

Typische Integrationsszenarien und deren Unterstützung durch Web Services und andere Technologien

Cornelia Boles, Jörg Friebe, Till Luhmann

OSC – IM Systems AG
Industriestr. 11
26121 Oldenburg

cornelia.boles@osc-ims.de, joerg.friebe@osc-ims.de, till.luhmann@osc-ims.de

Abstract. Die Integration existierender und neu einzuführender Anwendungen ist heute eine Kernaufgabe im Bereich der IT. Um für das jeweilige Integrations-szenario die geeignete Integrationstechnologie auswählen zu können, müssen die Integrationsszenarien und die daraus resultierenden Anforderungen an die Integrationstechnologien genauer untersucht werden. Der vorliegende Artikel beschreibt nach einer einführenden Betrachtung der möglichen Integrations-ebenen die charakteristischen Merkmale von Integrationsszenarien. Im Anschluss daran werden die wichtigsten Integrationstechnologien kurz vorgestellt und bezüglich ihrer Eignung für die einzelnen Merkmalsausprägungen der Integrations-szenarien bewertet. Der Artikel schließt mit einer Zusammenfassung und einem Ausblick.

1 Einleitung

Das Thema Web Services wird derzeit in vielen Publikationen und auf großen Konferenzen behandelt. Doch Web Services sind längst kein reines Forschungsthema mehr. Vielmehr werden sie vermehrt in Unternehmen eingesetzt, in erster Linie zur Integration von Anwendungen. Die Vielzahl der im Umfeld von Web Services entstehenden Standards (z.B. zu den Themen Sicherheit, Transaktionen oder Web Services Choreographien) zeigt, dass die Technologie heute einerseits noch nicht ganz ausgereift ist, sie aber andererseits als Basis angesehen werden kann, auf der aufbauend weitere Technologien entstehen werden. Die Kernaspekte von Web Services sind jedoch soweit standardisiert, dass sie sich auch in Produktivumgebungen einsetzen lassen.

Im Gebiet der Anwendungsintegration stellt sich die Frage, wie sich Web Services im Vergleich zu etablierten Integrationstechnologien und Werkzeugen (z.B. aus dem Enterprise Application Integration Umfeld) positionieren. Um diese Frage beantworten zu können, werden im vorliegenden Artikel Integrationsszenarien genauer untersucht und anhand ihrer charakteristischen Merkmale klassifiziert.

Der Artikel gliedert sich dabei wie folgt: Nach einer grundlegenden Betrachtung der Ebenen, auf denen Softwaresysteme integriert werden können, werden in die charakteristischen Merkmale von Integrationsszenarien diskutiert. Der letzte Abschnitt stellt bestehende Integrationstechnologien vor und geht darauf ein, für welche

Integrationsmerkmale die verschiedenen Technologien geeignet sind. Der Artikel schließt mit einer Zusammenfassung und einem Ausblick.

2 Integrationsebenen

Die Integration von Softwaresystemen in der Informationsverarbeitung kann mindestens auf den Ebenen der Daten, Anwendungen und Prozesse betrachtet werden [BFGH02]. Die dabei relevanten Aspekte werden im Folgenden dargelegt.

Datenintegration

Grundlegende Techniken zur Integration von Informationssystemen sind auf der Ebene der Datenbestände bzw. auf Basis der Schemata zugrunde liegender Datenbanken angesiedelt. Ziel ist fast immer die Sicherstellung anwendungsübergreifender Konsistenz der Daten. Eine erste naheliegende Lösung sieht die Einführung eines unternehmensweit integrierten Datenschemas vor. Die Idee ist hier, nur eine Datenhaltungskomponente und nur ein Datenschema zu verwenden und Anwendungen so zu entwickeln, dass sie auf dieses Datenschema aufsetzen. Somit können die Konsistenzsicherungsmechanismen des DBMS genutzt werden. Dieser Ansatz ist jedoch nur sinnvoll verwendbar, wenn Anwendungen neu entwickelt bzw. reimplementiert werden sollen oder können. Die oft in Unternehmen vorzufindende heterogene Systemlandschaft macht andere Ansätze erforderlich.

