Inhaltsverzeichnis

1. Einleitung

1.1 Überblick

Electronic Commerce ist ein hochaktuelles Anwendungsfeld der Informatik. Besonders die zunehmende Verbreitung des Internet in den letzten Jahren hat dazu beigetragen, dass zu Beginn des neuen Jahrtausends ein Boom im Bereich elektronischer Handelssysteme erwartet wird. Mehr und mehr private Haushalte verfügen über einen Internetanschluss. Das bietet Unternehmen die Möglichkeit, ihre Produkte „online" anzubieten und zu verkaufen.

1.1.1 Referenzmodell nach Merz

Merz [Merz99a] gibt mit seinem Referenzmodell einen Überblick über das große Gebiet des Electronic Commerce. Die unterste Ebene bilden bei ihm sogenannte Basistechnologien, zu denen das Internet und die Protokolle HTTP und TCP/IP zählen. Datenbanken werden ebenfalls dort eingeordnet, da sie in nahezu jedem Electronic Commerce System eingesetzt werden, um Kunden- und Produktdaten zu speichern. Aus dem Bereich der Softwaretechnologien sind Frameworks wie CORBA BOCA oder das San Francisco Framework von IBM von großer Bedeutung, aber auch Konzepte wie das komponentenbasierte Programmierparadigma, das in den in Abschnitt 2.2.1 vorgestellten Business Objects Anwendung findet. Nahezu jedes Teilgebiet der Informatik kann Technologien einbringen.

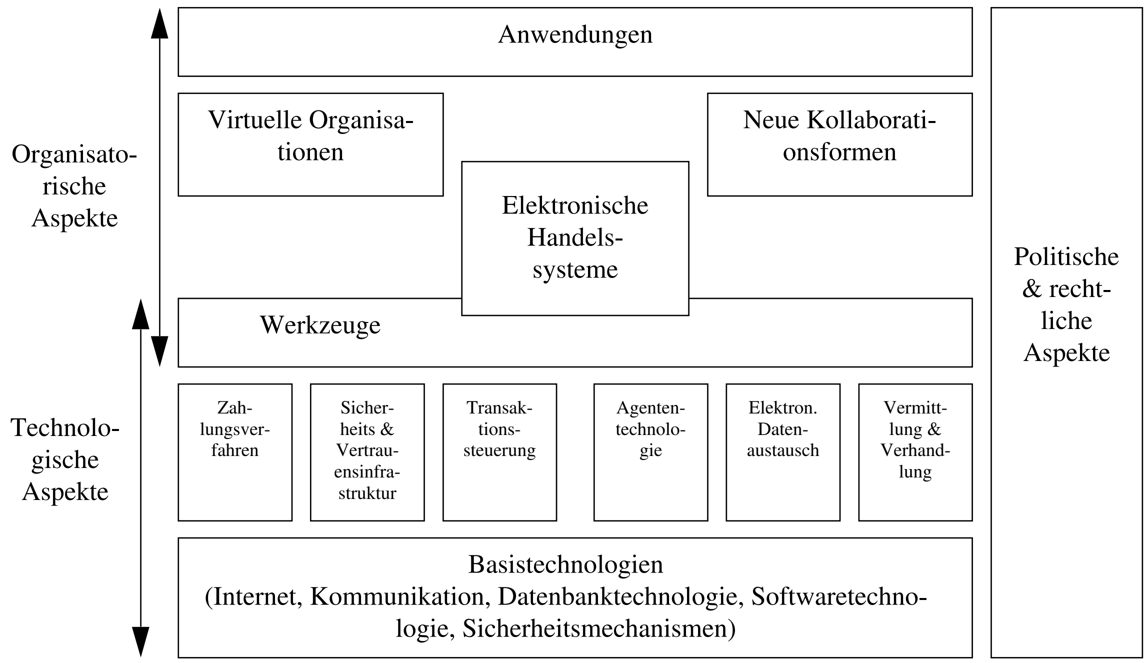

Abb. 1 Referenzmodell zum Electronic Commerce nach Merz

Die Basistechnologien werden dann auf der nächst höheren Ebene des Referenzmodells eingesetzt. Diese Ebene ist durch mehr Anwendungsnähe gekennzeichnet. Standards, wie zum Beispiel im Bereich elektronischer Zahlungsverfahren, verhelfen Electronic Commerce zu mehr und mehr Erfolg. Sicherheit und Vertrauen spielen eine ganz zentrale Rolle und werden daher schon durch Maßnahmen auf einer der niedrigsten Ebenen gewährleistet. Die Privatsphäre der

Internet-Nutzer muss geschützt werden, und der Fluss von Geld und Gütern muss sichergestellt sein. Für den elektronischen Datenaustausch müssen Dokumentformate und Protokolle standardisiert werden.
Auf der organisatorischen Ebene können auf diesen Grundlagen Electronic Commerce Anwendungen realisiert werden. Elektronische Handelssysteme wie Online-Shops nutzen die zur Verfügung stehenden Zahlungsverfahren, um ihren Kunden sichere Bezahlung gewährleisten zu können. Sie setzen Transaktionssteuerung ein, um die hohe Zahl von teilweise mehreren tausend Transaktionen pro Sekunde bewältigen zu können. Wünschenswert sind Werkzeuge, um Electronic Commerce Systeme aufzubauen und zu verwalten, da durch die integrierten grafischen Benutzeroberflächen neuen Anwendern der Zugang erleichtert werden kann.

1.1.2 Neue Kollaborationsformen

Ebenfalls auf der organisatorischen Ebene entwickeln sich neue Kollaborationsformen wie Auktionssysteme und Agentensysteme. Auktionssysteme erlauben eine effiziente Ressourcenallokation aufgrund der flexiblen Preisgestaltung. Anbieter und Nachfrager können sich auf einem virtuellen Marktplatz treffen und verhandeln, ohne physisch anwesend zu sein. Die Kosten eines Geschäfts können so deutlich verringert werden, denn es entfallen beispielsweise Reisekosten. Agentensysteme nutzen die im Referenzmodell aufgeführte Agententechnologie. Agenten sind Softwaresysteme, die in einem eingeschränkten Gebiet Aufgaben des Menschen übernehmen können, von der Informationssuche bis zu Verhandlung und Kauf. Der Nutzer eines Agentensystems kann nun einen Agenten programmieren, der dann mit anderen Agenten und Dienstanbietern in Kontakt tritt, um seinen Auftrag zu erfüllen.

1.1.3 Neue Organisationsformen

Es entwickeln sich interessante neue Organisationsformen, zum Beispiel die der Virtuellen Organisation. Unternehmen arbeiten zur Erbringung einer bestimmten Leistung zusammen und lösen ihren Verbund danach wieder. Ein Beispiel kann die Publikation eines Buches sein [Merz99b]. Der Autor kann einen Publikationsagenten beauftragen, der über eine Schnittstelle den Auftrag und das Manuskript erhält und für ihn die Koordination aller erforderlichen Dienstleistungen übernimmt. Es werden Herausgeber, Werbeagenturen, Korrekturleser und Kritiker kontaktiert, die alle eine Leistung zur Publikation beitragen. Möglich machen das der elektronische Datenaustausch und die oben angesprochene Agententechnologie.

1.1.4 Politische und rechtliche Aspekte

Über alle Ebenen hinweg spielen politische und rechtliche Aspekte eine Rolle. Beispielsweise müssen alle Geschäfte vertraglich abgesichert werden. Da viele Geschäfte länderübergreifend abgeschlossen werden, müssen sich die Regierungen auf Rechtsstandards in ihren Ländern verständigen. Der EU-Ministerrat hat kürzlich beschlossen, dass bei Geschäften mit Privatkunden das Recht des Heimatlandes des Privatkunden anzuwenden ist, bei Geschäften mit Unternehmen das Recht des Landes, in dem der Anbieter ansässig ist [CompZei99].

Electronic Commerce ist ein sehr umfassender Begriff. Um ihn über das Referenzmodell hinaus noch weiter zu durchleuchten, wird im folgenden Abschnitt eine gängige Art der Klassifikation vorgestellt [Merz99a, Merz99b]. Dazu werden die Beteiligten an einem Geschäftsprozess in die Gruppen Unternehmen und Konsumenten eingeteilt. Es werden Business-to-Business-Commerce (B2B) und Business-to-Consumer-Commerce (B2C) als die beiden bedeutendsten Ausprägungen von Electronic Commerce vorgestellt. Darüberhinaus gibt es noch Business-to-Administration-Commerce (B2A). Es wäre zum Beispiel denkbar, dass öffentliche Ausschreibungen im Internet zur Verfügung gestellt werden. Dieser Bereich ist aber zum heutigen Zeitpunkt noch von untergeordneter Bedeutung und soll daher nicht genauer betrachtet werden.

