

Joachim Schelp, Robert Winter

Enterprise Portals und Enterprise Application Integration

Unter den Stichworten „Enterprise Portals“ und „Enterprise Application Integration“ werden neue Herausforderungen an die Applikationsarchitektur der Unternehmungen gestellt. Bei beiden spielt die Integration vorhandener und das Einfügen neuer Applikationen eine große Rolle. Beiden Themen ist gemein, dass sie die Diskussion bestimmter technischer Konzepte aus den vergangenen Jahren fortsetzen: Portale stellen das moderne Frontend dar, über das die verschiedenen e-Business-Konzepte umgesetzt werden. Mit EAI-Systemen werden die Middlewarekonzepte der vergangenen Jahre fortentwickelt und eine Integrationsarchitektur für die Applikationslandschaft der Unternehmen aufgebaut. Dabei stellt eine solche Integrationsarchitektur eine ideale Ausgangsbasis zum Aufbau von Portalen jeder Art dar. Denn mit Portalen kann die Integration der Applikationen über die Grenzen des einzelnen Unternehmens hinaus fortgesetzt, die Unternehmens-IT gezielt und punktuell nach außen geöffnet werden.

Inhaltsübersicht

1	Einführung	2
1.1	<i>Begriffsbestimmung Portal, Begriffsbestimmung EAI</i>	2
1.2	<i>Portale und EAI – komplementäre Integrationskonzeptionen</i>	4
1.3	<i>Business Case „EAI“</i>	4
2	Portale – Öffnung der Systemlandschaft	5
2.1	<i>Erscheinungsformen</i>	5
2.2	<i>Potentiale, Vor- und Nachteile</i>	6
2.3	<i>Anwendungsbeispiele</i>	7
3	Enterprise Application Integration – Die Systemlandschaft integrieren	8
3.1	<i>Erscheinungsformen: Middleware von RPCs zu Web Services</i>	8
3.2	<i>Potentiale, Vor- und Nachteile</i>	12
3.3	<i>Anwendungsbeispiele</i>	13
4	Marktübersicht Portale	14
5	Zusammenfassung, Ausblick und Trends	16
6	Literatur	17

1 Einführung

Integrierte Prozessketten auch über Unternehmensgrenzen hinweg erfordern einen zunehmenden Datenaustausch zwischen Unternehmen. Dies setzt voraus, dass Unternehmen ihre betriebliche Informationsverarbeitung nach außen öffnen. Für ein integriertes Supply Chain Management oder andere Formen der B2B-Kommunikation sind nicht selten Daten aus unterschiedlichen betrieblichen Applikationen erforderlich, beispielsweise aus einem Warenwirtschaftssystem, einem Vertriebssystem oder einem Finanzbuchhaltungssystem. Statt diese Systeme einzeln mit externen Schnittstellen zu versehen, werden integrierte Schnittstellen nach „außen“ in Form von Portalen aufgebaut. Auch für die Unterstützung der internen Prozesse bietet eine Integration der vorhandenen, getrennten Systeme Nutzenpotentiale.

In dieser Einleitung werden kurz die Begriffe Enterprise Portals und Enterprise Application Integration (EAI) eingeführt, hinter denen sich die entsprechenden Technologien zur zwischen- und innerbetrieblichen Integration vorhandener Applikationen erschließen. Weiterhin wird kurz dargelegt, dass diese Technologien komplementär sind und ein kurz gefasster Business Case für die Synergiepotentiale aufgezeigt. In den nachfolgenden Hauptabschnitten werden beide Bereiche näher dargestellt und anschließend ausgeführt, welche Potentiale sich einem Unternehmen bei Einsatz von EAI-Systemen und darauf basierender Portale bieten. Mit einem kurzen Ausblick endet dieser Beitrag.

1.1 Begriffsbestimmung Portal, Begriffsbestimmung EAI

Die im Laufe der Jahre evolutionär entwickelte Applikationslandschaft einer Unternehmung ist typischerweise dadurch geprägt, dass sich die Applikationen „vertikal“ an den Funktionalbereichen orientieren (vgl. den linken Teil der Abbildung 1). So finden sich häufig jeweils dedizierte Informationssysteme für die Finanzbuchhaltung, die Materialwirtschaft, die Auftragsabwicklung etc. Oftmals ist dabei auch die Datenhaltung getrennt und führt im Extremfall dazu, dass in jedem dieser Systeme z.B. Kunden- oder Produktdaten redundant geführt werden. Bei einer voneinander getrennten Weiterverarbeitung der Daten kann es sogar vorkommen, dass beispielsweise Adressveränderungen nicht über alle Applikationen hinweg berücksichtigt werden. Es kann nachteilig sein, wenn die Kundendaten in verschiedenen Datenbanken vorgehalten werden, die Daten eines Kunden vom Unternehmen aber nicht diesem eindeutig zugeordnet werden können. Beispielsweise können Cross-Selling-Potentiale ungenutzt bleiben oder ein Kunde kann aus verschiedenen Applikationen heraus, jeweils auf der Basis unterschiedlicher Fragmente seiner Kundenhistorie, inkonsistent angesprochen werden.

Seit Beginn der 90er Jahre werden diese Nachteile durch Querschnittssysteme aufgefangen, in denen z.B. Kundendaten integriert verwaltet werden. Aufgrund der stark zunehmenden Bedeutung elektronischer oder telephoniebasierter Zugangskanäle werden diese Querschnittsanwendungen durch kanalspezifische Funktionalitäten aufgewertet. In diesem Kontext sind auch Portale einzuordnen: Es handelt sich um Anwendungstypen, die das Internet als Kommunikations- und insbesondere als Vertriebsmedium erschließen und sich stärker als die traditionellen

Applikationen an den Geschäftsprozessen und den Kunden orientieren. Allgemein definiert ist ein Portal eine Webanwendung, in der Inhalte, Dienste und Funktionen integriert werden [Chan et al. 2002, S. 115]. Diese Webanwendungen können zudem benutzerspezifisch angepasst („personalisiert“) werden, so dass sich z.B. ein Geschäftspartner einen individuellen Zugang zum Internetangebot eines Unternehmens schaffen kann.

In Abbildung 1 sind neuere Anwendungssysteme, wie sie beispielsweise mit den Portalen für das e-Business entstanden sind, rechts als horizontale Applikationen aufgeführt, da sie Absatzkanäle repräsentieren, die nicht entlang der klassischen Abteilungsgrenzen aufgebaut sind. In der unternehmensübergreifenden Integration bilden diese Applikationen die Schnittstelle zu den Applikationen anderer Unternehmen, die dann spiegelbildlich aufgebaut sind.

Enterprise Portals stellen diese und weitere Funktionen den Mitarbeitern innerhalb eines Unternehmens integriert und personalisiert zur Verfügung. Sie brechen ebenso die Abteilungsgrenzen zwischen den vertikalen Applikationen auf wie die nach außen gerichteten Portale. Und ebenso wie diese basieren sie auf der gleichen Voraussetzung: der Integration der vertikalen Applikationen, die über zusätzliche Schichten erfolgen kann.

