

September 2003

Implementierung von Kommunikationsdiensten zur Kopplung einer Verhandlungsanwendung an elektronische Marktplätze und ERP-Systeme

Michael Rebstock

Fachhochschule Darmstadt, rebstock@uni-koblenz.de

Michael Lipp

Fachhochschule Darmstadt

Follow this and additional works at: <http://aisel.aisnet.org/wi2003>

Recommended Citation

Rebstock, Michael and Lipp, Michael, "Implementierung von Kommunikationsdiensten zur Kopplung einer Verhandlungsanwendung an elektronische Marktplätze und ERP-Systeme" (2003). *Wirtschaftsinformatik Proceedings 2003*. 19.
<http://aisel.aisnet.org/wi2003/19>

This material is brought to you by the Wirtschaftsinformatik at AIS Electronic Library (AISeL). It has been accepted for inclusion in Wirtschaftsinformatik Proceedings 2003 by an authorized administrator of AIS Electronic Library (AISeL). For more information, please contact elibrary@aisnet.org.

In: Uhr, Wolfgang, Esswein, Werner & Schoop, Eric (Hg.) 2003. *Wirtschaftsinformatik 2003: Medien - Märkte - Mobilität*, 2 Bde. Heidelberg: Physica-Verlag

ISBN: 3-7908-0111-9 (Band 1)

ISBN: 3-7908-0116-X (Band 2)

© Physica-Verlag Heidelberg 2003

Implementierung von Kommunikationsdiensten zur Kopplung einer Verhandlungsanwendung an elektronische Marktplätze und ERP-Systeme

Michael Rebstock, Michael Lipp

Fachhochschule Darmstadt

Zusammenfassung: Wir berichten von der Implementierung netzbasierter Kommunikationsdienste zur Kopplung einer Verhandlungsanwendung an elektronische Marktplätze und betriebliche Anwendungssysteme (ERP-Systeme). Dazu werden zunächst konzeptionelle und technische Grundlagen hinsichtlich Web Services, elektronischer Märkte und Verhandlungen erläutert. Die den Anwendungskomponenten zugrunde liegende Gesamtanwendungsarchitektur wird vorgestellt. Anschließend wird das Transaktionsszenario der Dienste aufgezeigt und ein Überblick über deren Zusammenspiel gegeben. Danach werden ausgewählte implementierte Web Services der Anwendung näher dargestellt. Diskussion und Ausblick schließen den Beitrag ab.

Schlüsselworte: Web Services, Elektronische Verhandlungen, Elektronische Märkte, Elektronische Kataloge, Verhandlungsunterstützungssysteme, Enterprise Application Integration

1 Einleitung

Der derzeitige Funktionsumfang elektronischer Marktplatzanwendungen - in der Regel katalogbasierte Beschaffung, einattributive Auktionen und Ausschreibungen - ist aus Sicht der Unternehmenspraxis oft unbefriedigend, da mit ihm nur ein kleinerer Teil des Gesamtbeschaffungsvolumens abgedeckt werden kann. Weitergehende Verhandlungsfunktionalitäten fehlen, obwohl elektronische Verhandlungen bereits seit einigen Jahren Gegenstand von Forschungsprojekten [etwa BiK100, Bic+98, deP+01, Lee+97, Ste+99, Tei+99, Tsv+97] als auch von kommerziellen Entwicklungsprojekten [etwa Comm02; IBM02; Orac02] sind.

In diesem Beitrag berichten wir ausgewählte Aspekte der Implementierung von Kommunikationsdiensten zur Kopplung einer Verhandlungsplattform an elektronische Marktplatzanwendungen und betriebliche Anwendungssysteme (ERP-Systemen). Zielsetzung im Projektgesamtzusammenhang ist ein Abbau der beschriebenen Funktionalitätsdefizite elektronischer Marktplätze durch die Entwick-

lung von Anwendungskomponenten für elektronische Verhandlungen. Die realisierten Lösungen zielen nicht nur auf eine Steigerung der Effizienz der abgebildeten Verhandlungsprozesse ab, sondern auch auf die Erhöhung der Qualität der dabei ausgetauschten Informationen. Dazu ist ein Rückgriff auf Standardisierungskonzepte sowie die enge Integration der Verhandlungslösung in die übergeordnete Informationskette - eben auch unter Einschluss der internen operativen Systeme der beteiligten Unternehmen - von Bedeutung. Beide Aspekte werden bei der Konzeption der Dienste berücksichtigt.