1.2 Klassifikation

1.2.1 Business-to-Business-Commerce (B2B)

Der Einsatz von Electronic Commerce im Business-to-Business Bereich bietet Unternehmen die Möglichkeit, Kommunikation und Kooperation mit anderen zu fördern. Der elektronische Datenaustausch (EDI-*Electronic Data Interchange*) erleichtert die Abwicklung von Geschäftsvorfällen, da zum Beispiel Bestellungs- und Rechnungsformulare elektronisch verfügbar gemacht und über das Internet verschickt werden können. Auf dieser Grundlage von elektronischen Formularen können Geschäftsvorfälle sogar automatisiert werden, sowohl innerhalb eines Unternehmens als auch unternehmensübergreifend, zum Beispiel durch den Einsatz von Workflow-Management Systemen. Die gemeinsame Produktion wird effizienter. Es kann zum Beispiel Zulieferern jederzeit Information über die aktuelle Lagerausstattung und die aktuelle Auftragslage zugänglich gemacht werden. Die Zulieferer können dann bei Bedarf automatisch liefern, das belieferte Unternehmen muss sich nicht um Bestellungen kümmern und verringert dadurch seinen Arbeitsaufwand.

1.2.1.1 Technische Aspekte

Charakteristisch für den B2B-Bereich ist, dass sich kommunizierende Softwaresysteme gegenüberstehen, zum Beispiel bei den oben erwähnten automatisierten Geschäftsvorfällen. Da die Systeme keine standardisierten Schnittstellen besitzen, müssen zur Kommunikation bzw. zum Datenaustausch von allen anerkannte Nachrichtenformate verwendet werden. Seit den 80er Jahren entwickeln sich Standards im Bereich EDI, wie zum Beispiel EDIFACT (EDI For Administration, Commerce and Transport), die Nachrichtenformate und deren Semantik festlegen. In Zukunft könnte XML als Sprache des Datenaustausches eingesetzt werden. Es gibt schon erste Beispiele wie die Automobilhersteller DaimlerChrysler, Ford und General Motors, die im EDI-Bereich XML einsetzen.

1.2.1.2 Anwendungsbeispiele

Ein Anwendungsbeispiel für B2B-Commerce sind die sogenannten Extranets. So wie Intranets Informationen für die Mitarbeiter bieten, wird dasselbe für Geschäftspartner ermöglicht. Extranets sind geschützte Bereiche, der Zugriff ist nur autorisierten Partnern gestattet. Das oben erwähnte Beispiel, in dem der Zulieferer das Lager des Belieferten einsehen konnte, wäre auf diese Weise realisierbar.

Beschaffungssysteme als weiteres Anwendungsbeispiel bieten Unternehmen die Möglichkeit, gemeinsam am Markt aufzutreten und so Rabatte beim Einkauf zu erzielen. Man registriert die gewünschte Menge eines Produktes beim Beschaffungssystem. Der Kauf erfolgt, wenn eine Menge erreicht ist, die einen Rabatt zur Folge hat. General Motors, Cisco oder Hewlett Packard nutzen solche Beschaffungssysteme.

1.2.2 Business-to-Consumer Commerce (B2C)

Der Begriff Business-to-Consumer Commerce umfasst alle Geschäftsvorfälle zwischen Unternehmen und Privatkunden. Das Angebot ist sehr vielseitig und reicht vom Anbieten von Informationen über den Verkauf von Produkten bis zu Supportleistungen. Softwareprodukte sind geradezu prädestiniert für den Online-Handel, da sie auch online ausgeliefert werden können. Sehr großen Erfolg haben aber auch Bestellsysteme für Bücher, CDs, Videos oder DVDs, die dann über Versandsysteme ausgeliefert werden.
B2C-Commerce bietet Kunden die Möglichkeit, rund um die Uhr auf der ganzen Welt einzukaufen. Sie können sich schnell und umfassend auf den Web-Seiten der Anbieter informieren und Einkäufe tätigen, aber auch leicht Vergleiche anstellen, da die Anbieter nur einen Mausklick voneinander entfernt sind. Für die Anbieter besteht die Möglichkeit, dem Kunden konkrete Angebote zu unterbreiten, wenn dieser Informationen über sich zur Verfügung stellt. Ob diese Art der Interaktion über Web-Seiten allerdings ein Verkaufsgespräch in entsprechender Atmosphäre ersetzen kann ist fraglich. Sicherheitsaspekten muss ebenfalls immer wieder Beachtung geschenkt werden, wie schon im Referenzmodell angedeutet ist. Der Kunde gibt zum Beispiel beim Bezahlen Kreditkarteninformationen preis, mit denen großer Schaden angerichtet werden kann, wenn sie in die falschen Hände geraten.
Weiterhin besteht die Möglichkeit, Daten über die Kunden und ihr Kaufverhalten zu sammeln, was derzeit in hohem Maße getan wird. Man denke in diesem Zusammenhang auch an Data Warehouses und Data Mining.

1.2.1.1 Beispiele für erfolgreiche Anbieter im B2C-Bereich

Amazon, http://www.amazon.de
Bestellsystem für Bücher, CDs, Videos, DVDs; Amazon nimmt eine Vorreiterrolle im B2C-Commerce ein, da es eines der ersten Unternehmen war, das ausschließlich über das Internet verkauft hat.

Netzmarkt-Deutschlands erstes Internet-Kaufhaus, http://www.netzmarkt.de
Beispiel dafür, dass eine Vielzahl von unterschiedlichen Artikeln von technischen Produkten wie Computer oder Kameras bis zu verderblichen Produkten wie Blumen über das Internet verkauft werden kann.

Dell, http://www.dell.de
Dell war der erste Computerhersteller, der direkt an Endkunden verkauft hat und diese Firmenphilosophie im Internet konsequent fortsetzt.

Im weiteren Verlauf meiner Ausarbeitung werde ich mich auf Architekturen und Systeme im Bereich Business-to-Consumer-Commerce beschränken. Vor allem hier existieren enorm hohe Leistungsanforderungen, denn bei Anbietern wie Amazon greifen unter Umständen mehrere tausend Kunden zur gleichen Zeit auf das Online-Angebot zu. Das ist mit einer Vielzahl von Transaktionen verbunden: Web Seiten müssen für den speziellen Kunden personalisiert werden, jedes Suchen von Informationen und das Abwickeln eines Kaufes lösen zahlreiche Transaktionen aus. Lange Wartezeiten werden vom Kunden nicht akzeptiert und sorgen für ein schlechtes Image des Anbieters. Kommt es im Extremfall sogar zu Systemausfällen, so ist das mit hohen Verlusten für den Anbieter verbunden, da es zu Umsatzausfällen kommt. Daher sollten Electronic Commerce Systeme in der Lage sein, diesen hohen Leistungsanforderungen standzuhalten.
Die Basistechnologie HTTP führt zu einem Problem, das durch die Systeme ebenfalls gelöst

werden muss. HTTP ist ein zustandsloses Protokoll. Geschäftsvorfälle setzen sich in der Regel aus mehreren Transaktionen zusammen. Das erfordert die Speicherung des derzeitigen Verarbeitungszustandes über mehrere Verarbeitungsschritte hinaus. Da HTTP das aus dem gerade genannten Grund nicht leisten kann, müssen beispielsweise Cookies oder SessionIDs eingesetzt werden.

Im folgenden 2. Kapitel werden die Komponenten von Electronic Commerce Systemen vorgestellt. Das geschieht am Beispiel eines allgemeinen Geschäftsvorfalls. Im Zusammenhang mit einem Geschäftsvorfall wird auf das Integrationsproblem hingewiesen, das häufig auftritt, wenn Unternehmen eine neue Anwendung wie einen Online-Shop in das bestehende Anwendungssystem einbetten wollen. Nach der anschließenden Einführung allgemeiner Architekturen werden diese im 3. Kapitel genauer vorgestellt, genauso wie konkrete Anbieter von Electronic Commerce Lösungen, die eine der Architekturen einsetzen. Den Abschluss bildet das 4. Kapitel mit einer Möglichkeit zur Leistungsmessung von Electronic Commerce Systemen, dem Electronic Commerce Web Benchmark TPC-W. Unter 5. findet man dann eine Zusammenfassung.

2. Einführung in Electronic Commerce Systeme und Architekturen

Zunächst wird in diesem Kapitel ein allgemeiner Geschäftsvorfall im Bereich B2C vorgestellt. Daraus lässt sich das Integrationsproblem ableiten, das Unternehmen haben, wenn sie ihr bestehendes Geschäftssystem um einen Online-Shop erweitern wollen. Als mögliche zukünftige Lösung dieses Problems werden Business Objects und ein passendes Framework vorgestellt. Anschließend werden Varianten von Architekturen für Electronic Commerce Systeme vorgestellt, die anhand des Verarbeitungsortes der Applikationslogik unterschieden werden.

2.1 Modell eines allgemeinen Geschäftsvorfalls

Abb. 2 zeigt das Modell eines Geschäftsvorfalls im B2C-Bereich. Im Folgenden wird er anhand der Phasen Information, Verhandlung und Abwicklung charakterisiert.