Zur Datenintegration wird in heterogenen Systemlandschaften häufig Datenzugriffsmiddleware eingesetzt. Sie ermöglicht neben einer für Anwendungen transparenten Einbindung externer Daten den Erhalt vorhandener Zuständigkeiten für Datenpflege und -verwaltung sowie die Wahrung der Autonomie existierender Anwendungen. Verwendete Technologien sind hier Open Database Connectivity (ODBC) und deren Nachfolbertechnologien wie OLEDB und ADO.net sowie die Java Database Connectivity (JDBC) und deren weiterführende Entwicklungen wie Entity Beans im J2EE-Umfeld. Die Integration der Schemata erfolgt hier somit lediglich auf logischer Ebene. Als Voraussetzung für die Integration externer Datenbestände müssen Systemhersteller die Semantik der Datenschemata offen legen bzw. den direkten Zugriff auf den Datenbestand freischalten. Ist dieser Zugriff nur über spezielle Application Programming Interfaces (APIs) möglich, so sind Ansätze, die nur auf Mechanismen beteiligter (relationaler) Datenbankmanagementsysteme (DBMS) basieren, nicht anwendbar.

Anwendungsintegration

Neben der Integration von Schemata und Daten verschiedener Anwendungen ist jedoch auch die Integration der Anwendungen selbst wichtig. Ziel ist dabei einerseits die oben genannte anwendungsübergreifende Konsistenz der Daten. Andererseits wird die möglichst weitgehende Wiederverwendung und Nutzung von vorhandener Funktionalität aus den integrierten Systemen (funktionale Integration) sowie eine ein-

heitliche Benutzungsoberfläche (GUI-Integration) angestrebt. Als Basis für die Systemkopplung dienen Integrationsplattformen, welche die erforderlichen Dienste bereitstellen. Derzeit bieten sich mit .net [MSOFT03] und J2EE [Mon02] zwei Integrationsplattformen an, die die notwendige Infrastruktur bereitstellen. Mit Einschränkungen kann auch CORBA [OMG02] hier genannt werden, doch insbesondere in Bezug auf GUI-Integration (Nutzung von Web-Frontends) sowie Verfügbarkeit und Unterstützung der Schnittstellen durch Anwendungen und Komponenten (Einbettung von Visual Basic bzw. zahlreicher Java-Pakete und APIs) bieten J2EE und .net hier Vorteile.

Oft kommt bei der Anwendungsintegration Message-Oriented Middleware [Bern96] zum Einsatz. J2EE bietet hier das Konzept der Message Queues, Publish/Subscribe-Mechanismen und Message-Driven Beans. Die damit realisierbaren Architekturen verbinden funktionale Integration mit den Vorteilen asynchroner Kommunikation (Send and Forget) und erlauben eine lose Kopplung der beteiligten Anwendungen, denn oft besteht die Anforderung, dass Systeme durch Downzeiten angekoppelter Systeme nicht in ihrer eigenen Verfügbarkeit beeinträchtigt werden dürfen. Das Nachrichtenformat ist typischerweise XML-basiert, um die Einbindung weiterer Systeme zu erleichtern. Die funktionale Integration allein reicht jedoch nicht aus, da hierdurch zwar Funktionalität wiederverwendet werden kann, nicht aber Dialogelemente, die zur Parametrisierung bzw. Präsentation der Funktionalität nötig sind. Aus diesem Grund wird die Integration der Anwendungen auch auf der Ebene der Benutzungsoberflächen durchgeführt. Hierzu sind mehrere Wege denkbar. Der klassische Weg führt zur Integration von GUI-Controls, wie sie als Microsoft Foundation Classes, Java AWT oder Swing bekannt sind [Sun02]. Teilweise lassen sie sich auch im Java-Applet-Umfeld nutzen. Ist jedoch der Einsatz im Internet erforderlich (z.B. im B2B-Umfeld oder im Bereich Customer Self Service), so sind schlanke, nahezu überall lauffähige Frontends erforderlich, die auf den Client-Rechnern keine Arbeiten (Installation, Administration) erfordern. Die Wahl fällt hier auf HTML-Clients, die z.B. mit den Technologien ASP.net bzw. JSP realisiert werden können.

