹

Das wird auch in den Prozessebenenmodellen deutlich, die letztendlich in allen kommerziellen EAI-Toolarchitekturen als auch in den wissenschaftlich motivierten Referenzarchitekturen verwendet wird.

Die Prozesse müssen auf einer genügend abstrakten Ebene definiert werden sowie mit einer Modellierungssprache und (grafischen) Modellierungstools modellierbar sein. Durch die Trennung zwischen der Geschäftsprozessebene und den darunter liegenden Integrationsschichten können die Prozesse angepasst werden, ohne dass die darunter liegenden Implementierungen geändert werden müssen. Gleiches gilt für Änderungen in den unteren Integrationsschichten.

Die Komponente des Prozessmanagements ist für die Funktionalitäten Prozessmodellierung, Prozesssteuerung und Prozesskontrolle zuständig. In der Prozessmodellierung werden die Prozessmodelle definiert und ihre Zugriffe auf Anwendungen und Ressourcen, ihre Abläufe, ihre Informationsflüsse sowie die einbezogenen Personen beschrieben. Die modellierten Geschäftsprozesse bilden eine Abstraktionsebene oberhalb der darunter liegenden Anwendungen. Sie greifen auf Teilprozesse zu, die in den Anwendungen definiert sind. Die Prozesssteuerung koordiniert die Arbeitsabläufe, die Informationsflüsse und die Interaktion mit den Anwendungen gemäß dem Prozessmodell. Die Prozesskontrolle führt ein Prozessmonitoring und eine Prozessoptimierung durch.

5 Management modularisierter Architekturen

Nach der separaten Darstellung einer Flexibilisierung von Organisation und IT werden im folgenden Kapitel die Konzepte integrierter Unternehmensarchitekturen auf Methoden untersucht, die die Domänen im beschriebenen Kontext verbinden.

5.1 Definition Architektur

Die inzwischen oft verwendeten Begriffe der Architektur, des Architekturmanagements bzw. einer stringenten Architekturlehre folgen keiner einheit-

¹⁰⁹ KAIB (2002), S. 119.

lichen Definition.¹¹⁰ Die Literatur betrachtend muss festgestellt werden, dass die existierenden Definitionen stark von dem Kontext des jeweiligen Autors abhängen. Im Folgenden werden vor allem Argumente hervorgehoben, die von individuellen Situationen abstrahieren.¹¹¹

Elemente

Der erweiterten Definition eines Informationssystems von PICOT/MAIER folgend, umfasst eine Systemarchitektur nicht ausschließlich technische Elemente. Vielmehr wird von einem sozio-technischen System ausgegangen, das die gesamte technische Infrastruktur und deren Nutzer einbezieht.¹¹²

Ziele

Architekturbetrachtungen werden von verschiedenen Personengruppen durchgeführt. So führen Unternehmen, Softwaresystemhersteller, Beratungsunternehmen und Forschungseinrichtungen sehr unterschiedliche Ziele an. Dieser Beitrag stellt die Interessen von Unternehmen als Betreiber technischer Infrastrukturen in komplexen organisatorischen Umgebungen in den Mittelpunkt.

Nach KRCMAR steht die Nachhaltigkeit der Architektur als Hauptziel im Vordergrund. Zukünftige Entwicklungen sollen vorweggenommen werden. Das Thema der Nachhaltigkeit von IT wird in der Fachliteratur vielfältig diskutiert.¹¹³ Vor allem technische bzw. prozessorientierte Standards und die Orientierung an der Geschäftsstrategie werden in dieser Diskussion betont.¹¹⁴ Als Voraussetzung für Nachhaltigkeit gilt die Eigenschaft der Flexibilität.¹¹⁵ Je flexibler eine Architektur auf Änderungen eingehen kann, desto nachhaltiger ist sie.

¹¹⁰ Vgl. WALL (1996), S. 26.

¹¹¹ Die Autoren beziehen sich hier nur auf Definitionen, die sich im Spannungsfeld der Wirtschaftsinformatik, sprich zwischen Organisation und IT einordnen lassen, nicht auf oft verwendete Verweise auf andere Fachgebiete wie Ingenieurwissenschaften, Architektur im Sinne des Entwerfens und Bauens von Gebäuden etc.

¹¹² Vgl. PICOT/MAIER (1992), S. 923.

¹¹³ Siehe u. a. GRONAU (2000a).

¹¹⁴ Vgl. KRCMAR (1990), S. 395 ff.

¹¹⁵ Der Flexibilität von IT steht vor allem die Stabilität in der Ablaufumgebung gegenüber. Dazu mehr bei ALLEN/BOYNTON (1991), S. 436.

Eigenschaften

Um die Vergleichbarkeit bzw. Bewertbarkeit verschiedener Architekturkonzepte zu ermöglichen, sollte eine Architektur anhand ihrer wichtigsten Eigenschaften beschrieben werden können. Die Literatur bietet diverse Kriterienkataloge, um Architekturen zu spezifizieren.

Im Folgenden werden die Kriterien Abstraktion, Strukturbeschreibung, Muster, ganzheitliche Betrachtung und der Planungscharakter von Architekturen besprochen.¹¹⁶

WALL beschreibt das Kriterium der Abstraktion als die Konstruktion einer Architektur mit idealisierten Bausteinen, die von individuellen Details abstrahieren¹¹⁷. GÖRLIN und JÓKAY empfehlen im Zusammenhang mit der Abstraktion eine komplexitätsreduzierende Verkürzung der Modelleigenschaften, die vor allem darauf abzielt, die Funktionen und Systemschnittstellen herauszuarbeiten und in den Vordergrund zu heben.¹¹⁸ Architekturen im Sinne dieses Beitrages haben immer Modellcharakter, der von der realen Komplexität abstrahiert und nur die relevanten Eigenschaften und Zusammenhänge beschreibt.

KRCMAR versteht Architekturen „im Wesentlichen als eine Beschreibung von Strukturen“¹¹⁹. Auch MERTENS definiert den Architekturbegriff mit einer „stark verdichteten Sicht auf die Anordnung von Vorgängen bzw. Prozessen“¹²⁰. Prinzipiell entstehen Architekturen auf Grundlage von Prozessen und Abläufen, die funktionale Vorgaben und Grenzen der Architektur definieren.