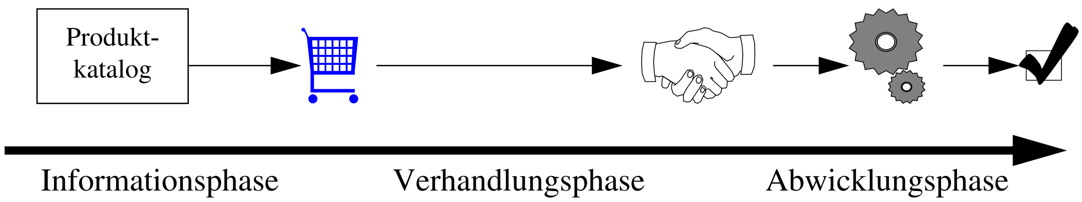

Abb. 2 Modell eines allgemeinen Geschäftsvorfalls

2.1.1 Informationsphase

In der Informationsphase wird über den Web-Server des Anbieters der Kontakt hergestellt. Der Kunde hat über die Web-Seite des Anbieters die Möglichkeit, auf einen Produktkatalog zuzugreifen, den er nach seinen Wünschen durchblättern kann. Das ist mit einer hohen Zahl von

Transaktionen verbunden, da die Seiten des Katalogs von der Datenbank angefordert werden müssen. Um das gewünschte Produkt schneller finden zu können, sollte die Suche durch Angabe von Kriterien möglich sein. Dem Kunden stehen in der Informationsphase nicht nur herstellerspezifische Produktkataloge zur Verfügung. Es gibt auch neutrale Informationsanbieter [Power00], die herstellerübergreifende Kataloge anbieten. Bei allen Anbietern wird Kunden, die bereits registriert sind, eine personalisierte Seite zur Verfügung gestellt, die ihren Interessen bzw. ihrem Informationsbedarf angepasst ist.
Entscheidet sich der Kunde für einen Artikel, den er im Produktkatalog gefunden hat, so kann er ihn in seinen elektronischen Warenkorb legen. Hier kommt die schon angesprochene Zustandslosigkeit des HTTP ins Spiel. Cookies erlauben es dem Anbieter, Informationen im Web-Browser des Kunden zu speichern, die dieser dann bei jedem Kontakt mit dem Web-Server mitsendet. So kann der letzte Zustand immer gesichert werden. Da allerdings Cookies vom Browser des Kunden abgelehnt werden können, ist es vorteilhaft, andere Verfahren zu verwenden. Es kann eine sogenannte SessionID beim Eintritt in den Shop vergeben werden, die dann in jede HTML-Seite integriert wird, die an den Browser des Kunden geschickt wird. Sie wird dann bei jedem darauffolgenden Mausklick zurück an den Web-Server übertragen und sorgt für eine Identifizierung des Kunden.

2.1.2 Verhandlungsphase

Die Verhandlungsphase wird oft schnell durchlaufen. Da der Preis in der Regel fest vorgegeben ist, muss nur noch im Falle von Hard Goods durch Kontakt mit dem Lager geklärt werden, ob der Artikel lieferbar ist. Unter Hard Goods versteht man Podukte, die man nicht auf elektronischem Weg liefern kann. Im Falle von Auktionssystemen kann die Verhandlungsphase länger andauern, denn es nimmt Zeit in Anspruch, bis sich die Parteien auf einen Preis geeinigt haben.

2.1.3 Abwicklungsphase

In der Abwicklungsphase steht zunächst die Wahl des Zahlungsverfahrens an. Es stehen unter anderem Lastschriftverfahren und Kreditkartenverfahren zur Verfügung. Welche Auswahl angeboten wird hängt vom Anbieter ab. Die Zahlung wird in der Regel über spezielle Anbieter abgewickelt, die über einen Payment-Gateway verbunden werden, zum Beispiel eine Internet Bank. Nach dem Bezahlen und dem Zustellen der Rechnung wird im Falle von Soft Goods Zugang zur Download-Area gewährt oder bei Hard Goods die Auslieferung veranlasst. Dazu ist Kontakt mit der Warenwirtschaft notwendig. Die Geschäftsdaten werden in der Regel noch an die angeschlossene ERP (Enterprise Ressource Planing)-Software weitergegeben, wie zum Beispiel SAP R/3.

2.1.4 Profiling

Während des gesamten Geschäftsvorfalls hat der Anbieter die Möglichkeit, Profilinformationen über seine Kunden zu sammeln, die diese freiwillig zur Verfügung stellen, wenn sie sich registrieren lassen, oder die aus Online-Sitzungen mittels geeigneter Software abgeleitet werden können. Diese Informationen können genutzt werden, um dem Kunden personalisierte Angebote zu unterbreiten, aber auch um Trends auf dem Markt zu erkennen, Kundenwünsche abzulesen und entsprechende Produkte zu entwickeln. Man kann den Einkaufskorb eines Kunden analysieren, aber auch die Web-Seiten verfolgen, die er besucht. Bei der Bonitätsprüfung vor dem Bezahlen gibt der Kunde ebenfalls Daten preis, die helfen, ihn einzuschätzen. Durch die Verarbeitung der gewonnenen Informationen erwarten sich die Unternehmen eine Absatzförderung.

Dem Datenschutz sollte im Zusammenhang mit Kundeninformationen Beachtung geschenkt werden. Die Anbieter müssen ihre gewonnenen Informationen vertraulich behandeln, wenn ein Kunde damit identifiziert werden kann.

In den Abschnitten Abwicklungsphase und Profiling wird deutlich, dass Electronic Commerce Anwendungen wie Online-Shops in das bestehende Anwendungssystem eines Unternehmens integriert werden müssen. Man denke an die Zusammenarbeit mit dem ERP-System oder die Verarbeitung der gewonnenen Kundeninformationen mit geeigneter Analysesoftware. Aufgrund der großen Bedeutung der Integration in der Praxis wird im nächsten Abschnitt gesondert darauf eingegangen. Es werden mit CORBA und CORBA BOCA Techniken vorgestellt, die das Integrationsproblem in Zukunft erleichtern könnten.

2.2 Das Integrationsproblem

Die in 2.1 erwähnte Integration von Electronic Commerce Systemen in das bestehende Anwendungssystem ist oft schwierig, da die Anwendungen keine standardisierten Schnittstellen haben. Die Kommunikation und der Datenaustausch zwischen Anwendungen wie Online-Shops und ERP-Software ist nicht ohne weiteres möglich. Daher muss eine geeignete Middleware eingesetzt werden, um diese Heterogenität zu überwinden. Das Integrationsproblem könnte in Zukunft durch den Einsatz von sogenannten Business Objects gelöst werden [Merz99b]. In den folgenden Abschnitten werden Business Objects vorgestellt, sowie CORBA und ein Framework namens CORBA BOCA, das eine Standardarchitektur für verteilte Softwaresysteme basierend auf Business Objects vorgibt.

2.2.1 Business Objects

Business Objects sind zunächst anwendungsneutrale Objekte, zum Beispiel Kunden, Adressen, Bestellungen, Mitarbeiter oder Abteilungen, die aus der realen Welt abgeleitet werden. Sie erhalten einen Namen, Attribute, Methoden, zeigen aktiv Verhalten, indem sie auf Events reagieren, und haben Beziehungen zu anderen Business Objects. Geschäftsprozesse können mit Hilfe von Business Objects nachgebildet werden, indem die Beziehungen der Objekte in der realen Welt auf Beziehungen zwischen Business Objects abgebildet werden. Wenn alle Anwendungen eines Unternehmens mit standardisierten Business Objects arbeiten, können leicht Geschäftsprozesse anwendungsübergreifend definiert werden, da die Business Objects gemeinsam von den Anwendungen genutzt werden können.
Durch das Konzept der Business Objects wird es einfach, neue Anwendungen in das bestehende Geschäftssystem zu integrieren, denn die neue Anwendung kann auf schon existierenden und neu eingeführten Business Objects aufgebaut werden. Will zum Beispiel eine Unternehmung einen Online-Shop eröffnen, so kann diese Anwendung bestehende Business Objects wie Kunden, Bestellungen, Rechnungen und deren Funktionen wiederverwenden, um sie auch im Online-Geschäft einzusetzen. Die Integration mit anderen Abteilungen, zum Beispiel der Warenwirtschaft, geschieht automatisch, da Business Objects desselben Typs im Online-Handel und im sonstigen Handel verwendet werden. Bestellungen über das Internet können wie Bestellungen, die auf schriftlichem Wege eingegangen sind, von der Warenwirtschaft bearbeitet werden, wenn sie im Format eines Business Objects Bestellung dort ankommen.