Prozessintegration

Daten- und Anwendungsintegration reichen nicht aus, wenn komplexe Geschäftsprozesse unterstützt werden müssen, die zudem einem ständigen Wandel unterliegen. In diesem Fall muss es mit einfachen Mitteln möglich sein, die Geschäftsprozesse um Aktivitäten zu erweitern oder anzupassen. Hierzu bedarf es jedoch einer übergeordneten Steuerungsebene, mit der anwendungsübergreifende Geschäftsprozesse realisiert werden können. Um dies umzusetzen, werden Konzepte aus den Bereichen Workflowmanagement (WFM) und Enterprise Application Integration (EAI) bemüht. Hierbei werden die Workflow-Schemadaten, Workflow-Instanzdaten sowie Akteure und Aktionen zentral in einem Prozesssteuerungssystem verwaltet. Je nach Art der Prozesse kann dies ein Workflowmanagementsystem (WfMS) oder eine EAI-Engine sein. Im Mittelpunkt der Prozessintegration stehen die betriebswirtschaftlichen und auf den Kunden ausgerichteten Geschäftsprozesse. Ziel der Prozessintegration ist die vollständige elektronische Bearbeitung aller kundenorientierten Prozesse innerhalb des Unternehmens. Zentrale Aspekte sind hier die automatischen Aktivitäten und die

Arbeitskorb-Metapher. Während erstere völlig ohne Eingriff des Anwenders durch das System gesteuert werden (durch asynchrone Nachrichtenkommunikation, in Einzelfällen auch durch synchronen Aufruf von z.B. Session Beans), stellt der Arbeitskorb die zentrale Schnittstelle zwischen Prozess und Benutzer dar. Auch hier steht die asynchrone Bearbeitung im Vordergrund. Wird z.B. in einem Geschäftsprozess nach Ausführung von automatischen Aktivitäten eine Bearbeitung durch einen Anwender notwendig, so erhält dieser (oder seine Rolle) im Arbeitskorb eine entsprechende manuelle Aktivität zugeordnet. Die Aktivität ist z.B. anhand von Web-Formularen durchzuführen, die vom System dem Anwender prozessgesteuert vorgelegt werden. Nach der Bearbeitung übernimmt das System die weitere Steuerung des Prozesses.

3 Charakteristika eines Integrationsszenarios

Betrachtet man die in konkreten Projekten auftretenden Integrationsszenarien genauer, so stellt man schnell fest, dass sie sich oftmals nicht genau einer Integrationsebene zuordnen lassen. Vielmehr wachsen die Integrationsebenen zusammen, so dass die Integration auf mehreren Ebenen erfolgt. Damit reicht die Klassifikation nach Integrationsebenen nicht aus, um Integrationsszenarien zu charakterisieren und die für das jeweilige Szenario geeignete Integrationstechnologie auszuwählen.

Um dennoch die Auswahl der geeigneten Technologie unterstützen zu können, haben die Autoren die charakteristischen Merkmale von Integrationsszenarien analysiert. Dabei haben wurden sechs Merkmale identifiziert: Automatisierungsgrad und Benutzerinteraktion, Strukturiertheit der Daten, Dynamik der Daten, Anzahl der zu integrierenden Systeme, Kommunikationsparadigma und Status und Zustandshaltung, die im Folgenden genauer erläutert werden.

Automatisierungsgrad und Benutzerinteraktion

Zwei fast immer in Integrationsszenarien relevante, zudem komplementäre Aspekte sind der Grad der Automatisierbarkeit von Prozessschritten und der Grad der benötigten Benutzerinteraktion. Ist der Grad der Automatisierbarkeit hoch, so können die einzelnen Prozessschritte bzw. Aktivitäten ohne Interaktion durch einen Bearbeiter durchgeführt werden. Ziel der Geschäftsprozessoptimierung ist es typischerweise, diesen Grad zu erhöhen. Ein hoher Automatisierungsgrad bedeutet zudem, dass Aspekte wie Benutzerverwaltung, Rollen, Authentifizierung weniger relevant sind. Einzelne Aktivitäten können vom System direkt ohne Rückkopplung durch den Anwender durchgeführt werden. Dies bedeutet auch, dass die Lokalisation der Aktivität unkritisch ist, und z.B. von Loadbalancing-Strategien bestimmt werden kann (serverbasierte Durchführung automatischer Aktivitäten) und dass synchrone Kopplung nicht notwendig ist (da kein Benutzer unmittelbar auf Rückmeldung wartet). Ein hoher Automatisierungsgrad erfordert effiziente Konzepte für Interprozesskommunikation, Multithreading und Schnittstellen zu zu steuernden Komponenten (APIs).