Die Verwendung des Musterbegriffes¹²¹ führt zu mehrdeutiger Auslegung. Es gibt in der einschlägigen Fachliteratur wenige Quellen, die für die Architekturbeschreibung von IT-Systemen Muster als grundlegendes Beschreibungskriterium anführen. ALEXANDER definiert Muster wie folgt:

„Each pattern [Muster, Übersetzung d. Autoren] describes a problem which occurs over and over again in our environment, and then describes the core of the solution to that problem, in such a way that you can use this solution a million times over, without ever doing it the same way twice.“¹²²

¹¹⁶ Unter anderem aus HORN/REINKE (2002), S. 13 ff.; BASS/CLEMENTS/KAZMAN (2003), S. 19 ff.; WALL (1996), S. 26 ff.; POHLAND (2000), S. 9 ff.

¹¹⁷ Bei WALL am Beispiel von Hardware-Architekturen.

¹¹⁸ Vgl. GÖRLIN/JÓKAY (1992), S. 3.

¹¹⁹ Vgl. KRCMAR (1990), S. 396.

¹²⁰ Vgl. MERTENS (1991), S. 16.

¹²¹ Im technischen Kontext oft *Patterns* genannt [Anm. d. Autoren].

¹²² ALEXANDER (1995), S. 12 ff.

Der Autor ist Architekt. Daher gilt es, diese Definition als abstrakt zu betrachten. Allerdings erscheint sie geeignet, auch andere Sachverhalte sehr zwingend zu beschreiben. Prinzipiell stellen Muster eine Möglichkeit dar, auf wiederholt auftretende Probleme standardisiert zu reagieren. Allerdings sind die Probleme und der Kontext selten identisch, vielmehr nur ähnlich. Die Kunst der Erkennung von allgemeingültigen Mustern in Architekturen ist wohl die Abgrenzung von Problemlösungsansätzen in entsprechend kleine Teillösungen, so dass die Verwendung der Muster für diverse Problemlösungen in der Praxis anwendbar ist.¹²³

Nach KAZMAN stehen Muster in direktem Zusammenhang mit der Eigenschaft der strukturellen Beschreibung, sprich: sie sind allgemein geltende Regeln, welche den logischen bzw. funktionalen Zusammenhang der Elemente der Architektur beschreiben.¹²⁴ Ein Beispiel für ein Muster ist das funktionale Abhängigkeitsverhältnis zwischen Architekturelementen in einer Client-Server-Infrastruktur. Es wird abstrakt definiert, nach welchen Prinzipien Dienste angeboten (Server) und genutzt (Client) werden. Dabei kann der Server eines Dienstes gleichzeitig auch Client eines anderen Dienstes sein. Das Muster beschreibt den allgemeingültigen Zusammenhang, nicht die individuelle Problemlösung.

Alle Architekturen haben Planungscharakter. Sie sind in der Regel das Ergebnis eines Planungsprozesses¹²⁵ und stellen nach ihrer Definition selbst einen Masterplan für die ganzheitliche Realisierung zukünftiger Maßnahmen dar.¹²⁶

Bis hier wurde Architektur bereichsunabhängig betrachtet. Im Folgenden soll nun der Kontext der Gestaltung effizienter Unternehmensstrukturen mit Hilfe von Informationstechnik eine größere Rolle spielen. Der Begriff Architektur ist auch in der Wirtschaftsinformatik nicht einheitlich definiert. Oft wird zwischen Geschäfts- und Applikationsarchitektur unterschieden.¹²⁷ Die Geschäftsarchitektur definiert betriebswirtschaftliche Ziele und Strukturen. Dagegen bildet die Applikationsarchitektur die IT-Systeme ab. Die Herausforderung besteht darin, Methoden zu finden, um beide Welten zu verbinden.

Das nächste Kapitel stellt Ansätze vor, die Unternehmens- und IT-Architekturen integrieren.

¹²³ Vgl. FOWLER (2003), S. 10.

¹²⁴ Vgl. BASS/CLEMENTS/KAZMAN (2003), S. 24 f.

¹²⁵ Vgl. KRCMAR (1990), S. 395.

¹²⁶ Vgl. SCHMALZL, Architekturmodelle zur Planung der Informationsverarbeitung von Kreditinstituten.

¹²⁷ Vgl. POHLAND (2000), S. 19.

5.2 Holistische Architekturkonzepte

Das Paradigma der Verbindung von strategischen Unternehmenszielen, Geschäftsprozessen und der technischen Infrastruktur wird in der Wirtschaftsinformatik bereits seit vielen Jahren diskutiert.¹²⁸ Dabei steht außer Frage, dass die Anforderungen an IT-Architekturen im Allgemeinen und Softwaresysteme im Speziellen aus den unternehmensindividuellen Bedürfnissen abgeleitet werden müssen.

Neben bzw. über der zielorientierten Gestaltung technischer Strukturen stehen die Definition, die Etablierung und die permanente Anpassung der Geschäftsprozesse eines Unternehmens an die Notwendigkeiten des Umfeldes.

Die Wechselwirkungen in dem skizzierten Spannungsfeld sind vielfältig, individuell und komplex. Daher fällt es schwer, allgemeine Regeln und Methoden zu entwickeln, die hier ein durchgängiges und nachhaltiges Vorgehen gewährleisten. Leider finden sich meist nur Ansätze, die vorwiegend prozessorientiert oder organisatorisch oder technisch motiviert sind.

Ziel der Betrachtung von Architekturkonzepten ist die Komplexitätsreduktion innerhalb des oben beschriebenen Spannungsfeldes.¹²⁹ Für dieses übergeordnete Ziel bieten Konzepte meist verschiedene Sichten auf Informationssysteme.¹³⁰ Die Beschreibung bestimmter Sichten auf IT-Systeme wird als Architekturmodell bezeichnet. Die Ziele eines Architekturmodells in diesem Sinne sind Vollständigkeit, Verständlichkeit und Flexibilität.¹³¹

IBM beschreibt das Strategic Architecture Framework (Abbildung 10) vor dem Hintergrund eines stringenten Architekturmanagements, das vor allem IT-Kosten senken soll. Nebeneffekte sind die Etablierung von Standards und kürzere Laufzeiten von IT-Projekten. Der Ansatz geht von einer gegenseitigen Beeinflussung von Geschäftsstrategie und IT aus und kann zur Veranschaulichung von Zusammenhängen verwendet werden.

Operative Aussagen für die Gestaltung der Architektur werden allerdings nicht explizit unterstützt. Die Literatur bietet neben praxisorientierten Konzepten auch wissenschaftlich ambitionierte Ansätze. Im Folgenden werden drei vorgestellt.