2.2.2 CORBA

CORBA (Common Object Request Broker Architecture) bietet eine Infrastruktur, die es den Business Objects ermöglicht, untereinander und mit den Anwendungen zu kommunizieren [Orfali96, OMG00]. CORBA stellt eine Schnittstellendefinitionssprache namens IDL (Interface Definition Language) zur Verfügung. Mit ihr werden die Schnittstellen der Business Objects implementiert, die eigentlichen Business Objects können in nahezu jeder beliebigen Sprache implementiert sein. Methodenaufrufe an Business Objects werden an einen ORB (Object Request Broker) geschickt. Der ORB hat die Aufgabe, den Aufruf an die richtige Stelle zu delegieren, denn Business Objects können durch den Einsatz von CORBA im System beliebig verteilt sein.
Aufbauend auf der Infrastruktur CORBA gibt es ein Framework namens BOCA (Business Object Component Architecture), das einen Rahmen für die Entwicklung von Business Objects bietet.

2.2.3 CORBA BOCA

Die Bausteine der BOCA sind in Abb.3 dargestellt [Merz99b]. Wie im vorherigen Abschnitt erwähnt bildet CORBA als Infrastruktur die Grundlage. Um Business Objects zu entwickeln, stellt die Business Object Facility (BOF) eine Plattform zu deren Modellierung bereit. Eine Spezifikationssprache namens CDL (Component Definition Language) ermöglicht Schnittstellen und Verhalten eines Business Objects, sowie Beziehungen zwischen Business Objects zu definieren. Die Schnittstellen werden dann wie in 2.2.2 erwähnt in IDL formuliert, der Rest eines Business Objects in einer beliebigen Programmiersprache implementiert.

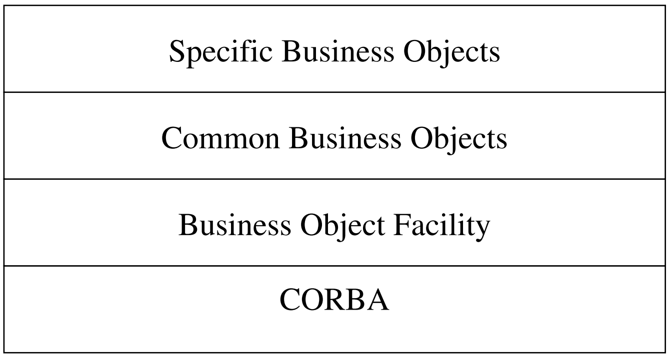

Abb.3 Bausteine der CORBA BOCA

Damit ein Unternehmen auf eine Basis von vordefinierten Business Objects zurückgreifen kann, gibt es die Common Business Objects (CBO). Sie sind standardisiert und stehen in jeder BOCA-Umgebung zur Verfügung. Softwarehersteller haben die Möglichkeit, in Zukunft gemeinsam ein Repertoire an CBOs zu entwickeln, auf denen die Anwendungen aller aufbauen können. CBOs bilden zusammen mit der BOF die Grundlage für Specific Business Objects (SBO). SBOs kann der Anwender selbst mit Hilfe von CDL entwickeln.
CORBA BOCA wurde 1998 veröffentlicht und wird bisher selten umgesetzt, da es bis zum heutigen Zeitpunkt außer wenigen Tools, um aus CDL-Spezifikationen Code zu generieren, kaum Werkzeugunterstützung gibt. Für die Zukunft steckt aber viel Potential in dieser Architektur, da Geschäftsprozesse komfortabel über die Interaktion von Business Objects modelliert werden können, was in einem Bereich wie dem Electronic Commerce von großer Bedeutung ist, da hier Geschäftsprozesse ständig erweitert und modifiziert werden müssen. Vielleicht einigen sich die führenden Softwarehersteller wirklich auf eine umfangreiche Standardbibliothek von CBOs.

2.2.4 Das IBM San Francisco Framework

Ein weiteres Framework, das CORBA BOCA sehr nahe kommt, ist das San Francisco Framework von IBM [Merz99b]. Die Verwandtheit kommt daher, dass IBM am BOCA-Standard mitgearbeitet hat. Dieses Framework basiert allerdings nicht auf CORBA und bietet nicht die Freiheit, eine Programmiersprache zur Implementation der Business Objects zu wählen. Stattdessen wird Java vorgeschrieben. Die Business Objects sollen in Zukunft durch Java Beans realisiert werden [DVDialog99].

Im Folgenden werden nun Architekturvarianten vorgestellt, die im Electronic Commerce eingesetzt werden, und die anhand des Verarbeitungsortes der Applikationslogik unterschieden werden können. Vorher werden allerdings die bisher genannten Kriterien zur Bewertung von Electronic Commerce Systemen zusammengefasst und ergänzt.

2.3 Kriterien zur Bewertung von Electronic Commerce Systemen

Integrierbarkeit: Sie stellt eines der bedeutendsten Kriterien dar, wie schon in 2.2 erläutert wurde. Mit Integrierbarkeit wird die Eigenschaft einer Anwendung bezeichnet, sich leicht in das bestehende Anwendungssystem eines Unternehmens einbauen zu lassen.

Sicherheit: Da bei Geschäftstransaktionen der Kunde vertrauliche Daten wie Kreditkarteninformationen weitergibt, muss eine sichere Übertragung und Speicherung der Daten ermöglicht werden.

Erweiterbarkeit: Ein Unternehmen muss in der Lage sein, sein Electronic Commerce Angebot ständig zu erweitern, um mit dem schnell wachsenden Markt und den rasch wechselnden Anforderungen mithalten zu können.

Verfügbarkeit: Die Kunden verlangen, dass das Angebot eines Unternehmens rund um die Uhr zur Verfügung steht, denn das ist gerade einer der großen Vorteile des Electronic Commerce. Serverausfälle sind aus Sicht des Anbieters auch deshalb katastrophal, da sie mit Umsatzeinbussen verbunden sind.

Skalierbarkeit: Die Leistungsanforderungen sind wie in 1.1 erwähnt sehr hoch und fordern geeignete Maßnahmen, um sie zu bewältigen. Skalierung bezeichnet die Eigenschaft, auch bei hohen Leistungsanforderungen ein akzeptables Leistungsverhalten zu zeigen. Ein Kriterium für die Beurteilung der Skalierbarkeit eines Systems ist die Möglichkeit, dieses in einer verteilten Umgebung einzusetzen. Durch Verteilbarkeit der Komponenten auf verschiedene Rechner kann die Leistungsfähigkeit gesteigert werden, da kein einzelner Rechner zum Flaschenhals wird. Somit sind Replikation und Verteilbarkeit zwei entscheidende Faktoren zur Leistungssteigerung. Wie die Leistungsfähigkeit eines Electronic Commerce Systems gemessen werden kann, wird in Kapitel 3 am Beispiel des TPC-W vorgestellt.

Werkzeugunterstützung: Ein Unternehmen, das in den Electronic Commerce einsteigt, will schnell ein lauffähiges System aufbauen und administrieren können, um vor den Wettbewerbern am Markt zu sein. Das kann durch geeignete Werkzeugunterstützung gewährleistet werden, denn grafische Benutzeroberflächen erlauben es auch unerfahrenen Anbietern, schnell Lösungen zu entwickeln.

2.4 Architekturvarianten von Electronic Commerce Systemen

Datenbanken spielen eine große Rolle im Bereich Electronic Commerce. Wenn man aus dieser Sicht Electronic Commerce Systeme als Datenbankanwendungen ansieht, kann man die verschiedenen Architekturvarianten an einem 3-Schichtenmodell für Datenbankanwendungen aufzeigen. Es gibt in diesem Modell die Präsentations-, Anwendungs- und Datenhaltungsschicht. Als Komponenten der einzelnen Schichten treten in Electronic Commerce Architekturen Web-Browser, Web-Server, Application-Server und DB-Server auf. Die in der folgenden Abbildung gezeigten drei Varianten sind prinzipiell möglich. Der markierte Bereich hebt jeweils die Anwendungsschicht hervor.