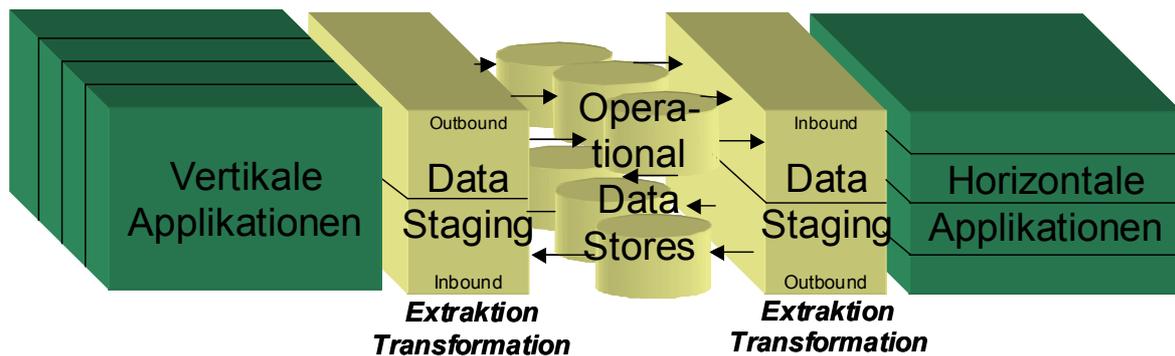


Abbildung 1: Datenorientierte Integration verschiedener Applikationstypen

Eine Integration verschiedener Applikationen kann in unterschiedlicher Form erfolgen. Beispielsweise auf der Präsentationsebene, indem weiterhin selbständige Applikationen über ein gemeinsames Frontend bedient werden. Sie kann ebenso auf der Backend-Seite erfolgen. In diesem Fall ist es möglich, zwischen der Integration über Daten bzw. Funktionen oder den Austausch von Nachrichten bzw. Ereignissen oder einer Kombination dieser Varianten zu unterscheiden. Diese Integrationsformen sind nicht neu, sondern reichen teilweise bis in die Zeit vor Einführung des Client-/Server-Computings zurück. Werden die dafür eingesetzten Technologien in Form einer zusätzlichen Softwareschicht realisiert, spricht man auch von Middleware. Die in der Literatur anzutreffenden Definitionen zielen überwiegend auf eine Softwareschicht ab, die eine Kommunikation zwischen verschiedenen Applikationen ermöglicht [z.B. Riehm 1997, S. 98; Ruh et al. 2001, S. 2; Linthicum 2001, S. 128]. Eine andere Integrationsform besteht darin, eine neue Applikation zu entwerfen, welche die bestehenden Systeme ersetzt. Diese „koppelnde“ Integration dominiert in vielen

Ansätzen, die in der Wirtschaftsinformatik in der Vergangenheit entwickelt worden sind.

Unter dem Stichwort Enterprise Application Integration (EAI) wird hingegen eine Integrationsform diskutiert, bei der die bestehenden Applikationen weiterhin unabhängig voneinander bleiben. Die Kopplung findet nur lose unter Einsatz von Middleware statt. Im Gegensatz zu den zuvor ebenfalls über Middleware durchgeführten punktuellen Integrationslösungen wird aber eine umfassende Lösung gesucht, welche bestehenden Eigenentwicklungen, standardisierten Softwarekomponenten wie Neuentwicklungen eine gemeinsame Integrationsplattform verschafft [Ruh et al. 2001, S. 2f; Linthicum 2001, S. 18].

1.2 Portale und EAI – komplementäre Integrationskonzeptionen

In welcher Beziehung stehen Enterprise Portals und Enterprise Application Integration zueinander? Enterprise Portals werden zumeist mit neuen Technologien realisiert, die als weitere Komponenten in die bestehende Applikationslandschaft eines Unternehmens eingefügt werden. Da sie sich an den Geschäftsprozessen orientieren, benötigen sie Daten bzw. Funktionen, die über die nach Abteilungen getrennten bestehenden Applikationen verteilt sind. Es handelt sich um Querschnittsanwendungen bzw. „horizontale“ Applikationen, wie sie im rechten Teil der Abbildung 1 angedeutet sind. So benötigt ein typischer Webshop Daten über Kunden und Warenbestände, die nicht selten in dedizierten Vertriebs- und Warenwirtschaftssystemen vorgehalten werden. Erfolgt eine Transaktion, so muss auch die Buchhaltungssoftware entsprechend aktualisiert werden. Diese Verbindungen lassen sich über 1:1-Schnittstellen zwischen den beteiligten Systemen leicht realisieren. Nimmt die Anzahl der miteinander zu koppelnden Applikationen zu, so steigen der Erstellungs- und noch mehr der Wartungsaufwand. Leichter lässt sich diese Integration über einen einheitlichen Bus realisieren, der die Anzahl der Schnittstellen deutlich reduziert. Ein solcher Bus oder Middleware-Layer wird im Rahmen eines Enterprise Application Integration-Projektes geschaffen. Besteht eine derartige Infrastruktur, fällt es leicht, weitere Applikationen einzubinden. Dies betrifft dann sowohl intern wie extern orientierte Applikationen.

1.3 Business Case „EAI“

Eine übergreifende Integrationsschicht aufzubauen ist eine Infrastrukturaufgabe, die in ihrer Wirkung weit über einzelnen Integrationsprojekte hinausgeht. Gerade hierin liegen die Vorteile, die den erhöhten Aufwand gegenüber schneller umsetzbaren Punkt-zu-Punkt-Lösungen bei isoliert voneinander betrachteten Integrationsprojekten rechtfertigen.

- Die Komplexität des Gesamtsystems wird verringert, da die Anzahl der Schnittstellen abnimmt. Im Extremfall reduziert sich die Anzahl der Schnittstellen bei einer Zwei-Wege-Kommunikation von $n*(n-1)$ auf $n*2$, sofern eine zentraler Bus aufgebaut wird.
- Durch die geringere Anzahl an Schnittstellen sinkt neben dem Erstellungs- auch der Wartungsaufwand.
- Vermeidung von Redundanz durch Datenintegration.

- Die Integration der Applikationen bzw. ihrer Funktionen entlang der Geschäftsprozesse ermöglicht auch eine bessere Nutzung vorhandener Daten, z.B. konsolidierter Kundendaten, deren gesamthafte Kenntnis erst den sinnvollen Einsatz eines Customer Relationship Management Systems (CRM) ermöglicht.

Diese Vorteile erschließen sich jedoch erst bei einer über ein einzelnes Projekt hinausgehenden Planung der Infrastruktur. Das entscheidende Argument für eine über Punkt-zu-Punkt-Lösungen hinausgehende Integration ist aber die Verringerung der Komplexität des Gesamtsystems. Externe Anforderungen wie Portallösungen oder die Einbindung in unternehmensübergreifende Prozessketten erleichtern es, die für eine umfassendere Lösung notwendigen Anstrengungen zu rechtfertigen.

2 Portale – Öffnung der Systemlandschaft

Vor der Diskussion ihrer Potentiale werden zunächst die Erscheinungsformen der Portale kurz skizziert. Den Abschluss bilden einige Anwendungsbeispiele.

2.1 Erscheinungsformen

Im Laufe der letzten Jahre hat die Interpretation des Begriffes Portal im Internet-Umfeld eine Veränderung erfahren: Von der Einstiegsseite einer Internet-Suchmaschine zur Unternehmens-Webanwendung, in der Inhalte, Dienste und Funktionen integriert werden [z.B. Dias 2001, S. 273ff.]. Die Spannweite reicht dabei immer noch vom Informationsportal bis hin zu interaktiven Anwendungen, von Kommunikationsplattformen gegenüber Kunden bzw. Geschäftspartnern bis zu den eigenen Angestellten.

Allen Varianten ist gemein, dass über ein vereinheitlichtes, üblicherweise Webbrowser-basiertes Frontend, ein Zugang zu unterschiedlichen Backend-Systemen geschaffen wird. Gegenüber unternehmensexternen Anwendern kann man so von der Öffnung der Systemlandschaft sprechen.

Eingesetzt werden diese Lösungen in verschiedenen Formen des e-Business. Bezogen auf die zu diskutierenden Enterprise-Portale spielen die drei Varianten B2B, B2C und B2E eine zentrale Rolle: *Business to business* (B2B) im Geschäftsverkehr zwischen Unternehmen, *business to customer* (B2C) im Umgang mit dem Kunden sowie *business to employee* (B2E) in Richtung der eigenen Angestellten.