Der Beitrag gliedert sich wie folgt: Im Anschluss an die Einleitung stellen wir zunächst einige Grundlagen im Hinblick auf netzbasierte Kommunikationsdienste, elektronische Märkte und Verhandlungen dar. Daraufhin werden die für die Konzeption der Kommunikationsdienste relevanten Rahmenbedingungen und das Anwendungsszenario eingeführt. Die implementierten Kommunikationsdienste werden in einer Gesamtübersicht vorgestellt, bevor zwei ausgewählte Dienste detaillierter beschrieben werden. Diskussion und Ausblick schließen den Beitrag ab.

2 Grundlagen

2.1 Netzbasierte Kommunikationsdienste

Neuartige Kommunikationsdienste, die auf aktuellen Internet-Standards und Web-Technologien basieren, sind unter dem Begriff *Web Services* bekannt geworden. Ein Web Service ist im weitesten Sinne ein Datendienst, der einem Service Requester von einem Service Provider offeriert wird. Innerhalb der Web-Service-Architektur erfolgt dabei die Vermittlung in der Regel über einen Service Broker. Web Services nutzen die vom World Wide Web Consortium [W3C02a] definierten Basistechnologien *XML* [W3C02b], *SOAP* [W3C02d] und *WSDL* [W3C02c, Chr+01] sowie den Verzeichnisdienst *UDDI* [UDDI02]. Als Transportinfrastruktur dienen die Internetdienste *HTTP*, *SMTP* oder *FTP* über *TCP/IP*. In Bezug auf Netzinfrastruktur, Transportprotokoll und Datenaustauschformat greifen Web Services damit auf vorhandene Technologien zurück. Einer der Hauptaspekte der Entwicklung von Web Services ist die Integration heterogener Anwendungslandschaften [Stal02].

Obwohl das Forschungs- und Entwicklungsfeld der Web Services sich erst stabilisiert, kristallisieren sich bereits einige Aspekte heraus, die allgemein als Merkmale von Web Services verstanden werden [Fre+02, Kreg01, Glas00]:

- **Modularität und Interoperabilität:** Web Services erfüllen eine bestimmte Aufgabe bzw. Aufgabenmenge und sind eigenständig oder in Kombination mit anderen Web Services einsetzbar, um auch komplexere Transaktionen auszuführen

ren. Einzelne Web Services lassen sich miteinander zu einem neuen Dienst kombinieren. Dieser wiederum tritt über seine Schnittstellendefinition nach außen hin als eigenständiger Service auf. (Die Datenkommunikation über Netzwerke legt dabei aufgrund des großen Kommunikationsaufwands ein grobgranulares Applikationsdesign, d.h. größere Informationspakete je Methodenaufruf, nahe.)

- Interpretierbarkeit durch strukturierte Beschreibung: Die vollständige Beschreibung eines Dienstes, seiner Autoren und der benutzten Schnittstellen liefert alle Informationen, die für seinen Aufruf erforderlich sind.
- Plattformunabhängigkeit: Aufgrund der textbasierten Nachrichten des Austauschprotokolls SOAP und der TCP/IP-basierten Transportinfrastruktur sind Web Services weitestgehend unabhängig von der eingesetzten Hardware und Software.

2.2 Elektronische Märkte und Verhandlungen

Elektronische Märkte stellen zentrale Elemente der elektronischen Geschäftsabwicklung dar. Ein elektronischer Markt ist eine auf elektronischen Kommunikationsdiensten beruhende Anwendung, die eine marktmäßige Koordination wirtschaftlicher Leistungen ermöglicht [ähnlich Schm93]. Dabei unterstützen elektronische Märkte im engeren Sinne alle, elektronische Märkte im weiteren Sinne auch nur einzelne Phasen einer Markttransaktion [Schm93]. Meist werden in diesem Zusammenhang vier [Schm99] bzw. fünf [Rebs00] Phasen unterschieden:

- Wissens- bzw. Informationsphase: Dient der Akquisition des Wissens, das für die Durchführung einer Transaktion notwendig ist.
- Absichts- bzw. Entscheidungsphase: Führt zum Entschluss, ob und in welcher Weise ein Gebot gegenüber Marktpartnern abgegeben wird.
- Vereinbarungsphase: Zielt auf die Einigung der Marktpartner und den Abschluss eines Kontrakts ab.
- Abwicklungsphase: Beinhaltet die Erbringung der im Kontrakt beschriebenen Leistungen.
- Betreuungsphase: Umfasst Betreuungs- und Serviceleistungen nach Erbringung der im Kontrakt beschriebenen Leistungen.