Dem entgegen steht der Grad der notwendigen Benutzerinteraktion bei der Bearbeitung von Geschäftsprozessen. Notwendige Benutzerinteraktionen reduzieren den

Grad der Automatisierbarkeit und sollten minimiert werden. Derartige Interaktionen können jedoch vielfältig sein, z.B. Synchronisationspunkte innerhalb weitgehend automatisierter Prozesse, an denen der Anwender automatisch erstellte Berechnungsergebnisse prüfen muss, bevor die nächste automatische Aktivität gestartet wird, Erfassung von zusätzlichen Daten über Bildschirmformulare oder manuelle Bearbeitung von Dokumenten (z.B. Textdokumente oder Grafiken). Benutzerinteraktionen erfordern einen Mechanismus, über den Aktivitäten einzelnen Anwendern oder Rollen zugeordnet werden können (Arbeitskorb-Metapher). Dieser Mechanismus wird insbesondere zur asynchronen Kopplung genutzt. Zur Einbindung von Formularen ist ein entsprechendes Framework nötig. Weiterhin sind clientseitige Schnittstellen zu Anwendungen notwendig, die zur Unterstützung bestimmter Aktivitäten (z.B. Dokumentbearbeitung) notwendig sind.

Der Grad der Automatisierbarkeit und Benutzerinteraktion variiert und ist vom Geschäftsprozess abhängig. In den seltensten Fällen liegen in Unternehmen die Extremsituationen (sehr hohe Automatisierung, sehr geringe Benutzerinteraktion bzw. sehr geringe Automatisierung, sehr hohe Benutzerinteraktion) vor.

Strukturiertheit der Daten

Ein wichtiger Aspekt bei der Integration ist die Struktur bzw. der Strukturierungsgrad der zu integrierenden Daten. In Unternehmen ist dabei typischerweise eine große Bandbreite an Daten anzutreffen, die in Geschäftsprozessen relevant sind. Zur Optimierung der Geschäftsprozesse ist es sinnvoll, den Anteil strukturierter Daten zu erhöhen und den Anteil an unstrukturierten Daten zu verringern, denn strukturierte Daten erlauben einen höheren Automatisierungsgrad, während unstrukturierte Daten fast immer erweiterte Benutzerinteraktionen, z.B. Interpretation, erfordern.

Beim Austausch auf Dateiebene sind die Daten häufig rudimentär formatiert, z.B. in Form kommaseparierter Werte. Diese lassen sich innerhalb einer Integrationsplattform durch File-Adapter einlesen und weiterverarbeiten. Ist die Struktur der Daten selbst komplex, wie z.B. bei EDI-Formaten, so gibt es heute zu einer XML-Darstellung praktisch keine Alternative, weil der individuelle Entwicklungsaufwand für das Einlesen und Parsen der Daten dadurch entfällt. Handelt es sich hingegen um Dokumente in proprietären Formaten, wie z.B. Word-Dateien, so werden die Daten in der Regel nur als Ganzes weitergeleitet oder archiviert. Zwar existieren inhaltliche Zugriffsmöglichkeiten über Objektmodelle, doch die individuellen Entwicklungsaufwände für eine inhaltliche Interpretation der Dokumente sind meist beträchtlich, so dass hierfür einfachere Mechanismen eingesetzt werden. Die letzte Gruppe der zu verarbeitenden Daten bilden die unstrukturierten Dokumente, wie z.B. Bilddateien, Faxe oder eingescannte Schriftstücke.

Dynamik der Daten

Als Datenbasis einer Integrationslösung kann im Prinzip die gemeinsame Datenbasis aller angeschlossenen Anwendungen angesehen werden. Ziel beim Betrieb einer Integrationsplattform sollte es sein, die Gesamtheit der Daten in einem konsistenten

Zustand zu halten. Da sich bei einer Vielzahl von angeschlossenen Systemen Datenredundanzen nicht vermeiden lassen, sind Abgleichsmechanismen zu implementieren. Hierbei muss sorgfältig zwischen der Häufigkeit von Datenänderungen, der Notwendigkeit einer Verfügbarkeit aktueller Daten in einzelnen Systemen und dem beim Datenabgleich entstehenden Kommunikationsaufwand abgewägt werden.

Bei bestimmten hochdynamischen Daten ist die Aktualität das entscheidende Wertmerkmal. Hierzu zählten Börsendaten, aber auch Wetterdaten. Entscheidend ist hier, dass die Daten nur in dem Moment der Messung bzw. Aufnahme wirklich aktuell sind. Daher müssen die Daten möglichst zeitnah beim Interessenten zur Wahrnehmung gelangen. Zum Beispiel werden hereinkommende Daten als Nachrichten entgegengenommen und sofort auf die Zielsysteme zum Abruf verteilt.