Der Begriff *Enterprise Portals* wird nun unterschiedlich weit interpretiert. In der engeren Form (Enterprise Portal i.e.S.) wird darunter nur die persönliche Arbeitsumgebung verstanden, die dem Angestellten als Webangebot alle Funktionalitäten bietet, die er für seine Aufgabenerfüllung benötigt [Österle 2000, S. 40]. Diese enge Eingrenzung deckt sich mit dem andernorts verwendeten Begriff *Enterprise Information Portal*, wie er beispielsweise bei [Alt et al. 2002a, S. 97;] zu finden ist.

Am anderen Ende des Spektrums wird das Enterprise Portal synonym zu e-Business Portalen verwendet, wie beispielsweise in [Chan et al. 2002, S. 114]. In dieser weiten Interpretation finden sich sowohl Portale zwischen Unternehmen wie zwischen Unternehmen und Endkunden, sowohl auf der Käufer- wie auf der Verkäuferseite. Wenngleich meist unter Marktplätzen subsumiert, fallen auch diese unter den

Portalbegriff [Merz 2002, S. 77]. Diese weite Interpretation wird im folgenden unter Enterprise Portal i.w.S. angesprochen.

Die Funktionalität eines Enterprise Portals kann eine weite Spanne abdecken. Ausgehend von der ursprünglichen Interpretation bilden auch heute Suchfunktionen eine Basisfunktionalität für ein Portal. Funktionen können darüber hinaus sein [z.B. Dias 2001, S. 276ff.; Eckerson 1999, S. 11ff.]:

- Personalisierung, d.h. es ist möglich, Benutzerprofile zu erstellen um beispielsweise häufig genutzte Funktionen komfortabel anzubieten und diese benutzerspezifischen Parameter persistent zu speichern.
- Flexible Suchfunktionen.
- Synchron und asynchrone Kollaboration.
- Zusammenführen interner und externer Informationen über eine Weboberfläche. Die internen Informationen können dabei sowohl aus den administrativen wie den dispositiven Systemen stammen.
- Eine einheitliche Benutzerschnittstelle für verschiedene Applikationen bilden, so dass geschäftsprozessbezogen alle für einen Bearbeiter notwendigen Funktionen leicht erreichbar sind.
- Content-Management-Funktionen, die sowohl im Sinne eines Knowledge-Management-Systems wie eines Web-Administrationstools verstanden werden können.

Diese Funktionen können sowohl in Enterprise Portalen im weiten wie im engeren Sinne Anwendung finden. Der tatsächliche Funktionsumfang der in der Realität anzutreffenden Portallösungen ist jedoch sehr unterschiedlich, da diese die angesprochene technische Entwicklung (vom Suchportal bis zur Unterstützung des Customer Resource Life Cycle) in ihren verschiedenen Ausprägungen widerspiegeln. Auch ergibt sich aus der unterschiedlichen Produkthistorie der Software für den Aufbau von Portallösungen ein unterschiedlich weit gefasstes Funktionsspektrum (vgl. auch die anderen Beiträge in diesem Heft). Dies kann soweit reichen, dass die Grenze zwischen Portalen und Backend-Systemen unschärfer wird. Dabei kann es sich in der Praxis sowohl um eine Integration der Backend-Systeme in ein Portal wie um die Erweiterung der Backend-Systeme um Portalfunktionen handeln. Ein Beispiel ist die zunehmende Integration dedizierter Suchmaschinen nicht nur in Portallösungen für externe Nutzer, sondern auch in bestehende unternehmensinterne Applikationsstrukturen.

2.2 Potentiale, Vor- und Nachteile

Sowohl Enterprise Portale im engeren wie weiteren Sinnen bieten eine Reihe eindeutiger Nutzenpotentiale [Alt/Fleisch 2002a, S. 73; Alt et al. 2002a, S. 95], von denen zwei exemplarisch genannt seien:

- Bearbeitungszeiten und -kosten können durch die Orientierung an den Geschäftsprozessen und den integrierten Zugriff auf verschiedene Backend-Systeme über nur ein Portal verringert, die Prozesseffizienz kann verbessert werden.

- Die Flexibilität wird erhöht, da über ein Portal z.B. schnell wechselnde Anforderungen an elektronische Produktkataloge bedient werden können.

Enterprise Portale im weiteren Sinne bieten insbesondere die Chance,

- die Pre- und After-sale-Aktivitäten gegenüber dem Kunden zu verbessern und den Customer Resource Life Cycle durchgehend zu unterstützen und
- die überbetriebliche Kooperation mit Zulieferern und Kunden im Rahmen des Supply Chain Managements zu vertiefen [Alt et al. 2002a, S. 93ff.].

Enterprise Portale i.e.S. bieten, bezogen auf die unternehmensinterne Sicht, aus den genannten Gründen u.a. folgende Potentiale

- Integrierter Zugriff auf alle Funktionen, die zur Bearbeitung der Aufgabenstellung eines Mitarbeiters erforderlich ist.
- Verbesserung der Teamarbeit durch eine geschäftsprozessorientierte Aufgabenverteilung und Funktionen zur Kommunikationsunterstützung.
- Geringerer Schulungsaufwand für die Mitarbeiter, da diese über den Web-Browser auf das ihnen bekannte Portal auf die Backend-Systeme zugreifen, ohne sich mit deren Benutzungskonzepten direkt auseinandersetzen zu müssen. Bei gelungener Integration in das Enterprise Portal i.e.S. kann sogar ein Austausch eines Backend-Systems für den Benutzer unmerklich erfolgen.

Mit Portalen im engeren wie weiteren Sinne sind jedoch nicht nur Vorteile verbunden, sondern auch Risiken bzw. Nachteile. So können Kunden den Geschäftspartner leicht wechseln, wenn die angebotenen Leistungen standardisiert erscheinen [Alt et al. 2002a, S. 81]. Um dem entgegenzuwirken, sind eine konstante Pflege der Webangebote, eine Ausweitung der Leistungen sowie das Einbinden von Partnern erforderlich, um sich gegenüber dem Wettbewerb abzuheben. Insbesondere aber sind schnelle Reaktionen erforderlich, um auf Massenmärkten schnell ein hohes Aufmerksamkeitsprofil zu gewinnen, eine kritische Masse an Kundenkontakten zu erreichen und zu erhalten [Alt/Fleisch 2002b, S. 323].

Mit der Anzahl der einzubindenden Backend-Systeme steigt auch die Komplexität der Portallösung selber. Hier bietet es sich daher an, die Synergiepotentiale zu nutzen, die sich durch eine EAI-Lösung ergeben können, da dies zu einer Verringerung der zu beachtenden Schnittstellen beiträgt (vgl. auch den nächsten Abschnitt). Eine Herausforderung stellt es aber dar, Änderungen in der Struktur der eingebundenen Backend-Systeme vorzunehmen, wenn dies von dem Benutzer unbemerkt erfolgen soll. Voraussetzung ist, dass derartige Veränderungen ohne Änderungen der Geschäftsprozesse vollzogen werden können. Jedoch ist dies umso schwieriger zu erreichen, als mit der zunehmenden Verwendung standardisierter Lösungen (z.B. ERP-Systeme wie SAP R/3) die bestehenden Geschäftsprozesse durchaus tangiert werden können.