Der allgemeine Begriff des elektronischen Marktes ist vom spezifischeren des Marktplatzes abzugrenzen [Rebs00]. Das Bild eines Marktplatzes impliziert das Zusammentreffen vieler Anbieter mit vielen Nachfragern. Ein elektronischer Marktplatz bringt damit die Angebote mehrerer Lieferanten und die Nachfrage mehrerer Kunden an einem virtuellen Ort zusammen. Elektronische Marktplätze sind technisch und organisatorisch intermediäre Applikationen, die im Unter-

schied zu unternehmenszentrierten Anwendungen in der Regel als eigenständige Plattformen von Dritten betrieben werden.

Elektronische Verhandlungen sind mögliche Prozesse innerhalb der Vereinbarungsphase einer elektronischen Markttransaktion. Eine elektronische Verhandlung ist ein auf gegenseitigem Informationsaustausch beruhender Prozess der Entscheidungsfindung zwischen zwei oder mehr Parteien, der eine Einigung (in Form eines Kontraktes) zum Ziel hat [ähnlich DaSm83; GuMa98; Zarn99]. Der Prozess besteht in der Regel aus einer Abgabe von Geboten und Gegengeboten so lange, bis die Einigung erzielt oder er von mindestens einem der beteiligten Partner abgebrochen wurde. Die Zahl der am Verhandlungsprozess beteiligten Partner sowie die logischen und temporalen Prozessstrukturen hängen vom gewählten Verhandlungsprotokoll ab.

Der vorliegende Beitrag entstand im Rahmen eines Projektes, das die Analyse, Konzeption und prototypische Realisierung von Anwendungskomponenten zur elektronischen Unterstützung *interaktiver bilateraler multi-attributiver Verhandlungen* in einem *interorganisationalen Kontext* zur Aufgabe hat [Mult03].

Beispielhafte Einsatzgebiete der im Rahmen des Projekts entwickelten Anwendungskomponenten sind komplexere Verhandlungen über:

- Konfigurierbare und semistrukturierte Produkte, wie etwa in der IT- (Informationstechnologie-), der Automobil-, der Chemie- oder der Nahrungs- und Genussmittelindustrie,
- Rahmenverträge sowie Dienstleistungsverträge aller Art,
- Verträge über kombinierte Güter- und Dienstleistungsbündel, wie etwa in der IT-Branche (Hardware, Software, Services) oder im Maschinen- und Anlagenbau.

Damit werden elektronische Verhandlungen mit einer von der Großzahl anderer Beiträge abweichenden, auf Unternehmensanwendungen fokussierenden Schwerpunktsetzung betrachtet:

Bilaterale Verhandlungen sind im Unternehmensbereich der Normalfall (was parallele bilaterale Verhandlungen mit mehreren Geschäftspartnern nicht ausschließt). Die Mehrzahl der Ansätze zu elektronischen Verhandlungen befasst sich jedoch nicht mit bilateralen, sondern mit einseitig oder beidseitig multilateralen Verhandlungen, also Auktionen, Ausschreibungen und Börsen [etwa BeRo01, Bic+98, CaOl01, ChLe01, Col+98, Ito+01, Tei+99]. Ausnahmen hiervon, die bilaterale Verhandlungen untersuchen, sind [BeKe00, deP+01, Koe+00].

Die im Unternehmensbereich anzutreffenden Beschaffungsentscheidungen lassen sich in den meisten Branchen nur in der geringeren Zahl der Fälle auf reine Preisentscheidungen reduzieren. Dies bedeutet, dass elektronische Anwendungen zur Unterstützung dieser Verhandlungen die Aushandlung multipler Attribute erlauben müssen, sollen sie in größerem Umfang praxistauglich sein. Die Abbildung

multipler Verhandlungsattribute stellt daher ebenfalls eine der Zielsetzungen des Projekts dar. Multiattributive Verhandlungen werden bisher eher selten untersucht; Ausnahmen sind etwa [BaLo01, TeKu02].

Eine überwiegende Zahl von Forschungsprojekten zu elektronischen Verhandlungen stellt die vollständige Automatisierung des Verhandlungsvorgangs in ihren Mittelpunkt. Diese Projekte konzipieren und entwickeln größtenteils agentenbasierte Systeme zur automatisierten Durchführung von Auktionen [etwa BeRo01, CaOl01, ChLe01, Cha+97, Cha+96, GuMa98, Ito+01, Wur+98]. Die Frage, in welchem Umfang die vollständige Automatisierung von Verhandlungsvorgängen möglich ist, wird nach wie vor kontrovers diskutiert [BeSe99; Ker+99; PrCo01], soll aber an dieser Stelle nicht thematisiert werden. Ansätze, die nicht auf Vollautomatisierung abzielen, finden sich - mit durchaus unterschiedlichen Hintergrund - bei [BeKe00, Koe+00, Rung00]. Ziel der Anwendungsentwicklung im hier berichteten Kontext ist nicht die Vollautomatisierung von Verhandlungsvorgängen, sondern deren elektronische Unterstützung. Während durchaus einzelne Teilaufgaben automatisiert werden, wird das Verhandlungsergebnis durch elektronisch gestützte, strukturierte Interaktion menschlicher Akteure ermittelt. In diesem Zusammenhang wird auch von *Verhandlungsunterstützungssystemen* gesprochen.