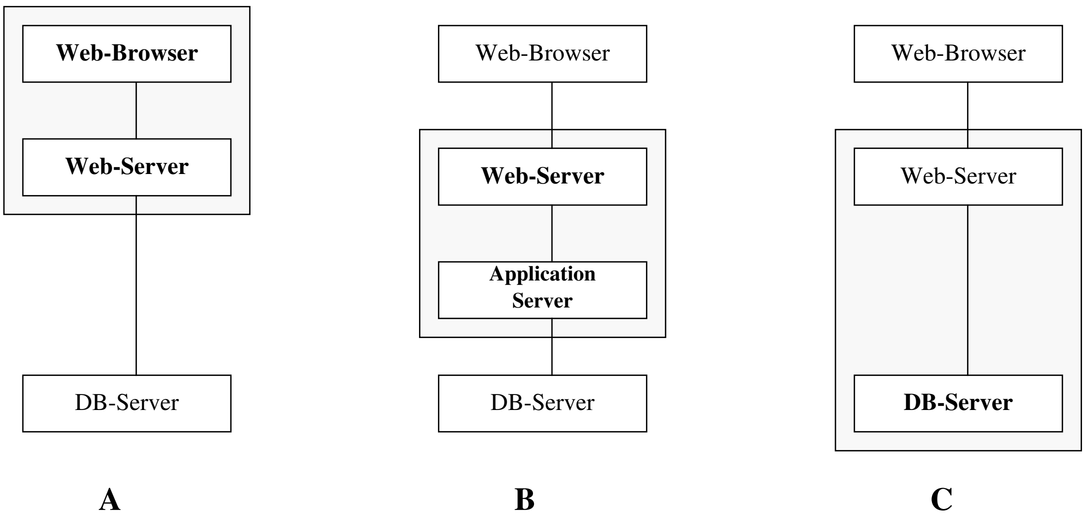

Abb. 4 Architekturvarianten

Die Präsentationsschicht bildet in jedem Fall der Web-Browser, den Zugang zur Datenhaltungsschicht der DB-Server. Interessant ist, welche Komponenten zur Anwendungsschicht zählen. Der Web-Server ist schwer einer Schicht zuzuordnen. Er stellt für den Kunden das Bindeglied zwischen ihm und dem Electronic Commerce Angebot eines Unternehmens dar. Der Web-Server hat unter anderem die Aufgabe, CGI-Programme mit den Eingaben von Kunden zu starten, und die von diesen Programmen dynamisch erstellten HTML-Seiten als Ergebnis an den Web-Browser zu schicken. Die CGI-Programme können als DB-Client fungieren und SQL-Anfragen an den DB-Server stellen. Eine Zuordnung des Web-Servers zur Anwendungsschicht ist vertretbar aber dennoch problematisch, da er eigentlich nur Aufrufe weiterleitet und die Ergebnisse an den Web-Browser des Kunden liefert und keine Applikationslogik verarbeitet.

Im Folgenden werden die in Abbildung 4 dargestellten Architekturvarianten detaillierter mit ihren Vor- und Nachteilen vorgestellt. Sie unterscheiden sich im Verarbeitungsort der Applikationslogik. Zu jeder Architekturvariante werden Anbieter vorgestellt, die sie einsetzen.

3. Anbieter und ihre Electronic Commerce Lösungen

In diesem Kapitel werden die in 2.4 vorgestellten Architekturen genauer erläutert und am Beispiel von Anbietern verdeutlicht. Die Lösungen der Anbieter sollen außerdem anhand der in 2.3 vorgestellten Kriterien untersucht werden.

3.1 Clientseitige Verarbeitung der Applikationslogik

In Teil A der Abbildung 4 liegt der Fall vor, in dem die Applikationslogik im Client integriert ist. Der Einsatz von Java-Applets ermöglicht die Verarbeitung der Applikationslogik im Web-Browser. Es kommt zu einer Überschneidung von Präsentationsschicht und Anwendungsschicht, denn der Web-Browser des Client war zuvor der erstgenannten Schicht zugeordnet worden.
Das Benutzen einer clientseitig ausgeführten Anwendung ist vor dem ersten Einsatz immer mit einem Download verbunden, bzw. die erforderliche Software müsste auf einem anderen Weg erworben werden. Das wiederum wäre mit zeitlichen Verzögerungen verbunden und verwehrt den Kunden die Möglichkeit, ohne eine gewisse Anlaufzeit ein Electronic Commerce Angebot zu nutzen. Allerdings kann dieser Nachteil wieder ausgeglichen werden, denn durch die Verarbeitung auf dem Client wird der Server des Anbieters entlastet und der Kunde profitiert von schnelleren Bearbeitungszeiten. Ein weiterer Vorteil besteht darin, dass persönliche Daten des Kunden bei ihm lokal gespeichert bleiben und verarbeitet werden. Das erhöht das Vertrauen des Kunden in die Sicherheit der Electronic Commerce Anwendung. Ein weiteres Problem dagegen stellen die verschiedenen Plattformen dar, die Kunden prinzipiell nutzen können. Die Anwendung müsste in verschiedenen Versionen verfügbar sein. Dieses Problem besteht nicht, wenn Java zur Entwicklung benutzt wird. Daher bietet Sun die Java Wallet Architektur an, die eine clientseitige Lösung darstellt. Sie wird im Folgenden vorgestellt.

3.1.1 Das Java Wallet von Sun

Sun bietet mit dem Java Wallet [Sun98, Sun00a] eine clientseitig-zentrierte Lösung an. Es bietet keine Komplettlösung, die ein Unternehmen direkt auf seinem System installieren und mit Hilfe einer Benutzeroberfläche konfigurieren kann. Stattdessen erhält der Anbieter die Möglichkeit, seine Electronic Commerce Lösung selbst zu implementieren. Dazu stellt das Java Wallet ein Framework bereit.

3.1.1.1 Das Java Wallet als Framework

Eine clientseitig zentrierte Electronic Commerce Lösung auf Basis des Java Wallet kann unter Ausnutzung der Standard Java APIs und speziellen Java Commerce APIs implementiert werden. Einerseits kann der Anbieter Applets programmieren, die im Browser des Kunden ausgeführt werden. Im Gegensatz zu den nicht dauerhaft im Client gespeicherten Applets können sogenannte Cassettes entwickelt und dauerhaft auf dem Client gespeichert werden, die dann zum Beispiel von Applets genutzt werden. Sie stehen nach einmaligem Download immer direkt auf dem Client zur Verfügung. Cassettes realisieren Komponenten in Form von Java Beans, unter anderem Protokolle, Anwendungen wie Zahlungsverfahren, Benutzeroberflächen oder Zertifikate. Da Cassettes digital signiert sind, kann der Kunde Sicherheit über die Herkunft erlangen. Die Cassettes haben standardisierte Schnittstellen, so dass ihre Wiederverwendung durch andere Anbieter unterstützt wird. Dieser Vorteil kann zum Beispiel bei elektronischen Zahlungsverfahren genutzt werden. Die Cassette zur Realisierung eines solchen Verfahrens kann vom Anbieter eines Zahlungsverfahrens allen zur Verfügung gestellt werden, die es einsetzen wollen. Es gibt bisher keine Schnittstellen zu anderen Systemen, lediglich einige Cassettes, die gängige Zahlungsverfahren implementieren.

Die Funktionsweise einer Electronic Commerce Lösung auf Basis des Java Wallet wird im folgenden Abschnitt am Modell des allgemeinen Geschäftsvorfalls aus 2.1 erläutert.

3.1.1.2 Realisierung eines allgemeinen Geschäftsvorfalls mit dem Java Wallet

Die folgende Grafik verdeutlicht die Funktionsweise von Electronic Commerce Lösungen auf Basis des Java Wallet.

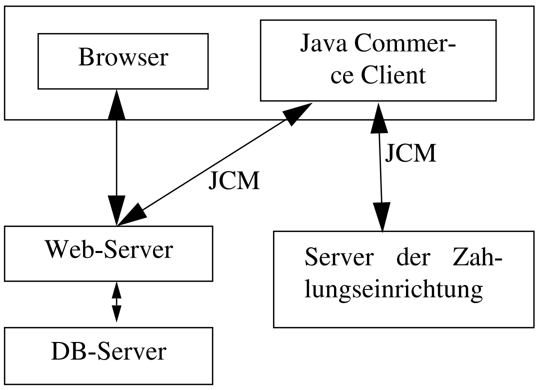

Abb. 5 Electronic Commerce Lösung auf der Basis des Java Wallet

Voraussetzung für einen Kunden ist, dass er den sogenannten Java Commerce Client (JCC) auf seinem System installiert hat. Dieser registriert schon vorhandene Cassettes und verwaltet die Daten seines Anwenders in einer Datenbank, zum Beispiel Adresse und Bankverbindungen, um sie bei Bedarf Autorisierten zur Verfügung zu stellen. Nachdem der Kunde über die Web-Seite des Anbieters den Kontakt hergestellt hat, können noch nicht vorhandene Cassettes wie zum Beispiel eine Benutzeroberfläche vom Anbieter per Download zum Kunden gelangen und im JCC integriert werden. Im JCC kann dann diese anbieterspezifische Benutzeroberfläche zur weiteren Interaktion ausgeführt werden. Die Kommunikation erfolgt über sogenannte Java Commerce Messages (JCM) zwischen Web-Server und JCC. Der Kunde kann nach Auswahl des Produktes den Kauf auslösen. Der Web-Server schickt als Reaktion eine JCM an den JCC des Kunden, die Informationen enthält, um die Abwicklung des Kaufs durchzuführen. Dazu zählen Informationen über die benötigten Cassettes und die möglichen Zahlungsverfahren. Jetzt kann der Download nicht im JCC vorhandener Cassettes erfolgen, die für den Kauf notwendig sind, zum Beispiel eine Cassette, die die Applikationslogik der Transaktion enthält. Die Transaktion wird dann auf dem Client ausgeführt, der Kunde schickt zum Beispiel seine Daten zum Anbieter, und es kann dann der Server einer Zahlungseinrichtung kontaktiert werden, die zur Abwicklung des Kaufes mit dem Anbieter des Produktes zusammenarbeitet.

Im nächsten Abschnitt wird nun die Architekturvariante mit Application-Server zur Verarbeitung der Applikationslogik vorgestellt.