2.3 Anwendungsbeispiele

Enterprise Portale i.w.S. zeigen heute schon, dass sich neue Vertriebsstrukturen schnell etablieren und – je nach Branche – die bestehenden Strukturen erheblich verändern bzw. deren Umfang schmälern werden. Ein Beispiel hierfür ist die Pharmabranche, in der sich aus Healthcare-Portalen eine Bedrohung des

traditionellen Vertriebs über Pharmareferenten ergeben kann [Puschmann et al. 2002a, S. 161]. Eine solche Entwicklung könnte den Umfang der bestehenden traditionellen Vertriebsstrukturen verringern und so die Kostenstrukturen erheblich verändern. Weiterhin kann sich durch die automatische Datenweitergabe auch die Datenqualität und damit die Effizienz der nachgelagerten Prozesse erhöhen, was positive Auswirkungen sowohl auf der Kosten- wie auf der Erlösseite bewirken kann.

Bei Enterprise Portalen i.e.S. können sich Effizienzgewinne ergeben sowie Einsparpotentiale bei Softwareschulungen, wenn in Umfeldern mit einer Vielzahl von Softwaresystemen die Anwender nur noch über ein standardisiertes Portal auf diese Anwendungen zugreifen. Aufgrund der Vielzahl der in Unternehmen anzutreffenden Einzelsysteme stellt die Integration aller zur Unterstützung der Geschäftsprozesse relevanten Systeme eine Herausforderung dar. In [Huber et al. 2002] findet sich das Beispielszenario mit 22 Anwendungssystemen, welche über noch 8 verschiedene Applikationstypen bedient werden. Eine Integration dieser Systeme steht allerdings noch aus [Huber et al. 2002, S. 178ff.].

In der Literatur finden sich mittlerweile Erfahrungsberichte realisierter Portalprojekte, die von Projektbeschreibungen mit Schilderung der realisierten Nutzeneffekte [Alt et al. 2002b] bis zu Beispielrechnungen z.B. in Form einer Kosten-/Nutzenanalyse der B2B Kommunikation [Merz 2002] reichen.

3 Enterprise Application Integration - Die Systemlandschaft integrieren

Systeme und Konzepte zur Integration bestehender wie neuer Applikationen gibt es schon länger. Die heute unter dem Begriff „Enterprise Application Integration“ diskutierten Systeme und Integrationskonzepte stellen ihre Weiterentwicklung dar. Viele EAI-Produkte bestehen aus einer Reihe von Komponenten bzw. Modulen. Diese reflektieren ihrerseits die unterschiedlichen, in den Jahrzehnten zuvor entwickelten Integrationstechniken. Der nachfolgende Abschnitt greift daher zunächst diese grundlegenden Techniken bzw. technischen Konzepte auf und ordnet sie dann in die aktuellen EAI-Konzepte ein. Im Anschluss daran findet sich eine Diskussion der Potentiale dieser Konzepte. Zum Abschluss wird dann auf einige Anwendungsbeispiele eingegangen.

3.1 Erscheinungsformen: Middleware von RPCs zu Web Services

Wie in der Einführung dargestellt, reicht die Spannweite der Integration von der Verschmelzung in einem (neuen) Anwendungssystem bis zur losen Kopplung bestehender und getrennt bleibender Systeme. Neu entwickelte Anwendungssysteme, die oft als Client-Server-Systeme aufgebaut werden, greifen dabei zum Teil auf die gleichen Technologien zurück, mit denen eine solche lose Kopplung realisiert wird. Sofern diese Technologien in Form zusätzlicher Softwareschichten verwendet werden, handelt es sich um Middleware. Die Abgrenzung der unterschiedlichen Konzepte hierzu ist in der Literatur nicht einheitlich. Im folgenden werden daher eine Reihe grundlegender Konzepte kurz dargelegt, die sich in den unterschiedlichsten Ansätzen wiederfinden lassen.

Mit *Remote Procedure Calls* (RPC) können Funktionen aus anderen Programmsystemen heraus aufgerufen werden. Der Unterschied zum Aufruf einer Funktion, die dynamisch gegen eine lokale Programmbibliothek gelinkt ist, besteht darin, dass dieser Aufruf auch über ein Netzwerk erfolgen kann. Diese Technologie bildet die Grundlage aller Client-Server-Systeme. Von Nachteil synchroner RPC-Systeme ist, dass das Client-Programm während des Prozeduraufrufs solange blockiert ist, bis das Server-Programm geantwortet hat [Britton 2001, S. 26]. Da die Definition des Funktionsaufrufs während der Programmerstellung fest in den Programmcode eingefügt wird, erfordert eine Aktualisierung einer Komponente oftmals die Neuerstellung aller abhängigen Systeme. Bei einer größeren Anzahl von Applikationen steigt der Wartungsaufwand entsprechend, so dass diese Technologie bei alleiniger Verwendung für komplexe Systeme mit Nachteilen verbunden ist. Darüber hinaus müssen Mechanismen zur Gewährleistung von Sicherheit, Performance, Verteilung etc. explizit vom Anwendungsprogrammierer berücksichtigt werden. Dennoch stellen RPCs ein wichtiges Konzept dar und bilden eine der Grundlagen der im weiteren aufgeführten Systeme [vgl. z.B. auch Riehm 1997, S. 144].

Der *Remote Data Access* ermöglicht über eine (standardisierte) Abfragesprache den Zugriff auf eine unabhängig vom jeweiligen Anwendungssystem geführte Datenhaltung [Britton 2001, S. 27]. Die bekannteste Form des Remote Data Access ist die von Microsoft definierte ODBC-Schnittstelle, mit der auf die gebräuchlichen Datenbanksysteme zugegriffen werden kann. Durch die Definition zusätzlicher Softwareschnittstellen reduziert sich der im Vergleich zu RPCs erforderliche Wartungsaufwand bei komplexeren Systemen. Von Nachteil ist jedoch, dass hier nur die Datenintegration angesprochen wird. Zwar kann in vielen Datenbanksystemen z.B. in Form von Stored Procedures oder ähnlichen Verfahren auch eine entfernte Verarbeitungslogik ausgelöst werden, doch sind dies zumeist proprietäre Lösungen, die eine Migration solcher Systeme erschweren.

Das *Distributed Transaction Processing* zielt ähnlich den Remote Procedure Calls auf den Zugriff auf andere Programmsysteme über ein Netzwerk ab. Der Unterschied zu RPC liegt darin, dass vollständige Transaktionen und nicht mehr nur einzelne Funktionen betrachtet werden. Wichtig ist dabei das Einhalten der aus dem Datenbankbereich bekannten ACID-Kriterien, die hier auf Transaktionen in verteilten Anwendungsumgebungen angewendet werden und entsprechend komplexe Anforderungen stellen [Britton 2001, S. 30ff.]. Die ACID-Kriterien ebenso wie Mechanismen zur Gewährleistung von Sicherheit, Verteilung etc. werden mit diesen gegenüber den RPC erweiterten Ansätzen adressiert. Das Distributed Transaction Processing kam in den 80er Jahren im Zusammenhang mit der Client-Server-Diskussion als Standardisierungsvorschlag der X/Open-Group auf und reicht in seinen Wurzeln in die Transaktionsmonitore früher Host-Systeme zurück.

Die *Message Oriented Middleware* (MOM) basiert auf dem Austausch von Nachrichten und ermöglicht sowohl eine synchrone Kommunikation wie bei den RPCs als auch eine asynchrone gesicherte Kommunikation. Für letztere werden sogenannte Message Queues aufgebaut. Durch das Zwischenspeichern in Warteschlangen kann die Verarbeitung auch dann fortgesetzt werden, wenn das zum Nachrichtentransport verwendete Netzwerk zwischenzeitlich ausgefallen ist. Die Anforderungen an die

Transaktionssicherheit im Sinne der ACID-Kriterien sind jedoch höher, da durch die asynchrone Verarbeitung ein höherer Koordinationsaufwand erforderlich ist [Riehm et al. 1996, S. 78ff.] Erweiterungen finden solche Systeme durch Publish- und Subscribe-Mechanismen, mit denen die Nachrichtenübertragung auf Gruppen ausgedehnt werden kann. Dabei wird geregelt, ob die einzelnen Akteure als Abonnenten oder als Herausgeber auftreten [Riehm 1997, S. 147f.]. Die aktuellen Systeme gehen darüber hinaus, indem sie in Form von Message Brokern auch Funktionen zur Nachrichtentransformation, Routing, Anwendung von Regeln, zum Vorhalten von Nachrichten, etc. bereitstellen und so eine eigene Anwendungsschicht definieren. Diese wird gelegentlich auch als „middleware of middleware“ [Linthicum 2001, S. 233] bezeichnet.