KRCMAR diskutiert bereits seit Ende der 1980er Jahre die Notwendigkeit, aus den strategischen Zielen und der daraus definierten Aufbau- und Ablaufor-

¹²⁸ Vgl. WALL (1996), S. 47.

¹²⁹ Vgl. POHLAND (2000), S. 46.

¹³⁰ Vgl. WALL (1996), S. 38.

¹³¹ Vgl. KRCMAR (1990), S. 398.

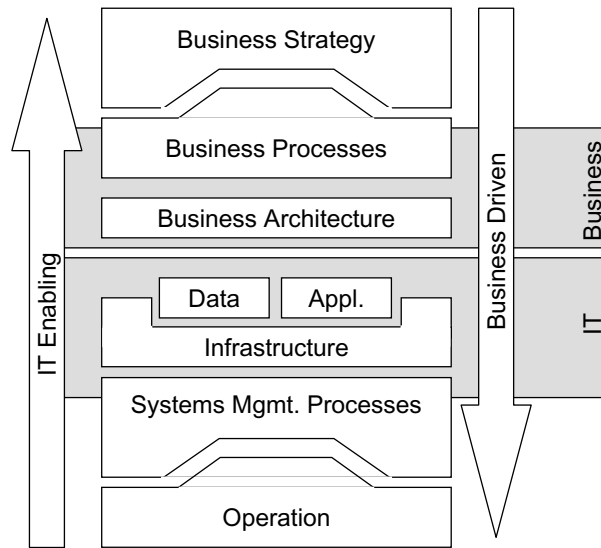


Abbildung 10. IBM Strategic Architecture Framework, vgl. HARBEKE (2002), S. 15

ganisation eines Unternehmens, alle Schichten des IT-Architekturmodells abzuleiten. Dem oft diskutierten Anachronismus zwischen fachlichen Anforderungen und IT-Infrastrukturen trägt er in der Abbildung durch die Wahl eines Kreisels Rechnung. Nur wenn alle Teile des Modells ausbalanciert sind, erreicht man ein Gleichgewicht.¹³² KRCMAR geht von einer Rückkopplung der Möglichkeiten einer vorhandenen IT-Infrastruktur auf die strategischen Ziele eines Unternehmens aus.¹³³

KRCMAR weist zwar ausdrücklich auf einen Zusammenhang zwischen den beschriebenen Ebenen hin, liefert aber keinen Lösungsansatz für die Versöhnung derselben.¹³⁴

ZACHMANN unterscheidet in seinem Architekturmodell in Ebenen und Objekte. Die Ebenen beachten den Zusammenhang zwischen Zielen des Unternehmens und der Entwicklung von Software. Objekte sind technischer Natur und gehen auf abgebildete Daten, Prozesse und Netzwerke ein. Nachteil ist die fehlende Betrachtung bereits vorhandener Infrastruktur. Allerdings bietet sich die iterative Überführung unternehmensrelevanter Kriterien in technische Infrastruktur für weitergehende Betrachtungen an.¹³⁵

¹³² Vgl. POHLAND (2000), S. 46.

¹³³ Vgl. dazu den Begriff des *organizational imperative* S. 19.

¹³⁴ Vgl. WALL (1996), S. 41.

¹³⁵ Vgl. POHLAND (2000), S. 47.

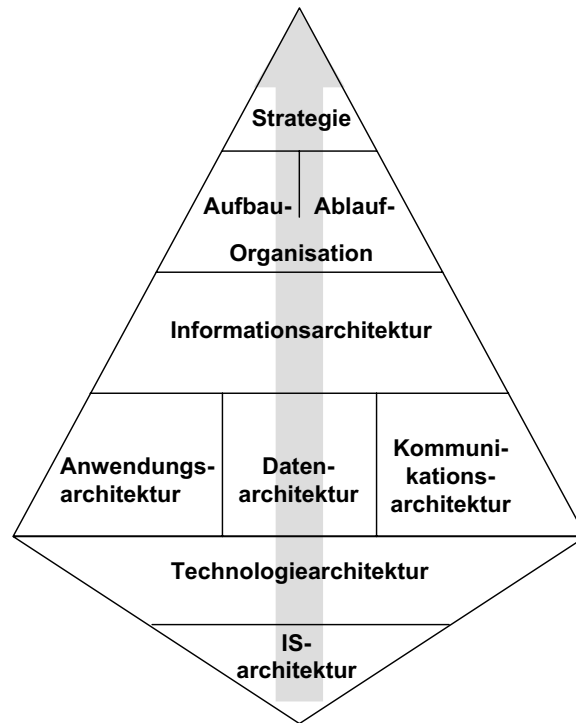


Abbildung 11. Architekturkonzept nach KRCMAR, KRCMAR (1990), S. 399

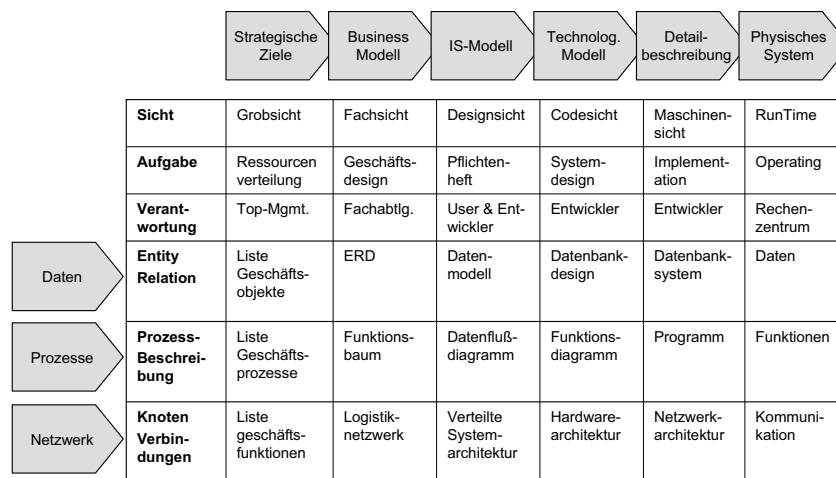


Abbildung 12. Architekturkonzept nach ZACHMANN, IMMON/ZACHMANN/GEIGER (1997)

Ähnlich wie KRCMAR lässt ZACHMANN Aspekte offen. Vor allem die Zusammenhänge zwischen den definierten Ebenen und Objekten werden nicht deutlich.