3 Konzeptionelle Aspekte

3.1 Umfeld und Rahmenbedingungen

Zielsetzung des Gesamtprojektes ist es, ein generisches Anwendungsmodell elektronischer Verhandlungen zu entwickeln, das in verschiedenen Branchen und von unterschiedlichen Unternehmenstypen eingesetzt werden kann. Auch darin unterscheidet sich das hier zugrunde liegende Konzept von der Großzahl anderer Ansätze, die in der Regel auf eine Branche oder einen Anwendungsfall hin zugeschnitten sind [Add+00, Col+98, TeKu02, vaRi98].

Die Anforderungsanalyse basierte auf einer strukturierten Analyse der Anforderungen interorganisationaler elektronischer Verhandlungen im Unternehmensbereich nach den sechs Merkmalen Protokollkategorie, Automatisierungsgrad, Anzahl Attribute, Anzahl Positionen, Anzahl Verhandlungen, Mediationstyp [Rebs01c]. Aus Gründen des Umfangs wird diese Analyse hier nicht wiedergegeben. Die Evaluation vorliegender Ansätze zeigte auf, dass für diesen Kontext deutliche funktionale Defizite bestehen [Rebs01c].

Die Konzeption der Anwendungsarchitektur verfolgte insbesondere die Ziele der Modularität, Offenheit, Skalierbarkeit, Robustheit, Sicherheit und Wartbarkeit. Die einzelnen Gestaltungsentscheidungen, die zur aktuellen Anwendungsarchitek-

tur geführt haben, lassen sich an dieser Stelle aus Gründen des Umfangs ebenfalls nicht vollständig wiedergeben; es wird dazu auf andere Beiträge verwiesen [Rebs01a; Rebs01b; ReAm02]. Die Systemkonzeption erfolgte nach dem klassischen Wasserfallmodell unter durchgängiger Nutzung der Modellierungssprache UML.

Die entwickelte Gesamtarchitektur der Verhandlungsanwendung unterscheidet drei Anwendungsebenen, (1) die Meta-Verhandlungsebene, (2) die Verhandlungsebene und (3) die Kommunikationsebene. Jede der Ebenen wird durch eine der drei Hauptkomponenten der Anwendungsarchitektur repräsentiert (Abbildung 1):

- "Business Object Framework (BOF) Engine": Diese Komponente unterstützt die Verwaltung der anwendungsspezifischen Objekt- und Dokumentstrukturen (Business Object Framework) (Ebene 1).
- "Negotiation Engine": Diese Komponente unterstützt den eigentlichen Prozess interaktiver, bilateraler multiattributiver Verhandlungen (Ebene 2).
- "Communication Engine": Diese Komponente versorgt die anderen beiden Komponenten mit den zur Abwicklung des Verhandlungsprozesses notwendigen Kommunikationsfunktionen. Eingehende und ausgehende Nachrichten (Dokumente) werden verwaltet, authentifiziert und gegebenenfalls ver- bzw. entschlüsselt. Die Komponente stellt außerdem grundlegende Workflow-Funktionalitäten zur Einbindung in übergeordnete, organisationsübergreifende Workflows zur Verfügung (Ebene 3).

Alle drei Elemente sind als grobgranulare Komponenten mit standardisierten, XML-basierten Schnittstellen konzipiert und setzen sich wiederum aus Teilkomponenten zusammen. Andere Marktfunktionalitäten wie allgemeine Katalogdienste, Anbieterverzeichnisse oder Zahlungsdienste werden als eigenständige Komponenten verstanden und sind nicht Teil dieser Anwendung. Die Dienste, über die wir im Rahmen dieses Beitrags berichten, sind Bestandteile der Communication Engine.

3.2 Anwendungsszenario

Das in diesem Beitrag berichtete Zusammenspiel einer Marktplatzapplikation mit einer Verhandlungsanwendung ist eingebettet in ein Gesamtszenario, das auch als *Dynamic Business Web* bezeichnet wird [Arnd02]. Die beteiligten Akteure und Anwendungen (Verkäufer, Käufer, elektronischer Marktplatz, interne Systeme und Verhandlungslösung) werden darin zu einer integrierten und gleichzeitig flexiblen Informationskette verknüpft.