Aus der Diskussion objektorientierter Konzepte sind auch Ansätze für eine *Object Middleware* entstanden. Stellvertretend für verschiedene am Markt vorhandene Konzepte wird hier nur die aus einem Standardisierungsvorschlag entstandene Common Object Request Broker Architecture (CORBA) angesprochen, für die sich Implementierungen unterschiedlicher Hersteller finden lassen. Wie bei RPCs und dem Distributed Transaction Processing geht es im Kern der CORBA-Implementierungen um den Aufruf (entfernter) Anwendungen. In der objektorientierten Terminologie rufen Clients Funktionen über Objekte auf. Die zusätzlichen Schichten, die in dieser Architektur definiert werden, stellen dabei Dienste u.a. zur Koordination der Transaktionen, Ereignisse, Sicherheit der Übertragung und zur Bekanntmachung der Dienste zur Verfügung. Der gegenüber anderen Ansätzen zusätzlich erforderliche Koordinations- und Kommunikationsbedarf muss aber nicht zwangsläufig zur Verlangsamung führen. Beispielsweise können die Object Request Broker als zwischen Client und Objekt geschaltete Instanzen häufig aufgerufene Objekte im Hauptspeicher vorhalten und nicht erst bei jedem Neuaufruf neu laden.

Objektorientierte Konzepte sowie die Berücksichtigung der Internetstandards standen Pate bei der Konzeption komponentenorientierter Middleware (*Component Middleware*). Zur Zeit stehen hier zwei Architekturvarianten verschiedener Herstellergruppen einander gegenüber: Auf der einen Seite die von verschiedenen Herstellern implementierten Enterprise Java Beans (EJB) und auf der anderen Seite das von Microsoft definierte Distributed Component Object Model (DCOM/COM+). Bezogen auf die objektorientierten Middlewarekonzepte üben sie die gleiche Funktion aus wie die Transaktionskonzepte für die vorhergehenden Ansätze [Britton 2001, S. 67]. Durch eine zusätzliche Abstraktionsebene wird die Entwicklung flexibel einsetzbarer Komponenten deutlich erleichtert. Beide Ansätze sind jeweils in übergreifende Architekturen (J2EE bzw. .NET) eingebettet, mit denen umfassende, verteilte Applikationen entwickelt werden können.

Internetdienste stehen im Mittelpunkt der *Web Services*. In der Literatur ist dieser jüngste Begriff längst nicht einheitlich definiert. Im Kern handelt es sich jedoch um einen von einem Webserver dynamisch generierten Datenstrom, der in einem XML-Format (eXtensible Markup Language) vorliegt und von dedizierten Client-Anwendungen aufgerufen werden kann [Schulz 2002, S. 236]. Die zugrundeliegenden Mechanismen sind die gleichen wie bei normalen Webseiten (Web-Server und Browser), jedoch werden die bereitgestellten Funktionalitäten in

einheitlich strukturierter Form (Simple Object Access Protocol, SOAP) über standardisierte Beschreibungen (Web Service Description Language, WSDL) und Repositories (Universal Description, Discovery and Integration, UDDI) gegenüber den Clients publik gemacht. Die unterschiedliche Semantik verschiedener Geschäftsdokumente soll dabei durch eine standardisierte Beschreibung der möglichen Inhalte (Electronic Business XML, ebXML) erschlossen werden [Kotok et al. 2002, S. 48, S. 195ff.]. Da zur Zeit in vielen Branchen Anstrengungen unternommen werden, einen Nachfolger des EDI-Standards (Electronic Data Interchange) auf XML-Basis zu schaffen, konkurrieren zur Zeit eine Reihe von Branchenstandards miteinander, die von unterschiedlichen Herstellern getragen werden. Prinzipiell lassen sich die für Web Services verwendeten Technologien auch zur unternehmensinternen Integration vorhandener Applikationen verwenden. Die Internetorientierung der Web Services spricht eher für ihre Verwendung in der Kommunikation mit unternehmensexternen Partnern. Dennoch stellt sich die Überlegung, ob „außen“ verwendete Standards nicht auch in der unternehmensinternen Kommunikation verwendet werden können. Die dabei entfallende Transformation vom unternehmensinternen zum -externen Standard trägt sicherlich zu einer Komplexitätsreduktion bei.

Auf diesen Konzepten basieren die Softwarekomponenten bzw. -systeme der verschiedenen EAI-Lösungen, die von den unterschiedlichen Herstellern zur Zeit am Markt angeboten werden. Im Vordergrund der aktuellen Diskussion stehen häufig Messaging-Konzepte sowie insbesondere die Web-Services. Werden im Rahmen eines Integrationsprojektes zusätzlich Data Warehouse-Daten in die operativen Systeme zurückgeführt - beispielsweise um Analyseergebnisse in Customer Relationship Management-Systemen (CRM) zu verwenden - bildet die Datenintegration über separate Datenspeicher einen weiteren Lösungsansatz. Eine solche redundante Datenhaltung für operative Zwecke wird im Zusammenhang mit Data Warehouse-Systemen schon länger unter dem Begriff Operational Data Stores diskutiert. Die im Rahmen der EAI-Lösung verwendeten Softwarekomponenten bzw. -systeme bilden einen weiteren Bestandteil der aufzubauenden Integrationsarchitektur wie die zu integrierenden operativen Systeme und ggf. analytischen Systeme.

Neben der funktional orientierten Punkt-zu-Punkt-Integration und den genannten message- oder datenzentrierten Integrationslösungen sind auch gemischte Architekturen denkbar. Beispielsweise ist eine Anbindung einer Portallösung an bestehende transaktionale Applikationen über ein zentrales Messaging-System problematisch, wenn die transaktionsorientierten Systeme schon an ihrer Kapazitätsgrenze arbeiten und nur mit merklichen Verzögerungen Daten zurückliefern. Langsam aufgebaute Webseiten in Portalanwendungen sind für Anwender jedoch eine wenig befriedigende Arbeitsumgebung. In solchen Fällen bietet es sich an, zusätzlich Teile der Daten in einem separaten Datenspeicher redundant zwischenspeichern und lediglich ausgewählte Daten wie z.B. Bestandsinformationen aus den transaktionalen Systemen direkt abzufragen. Kundenprofile hingegen können auch aus einem Zwischenspeicher entnommen werden. Eine so aufgebaute Architektur findet sich auch in Abbildung 2 wieder.