Am weitesten geht SCHEER in dem von ihm vorgeschlagenen Architekturmodell.¹³⁶ Er beschreibt eine betriebswirtschaftliche Methode für die Entwicklung von Informationssystemen. Das Konzept sieht vier Sichten vor: die Organisationssicht, die Datensicht, die Vorgangssicht und die Steuerungssicht. Jede dieser Sichten unterscheidet die Ebenen Fachkonzept, DV-Konzept und Implementierung. SCHEER stellt im Gegensatz zu KRCMAR einen expliziten Zusammenhang zwischen der Organisationsstruktur und der IT her. Die Steuerungssicht verbindet die Ebenen der verbleibenden drei Sichten. Es werden dabei logische Zusammenhänge zwischen der Organisation, ihren Prozessen, den korrespondierenden Funktionen der Software und den notwendigen Daten hergestellt.¹³⁷ Abbildung 13 zeigt das Architekturkonzept nach SCHEER.

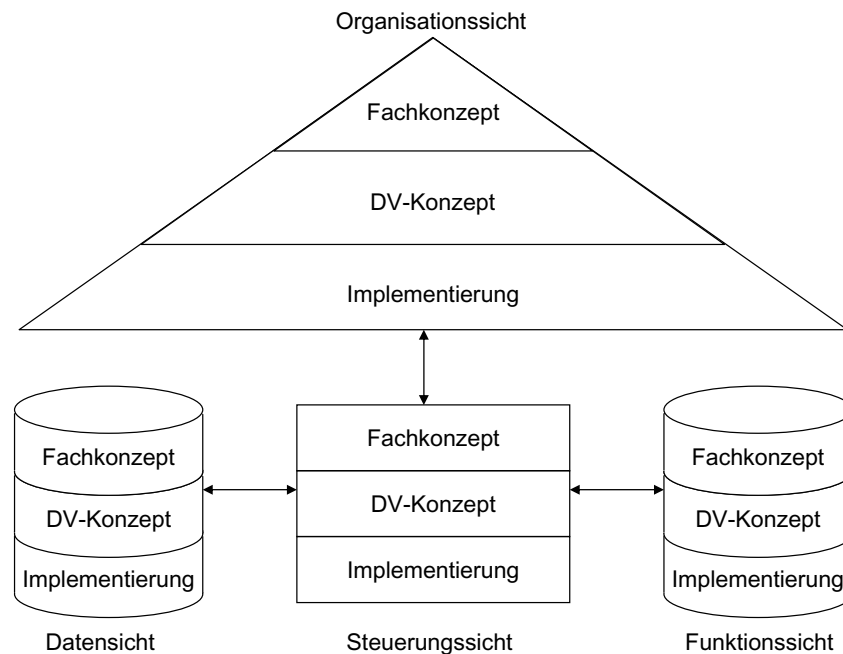


Abbildung 13. Architektur integrierter Informationssysteme, vgl. SCHEER (1991), S. 18

¹³⁶ Vgl. SCHEER (1991), S. 3.

¹³⁷ Vgl. SCHEER (1991), S. 92 ff.

Zusammenfassend kann gesagt werden, dass von einer gegenseitigen Abhängigkeit zwischen den Ebenen Organisationsgestaltung und IT-Systemlandschaft ausgegangen werden muss. Die Gestaltung der technischen Systeme wird durch geschäftliche Ziele getrieben. IT spielt eine große Rolle als „enabling-factor“. Architekturkonzepte stellen nur wenig operationale Methoden zur Verfügung, die zur nachhaltigen Versöhnung zwischen Organisation und unterstützender IT in der Lage wären.

6 Fazit

Dieser einführende Beitrag hat bewusst die Problemfelder im Überblick angesprochen. Auf einige Fragen gibt es bisher keine zwingenden Antworten. Vor allem in technisch getriebenen EAI-Projekten wird schnell deutlich, dass ohne fachliche Konzeption tatsächlich nur Schnittstellen abgelöst werden. Die Komplexität der IT-Infrastruktur erhöht sich durch die Einführung eines neuen Systems – der zentralen EAI-Plattform. Eine Flexibilisierung der Abläufe stellt sich vordergründig ein. Wirkliche Flexibilisierung von Prozessen wird erst erreicht, wenn auf fachlicher Ebene Abläufe modularisiert werden und die technische Infrastruktur in der Lage ist, diese Module effizient abzubilden und ablauffähig zur Verfügung zu stellen. Es dauert Jahre, bis sich zentrale Lösungen über Einsparungen bei der Administration von Punkt-zu-Punkt-Schnittstellen amortisieren. Selbst wenn die EAI-Maßnahmen wirtschaftlich erfolgreich sind, stellt sich die Frage, ob eine nachhaltige Erhöhung der Flexibilität des Gesamtsystems Unternehmung erreicht wurde.

Zur Lösung dieses Spannungsfeldes müssen holistische Architekturkonzepte implementiert werden, die mit EAI-Komponenten arbeiten, zusätzlich aber weitere Methoden und Technologien einsetzen, die auf Standardisierung und Wiederverwendung basieren.

Ein in diesem Zusammenhang derzeit viel diskutiertes Paradigma ist die Dezentralisierung von Integrationsarchitekturen. Zentrale Konzepte von EAI-Plattformen lassen sich nur mangelhaft mit organisatorischen Anforderungen von Unternehmen vereinbaren. In den Beiträgen von HAGEN und GALLAS in diesem Buch werden sowohl organisatorische als auch technische Grundlagen für die Implementierung dezentraler Infrastrukturen thematisiert. Die Etablierung serviceorientierter Architekturen (SOA) wird zur Zeit von vielen Unternehmen als Chance identifiziert, nachhaltige Lösungen zu implementieren.

Die oben erläuterte Definition von Clustern in Makro- und Mikroebene wurde auf dem EAI-Expertentag im Februar 2003 durch mehrere Vorträge bespro-

chen. Inzwischen betonen Experten die Notwendigkeit des Einsatzes zentraler EAI-Lösungen innerhalb dezentraler Gesamtsystemarchitekturen. Auf der Mikroebene innerhalb eines Clusters bieten sich zentrale Lösungen an. Diese müssen so konzipiert sein, dass sie innerhalb einer clusterübergreifenden Architekturebene serviceorientiert eingebunden werden können. Dem Trend folgend, bieten inzwischen die meisten Anbieter kommerzieller EAI-Plattformen ihre Funktionen auch als Service an. Der Beitrag von KELLER in diesem Buch zeigt Erfahrungen der Generali Gruppe bei der Einführung zentraler Infrastrukturen innerhalb eines Clusters.