3.2 Application-Server zur Verarbeitung der Applikationslogik

Es wurde bereits erwähnt, dass eine schnelle Transaktionsverarbeitung ein entscheidender Erfolgsfaktor für das Electronic Commerce Angebot eines Unternehmens ist. Daher bietet es sich an, die Anwendungen auf einem Application-Server auszuführen, um durch diese zusätzliche Komponente den Web-Server zu entlasten und kurze Bearbeitungszeiten zu garantieren. Diese Variante ist in Teil B der Abbildung 4 dargestellt. Der Web-Server muss keine Server-Erweiterungen mehr starten, um dynamisch HTML-Seiten zu erzeugen, denn das übernimmt der Application-Server, genauso wie das Öffnen von DB-Verbindungen und das Stellen von SQL-Anfragen. Der Web-Server leitet die Aufrufe der vom Client ausgelösten Transaktionen an den Application-Server weiter. Das kann durch den Einsatz von Servlets geschehen. Der Einsatz von CORBA als Middleware bietet sich ebenfalls an. Sie erlaubt wie RPC den ortstransparenten Aufruf von Methoden auf anderen Rechnern in einem verteilten System. Aufrufende und aufgerufene Methoden können dabei in unterschiedlichen Programmiersprachen implementiert sein (2.2.2). Das wird möglich durch IDL (Interface Definition Language), mit der man eine Schnittstellen für eine Methode schreiben kann, die unabhängig von der verwendeten Programmiersprache ist. Die Aufrufe werden an einen ORB (Object Request Broker) geschickt, der sie dann an die entsprechende Stelle weitervermittelt. Web-Server und Application-Server können über einen ORB kommunizieren.
Die Anwendungen bzw. die Transaktionsprogramme selbst laufen auf dem Application-Server. Damit trägt er die Hauptlast der Applikationslogik. Zur weiteren Leistungssteigerung bietet es sich an, Web- und Application-Server oder auch mehrere Application-Server Instanzen physisch zu verteilen, um einen Rechner nicht zu sehr zu belasten.
Die gerade vorgestellte Architekturvariante mit Application-Server ist für Electronic Commerce Systeme die bedeutendste. Die schon erwähnten Marktführer Intershop und BEA setzen Architekturen mit Application-Server ein. Im Folgenden wird das Angebot von INTERSHOP etwas genauer vorgestellt.

3.2.1 INTERSHOP Enfinity

Intershop ist Anbieter einer Komplettlösung für ein Electronic Commerce System [Inter99]. Web-Server und DB-Server müssen von einem anderen Anbieter erworben werden, ansonsten bietet Intershop eine Application-Server-basierte Electronic Commerce Lösung an, beginnend bei Servern, um Transaktionen zu verarbeiten, über Tools, um das System zu konfigurieren und zu administrieren, bis zu Zusatzmodulen für Marketing und Kundendatenanalyse. Intershop wirbt mit drei Prinzipien, die die entscheidenden Faktoren im Electronic Commerce sein sollen [Inter99]: Das erste lautet „Sell anywhere“. Ein Anbieter soll die Möglichkeit haben, seine Produkte überall im Internet zu verkaufen, das heißt sowohl auf seiner eigenen Web-Seite als auch auf fremden Seiten und im B2B-Bereich. Das zweite Prinzip lautet „Integrate everything“. Dadurch soll ausgedrückt werden, welch große Bedeutung der Integration der Electronic Commerce Lösung in das bestehende Geschäftssystem der Unternehmung gegeben wird. Mit dem Prinzip „Today and tomorrow“ wird auf die Erweiterbarkeit hingewiesen, die von einem Electronic Commerce System erwartet wird, um den zukünftigen Anforderungen des Marktes gewachsen zu sein.

3.2.1.1 Architektur von INTERSHOP Enfinity

Abb.6 zeigt die Architektur von INTERSHOP enfinity [Inter99]. Die realisierte Variante benutzt wie oben erwähnt einen Application Server. Dieser bietet eine Ausführungsumgebung für den Catalogue-Server und den Transactivity-Server. Die möglichen Transaktionen werden in zwei Gruppen aufgeteilt. Transaktionen, die Funktionen des Produktkatalogs nutzen, werden vom Catalogue-Server verarbeitet, alle anderen Transaktionen vom Transactivity-Server. Der Web-Adapter ist eine Erweiterung des Web-Servers, die dafür verantwortlich ist, einen Transaktionsaufruf an den zuständigen Server weiterzuleiten.

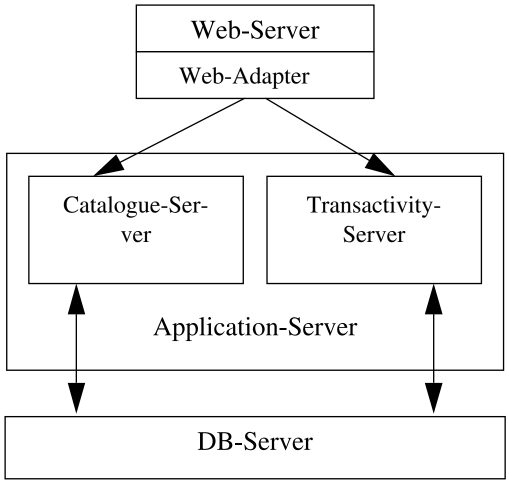

Abb. 6 Intershop Architektur

Die Architekturvariante von INTERSHOP nutzt die in 3.2 genannten Vorteile des Einsatzes eines Application-Servers, zum Beispiel die Entlastung des Web-Servers. Durch die Aufteilung in Catalogue-Server und Transactivity-Server kann eine Leistungssteigerung im Bereich der Abwicklungsphase erreicht werden, da der Transactivity-Server nicht von Kunden belastet wird, die sich in der Informationsphase befinden und den Produktkatalog durchsuchen. Da die Informationsphase lang andauern kann und mit vielen Transaktionen verbunden ist, kann dadurch eine hohe Belastung des Application-Servers entstehen, die mit der INTERSHOP Variante verteilt wird.

3.2.1.2 Transaktionen

Transaktionen werden bei INTERSHOP auf sogenannte Pipelines abgebildet, die wiederum aus Pipelets aufgebaut werden. Ein Repertoire an Pipelets ist bereits im Lieferumfang enthalten, zum Beispiel zur Führung und Manipulation eines Warenkorbs oder des Produktkatalogs. Weiterhin stehen Tools zur Verfügung, um Pipelets selbst in Java zu entwickeln. Die vorgefertigten Pipelets und die selbst in Java entwickelten können dann mit einem weiteren Tool, dem Visual Pipeline Manager, per Drag&Drop zu Pipelines zusammengesetzt werden und realisieren so Transaktionen. Auf diese Weise ist es für ein Unternehmen möglich, schnell und effektiv unter Nutzung der vorhandenen Pipelets die in einem Online-Shop nötigen Transaktionen aufzubauen, da die vorhandene Benutzeroberfläche leicht zu bedienen ist und zunächst keine Pipelets implementiert werden müssen. Das Kriterium Werkzeugunterstützung ist durch den Visual Pipeline Manager für den Bereich Transaktionen erfüllt.

3.2.1.3 Integration

INTERSHOP enfinity bietet Softwareherstellern eine sogenannte Cartridge API an, um Schnittstellen zu ihren Produkten zu implementieren. Aus dem Bereich der ERP-Software existiert unter anderem eine Schnittstelle zu SAP R/3, aus dem Bereich der Zahlungsverfahren zu CyberCash. Produkte, zu denen Schnittstellen existieren, können mit Hilfe des Visual Pipeline Managers integriert werden, da für diese dann Pipelets zur Anbindung existieren. Die Integration eines Electronic Commerce Systems in das bestehende Geschäftssystem scheint damit einfach, wenn es eine Schnittstelle zu den bereits benutzten Produkten gibt. Ansonsten besteht zumindest die Möglichkeit, mit Hilfe der Cartridge API selbst die Integration vorzunehmen.

3.2.1.4 Skalierbarkeit und Verfügbarkeit

Laut [Inter99] sind alle Komponenten der INTERSHOP Architektur aus Abb. 6 replizierbar und verteilbar. Das trägt zur Erfüllung des Kriteriums der Skalierung bei, denn es bedeutet, dass je nach Auslastung des Systems weitere Catalogue-Server und Transactivity-Server gestartet werden können, um für Entlastung der anderen Server zu sorgen, und dass diese zusätzlichen Server auf anderen Rechnern laufen können. Durch Replikation von Komponenten kann außerdem Redundanz eingeführt werden, um die unter den Kriterien geforderte Verfügbarkeit zu sichern.

3.2.2 Electronic Commerce Lösung von BEA

BEA bietet ebenfalls eine Electronic Commerce Lösung an, in deren Architektur ein Application-Server eingesetzt wird. Die Aufteilung in Catalogue-Server und Transactivity-Server wird nicht vorgenommen, ansonsten sind sich die Architekturen sehr ähnlich. Ein Überblick über die BEA-Lösung kann in [BEA99] gewonnen werden. BEA wird unter anderem von Amazon (1.2) eingesetzt.