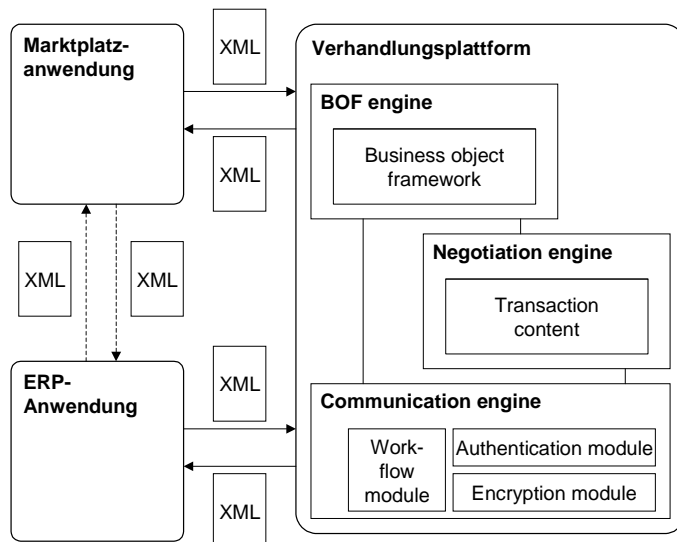


Abbildung 1: Gesamtanwendungsarchitektur

Das der Anwendungsentwicklung zugrunde liegende Ablaufszenario stellt sich wie folgt dar:

- Ein Unternehmen vertreibt Produkte und Dienstleistungen internetbasiert und bietet in diesem Zusammenhang verschiedene Web Services an. Hierzu zählen die Veröffentlichung eines elektronischen Katalogs in einem standardisierten XML-Format (z.B. BMEcat [ScKe01]) und eine Lieferfähigkeitsauskunft über im WSDL-Format definierte Schnittstellen. Das Unternehmen präsentiert sich unter Referenzierung der WSDL-Definitionen in einem UDDI-Repository.
- Anbieter können dem Marktplatz ihre Angebote zur Aufnahme bekannt geben oder werden aktiv vom Marktplatz identifiziert. Das öffentlich zugängliche UDDI-Repository liefert die entsprechenden Informationen. Der Marktplatz speichert die Web-Service-Schnittstellen (WSDL-Beschreibungen) der ausgewählten Unternehmen.
- Diejenigen Web Services, die für den Marktplatz interessant sind (Dienste zu Katalogdaten und Lagerbestandsdaten), werden selektiert und die notwendigen Web-Service-Clients werden implementiert. Die Katalogdaten werden in die Datenbank (Hyperkatalog) des Marktplatzes importiert.
- Ein registrierter potentieller Käufer durchsucht den Hyperkatalog des Marktplatzes und selektiert einen Anbieter sowie die für ihn interessanten Produktkategorien.