Die Grenzen zwischen der Integration innerbetrieblicher Cluster und unternehmensübergreifender Szenarien können bei dezentralen Lösungsansätzen als fließend beschrieben werden. Die Möglichkeiten prozessorientierter Integration über Unternehmensgrenzen hinweg stellt vor allem in der nahen Zukunft der Finanzdienstleistungsbranche einen entscheidenden Erfolgsfaktor dar. Hier werden bald spezialisierte Anbieter zentrale Leistungen übernehmen, die deutsche Allfinanzbanken derzeit noch ineffizient in Eigenregie erstellen. Ohne konsequent durchgehende Prozesse ist ein solches Prozess-Outsourcing allerdings nicht realisierbar. Der Beitrag von DIETRICH in diesem Buch thematisiert in dem Zusammenhang die Verwendung bekannter B2B-Standards für die analoge Anwendung im EAI-Bereich.

Eine weitere wichtige Komponente holistischer Architekturkonzepte sind Modellierungsmethoden, -notationen und -werkzeuge. Für die durchgehende Modellierung und Implementierung von Geschäftsprozessen in die Ebene der IT-Systeme werden zur Zeit Methoden und Standardnotationen für verschiedene Plattformen entwickelt. Die Beiträge von GALLAS und DIETRICH in diesem Buch besprechen in diesem Zusammenhang unterschiedliche Ansätze.

Weiterhin kann die Model Driven Architecture (MDA) als eine seit einigen Jahren diskutierte Vorgehensweise zur modellbasierten Neuimplementierung von Systemen auch für die Integration vorhandener Infrastrukturen eingesetzt werden. Der Beitrag von PETTKOFF in diesem Buch beschäftigt sich auf Grundlage von MDA mit der sogenannten Model Driven Architecture Integration (MDAI).

Literaturverzeichnis

- ALEXANDER, C.: Eine Muster-Sprache – A Pattern Language. Städte – Gebäude – Konstruktion. Wien: Löcker, 1995.
- ALEXANDER, S.: EAI – Hoffnungsträger mit noch unklarer Zukunft. In: Computerwoche 2000, Nr. 30, S. 9–10.
- ALLEN, B. R./BOYNTON, A. C.: Information Architecture: In search of efficient flexibility. In: MIS Quarterly 15 1991, Nr. 4, S. 435–445.
- APICELLA, M.: Side by side in perfect harmony? In: InfoWorld February 2002 (URL: http://www.infoworld.com/article/02/02/22/020225fedata_1.html) – Zugriff am 12.03.2003.
- APPLEGATE, L. M./CASH, J. I./MILES, D. Q.: Information Technology and Tomorrow's Manager. In: Harvard Business Review 66 November–December 1988, S. 128–136.
- ASHBY, W. R.: An Introduction to cybernetics. 5. Auflage, London: Chapman & Hall, 1970.
- BASS, L./CLEMENTS, P./KAZMAN, R.: Software Architecture in Practice. 2. Auflage, Boston: Pearson Education Inc., 2003.
- BLEICHER, K.: Unternehmenskultur und strategische Unternehmensführung. In: HAHN, D./TAYLOR, B. (Hrsg.): Strategische Unternehmensplanung – Strategische Unternehmensführung. 8. Auflage, Heidelberg: Physica, 1999, S. 223–265.
- BUHL, L./CHRIST, J./PAPE, U.; DANGELMAIER, W./BOHNER, M. (Hrsg.): Marktstudie: Softwaresysteme für Enterprise Application Integration. Paderborn: ALB-HNI, 2001.

- BURGFELD, B.: Organisationstheorie und Informationstechnologie. Wiesbaden: DUV, 1998 (zugl. Dissertation, Universität Bielefeld, 1998).
- CHANDLER, A. D.: Strategy and Structure. Chapters in the History of the Industrial Enterprise. 3. Auflage, Cambridge, Mass. et al.: MIT Press, 1966.
- CORSTEN, H.: Komponenten und Instrumente der produktionswirtschaftlichen Flexibilität. In: WISU 17 1988, Nr. 10.
- DANGELMAIER, W. ET AL.: Klassifikation von EAI-Systemen. June 2002 [⟨URL: http://www.competence-site.de/eaisysteme.nsf/f1b7ca69b19cbb26c12569180032a5cc/77f3c9de96765715c1256cf3006bcc11!OpenDocument⟩](http://www.competence-site.de/eaisysteme.nsf/f1b7ca69b19cbb26c12569180032a5cc/77f3c9de96765715c1256cf3006bcc11!OpenDocument) – Zugriff am 24.06.2003, Auch in HMD – Praxis der Wirtschaftsinformatik. dpunkt, Heft 225, Juni 2002, S. 61–71.
- DERSZTELER, G.: Prozessmanagement auf Basis von Workflow-Systemen. Lohmar, Köln: Josef Eul, 2000 (zugl. Dissertation, Technische Universität Berlin, 1998).
- ERLIKH, L.: Integrating Legacy into Extended Enterprise: Using Web Services. 2003 [⟨URL: http://objectz.com/whitepaper/integratinglegacy.pdf⟩](http://objectz.com/whitepaper/integratinglegacy.pdf) – Zugriff am 12.07.2003, Relativity Technologies.
- FÖHR, S./LENZ, H.: Unternehmenskultur und ökonomische Theorie. In: STAEHLE, W. H./CONRAD, P. (HRSG.): Managementforschung 2. Berlin, New York: De Gruyter, 1992, S. 111–162.
- FOWLER, M. (HRSG.): Patterns of Enterprise Application Architecture. Boston: MITP, 2003.
- FRANK, H./GRONAU, N.: Vorgehensmodell der Systemanalyse. In: KRALLMANN, H./FRANK, H./GRONAU, N. (HRSG.): Systemanalyse im Unternehmen – Vorgehensmodelle, Modellierungsverfahren und Gestaltungsoptionen. 4. Auflage, München, Wien: Oldenbourg, 2002, S. 47–114.
- FRESE, E./STÖBER, H. (HRSG.): E-Organisation. Wiesbaden: Gabler, 2002.
- FRESE, E. (HRSG.): Handwörterbuch der Organisation. 3. Auflage, Stuttgart: Poeschel, 1992.
- FRESE, E.: Grundlagen der Organisation: Konzept – Prinzipien – Strukturen. 8. Auflage, Wiesbaden: Gabler, 2000.
- FRESE, E.: Theorie der Organisationsgestaltung und netzbasierte Kommunikationseffekte. In: FRESE, E./STÖBER, H. (HRSG.): E-Organisation. Wiesbaden: Gabler, 2002, S. 191–241.