In anderen Fällen können zum automatisierten Datenabgleich mit mehreren Zielsystemen Publish-Subscribe-Mechanismen eingesetzt werden. Sofern die Änderungen plattformseitig gesammelt werden und der Abgleich nur selten stattfindet, ist es sinnvoll, die in einem System verfügbaren Daten mit einem Aktualitätszeitstempel zu versehen. Betriebsintern kann dem Bearbeiter die Möglichkeit eingeräumt werden, den Datenabgleich bei Bedarf manuell anzustoßen. Gängige EAI-Plattformen bieten hierfür entsprechende GUIs an.

Anzahl der zu integrierenden Systeme

Soll eine zentrale Integrationsplattform eingesetzt werden, um die Anzahl individueller Systemschnittstellen (Punkt-zu-Punkt-Verbindungen) zu reduzieren, so fällt auf, dass bei wenigen Systemen häufig mehr Schnittstellen von und zu der Integrationsplattform zu implementieren sind als ursprünglich Punkt-zu-Punkt-Verbindungen existieren. In der Regel ist nämlich nur ein Bruchteil aller möglichen Punkt-zu-Punkt-Schnittstellen sinnvoll und daher auch realisiert, wohl aber müssen als Voraussetzung für die Integration Kopplungen aller Systeme zur zentralen Plattform erstellt werden. Das Argument der Aufwandsreduktion ist somit erst ab einer größeren Anzahl von Systemen wirksam. Wird hingegen auf die höhere Qualität der Systemanbindungen und die Mehrwertdienste einer Integrationsplattform Wert gelegt, so tritt das Aufwandsargument in den Hintergrund. Daher kann eine plattformgestützte Integrationslösung bereits bei nur zwei Systemen sinnvoll sein.

Werden sehr viele Systeme integriert, so muss beachtet werden, dass mit der Anzahl der Systeme sich in der Regel auch die Technologien der Adapter immer stärker diversifizieren. Setzt die Plattform schwerpunktmäßig auf individuelle Adapterentwicklung, so können hier erhebliche Aufwände schlummern. Im Vorteil sind hier Plattformen, die bereits einen großen Satz von konfigurierbaren technologiespezifischen Adaptern mitbringen. Alternativ dazu bietet es sich an, Adapter auf Basis von Web Services zu realisieren. Hierdurch schafft man eine Anwendungskonnektivität, die unabhängig von der gewählten Integrationsplattform ist.

Sind die Systeme auf verschiedene, durch Firewalls getrennte Netzwerke verteilt, so kann die Integrationsplattform in jedem Netzwerk Kommunikationsserver platzieren, die über freigeschaltete Ports kommunizieren und damit eine Art Backbone für die Prozesskommunikation darstellen. Dies erlaubt es der Plattform theoretisch, Prozesse zentral administrierbar und überwachbar zu halten. Eine Alternative

hierzu stellt ein zentraler Server dar, der über ein VPN direkt mit den verschiedenen Systemen kommuniziert.

Kommunikationsparadigma

Eine besonders lose Form der Kommunikation ist der Export und Import von Austauschdateien: Die Quellanwendung schreibt ihre Daten in ein File in einem Austauschverzeichnis, welches von der Zielanwendung oder von einem Importskript im Batchbetrieb regelmäßig geprüft wird („push and poll“). Gerade Altsysteme bieten häufig nur diese Art des Datenaustausches an. Auch beim Einsatz einer Integrationsplattform muss die Ankopplung dieser Systeme asynchron über individuell konfigurierte File-Adapter erfolgen. Insbesondere die koordinierte Verteilung der Daten auf mehrere Zielsysteme (z.B. beim Stammdatenabgleich) wird durch die Plattform besser unterstützt als z. B. auf der Ebene der Shellskripte. Als weitere Form der asynchronen Kommunikation z.B. mit externen Systemen werden E-Mails eingesetzt. Eingehende E-Mails werden über spezielle Adapter interpretiert und weiterverarbeitet, ebenso werden E-Mails plattformseitig inhaltlich generiert und versandt.