3.2.3 Informix i.Sell

Informix i.Sell bietet ebenfalls Anwendungen zur Unterstützung eines Geschäftsvorfalls, wie er in 2.1 vorgestellt wurde. Unter anderem werden die Personalisierung von Web-Angeboten, die Abwicklung von Bestellungen, die Bezahlung und die Analyse von Kundendaten ermöglicht [Informix99b]. Informix setzt dabei wie BEA und INTERSHOP eine Architektur mit Application-Server ein.

3.2.4 Electronic Commerce Produkte von Oracle

Oracle bietet eine Suite von Produkten zur Unterstützung eines allgemeinen Geschäftsvorfalls an. Oracle iStore ermöglicht die Erstellung eines Online-Shops mit Personalisierung von Angeboten. Dazu gibt es noch Erweiterungen wie Oracle iPayment, das Zahlungsverfahren zur Verfügung stellt [Oracle99]. Die Produkte setzen den Einsatz eines Oracle Application-Servers voraus, bei iStore ist dieser im Lieferumfang enthalten. Das deutet auf eine Architektur hin, wie sie in Abb. 4 B dargestellt ist. Die technologischen Möglichkeiten von Oracle 8i erlauben aber eine enge Kopplung von Application Server und Datenbanksystem.

Der nächste Abschnitt beschäftigt sich nun mit der dritten möglichen Architekturvariante für Electronic Commerce Systeme, wie sie in Abb. 4 C dargestellt ist.

3.3 Integration der Applikationslogik im DB-Server

Es besteht wie in Abb. 4 C dargestellt die Möglichkeit, Applikationslogik im DB-Server zu integrieren. Das wird durch die in SQL3 eingeführten „stored procedures“, bzw. UDFs (*User Defined Function*) möglich. Sie bestehen aus SQL-Befehlen oder prozeduraler Logik, werden mit einem Namen versehen und neben den Daten in der Datenbank gespeichert. Nach ihrem Aufruf werden sie auf dem DB-Server ausgeführt. Diese Ausführung nahe an den Daten hat den Vorteil, dass weniger Kommunikation zwischen Anwendungen und dem DB-Server nötig ist. Es muss nicht jeder SQL-Aufruf einzeln an den DB-Server geschickt werden, sondern immer wieder auftretende Folgen von SQL-Befehlen, die sich nur in Parametern unterscheiden, können zusammen mit der dazugehörigen Applikationslogik in Prozeduren zusammengefasst werden. Nur der Prozeduraufruf mit entsprechenden Parametern muss dann an den DB-Server geschickt werden. Ein Beispiel ist die Personalisierung einer HTML-Seite, nachdem sich der Kunde identifiziert hat. Es könnte eine UDF aufgerufen werden, die als Parameter die Identifikation des Kunden erhält und aus den zu diesem Kunden gespeicherten Profilinformationen eine personalisierte Seite generiert.
Bisher stellt es ein Problem dar, dass es keine einheitliche Sprache gibt, mit der man UDFs für ein Datenbanksystem eines beliebigen Herstellers implementieren konnte. Man muß die Sprache und API benutzen, die der jeweilige Hersteller vorgeschrieben hat. Java und JSQL (Java-SQL) könnten sich in diesem Bereich als Standard durchsetzen und dieses Problem lösen. Die Bemühung gehen in die Richtung, die Ausführung beliebiger Java-Funktionen und sogar ganzer Java-Applikationen in das Datenbanksystem zu integrieren. Denkbar ist die Integration ganzer Server-Dienste wie Web- oder Application-Server in den DB-Server. Trotz einiger Bemühungen wird in heutigen Electronic Commerce Systemen aufgrund nicht verfügbarer Produkten, die Applikationslogik nicht in den DB-Server integriert, sondern höchstens eine Datenbanknahe Ausführung erreicht. Ein Ansatz zur Intergration der Applikationslogik in den DB-Server wird im nächsten Abschnitt mit dem Informix Web DataBlade vorgestellt.

3.3.1 Informix Web DataBlade Module

Informix bietet zum Informix Dynamic Server 2000, einem DB-Server, als Erweiterung das Web DataBlade Modul an, mit dem man DB-Anwendungen für das Web entwickeln kann, also auch Anwendungen im Bereich Electronic Commerce [Informix99a].
Als Web-Server werden Produkte von Netscape, Microsoft und Apache vorgeschlagen, es kann aber auch ein beliebiger anderer Web-Server benutzt werden. Der Web-Server wird um ein Modul namens Webdriver erweitert, der DB-Server neben anderen um eine stored procedure namens WebExplode(). Abb. 7 zeigt die Architektur im Überblick.

Die Funktionsweise soll an einem Beispiel verdeutlicht werden. Die Seite eines Anbieters soll für einen speziellen Kunden personalisiert werden. Der Kunde identifiziert sich und fordert seine persönliche Seite an. Der Webdriver ruft die WebExplode() Funktion auf und übergibt als Parameter die Kundendaten und die Identifikation einer sogenannten Application Page, die als Template für die persona-

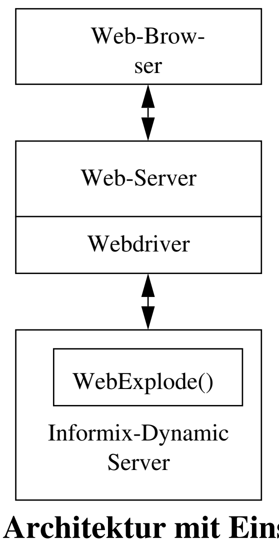

Abb. 7 Architektur mit Einsatz des Web-Data Blade Moduls von Informix

lisierte Seite dient. Die WebExplode() Funktion holt die angeforderte Application Page aus der Datenbank, führt die enthaltenen SQL-Befehle in den Tags aus und bildet aus den Ergebnissen die personalisierte HTML-Seite, die sie dann an den Webdriver zurückliefert. Der wiederum übergibt sie dem Web-Server und der letztendlich dem Web-Browser zur Darstellung. Mit Hilfe der Tags können unter anderem Bedingungen geprüft werden und Variable in der Datenbank manipuliert werden, zum Beispiel um den Zustand eines Warenkorbes zu speichern, den ein Kunde gerade füllt.

Durch die Benutzung des Web DataBlade können also Anwendungen entwickelt werden, deren Logik in der Informix Datenbank von der WebExplode() Funktion ausgeführt wird. Wir haben es daher mit der in 3.3 bzw. Abb. 4 C vorgestellten Architekturvariante zu tun. Es ist allerdings schwierig, mit dem Web DataBlade eine komplette Electronic Commerce Anwendung wie einen Online-Shop zu realisieren, denn es muss von Grund auf alles neu entwickelt werden. Dabei ist es nicht leicht, Zahlungsmöglichkeiten zur Verfügung zu stellen, da das Web DataBlade keine Sicherheitsmechanismen dazu bereitstellt.

Zum Abschluss meiner Ausarbeitung möchte ich im folgenden Abschnitt kurz den Web-Benchmark TPC-W vorstellen, der hilft, die Leistungsfähigkeit eines Electronic Commerce Systems zu beurteilen.

4. Der Electronic Commerce Web Benchmark TPC-W

Der TPC-W ist ein Benchmark zur Messung der Leistung eines Electronic Commerce Systems. Er wurde von den Mitgliedern des TPC (Transaction Processing Performance Council) entwikkelt, zu denen zum Beispiel BEA, Oracle oder Informix gehören.
Der TPC-W Benchmark simuliert einen Online-Shop. Kunden können Informationen zu Büchern suchen und diese bestellen. Die dazu notwendigen Funktionen müssen im zu testenden System implementiert sein. An die interne Realisierung des zu testenden Systems werden keine Anforderungen gestellt, außer dass nur die Möglichkeiten von HTML genutzt werden können, um die Transaktionen auszulösen, Java Applets können nicht genutzt werden. Die Kunden werden in Form von Browsern simuliert, die sogenannte Web Interactions auslösen. Die Auswahl der nächsten auszulösenden Web Interaction erfolgt zufällig durch Generierung einer Zufallszahl. In Abhängigkeit von der Zufallszahl und der zuletzt ausgelösten Web Interaction wird aus einer fest vorgegebenen Tabelle die nächste Web Interaction ausgewählt. Als Netzwerkprotokoll zur Kommunikation mit dem Web-Server wird TCP/IP verlangt, um Internetbedingungen zu schaffen.

In den folgenden Abschnitten werden einige Details zu den Metriken, dem Transaktionsmix und der Dokumentation aus der Spezifikation des TPC-W [TPC99] vorgestellt.