- GERYBADZE, A.: Strategisches Management und dynamische Konfiguration der Unternehmens-Umwelt-Beziehungen. In: LEISTEN, R./KRCAL, H.-C. (HRSG.): Nachhaltige Unternehmensführung – Systemperspektiven. Wiesbaden: Gabler, 2003, S. 83–100.
- GLASER, H./SCHRÖDER, E. F./V. WERDER, A. (HRSG.): Organisation im Wandel der Märkte. Wiesbaden: Gabler, 1998.
- GÖRLIN, H./JÓKAY, Z.: Rechnerarchitektur – ein Überblick. In: HMD 29 1992, Nr. 164, S. 3.
- GOSAIN, S./THILLAIRAJAH, V.: EAI: The Business Drivers and Technical Challenges – A Whitepaper. 2001 (URL: http://eai.ebizq.net/str/gosain_1a.html) – Zugriff am 10.01.2003.
- GRONAU, N.: Nachhaltige Architekturen industrieller Informationssysteme bei organisatorischem Wandel. Habilitation Technische Universität Berlin, 2000a.
- GRONAU, N.: Modellierung von Flexibilität in Architekturen industrieller Informationssysteme. In: SCHMIDT, H. (HRSG.): Modellierung betrieblicher Informationssysteme. Proceedings der MobIS-Fachtagung. Siegen, 2000b, S. 125–145.
- GRONAU, N.: Wandlungsfähige Informationssystemarchitekturen – Nachhaltigkeit bei organisatorischem Wandel. Berlin: Gito, 2003.
- GROUP OF TEN: Report on Consolidation in the Financial Sector. 2001 (URL: <http://www.bis.org/publ/gten05.pdf>) – Zugriff am 05.09.2003.
- GRUDÉN, A./STRANNEGARD, P.: Business Process Integration – The Next Wave. In: EAI Journal January 2003, S. 8–12.
- GRUHN, V./SCHÖPE, L.: Integration von Legacy-Systemen mit Electronic Commerce Anwendungen. 2002 (URL: <http://ls10-www.informatik.uni-dortmund.de/~schoepe/artikel/memo124.pdf>) – Zugriff am 05.09.2003, Internes Memorandum des Lehrstuhls für Software-Technologie, Nr. 124, Fachbereich Informatik, Universität Dortmund.
- GULP: Gulp Knowledge Base: Definition – EAI. 2002 (URL: <http://www.gulp.de/kb/pt/techexpert/mysap.html>) – Zugriff am 08.01.2003.
- HAHN, D./HUNGENBERG, H.: PuK – Wertorientierte Controllingkonzepte. 6. Auflage, Wiesbaden: Gabler, 2001.

- HAHN, D./TAYLOR, B. (HRSG.): Strategische Unternehmensplanung – Strategische Unternehmensführung. 8. Auflage, Heidelberg: Physica, 1999.
- HAHN, D.: Konzepte Strategischer Führung – Entwicklungstendenzen in Theorie und Praxis unter besonderer Berücksichtigung der Globalisierung. In: HAHN, D./TAYLOR, B. (HRSG.): Strategische Unternehmensplanung – Strategische Unternehmensführung. 8. Auflage, Heidelberg: Physica, 1999, S. 1037–1057.
- HAMMER, M./CHAMPY, J.: Reengineering The Corporation – A Manifesto for Business Revolution. New York: HarperCollins, 2001.
- HAMPEL, A.: Integration von IT-Systemen. 2002 (URL: http://www.pri.univie.ac.at/~hampel/WI_EIS/vorlesung/V0{\%}20WI_EIS{\%}20-{\%}2012{\%}20-{\%}20Integration.pdf), Unterlagen zur Vorlesung „Entwicklung und Einführung betrieblicher IT-Systeme“, Universität Wien, Institute for Computer Science and Business Informatics.
- HARBEKE, A.: Kostenoptimierung in der IT. IBM, 2002.
- HASSELBRING, W.: Information System Integration. In: Communications of the ACM 43 2000, Nr. 6, S. 33–38.
- HILL, W./FEHLBAUM, R./ULRICH, P.: Organisationslehre 1: Ziele, Instrumente und Bedingungen der Organisation sozialer Systeme. 5. Auflage, Bern, Stuttgart, Wien: Haupt, 1994.
- HILL, W./FEHLBAUM, R./ULRICH, P.: Organisationslehre 2: Theoretische Ansätze und praktische Methoden der Organisation sozialer Systeme. 5. Auflage, Bern, Stuttgart, Wien: Haupt, 1998.
- HORN, E./REINKE, T.: Softwarearchitektur und Softwarebauelemente: Eine Einführung für Softwarearchitekten. München, Wien: Hanser, 2002.
- HORVÁTH, P.: Controlling. 7. Auflage, München: Vahlen, 1998.
- IMAI, M.: Kaizen. Frankfurt a. M., Berlin: Ullstein, 1993.
- IMMON, W. H./ZACHMANN, J. A./GEIGER, J. G. (HRSG.): Data stores, Data Warehouseing and the Zachmann framework: manageing enterprise knowledge. New York: McGraw-Hill, 1997.
- JURKOVICH, R.: A Core Typology of Organizational Environments. In: Administrative Science Quaterly 19 1974, S. 380–394.
- KAIB, M.: Enterprise Application Integration: Grundlagen, Integrationsprodukte, Anwendungsbeispiele. Wiesbaden: DUV, 2002 (zugl. Dissertation, Universität Marburg, 2002).