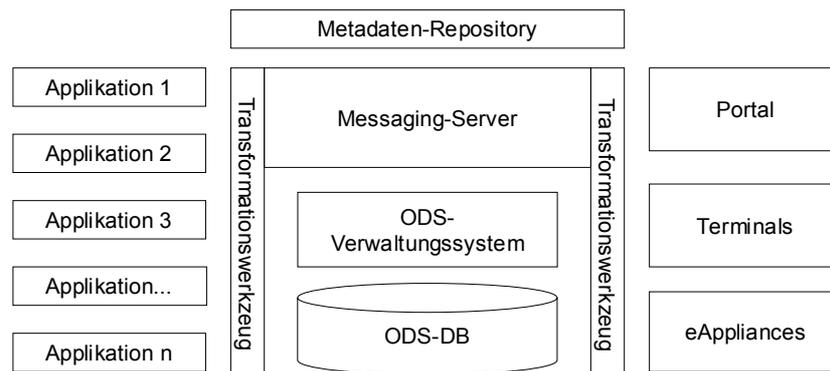


Abbildung 2: Beispiel einer gemischten EAI-Architektur

Die in Abbildung 2 zusätzlich genannten Komponenten sind die Transformationswerkzeuge sowie ein Metadatenrepository. Die Transformationswerkzeuge wandeln die verschiedenen Datenformate der Quell- und Zielsysteme ineinander über. Die Informationen über die Datenquellen und -ziele, die anzuwendenden Transformationsregeln, Datenflussrichtungen etc. werden in dem Metadatenrepository vorgehalten. Beide Systeme übernehmen die gleichen Funktionen wie im Data Warehousing und sind wie dort von zentraler Bedeutung.

In der Praxis finden sich die Komponenten einer EAI-Architektur allerdings nicht so deutlich getrennt wieder wie in Abbildung 2 skizziert. Metadaten finden sich in den unterschiedlichsten Werkzeugen wieder und werden teilweise dezentral, teilweise zentral vorgehalten. Auch unterscheiden sich die Formate der Metadaten voneinander, zum Teil sind gar Transformationswerkzeuge zur Überführung der verschiedenen Metadatenformate ineinander erforderlich (meta data bridges). Auch sind unterschiedliche Transformationswerkzeuge verwendbar, wie auch die Transformation teilweise von den Applikationen selber vorgenommen werden können. Ein Beispiel hierfür stellt das SAP R/3-System dar, das über die Sprache ABAP/4 entsprechend erweitert werden kann.

Schließlich können neben den skizzierten daten- und/oder funktionsorientierten EAI-Ansätzen auch Prozess- oder Workflow-Managementsysteme den Kern einer EAI-Lösung bilden [Kloppmann et al. 2000]. Wie eingangs dargestellt, werden die hier angesprochenen Begriffe weder in der Literatur noch in der Praxis einheitlich verwendet und die Übergänge sind fließend. Gerade die komponentenorientierten Middleware-Ansätze werden fortlaufend erweitert und viele Elemente der Web Services fließen in diese ein. Die Vertreter der ‚traditionellen‘ Systeme wiederum nehmen ihrerseits neuere Entwicklungen wie XML-Standards auf und erweitern ihre Ansätze. Da die Hersteller teilweise die gleichen sind, fließen Elemente aus den transaktions- oder message-orientierten Systemen in ihre jeweils anderen Konzepte ein – und zurück. Der Umfang der einzelnen am Markt vertretenen EAI-Lösungen variiert somit, was eine klare und eindeutige Abgrenzung der verschiedenen Ansätze wie konkreten Produkte bzw. Produktfamilien erschwert.

3.2 Potentiale, Vor- und Nachteile

Das Ziel der Enterprise Application Integration ist die Kopplung bisher voneinander isoliert betriebener betrieblicher Applikationen. Dabei soll die Komplexität der

Kopplung so gering wie möglich gehalten werden. Folglich bietet sich statt der 1:1 Verknüpfung die Schaffung einer Integrationsschicht an. Die aufgezeigten Middleware-Technologien beinhalten durchgehend die Möglichkeit, von den zugrundeliegenden Betriebssystem- und Netzwerkschichten zu abstrahieren und die Komplexität der zu entwickelnden Applikationen zu reduzieren. So lassen sich u.a.

- die Komplexität des Gesamtsystems, insbesondere aber die Schnittstellenkomplexität verringern,
- Wartbarkeit und Portabilität durch Abstraktion von Betriebssystem- und Netzwerkschicht erhöhen,
- durch Prozessorientierung zusätzliche Funktionalität durch Aufheben der Trennlinien zwischen den Applikationen erschließen und zugleich der
- Kundennutzen erhöhen durch Orientierung an den Geschäftsprozessen statt an den abteilungsspezifischen Applikationsgrenzen.

Insbesondere der Komplexitätsverringering kommt eine hohe Bedeutung zu. Das Einsparpotential z.B. bei der Implementierung der Schnittstellen kann im Idealfall der Verringerung der Schnittstellenzahl von $n*(n-1)$ auf $2*n$ entsprechen.

Allerdings können nicht alle Vorteile leicht realisiert bzw. es müssen einige Risiken in Betracht gezogen werden:

- Die Portabilität hängt von den verwendeten Architekturen ab: Nicht alle Hersteller unterstützen jede Plattform oder mehrere Betriebssysteme.
- Die Abstraktion von den „unteren“ Ebenen wird durch zusätzliche Softwareschichten realisiert, die jeweils eigenen Wartungszyklen unterliegen und somit Kompatibilitätsprobleme bergen. Diesem kann mit einem eigenen Release- bzw. Update-Management begegnet werden.
- Die unterschiedlichen Standardisierungsbemühungen sind noch nicht beendet. Welche XML-Dialekte für ein Unternehmen Relevanz haben, wird sich ggf. erst in einigen Jahren herausstellen. Unter Umständen wird sich der für die unternehmensinterne Kommunikation verwendete Standard als inkompatibel mit dem unternehmensextern verwendeten herausstellen.
- Die höhere Komplexität der Gesamtarchitektur vergrößert die Auswirkungen von Störungen. Je komplexer das Gesamtsystem aufgebaut ist, desto höher können die Ausfallzeiten werden.

Die genannten Vor- und Nachteile stellen nur einen Ausschnitt möglicher Beurteilungskriterien konkreter Projekte dar. Sie hängen zu sehr von der Ausgangssituation und den gewählten Technologien in der Zielarchitektur ab, als dass sie auf diesem begrenztem Raum abschließend aufgeführt werden könnten. Daher werden im nachfolgenden Abschnitt drei kurze Anwendungsbeispiele skizziert, die einen Eindruck von der Leistungsfähigkeit einer EAI-Lösung vermitteln.

3.3 Anwendungsbeispiele

Ein einfaches Beispiel ist die Zusammenführung getrennt geführter Adressbestände in einem Unternehmen oder Unternehmensverbund. Über einen gemeinsamen

Datenspeicher in Form eines Operational Data Store kann im einfachsten Fall eine Konsolidierung der Adressbestände herbeigeführt werden. Auf dieser technischen Basis kann dann die Reinigung der Datenbestände erfolgen, wobei die aus dem Data Warehousing bekannten Techniken eingesetzt werden. Die Vorteile sind offensichtlich: Kunden können über das gesamte Unternehmen hinweg einzeln identifiziert werden, eine einheitliche Kundenansprache wird ermöglicht und die grundlegende Basis für ein verbessertes Kundenbeziehungsmanagement wird gelegt.

Etwas komplexer wird das Szenario, wenn diese Vorarbeit durchgeführt wurde und Kundendaten beispielsweise in einem Data Warehouse analysiert werden. Die dabei gewonnenen Erkenntnisse z.B. über das Zahlungsverhalten von Kunden kann in den operativ betriebenen Customer Relationship Management-Systemen (CRM) genutzt werden [z.B. Biermann et al. 2001, S. 247]. Voraussetzung ist dafür die Rückführung der analyseorientierten Daten in die operativen Systeme. Je nach Berechnungsdauer kann es sich anbieten, diese Daten online über Messaging-Tools auszutauschen oder periodisch aktualisiert in einem physischen Zwischenspeicher wie einem ODS vorzuhalten.