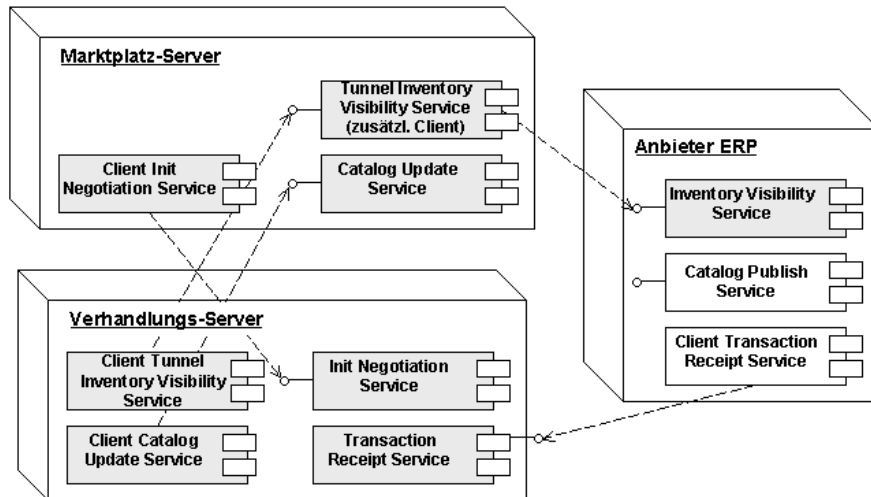


Abbildung 2: Deployment Diagram – Web Service Interaktion

- Bestehende Marktplatzanwendungen bieten dazu eine Warenkorbfunktionalität an, über die ausgewählte Produkte anschließend ohne weitere Verhandlungen gekauft werden können. Das dem Forschungsprojekt zugrunde liegende Szenario sieht dagegen, statt der bloßen Übernahme der Produkte in einen Warenkorb, die Möglichkeit der interaktiven bilateralen Aushandlung der spezifischen Attribute der Transaktion (produktbezogene, finanzielle und logistische Attribute) vor.
- Nach Aktivieren der Verhandlungsfunktionalität wird dazu ein Katalogauszug (ähnlich der gewählten Produkte eines Warenkorbs) generiert und mittels einer Web-Service-Schnittstelle zur Verhandlungsanwendung übertragen. Ebenso werden die Referenzinformationen der Verhandlungspartner (Anbieter und Nachfrager) automatisch eingestellt.
- Die Verhandlungsanwendung wird aktiviert. Die Verhandlungsanfrage wird auf Basis der übertragenen Daten dynamisch vorkonfiguriert eingestellt. Die weitere Verhandlung erfolgt innerhalb der Verhandlungsanwendung, deren eigene Dienste durch externe Web Services ergänzt werden. Bei bereits bestehenden Geschäftsbeziehungen können die Verhandlungen auch unmittelbar, ohne vorherige Einschaltung des Marktplatzes, aufgenommen werden.
- Zur Beurteilung der Verhandlungssituation ist via Web Service zu den verhandelten Positionen die aktuelle Verfügbarkeit der Produkte ersichtlich. Bei Bedarf, auch während der Verhandlung, können die Beteiligten den eingestellten Katalogauszug über eine weitere Web-Service-Schnittstelle aktualisieren.

- Nach erfolgreicher Verhandlung werden die ausgehandelten Transaktionsdaten mittels einer Web-Service-Schnittstelle an die internen Systeme der beteiligten Parteien übermittelt. Bei diesen Anwendungen handelt es sich in der Regel um betriebswirtschaftliche Standardsoftwaresysteme, d.h. ERP-(Enterprise Resource Planning)-Systeme. Die internen Systeme der Verhandlungspartner verarbeiten die Transaktionsdaten und erstellen daraus die notwendigen internen Belege (Bestellungen, Aufträge, Rechnungen, Lieferavise etc.).

Innerhalb des integrierten Verhandlungsprozesses fungiert die Verhandlungsanwendung als dezentraler Datenschnittpunkt, an dem die verschiedenen Web Services dynamisch zusammenlaufen. Designziel der hier beschriebenen Web-Service-Entwicklung war es dabei, möglichst schmale Schnittstellen mit geringen Abhängigkeiten zwischen den beteiligten Applikationen zu konzipieren.

Das UML Deployment Diagram in Abbildung 2 fasst die beteiligten Hard- und Software-Komponenten zusammen.

Die konzipierte Interaktion integriert verschiedene Web Services der beteiligten Parteien bzw. Anwendungen. In der hier beschriebenen Version basieren die Web Services auf *RPC (Remote Procedure Call)*, sind jedoch strukturell gleich mit messagebasierten Web Services und wären ebenfalls als solche zu realisieren. Dies wurde durch die Verwendung von Operationen (Methoden) erreicht, die nur einen einzigen String-Parameter - das zu übertragende XML-Dokument selbst - erfordern. Verwendet wurde das SOAP-Protokoll in der Version 1.2 [W3C02d], für die Schnittstellenbeschreibung diente die WSDL Spezifikation 1.1 [Chr+01].

4 Implementierung

4.1 Plattform

Als Entwicklungsplattform diente die objektorientierte Programmiersprache Java. Die Begründung für diese Auswahlentscheidung lieferten die wesentlichen Stärken dieser Sprache, die insbesondere in der Plattformneutralität, der konsequenten Objektorientierung und der Größe der weltweiten Entwicklergemeinschaft liegen. In Unternehmen ist der Einsatz der *Java Enterprise Version (J2EE)* bereits weit verbreitet. Existierender J2EE-Code kann dabei leicht in einen Web Service umgesetzt werden [Sun02a].