- KELLER, W.: Enterprise Application Integration – Erfahrungen aus der Praxis. Heidelberg: Dpunkt, 2002.
- KIESER, A./KUBICEK, H.: Organisation. 3. Auflage, Berlin, New York: De Gruyter, 1992.
- KIESER, A. (HRSG.): Organisationstheorien. 5. Auflage, Stuttgart: Kohlhammer, 2002.
- KNEER, G./NASSEHI, A.: Niklas Luhmanns Theorie sozialer Systeme. 4. Auflage, München: Wilhelm Fink, 2000.
- KNOF, H.-L.: CIM und organisatorische Flexibilität. München: Florentz, 1992 (zugl. Dissertation, Universität Bochum, 1991).
- KRALLMANN, H./FRANK, H./GRONAU, N. (HRSG.): Systemanalyse im Unternehmen – Vorgehensmodelle, Modellierungsverfahren und Gestaltungsoptionen. 4. Auflage, München, Wien: Oldenbourg, 2002.
- KRCAL, H.-C.: Systemtheoretischer Metaansatz für den Umgang mit Komplexität und Nachhaltigkeit. In: LEISTEN, R./KRCAL, H.-C. (HRSG.): Nachhaltige Unternehmensführung – Systemperspektiven. Wiesbaden: Gabler, 2003, S. 3–30.
- KRCMAR, H.: Bedeutung und Ziele von Informationssystem-Architekturen. In: Wirtschaftsinformatik 32 1990, Nr. 5, S. 395–402.
- KRÜGER, W.: Management permanenten Wandels. In: GLASER, H./SCHRÖDER, E. F./v. WERDER, A. (HRSG.): Organisation im Wandel der Märkte. Wiesbaden: Gabler, 1998, S. 227–249.
- KRÜGER, W.: Auswirkungen des Internet auf Wertketten und Geschäftsmodelle. In: FRESE, E./STÖBER, H. (HRSG.): E-Organisation. Wiesbaden: Gabler, 2002, S. 63–89.
- KRYSTEK, U.: Vertrauen als Basis erfolgreicher strategischer Unternehmensführung. In: HAHN, D./TAYLOR, B. (HRSG.): Strategische Unternehmensplanung – Strategische Unternehmensführung. 8. Auflage, Heidelberg: Physica, 1999, S. 266–288.
- LANG, C./UTIKAL, H.: Organisatorische Impulse durch Internet – Technologie und technologieinduzierte Strategien. In: FRESE, E./STÖBER, H. (HRSG.): E-Organisation. Wiesbaden: Gabler, 2002, S. 155–189.
- LANG, K.: Gestaltung von Geschäftsprozessen mit Referenzprozeßbausteinen. Wiesbaden: Gabler, 1997.
- LAUPENMÜHLEN, M./MOSCHITZ, R.: Herausforderung Basel II – Anforderungen des Basler Ausschusses für Bankenaufsicht auf der

- Grundlage des 2. Konsultationspapiers und aktuelle Entwicklungen. 2002 [⌈URL: http://www.ey.com/global/download.nsf/Germany/Broschuere_Herausforderung_Basel_II/\\$file/Herausforderung_Basel_II_09_2002.pdf⌋](http://www.ey.com/global/download.nsf/Germany/Broschuere_Herausforderung_Basel_II/$file/Herausforderung_Basel_II_09_2002.pdf) – Zugriff am 02.05.2003.
- LEAVITT, H./WHISLER, T.: Management in the 1980s: New Information Flows Cut New Organization Flows. In: Harvard Business Review 36 1958, S. 41–48.
- LEGNER, M.: Enterprise Application Integration. 2000a [⌈URL: http://www.informatik.uni-stuttgart.de/ipvr/as/personen/ruetschlin/seminars/Wrapping.pdf⌋](http://www.informatik.uni-stuttgart.de/ipvr/as/personen/ruetschlin/seminars/Wrapping.pdf), Seminararbeit, Universität Stuttgart.
- LEGNER, M.: Wrapping. In: LEGNER, M.: Enterprise Application Integration. 2000a [⌈URL: http://www.informatik.uni-stuttgart.de/ipvr/as/personen/ruetschlin/seminars/Wrapping.pdf⌋](http://www.informatik.uni-stuttgart.de/ipvr/as/personen/ruetschlin/seminars/Wrapping.pdf), Seminararbeit, Universität Stuttgart [⌈URL: http://www.informatik.uni-stuttgart.de/ipvr/as/personen/ruetschlin/seminars/Wrapping.pdf⌋](http://www.informatik.uni-stuttgart.de/ipvr/as/personen/ruetschlin/seminars/Wrapping.pdf), S. 5–6, Seminararbeit, Universität Stuttgart.
- LEISTEN, R./KRCAL, H.-C. (HRSG.): Nachhaltige Unternehmensführung – Systemperspektiven. Wiesbaden: Gabler, 2003.
- LEWIN, A. Y./HUNTER, S. D.: Information Technology & Organizational Design: A Longitudinal Study of Information Technology Implementations in the U.S. Retailing Industrie, 1980–1996. In: GLASER, H./SCHRÖDER, E. F./V. WERDER, A. (HRSG.): Organisation im Wandel der Märkte. Wiesbaden: Gabler, 1998, S. 251–286.
- LIEFER, R.: Matching Computer-Based Information Systems with Organizational Structures. In: MIS Quarterly 12 March 1988, Nr. 1, S. 63–73.
- LINTHICUM, D. S.: Enterprise Application Integration. Amsterdam: Addison-Wesley Longman, 2000.
- LUBLINSKY, B./FARRELL, M.: 10 Misconceptions About Web Services. In: EAI Journal February 2003, S. 30–33.
- MARKUS, M./ROBEY, D.: Information Technology and Organizational Change: Causal Structure in Theory and Research. In: Management Science 34 1988, Nr. 5, S. 583–589.
- MAY, M.: The Creative Destruction of EAI. In: EAI Journal October 2001, S. 56–58.

- MERTENS, P.: Integrierte Informationsverarbeitung. 8. Auflage, Wiesbaden: Gabler, 1991.
- MISCHE, M. A.: Defining Systems Integration. In: MYERSON, J. M. (HRSG.): Enterprise Systems Integration. Boca Raton et al.: Auerbach, 2002, S. 3–10.
- MOORMANN, J. (HRSG.): Handbuch Informationstechnologie in Banken. Wiesbaden: Gabler, 1999a.
- MOORMANN, J.: Umbruch in der Bankinformatik – Status quo und Perspektiven für eine Neugestaltung. In: MOORMANN, J. (HRSG.): Handbuch Informationstechnologie in Banken. Wiesbaden: Gabler, 1999a, S. 4–20.
- MORGAN, G.: Bilder der Organisation. Stuttgart: Klett-Cotta, 1997.
- MYERSON, J. M. (HRSG.): Enterprise Systems Integration. Boca Raton et al.: Auerbach, 2002.
- NARSU, U./MURPHY, P.: Web Services Adoption Outlook Improves. 2003
 (URL: <ftp://ftp.software.ibm.com/software/websphere/webservices/gigawebservicesoutlook.pdf>) – Zugriff am 12.07.2003, Giga Information Group.
- NEWCOMER, E.: Understanding Web services: XML, WSDL, SOAP and UDDI. Amsterdam: Addison-Wesley Longman, 2002.
- OESTEREICH, B.: Objektorientierte Softwareentwicklung – Analyse und Design mit der Unified Modeling Language. 4. Auflage, München: Oldenbourg, 1998.
- ÖSTERLE, H./RIEHM, R./VOGLER, P. (HRSG.): Middleware – Grundlagen, Produkte und Anwendungsbeispiele für die Integration heterogener Welten. Braunschweig et al.: Vieweg, 1996.
- ÖSTERLE, H.: Business Engineering: Prozeß- und Systementwicklung. Band 1, 2. Auflage, Berlin et al.: Springer, 1995.
- PICOT, A./MAIER, M.: Computergestützte Informationssysteme. In: FRESE, E. (HRSG.): Handwörterbuch der Organisation. 3. Auflage, Stuttgart: Poeschel, 1992, S. 923.
- PICOT, A./NEUBURGER, R.: Informationsbasierte (Re-)Organisation von Unternehmen. In: WEIBER, R. (HRSG.): Handbuch Electronic Business. 2. Auflage, Wiesbaden: Gabler, 2002a, S. 549–568.
- PICOT, A./REICHWALD, R./WIEGAND, R. T.: Die Grenzenlose Unternehmung – Information, Organisation und Management. 4. Auflage, Wiesbaden: Gabler, 2001.