Schließlich sei nochmals auf die Schnittstellenzahl hingewiesen, die sich durch Einführung einer EAI-Schicht in der Applikationsarchitektur einer Unternehmung entscheidend verringern kann. Die Einsparungen werden dabei nicht nur bei der Implementierung der geringeren Zahl von Schnittstellen erzielt, sondern auch bei der Wartung. In [Puschmann et al. 2002b] findet sich ein eindrucksvolles Beispiel aus einem realen EAI-Projekt. In der dort skizzierten Applikationsarchitektur waren 27 Softwarekomponenten zu integrieren, so dass sich $27 \cdot (27-1) = 702$ Schnittstellen ergaben. Durch die Einführung eines integrierten Business Bus reduzierte sich diese Zahl auf $2 \cdot 27 = 54$ Schnittstellen. Das in dieser Quelle vorgerechnete Einsparpotential bei der Implementierung betrug ca. zwei Millionen DM, bei der Wartung ca. sechseinhalb Millionen DM p.a. [Puschmann et al. 2002b, S. 296f.].

4 Marktübersicht Portale

Der Markt für Portalapplikationen ist nicht leicht abzugrenzen, da die Trennlinie zu ERP-, EAI- und anderen Systemen nicht scharf gezogen ist. Folglich beinhaltet die nachfolgende Marktübersicht nur einen Ausschnitt dieses dynamischen Marktes und kann keinerlei Anspruch auf Vollständigkeit erheben.

Als Kernprodukt wird Portalsoftware unter anderem von Epicentric (Epicentric Foundation Server), Plumtree (Corporate Portal), SAP Portals/TopTier (SAP Enterprise Portals) und Viador (E-Portal Framework) angeboten. Das in Abschnitt 2 genannte Funktionsspektrum einer Portallösung kann mit diesen Produkten weitestgehend vollständig realisiert werden und geht teilweise sogar darüber hinaus. Innovative Erweiterungen sind tendenziell eher bei diesen Produkten zu finden. Von Nachteil kann dabei sein, dass die Anknüpfung an die Backend-Systeme teilweise über 1:1-Verknüpfungen erfolgen muss [Baldwin 2002].

Portalfunktionalität wird zunehmend aber auch anderen Applikationstypen hinzugefügt und entsprechend dringen Anbieter anderer Applikationstypen in den Portalmarkt ein. Beispiele hierfür sind BEA Systems (WebLogic E-Business Platform), IBM (Websphere Portal Server, Enterprise Information Portal), Microsoft (SharePoint

Portal Server), Oracle (9iAS Portal) oder Sun Microsystems (iPlanet Portal Server), die über eine Erweiterung ihrer EAI-Werkzeuge bzw. ihrer Applikationsserver zu den Portalanbietern gezählt werden können. Dies bedeutet jedoch nicht, dass die für Portale typischen Funktionen mit diesen Produkten nicht realisiert werden können. Der besondere Vorteil dieser Produkte liegt aber in der potentiell besseren Integrationsmöglichkeit an die Backend-Systeme. Bei den aus Applikationsservern oder gar ERP-Systemen heraus entwickelten Portallösungen ist die Integration mit den proprietären Applikationen des gleichen Herstellers leichter, bei den aus EAI-Produkten entwickelten Portallösungen ist die Integration in heterogene Umgebungen potentiell leichter zu realisieren. Umgekehrt fließen mittlerweile entsprechende Funktionalitäten der EAI-Systeme oder der Applikationsserver auch in die zuvor genannten dedizierten Portallösungen ein.

Hersteller	Produktbezeichnung
BEA Systems	WebLogic E-Business Plattform
BroadVision	InfoExchange Portal
Computer Associates	CleverPath Portal
Epicentric	Epicentric Foundation Server
Hummingbird	Hummingbird EIP
IBM	Websphere Portal Server, Enterprise Information Portal
Microsoft	SharePoint Portal Server
Oracle	9iAS Portal
Plumtree	Corporate Portal
SAP Portals	SAP Enterprise Portals
Sun Microsystems	iPlanet Portal Server
Sybase	Sybase Enterprise Portal
Viador	E-Portal Framework

Tabelle 1: Hersteller und Produkte von Portalapplikationen

Die Abgrenzung des Marktes wird dadurch erschwert, dass weitere Produkte wie z.B. Workflow Management-Systeme oder andere Applikationen zur Unterstützung der Kollaboration in ihrem Funktionsumfang um entsprechende Portalelemente erweitert werden und somit weitere Marktteilnehmer zu berücksichtigen sind.

Auf dem Markt für Portal-Software agieren z.Zt. ca. 43 verschiedene Anbieter [Schaller 2001]. Dieser Markt wird gegenwärtig von keinem der Anbieter dominiert. Diese Situation kann sich ändern, sobald die schon stattfindende Konsolidierung ein größeres Ausmaß annimmt. Die Anbieter von ERP-Systemen, Groupware- oder Collaboration-Tools ebenso wie andere Middleware-Anbieter tragen zu dieser Konsolidierung bei, indem kleinere Spezialanbieter sowohl im Portal- wie EAI-Umfeld aufgekauft werden und das jeweilige Produktspektrum abrunden. Als Käufer agierten u.a. SAP (ehemals TopTier Software), SUN (Forte), IBM (Cross Worlds), Sybase (Neon) [Baldwin 2002].

Einen umfassenderen und aktuelleren Überblick über Portalanbieter und ihre Produkte ebenso wie über Software für EAI-Lösungen bieten aufgrund der vorhandenen Marktdynamik freie Informationsportale wie z.B. <http://www.middleware.info/> im Internet.

5 Zusammenfassung, Ausblick und Trends

Portallösungen benötigen eine vorgelagerte Integration der betrieblichen Applikationen, wenn sie Informationen aus zahlreichen Quellen integriert darstellen sollen und die Komplexität des Gesamtsystems beherrschbar sein soll. Die Integration erst innerhalb der Portallösung führt zu einer nur punktuellen Integration mit den vorgelagerten Systemen und ist nur dann sinnvoll, wenn keine weitere Applikation auf die integrierte Sicht zugreifen soll und die Anzahl der zu integrierenden Systeme gering ist. Aufgrund der zahlreichen Potentiale, die sich für die innerbetriebliche Applikationsintegration ergeben, ist der Aufbau einer zusätzlichen Integrationsschicht naheliegend. Geht die Portallösung über ein Enterprise Portal i.e.S., d.h. nur für die eigenen Mitarbeiter, hinaus, so steht mit der expliziten Integrationsschicht eine technische Basis zur Verfügung, die leicht ausgebaut werden kann. So wird es zudem leichter, der Forderung nach einer integrierten Sicht z.B. auf die Kundendaten zu entsprechen, die bei aller Differenzierung der Vertriebskanäle an ein CRM gestellt wird. Moderne Portale, die Web Services für B2B, B2C, B2E etc. anbieten, bauen letztlich auf Middlewarekomponenten auf, die auch für EAI-Lösungen unternehmensintern eingesetzt werden können [Linthicum 2001, S. 20]. Eine EAI-Architektur schafft letztlich Flexibilitätsspielräume, indem potentiell schnell neue Distributionskanäle oder Koordinationsfunktionen mit externen Partnern (z.B. im Supply Chain Management) aufgebaut werden können. Da Portale im weiten Sinne eine große Bandbreite der betrieblichen Prozesse abdecken können und EAI-Technologien die Integration mit den vorhandenen betrieblichen Applikationen erleichtern, werden sich beide Systeme ergänzen und in ihrer Verbreitung tendenziell gegenseitig verstärken. In ihrer Funktionalität werden sie sich dabei weiter annähern [Luce 2002].