Verfügt eine Anwendung über ein Remote API oder handelt es sich um eine datenbankgestützte Anwendung, so stehen verschiedene Mechanismen der synchronen Kommunikation offen: Zunächst kann die Anwendung über einen individuell zu entwickelnden bzw. proprietären Systemadapter der Integrationsplattform angesprochen werden. Alternativ dazu bietet sich eine Systemanbindung mittels Web Services an: Für verschiedene Programmiersprachen (z.B. Java und .net-Sprachen) stehen heute mächtige Tools zur Verfügung, mit denen aus einem Remote API anwendungsspezifische Web Service Schnittstellen inkl. WSDL-Beschreibungen generiert werden können. Bei einer datenbankgestützten Anwendung ohne eigenes API ergibt sich als weitere synchrone Kommunikationsmöglichkeit der pragmatische Weg des direkten Datenbankzugriffs. Der zugehörige individuell konfigurierte Datenbankadapter kann seine Funktionsschnittstelle plattformseitig z.B. über ein Web Service „Facade“ anbieten.

Mit der Anzahl der integrierten Systeme steigt auch der Kommunikationsaufwand innerhalb der Plattform an. Eine Methode, um die Kommunikation zu reduzieren, ist der von Java her bekannte Publish-Subscribe-Mechanismus, der auch von heutigen Integrationsplattformen unterstützt wird. Dabei melden sich nur „interessierte“ Anwendungen beim Verteiler bestimmter, von der Plattform definierter Nachrichten an. Nützlich ist dieses Verfahren z.B. bei komplexeren Koordinierungsaufgaben, wie dem oben erwähnten Stammdatenabgleich.

Status und Zustandshaltung

Der Bearbeitungsstand von Geschäftsprozessen wird klassischerweise manuell über Auftrags- oder Laufzettel erfasst, die von Bearbeitungsinstanz zu Bearbeitungsinstanz weitergereicht werden. Werden solche Vorgänge automatisiert und mittels einer Integrationsplattform gesteuert, so sollten auch die Zustände der einzelnen Prozessinstanzen über diese Plattform fortgeschrieben werden. Hierbei können drei Arten von

Zustandsinformation unterschieden werden: Zunächst ist für jede Prozessinstanz ein globaler Prozesszustand (z.B. „aktiv“) definiert. Ergänzt wird dieser durch ein Protokoll der im Prozessablauf durchgeführten Aktivitäten und der beteiligten Akteure (Systeme, Bearbeiter), d.h. durch eine Beschreibung des Weges, auf dem der aktuelle Prozesszustand erreicht wurde (Historienfortschreibung). Hinzu kommen prozessspezifische Informationen, für welche das System die Möglichkeit einer individuellen Festlegung anbieten muss. Sämtliche Zustandsinformationen sollten durch das System automatisch instanziiert, initialisiert und aktualisiert werden.

Ein weiterer Faktor ist die Persistierung der Statusinformation, ohne die z.B. keine langlaufenden Geschäftsprozesse abgewickelt werden könnten (im J2EE-Umfeld wird diese Aufgabe standardmäßig von Stateful Session-Beans erfüllt). Sinnvoll ist ferner, dass ein Integrationssystem bei Prozessänderungen noch aktive alte und bereits gestartete neue Prozessinstanzen parallel betreiben kann. Hierfür muss das System die gleichzeitige Verwaltung der individuellen Zustandsinformationen beider Prozessversionen unterstützen.

4 Integrationstechnologien

Die Entscheidung für eine konkrete Integrationstechnologie ist abhängig von den charakteristischen Merkmalen des Integrationsszenarios. Prinzipiell lassen sich die Integrationstechnologien in die Gruppen „klassische“ EAI-Werkzeuge, Middleware-Technologien, Web Services und Web Services in Kombination mit Business Process Modeling Languages unterteilen. Tabelle 1 gibt einen Überblick über die verschiedenen Integrationstechnologien und zeigt, welche Integrationscharakteristika die einzelnen Technologien unterstützen. Zum Vergleich wird in der Tabelle zusätzlich die Punkt-zu-Punkt-Integration den übrigen Integrationstechnologien gegenübergestellt.

	EAI-Werkzeuge	WfMS	Middleware	Web Services	Web Services + BPML	Punkt-zu-Punkt
Hoher Automatisierungsgrad	++	+	-	-	++	+ ¹
Viele Benutzerinteraktionen	0	++	-	-	+	+ ¹
Unterstützung für strukturierte Daten	++	0	++	++	++	++
Unterstützung für unstrukturierte Daten	0	++	0	0	+	0
Eignung für Real Time Szenarien	+	+	++	0	0	++

¹ Bestandteil der beteiligten Anwendungen

Eignung für Batch-Prozesse	++	0	+	0	0	+ ¹
Integration vieler Anwendungen	++	+	++	++	++	--
Synchrone Kommunikation	++	+	++	++	++	+
Asynchrone Kommunikation	++	++	++	0	+	+
Unterstützung von Zustandshaltung	++	++	0	-	+	+ ¹