4.1 Metriken

Es gibt zwei primäre Metriken. Die Leistung wird durch WIPS (Web Interactions Per Second) gemessen. Eine Web Interaction dauert vom Auslösen einer Transaktion im Browser bis zum Erhalt des Ergebnisses vom Web-Server. Außerdem wird eine Maßzahl für den Quotienten Preis/Leistung angegeben. Dazu muss der Softwareanbieter den Preis des Electronic Commerce

Systems angeben, der sich aus Hardwarekosten, Softwarekosten und Wartungskosten für die ersten drei Jahre zusammensetzt. Außer den sogenannten primären Metriken WIPS und Preis/Leistung gibt es noch die sekundären Metriken WIPSB (Web Interactions Per Second Browsing) und WIPSO (Web Interactions Per Second Ordering). Mit diesen weiteren Metriken soll die Reaktion des Systems auf unterschiedliches Lastverhalten gemessen werden. Zur Leistungsmessung wird ein Transaktionsmix aus den Bereichen Browsing und Bestellung zusammengestellt, der für die Messung von WIPS, WIPSB und WIPSO jeweils unterschiedlich ist.

4.2 Transaktionsmix

Der Transaktionsmix zur Messung von WIPS setzt sich zu 80 Prozent aus Browsing-Transaktionen und 20 Prozent aus Bestelltransaktionen zusammen. In den Browsing-Bereich fallen das Blättern im Produktkatalog und das Suchen nach konkreten Produkten. Die 20 Prozent aus dem Bestellbereich setzen sich aus Transaktionen zum Führen eines elektronischen Warenkorbes, Benutzerregistrierung und dem Kauf zusammen. In jedem Bereich ist exakt festgelegt, welche Transaktionen das System in seinem Funktionsumfang haben muss. Die Aufteilung des Transaktionsmixes ist dadurch zu erklären, dass wie schon in 2.1 erwähnt die Informationsphase, zu denen die Browsing-Transaktionen hauptsächlich zählen, längere Zeit in Anspruch nimmt und mit vielen Anfragen des Kunden an die Produktdatenbank verbunden ist. Die zur Bestellung notwendige Anzahl von Transaktionen ist dagegen eher gering, wodurch der Anteil von 20 Prozent zustande kommt. Zur Messung von WIPSB wird ein Transaktionsmix verwendet, der sich zu 95 Prozent aus Browsing-Transaktionen zusammensetzt, zur Messung von WIPSO beträgt dieser Anteil nur 50 Prozent. Durch diese Variation soll wie in 4.1 erwähnt das Lastverhalten des Systems gemessen werden, wenn Browsing-Transaktionen oder Bestelltransaktionen einen überdurchschnittlich hohen Anteil erreichen.

4.3 Dokumentation der Messung und Auditing

Der TPC-W schreibt vor, wie die Durchführung einer Messung dokumentiert werden muss, um die Ergebnisse jederzeit nachvollziehen zu können. Ein solcher Report enthält nicht nur die gemessenen Werte der in 4.1 vorgestellten Metriken, sondern auch detaillierte Angaben über die eingesetzte Hardware und die Implementierung. Der Report muss außerdem von einem unabhängigen Auditor überprüft werden.

4.4 Beurteilung

Abschließend soll der TPC-W noch kurz beurteilt werden. Ein Kritikpunkt ist, dass Java Applets zur Implementierung nicht eingesetzt werden dürfen, obwohl sie in der Praxis häufig Anwendung finden. Es muss angezweifelt werden, ob der Benchmark alle Anforderungen an ein Electronic Commerce System in der Realität wiederspiegelt, wenn er Java-Applets nicht zulässt. Den in 2.1 vorgestellten Geschäftsvorfalls spiegelt der Benchmark dagegen gut wieder. Die Informationsphase wird durch die Browsing-Transaktionen und die Manipulation des Warenkorbes abgebildet, Verhandlungsphase und Abwicklungsphase durch die restlichen Transaktionen. Ein Anbieter eines Zahlungsservice wird von außerhalb über einen Payment Gateway zugeschaltet, wie es auch in der Praxis die Regel ist.
Ein weiterer Kritikpunkt ist, dass der TPC-W nicht die Möglichkeit bietet, die Leistung der in 1.1.2 vorgestellten Auktionssysteme und Agentensysteme zu messen. Der zunehmende Einsatz dieser neuen Kollaborationsformen erfordert aber eine Möglichkeit, auch solche Systeme zu bewerten. Es ist deshalb wünschenswert, dass der Funktionsumfang des TPC-W in Zukunft so er-

weitert wird, dass auch Auktionssysteme und Agentensysteme getestet werden können.
Sun arbeitet derzeit an einem Benchmark namens ECperf. Über den aktuellen Stand kann man sich unter [Sun00b] informieren.

5. Zusammenfassung

In der Einleitung der Ausarbeitung wurde zunächst versucht, anhand des Referenzmodells von Merz [Merz99a] einen Überblick des Gebietes Electronic Commerce zu geben. Die Grundlage bilden sogenannte Basistechnologien, die eingesetzt werden, um auf höheren Ebenen Anwendungen zu realisieren. Es wurde auch auf interessante Entwicklungen im Electronic Commerce Bereich wie neue Kollaborations- und Organisationsformen eingegangen (1.1.2, 1.1.3). Im weiteren Verlauf der Einleitung wurden mit Business-to-Business- und Business-to-Consumer-Commerce die beiden bedeutendsten Ausprägungen des Electronic Commerce vorgestellt.

Ab dem 2. Kapitel wurde dann eine Einschränkung auf den Bereich Business-to-Consumer-Commerce vorgenommen. Das Modell eines allgemeinen Geschäftsvorfalls bildete den Einstieg in den Bereich von Electronic Commerce Systemen und Architekturen. Ein Abschnitt wurde dem Integrationsproblem gewidmet, um der großen Bedeutung der Einbettung von Electronic Commerce Anwendungen in das bestehende Anwendungssystem eines Unternehmens Rechnung zu tragen.

Nach der Vorstellung allgemeiner Architekturvarianten wurde dann im 3. Kapitel detaillierter auf diese eingegangen. Es traten drei Varianten auf, die anhand des Verarbeitungsortes der Applikationslogik unterschieden wurden. Die Applikationslogik kann wie im Beispiel Java Wallet [Sun98, Sun00a] im Client integriert werden, in einem Application Server wie bei INTERSHOP Enfinity [Inter99] oder in der Datenbank, wie beim Informix Web DataBlade [Informix99a]. Es stellte sich hier heraus, dass die Variante mit Application-Server in der Praxis die am häufigsten eingesetzte ist. INTERSHOP, BEA, Informix und Oracle setzen Architekturen mit Application-Server ein.

Im 4. Kapitel wurde mit dem TPC-W ein Benchmark zur Leistungsmessung von Electronic Commerce Systemen vorgestellt. Anbieter können mit ihrer Electronic Commerce Lösung einen Shop nach den Vorgaben des Benchmarks entwickeln und Messungen durchführen. Die Ergebnisse können potentiellen Kunden angeboten werden. Die Spezifikation des Benchmark schreibt dabei genau vor, welche Funktionen eines Shops implementiert sein müssen, und wie sich die simulierten Kunden verhalten.

Referenzen

[BEA99] *„BEA Electronic Commerce Transaction Platform"*, White Paper, 1999, http://www.beasys.com

[CompZei99] *„Ministerrat beschließt Regeln für E-Commerce"*, Computer Zeitung Nr.50, 16. Dezember 1999

[DVDialog99] *„Ein erneutes Redesign in Richtung Javabeans"*, DVDialog 12/99

[Informix99a] *„Creating Web-Enabled Database Applications Using the Informix Web Data-BladeModule"*, Informix White Paper, 1999, http://www.informix.com

[Informix99b] *„Informix i.Sell Data Sheet"*, 1999, http://www.informix.com

[Inter99] *„INTERSHOP Enfinity"*, White Paper, 1999, http://www.intershop.de

[Merz99a] Michael Merz, *„Electronic Commerce-Technologische und organisatorische Grundlagen"*, Informatik Spektrum, Oktober 1999

[Merz99b] Michael Merz, *„Electronic Commerce"*, dPunkt Verlag, 1999

[OMG00] Object Management Group, http://www.omg.org/corba/beginners.html

[Oracle99] Oracle Electronic Commerce, http://www.oracle.com/ecommerce/index.html

[Orfali96] Orfali, Harkey, Edwards, *„The Essential Client/Server Survival Guide"*, Wiley Computer Publishing, 1996

[Power00] Powershopping, http://www.powershopping.de

[Sun98] *„The Java Wallet Architecture"*, White Paper, 1998, http://www.java.sun.com/products/commerce

[Sun00a] Java Wallet Architecture FAQ, http://java.sun.com/products/commerce/faq.ht-ml#javawallet

[Sun00b] SunECperf, März 1999 http://java.sun.com/aboutJava/communityprocess/jsr/jsr_004_ecperf.html

[TPC99] *„TPC-W Benchmark Draft Specification"*, 1999, http://www.tpc.org/miscelaneous/TPC_W.folder/TPCW12.15.97/