	Frontend-Dienste	Backend-Dienste
Marktplatz-anwendung	Anmeldung Marktplatz Übersicht der Anbieterkataloge Übersicht und Auswahl der Kategorien eines Anbieterkatalogs Übersicht und Auswahl der Produkte einer Katalogkategorie	Verhandlungsinitialisierung (Generierung <i>Negotiation Request</i> , <i>Client Init Negotiation Service</i> für den automatischen Aufruf der Verhandlungsanwendung) Katalogausschnittaktualisierung (<i>Selective Catalog Update</i> , <i>Catalog Update Service</i>) Durchgriff Verfügbarkeitsdaten ERP (<i>Tunnel Inventory Visibility Service</i>)
Verhandlungs-anwendung	Einblenden der Lagerbestandsinformationen	Initialisierung Verhandlungsplattform, Datenübergabe (<i>Init Negotiation Service</i>) Aufruf Katalogausschnittaktualisierung und Aktualisierungslogik (<i>Client Catalog Update Service</i>) Aufruf Durchgriff Verfügbarkeitsdaten ERP (<i>Client Tunnel Inventory Visibility Service</i>) Generierung Kontraktdateien für ERP (Generierung <i>Transaction Receipt</i> , <i>Transaction Receipt Service</i>)
Anbieter-anwendung (ERP)		Verfügbarkeitsdaten ERP (<i>Inventory Visibility Service</i>) Übernahme Kontraktdateien (<i>Client Transaction Receipt Service</i>)

Tabelle 1: Überblick implementierte Dienste

Aktuelle Weiterentwicklungen der Java-Plattform wie die *Java Connector Architecture (JCA)* befassen sich mit der Verknüpfung von J2EE-Anwendungen und proprietären Softwarelösungen im ERP-Umfeld. JCA ist Bestandteil der J2EE 1.3 Spezifikation und kann im Bereich der *Enterprise Application Integration (EAI)* und somit auch für die Java-Web-Service-Technologie Vorteile bringen. Darüber hinaus arbeitet Sun Microsystems momentan an einer *Java Specification Request*, die grundlegende Programmiermuster für die effiziente Java-Web-Service-Entwicklung zusammenstellt [Sun02b].

4.2 Funktionsübersicht der implementierten Dienste

Das prototypische Szenario umfasst drei Anwendungen: elektronischer Marktplatz, Verhandlungsplattform und die interne Anwendung (ERP-System) eines Verhandlungspartners (Abbildung 1). Alle Dienste zur Abwicklung der Kommunikationen zwischen den drei Anwendungen im Rahmen des oben dargestellten Szenarios wurden konzipiert, realisiert und implementiert. Die implementierten Web Services können jeweils in Frontend- und Backend-Funktionen eingeteilt werden. Tabelle 1 gibt einen Überblick.

Im Folgenden werden lediglich die zentralen Backend-Interaktionen im Rahmen der Verhandlungsinitialisierung auf Seiten des Marktplatzes und der Verhandlungsplattform näher beleuchtet. Diese Dienste stellen einen der Kernbereiche der Realisierung dar. Auf die nähere Spezifikation der weiteren Backend- sowie der Frontend-Dienste wird aus Gründen des Umfangs hier verzichtet.

4.3 Exemplarische Dienstabeschreibung 1: Verhandlungsinitialisierung Marktplatzanwendung

Die Abläufe einer Verhandlungsinitialisierung werden durch das Sequenzdiagramm in Abbildung 3 verdeutlicht. (Die in UML-Sequenzdiagrammen übliche diagonale Darstellung von Methodenfernaufrufen wurde aus technischen Gründen durch eine waagrechte Abbildung ersetzt. Um die Übersichtlichkeit der Darstellung zu wahren, wurden nur die wesentlichen Objekte und Methodenaufrufe berücksichtigt.) Die Abläufe finden fast vollständig im Hintergrund statt und bleiben dem Anwender verborgen.

Der Anwender initialisiert den Prozess durch Aktivierung der Verhandlungsanwendung aus dem User-Interface der Marktplatzanwendung heraus. Alle zur weiteren Verhandlung notwendigen Daten (Kategoriedaten, Katalogmetadaten, Produktdaten mit verhandelbaren Attributen und Anbieterdaten) werden über entsprechende Methoden aus der Datenbank ausgelesen und in *Bean*-Objekte gespeichert.