Tabelle 1. Eignung der Integrationstechnologien für die verschiedenen Merkmalsausprägungen der Integrationsszenarien

Klassische EAI-Werkzeuge

EAI-Werkzeuge ermöglichen die Integration existierender Anwendungen auf Daten-, Anwendungs- und Prozessebene. Bei der Integration mittels eines EAI-Werkzeugs ist typischerweise für jede Anwendung, die integriert werden soll, ein Adapter zu erstellen, der die Transformation der Daten und der entfernten Systemaufrufe realisiert. Diese Adapter basieren in der Regel auf proprietären Klassenbibliotheken.

Neben der reinen Transformation der Daten und Systemaufrufe stellen die meisten EAI-Werkzeuge Funktionalität zur Lastverteilung, Ausfallsicherheit und zum Monitoring der EAI-Aktivitäten zur Verfügung. EAI-Werkzeuge eignen sich in erster Linie für vollautomatisierte Arbeitsabläufe, die auf strukturierten Daten arbeiten. Dabei wird der Bearbeitungsstand (Status) der einzelnen Arbeitsabläufe in der Regel vom EAI-Werkzeug registriert und dann über das EAI-Werkzeug überwacht.

Mit der wachsenden Popularität der Web Services haben auch die EAI-Anbieter das Potential dieser Technologie erkannt. So bieten heute viele EAI-Werkzeuganbieter die Integration nicht mehr nur mittels proprietärer Adapter an, sondern ermöglichen es den Integratoren, Anwendungen über Web Services an das EAI-Werkzeug anzubinden.

Workflow Management Systeme

Ebenso wie klassische EAI-Systeme dienen auch WfMS der Automatisierung von Arbeitsabläufen. Der wesentliche Unterschied zwischen diesen beiden Systemklassen liegt jedoch in der Art der Anwendungen, die automatisiert werden sollen. Während bei EAI-Systemen vollautomatisierte Prozesse im Vordergrund stehen, konzentrieren sich WfMS auf Arbeitsabläufe, die Benutzerinteraktionen erfordern. Der Bedarf an Benutzerinteraktionen ist auch dadurch begründet, dass WfMS mit unstrukturierter Daten arbeiten. Um diese Daten geeignet interpretieren und verarbeiten zu können, ist meist der Eingriff eines Menschen erforderlich.

Middleware-Technologien

Middleware-Technologien für sich genommen sind nicht ausreichend für die Integration existierender Anwendungen. Sie bieten jedoch eine Infrastruktur, mittels derer Anwendungen miteinander kommunizieren, Daten ausgetauscht und entfernte Methoden einer anderen Anwendung aufgerufen werden können. Damit eignen sich Middleware-Technologien für eine Integration auf Daten- und/oder Anwendungsebene. Die Prozessintegration wird durch die Middleware-Technologien nur insoweit unterstützt, dass entfernte Systemaufrufe erfolgen können. Wann, wie und in welcher Reihenfolge diese Aufrufe erfolgen, wird von den beteiligten Anwendungen bestimmt. Die Middleware selbst hat von Abhängigkeiten zwischen den einzelnen Systemaufrufen und damit vom Status der Arbeitsabläufe keinerlei Kenntnis.

Web Services

Die Web Services Standards bieten standardisierte Protokolle (SOAP) und Schnittstellenbeschreibungen (WSDL) für das Auffinden (UDDI) und die Nutzung entfernter Dienste. So kann ein Anbieter Anwendungsfunktionalität als Dienst über einen Web Service zur Verfügung stellen, der von beliebigen Nutzern (Personen oder Softwaresystemen) genutzt werden kann. Web Services eignen sich damit für eine lose Kopplung von Softwaresystemen.

Die Web-Services-Technologie erlaubt es nicht, verschiedene aufeinanderfolgende Dienstaufrufe zueinander in Beziehung zu setzen (ausser die beteiligten Anwendungen regeln dies), oder eine Abfolge von verschiedenen Dienstaufrufen festzulegen. Sie befasst sich immer nur mit genau einer Nutzung eines Dienstes.