- POHLAND, S.: Globale Unternehmensarchitekturen – Methode zur Verteilung von Informationssystemen. Berlin: Weißensee-Verlag, 2000.
- POLLOCK, J. T.: The Big Issue: Interoperability vs. Integration. In: EAI-Journal October 2001, S. 48–52.
- REICHWALD, R./WILDEMAN, H. (HRSG.): Kreative Unternehmen – Spitzenleistungen durch Produkt- und Prozessinnovationen. Stuttgart: Schäffer-Poeschel, 1995.
- RENK, T.: Auf der Suche nach dem Return-on-Investment. Ein Kostenvergleich zwischen EAI und herkömmlicher Integration. In: Computerwoche 36 2001, Nr. 28, S. 62–63.
- REN, F.: The Marketplace of C. 2002 [URL: http://www.public.asu.edu/~sim\\$mbfr2047/eai.html](http://www.public.asu.edu/~sim$mbfr2047/eai.html) – Zugriff am 22.05.2002.
- RIEHM, R./VOGLER, P.: Middleware: Infrastruktur für die Integration. In: ÖSTERLE, H./RIEHM, R./VOGLER, P. (HRSG.): Middleware – Grundlagen, Produkte und Anwendungsbeispiele für die Integration heterogener Welten. Braunschweig et al.: Vieweg, 1996, S. 47–48.
- RING, K.: EAI: Making the right Connections. Boston: Ovum Reports, 2000.
- ROCH, E.: Application Integration – Business and Technology Trends. In: EAI-Journal August 2002, S. 34–39.
- ROCKART, J. F./SHORT, J. E.: IT in the 90's: Managing Organizational Independence. In: Sloan Management Review Winter 1989, S. 7–17.
- ROMBACH, D.: Software nach dem Baukastenprinzip. In: Fraunhofer Magazin 2003, Nr. 1, S. 30–31.
- RUH, W. A./MAGINNIS, F. X./BROWN, W. J.: Enterprise Application Integration. New York: Wiley, John, & Sons, 2001.
- SCHECKENBACH, R. (HRSG.): Semantische Geschäftsprozessintegration. Wiesbaden: DUV, 1997 (zugl. Dissertation, Universität Würzburg, 1997).
- SCHEER, A.-W.: Architektur integrierter Informationssysteme. Berlin et al.: Springer, 1991.
- SCHEER, A.-W.: ARIS-Toolset – Die Geburt eines Softwareprodukts. In: REICHWALD, R./WILDEMAN, H. (HRSG.): Kreative Unternehmen – Spitzenleistungen durch Produkt- und Prozessinnovationen. Stuttgart: Schäffer-Poeschel, 1995, S. 307–337.
- SCHEIN, E. H.: Organizational Culture and Leadership. San Francisco/Washington/London: Jossey-Bass, 1985.

- SCHERER, A. G.: Kritik der Organisation oder Organisation der Kritik? – Wissenschaftstheoretische Bemerkungen zum kritischen Umgang Organisationstheorien. In: KIESER, A. (HRSG.): Organisationstheorien. 5. Auflage, Stuttgart: Kohlhammer, 2002, S. 1–37.
- SCHILL, A.: Middleware im Vergleich. 2000 (URL: [http://www.competence-site.de/eaisysteme.nsf/C937534DE4B6BC14C1256A14005E5D60/\\$File/middleware.pdf](http://www.competence-site.de/eaisysteme.nsf/C937534DE4B6BC14C1256A14005E5D60/$File/middleware.pdf)) – Zugriff am 08.01.2003.
- SCHMIDT, H. (HRSG.): Modellierung betrieblicher Informationssysteme. Proceedings der MobIS-Fachtagung. Siegen, 2000b.
- SOLARO, R.: Trends und Strömungen bei Rechnereinsatz und Organisation im Industrieunternehmen 1950–1997. Diplomarbeit, Technische Universität Berlin, Fachbereich Informatik, 1998.
- STAEHLE, W. H./CONRAD, P. (HRSG.): Managementforschung 2. Berlin, New York: De Gruyter, 1992.
- STÜNZER, L.: Systemtheorie und betriebswirtschaftliche Organisationsforschung. Berlin: Duncker & Humblot, 1996 (zugl. Dissertation, Universität Trier, 1996).
- TAYLOR, D. A.: Business Engineering with Object Technology. New York et al.: John Wiley & Sons, 1995.
- TEUBNER, R. A.: Organisations- und Informationssystemgestaltung. Wiesbaden: DUV, 1999 (zugl. Dissertation, Westf. Universität Münster, 1997).
- THOM, N.: Betriebliches Vorschlagswesen: ein Instrument der Betriebsführung und des Verbesserungsmanagements. 5. Auflage, Berlin et al.: Lang, 1996.
- ULRICH, P.: Systemsteuerung und Kulturentwicklung. In: Die Unternehmung 1984, Nr. 4, S. 303–325.
- WALL, F.: Organisation und betriebliche Informationssysteme – Elemente einer Konstruktionslehre. Wiesbaden: Gabler, 1996 (zugl. Habilitation, Universität Hamburg, 1996).
- WEIBER, R. (HRSG.): Handbuch Electronic Business. 2. Auflage, Wiesbaden: Gabler, 2002a.
- WILKENS, L.: Application Integration – Strategies and Technologies. Butler Group, United Kingdom, 1999.