Die Anwendungsintegration wird von einer zunehmenden Zahl von Anwenderunternehmen als relevantes Thema erkannt, wenngleich erst wenige Unternehmen sich dem Thema aktiv stellen. Beispielsweise wird in der Fachpresse eine Zahl von 6% der Anwenderunternehmen genannt [Freimark 2002], die zur Zeit EAI-Projekte umsetzen oder dies schon getan haben. Solche Zahlen müssen allerdings dahingehend relativiert werden, dass diese Projekte unter dem Stichwort EAI durchgeführt werden, aber nicht unbedingt alle Integrationsprojekte unter der Rubrik EAI durchgeführt werden. Die in der jüngeren Vergangenheit aufgestellten Marktaussichten scheinen ein gewisses Wachstum erwarten zu lassen [IDC 2001], doch muss abgewartet werden, ob diese Erwartungen auch unter den Bedingungen einer abgekühlten Konjunktur Bestand haben.

Enterprise Application Integration ist aber nicht nur ein aus technischer Sicht zu behandelndes Thema. Ähnlich wie im Data Warehousing werden organisatorische Aspekte beim Aufbau von EAI-Lösungen eine zunehmende Bedeutung erlangen. Die Implementierungen der Hersteller bergen keine Lösung beispielweise für die Fragen

nach der Daten-, Funktions- oder Prozesshoheit, die bei einer Integration der Systeme berührt werden. Auch erfordert der Betrieb einer integrierten Lösung den Aufbau einer entsprechenden Organisation, mit der nicht nur die IT-, sondern auch die Fachabteilungen eines Unternehmens gefordert sind.

6 Literatur

Eine erweiterte Fassung dieses Beitrags mit ausführlichen Literaturangaben erscheint als Arbeitsbericht des Institutes für Wirtschaftsinformatik der Universität St. Gallen im Juli 2002. Aus redaktionellen Gründen konnte hier nur ein Auszug angegeben werden.

[Alt/Fleisch 2002a] Alt, Rainer; Fleisch, Elgar: *Netzwerkfähigkeit von Unternehmen*, in: [Österle et al. 2002], Kap. 4, S. 63-73.

[Alt/Fleisch 2002b] Alt, Rainer; Fleisch, Elgar: *Kritische Erfolgsfaktoren des Business Networking*, in: [Österle et al. 2002], Kap. 16, S. 317-331.

[Alt et al. 2002a] Alt, Rainer; Puschmann, Thomas; Reichmayr, Christian: *Strategien zum Business Networking*, in: [Österle et al. 2002], Kap. 5, S. 77-101.

[Alt et al. 2002b] Alt, Rainer; Reichmayr, Christian; Zurmühlen, Rudolf: *Electronic Commerce und Supply Chain Management bei der Swatch Group*, in: [Österle et al. 2002], Kap. 6, S. 103-117.

[Baldwin 2002] Baldwin, Howard: *How integration needs are driving portals growth*, in: ZDnet / Enterprise, 18. März 2002. URL: <http://techupdate.zdnet.com/>

[Biermann et al. 2001] Biermann, Andreas; Bochow, Holger; Janning, Thorsten: *Perspektiven des Kundenbeziehungsmanagements über elektronische Medien*, in: Buhl, Hans Ulrich; Kreyer, Nina; Steck, Werner (Hrsg.): *e-Finance – Innovative Problemlösungen für Informationssysteme in der Finanzwirtschaft*, Berlin et al. 2001, S. 235-253.

[Britton 2001] Britton, Chris: *IT Architecture and Middleware – Strategies for Building Large, Integrated Systems*, Upper Saddle River 2001.

[Chan et al. 2002] Chan, Michael F. S. ; Chung, Walter W. C. : *A framework to develop an enterprise information portal for contract manufacturing*, in: International Journal of Production Economics 75 (2002) 1-2, S. 113-126.

[Dias 2001] Dias, Cláudia: *Corporate portals: a literature review of a new concept in Information Management*, in: International Journal of Information Management 21 (2001) 4, S. 269-287.

[Eckerson 1999] Eckerson, Wayne W.: *Business Portals – Drivers, Definitions, and Rules*, White Paper WP02-499, Patricia Seybold Group, Boston, MA 1999.

[Freimark 2002] Freimark, Alexander: *Die EAI-Karten werden neu gemischt*, in: Computerwoche Online, 8. April 2002. URL: <http://www.computerwoche.de/>

[Huber et al. 2002] Huber, Thomas; Alt, Rainer; Barak, Vladimir; Österle, Hubert: *Entwurf einer Applikationsarchitektur für die Pharmaindustrie*, in: [Österle et al. 2002], Kap. 9, S. 165-183.

[IDC 2001] International Data Corporation(IDC): *The Enterprise Application Integration Market Simmers with Robust Growth Expectations, According to IDC*, Framingham MA 2001. URL: <http://www.idc.com/>

[Kloppmann et al. 2000] Kloppmann, Matthias; Leymann, Frank; Roller, Dieter: *Enterprise Application Integration mit Workflow Management*, in: HMD 213/2000, S. 23-30.

[Kotok et al. 2002] Kotok, Alan; Webber, David R. R.: *ebXML – The New Global Standard For Doing Business Over The Net*, Indianapolis 2002.

[Linthicum 2001] Linthicum, David S.: *B2B Application Integration – e-Business–Enable Your Enterprise*, Boston et al. 2001.

[Luce 2002] Luce, Charles: *Three trends in enterprise information portals*, in: ZDnet / Enterprise, 19. März 2002. URL: <http://www.zdnet.com/>

[Merz 2002] Merz, Michael: *Eine Kosten-/Nutzenanalyse der standardisierten B2B-Kommunikation am Beispiel des papiNet-Projekts*, in: HMD – Praxis der Wirtschaftsinformatik 39 (2002) 223, Februar 2002, S. 76-87.

[Meyerson 2001] Meyerson, Judith M.: *Web Services Architectures – How they stack up*, <http://www.webservicesarchitect.com/content/articles/myerson01.asp>, 23. Januar 2002, Download vom 18. Februar 2002.

[Österle et al. 2002] Österle, Hubert ; Fleisch, Elgar ; Alt, Rainer (Hrsg.): *Business Networking in der Praxis – Beispiele und Strategien zur Vernetzung mit Kunden und Lieferanten*, Berlin et al. 2002

[Österle 2000] Österle, Hubert: *Geschäftsmodell des Informationszeitalters*, in: Österle, Hubert; Winter, Robert (Hrsg.): *Business Engineering – Auf dem Weg zum Unternehmen des Informationszeitalters*, Berlin et al. 2000, S. 21-42.

[Puschmann et al. 2002a] Puschmann, Thomas; Alt, Rainer; Huber, Thomas; Barak, Vladimir: *Customer Relationship Management in der Pharmaindustrie*, in: [Österle et al. 2002], Kap. 8, S. 139-162.

[Puschmann et al. 2002b] Puschmann, Thomas; Alt, Rainer; Sassmannshausen, Dirk: *Enterprise Application Integration bei Robert Bosch*, in: [Österle et al. 2002], Kap. 14, S. 271-298.

[Riehm et al. 1996] Riehm, Rainer ; Vogler, Petra : *Middleware: Infrastruktur für die Integration*, in : Österle, Hubert ; Riehm, Rainer ; Vogler, Petra: *Middleware – Grundlagen, Produkte und Anwendungsbeispiele für die Integration heterogener Welten*, Braunschweig et al. 1996, S. 25-135.

[Riehm 1997] Riehm, Rainer: *Integration von heterogenen Applikationen*, Universität St. Gallen, Dissertation, Bamberg 1997.

[Ruh et al. 2001] Ruh, William A. ; Maginnis, Francis X.; Brown, William J.: *Enterprise Application Integration*, New York et al. 2001

[Schaller 2001] Schaller, Katrin: *Der Markt für Portal-Software*, in: e-business.de, 19. September 2001, URL: <http://www.e-business.de/>

[Schulz 2002] Schulz, Hajo: *Weltweites Werkeln – Was bringen Web Services für Anwender und Entwickler*, in: c't 6/02, S. 236-241.