Web Services in Kombination mit Business Process Modeling Languages

Zur Spezifikation von Geschäftsprozessen sind in den letzten Jahren eine Reihe von Standards wie BPEL4WS, BMPL, XPDL, UML entstanden, wobei BPEL4WS, BMPL und XPDL XML-basierte Sprachen sind [BPML03, BPEL03]. BPEL4WS (Business Process Execution Language for Web Services) ist speziell auf die Koordination von Web Services ausgerichtet. Gesteuert und überwacht wird die Ausführung eines in BPEL4WS spezifizierten Geschäftsprozesses von einer sogenannten „BPEL Engine“. Derzeit ist die Entwicklung der Web Services noch nicht ausgereift. So gibt es noch einige offene Fragen bzgl. der Sicherheit und Transaktionen im Zusammenhang mit Web Services. Diese Probleme wurden jedoch erkannt, und es sind Standards im Entstehen, die sich mit den genannten Fragestellungen befassen. Für die Zukunft bietet die Kombination aus BPEL4WS und Web Services damit eine echte Alternative zu den klassischen EAI-Werkzeugen. Andererseits ist es auch denkbar, dass EAI-Werkzeuge um eine BPEL Engine erweitert werden.

Prinzipiell lassen sich mit Web Services und Business Process Modeling Languages Integrationsszenarien jeglicher Art realisieren, da sie alle oben genannten Integrationsmerkmale unterstützen.

5 Zusammenfassung und Ausblick

Dass Integration heute ein wichtiges Thema in fast allen Firmen ist, zeigen aktuelle Statistiken (vgl. [Born03]). Als Reaktion auf diesen hohen Bedarf nach Integrationslösungen haben Hersteller aus verschiedenen Bereichen ihre Werkzeuge um (standardisierte) Integrationsschnittstellen erweitert bzw. eigenständige Integrationswerkzeuge auf den Markt gebracht. Neben diesen reinen Integrationswerkzeugen haben sich in den letzten Jahren Basistechnologien für die Integration (z.B. CORBA, Web Services) etabliert. Die damit entstandene Vielfalt an Technologien und Werkzeugen stellt mit der Integration beauftragte Personen vor eine schwierige Aufgabe.

Um das geeignete Integrationswerkzeug bzw. die geeignete Integrationstechnologie auswählen zu können, muss das Integrationsszenario genauer untersucht werden, und es sollte eine Vision entwickelt werden, welche weiteren Integrationen in den nächsten Monaten und Jahren angestrebt werden. Auf der Basis dieser Vision kann dann die derzeit für diese Vision geeignete Integrationslösung ausgewählt werden.

Insgesamt lässt sich feststellen, dass die verschiedenen Integrationstechnologien immer weiter zusammen wachsen. So unterstützen heute bereits eine Reihe von EAI-Werkzeugen Web Services, und Workflow Management Systeme setzen auf standardisierte Business Process Modeling Languages. Solange der Prozess des Zusammenwachsens jedoch noch nicht abgeschlossen ist, steht der Integrationsverantwortliche vor der Qual der Wahl.

Literatur:

- [Bern96] Bernstein, Philip A. "Middleware: A Model for Distributed System Services," Communications of the ACM, February 1996, Vol. 39, No.2, 86-98.
- [BFGH02] Bunjes, B., Friebe, J., Götze, R. und Harren, A., Integration von Daten, Anwendungen und Prozessen am Beispiel des Telekommunikationsunternehmens EWE TEL, Wirtschaftsinformatik, 44, 2002, 5, 415-423
- [Born03] Born, A. und Diercks, J.: „Anwendungsintegration: grenzenlose Zusammenarbeit“, iX 11 (November 2003): 87-98
- [OMG02] Object Management Group: „Common Object Request Broker Architecture (CORBA/IIOP)“, http://www.omg.org/technology/documents/corba_spec_catalog.htm
- [BPML03] Business Process Modeling Language, <http://www.bpml.org>
- [BPEL03] BPEL4WS, <http://www-106.ibm.com/developerworks/webservices/library/ws-bpel/>
- [Sun02] Sun Microsystems, Inc: „The Java Platform: Five Years in Review.“ <http://java.sun.com/features/2000/06/time-line.html>
- [MSOFT03] Microsoft, Inc: „Enterprise Solution Patterns Using Microsoft .NET“, <http://msdn.microsoft.com/practices/type/Patterns/Enterprise>
- [Mon02] Monson-Haefel, Richard: „Enterprise Java Beans“, O'Reilly, 2001
- [Nus00] R. Nussdorfer, Das EAI-Buch, E-Business und EAI, Integration von Anwendungen, Trends, Technologie und Lösungen, CSA Consulting, 2000