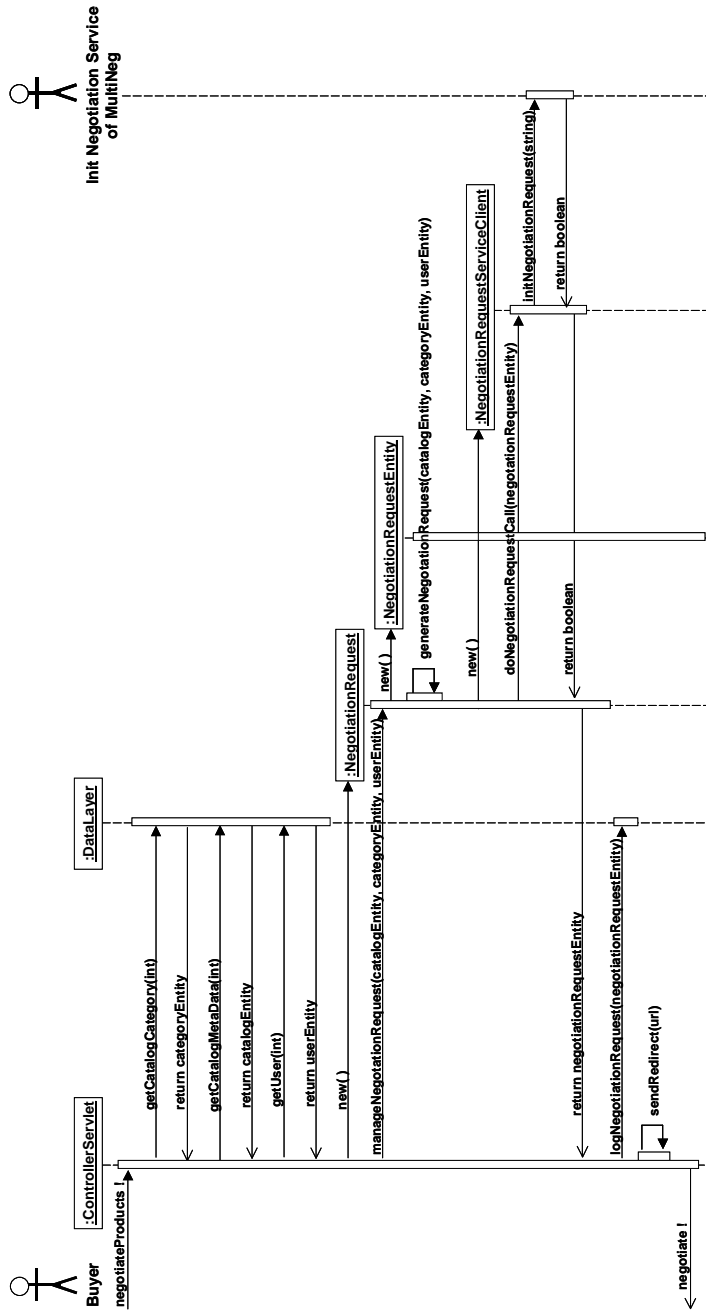


Abbildung 3: Sequenzdiagramm – Verhandlungsinitialisierung

Danach wird durch das *ControllerServlet* ein Objekt des Typs *NegotiationRequest* erzeugt, mittels dessen Methode *manageNegotiationRequest()* die Web-Service-Interaktion gesteuert wird. Die Hilfsmethode *generateNegotiationRequest()* hüllt das *Negotiation Request* als String-Eigenschaft in ein Objekt der Bean-Klasse *NegotiationRequestEntity*. Sind diese Prozesse erfolgreich abgelaufen, wird das Objekt über die Methode *doNegotiationRequestCall()* an das Web-Service-Client-Objekt vom Typ *NegotiationRequestServiceClient* weitergereicht.

Hier erfolgt der eigentliche Web-Service-Aufruf des *Init Negotiation Service* der Verhandlungsanwendung. Die Klasse *NegotiationRequestServiceClient* erzeugt hierzu ein Service-Objekt und führt den Aufruf nach Festlegung der Parameter und Parameterdatentypen über ein Call-Objekt aus.

Nach fehlerfreier Übertragung und Initialisierung der Verhandlungsanwendung wird der boolesche Rückgabewert `<true>` an das *ControllerServlet* zurückgegeben. Die Methode *logNegotiationRequest()* des *DataLayer*-Objekts erstellt den Datensatz in der Tabelle *negotiation_request*. Schließlich wird die Verhandlungsanwendung mit den Anmeldeparametern über die Methode *sendRedirect()* aufgerufen.

4.4 Exemplarische Dienstbeschreibung 2: Initialisierung Verhandlungsplattform

Innerhalb des prototypischen Szenarios weist die Verhandlungsanwendung die komplexeste Struktur auf. An dieser Stelle beschreiben wir lediglich die Initialisierung der Anwendung. Der Ablauf ist aus dem Sequenzdiagramm in Abbildung 4 ersichtlich.

Die Initialisierung der Verhandlungsanwendung wird von der Klasse *Negotiation-Initializer* gesteuert. Sie stellt die Methoden zum Extrahieren der Daten aus dem Objekt *NegotiationRequest* zur Verfügung und bildet die Schnittstelle zu den notwendigen Konfigurationsprozessen der Anwendung auf der Datenebene. Bei einer Anfrage des Marktplatz-Clients wird die Methode *initNegotiationRequest()* aufgerufen, die mittels der Hilfsmethode *initNegotiationInitializer()* ein *DataHandler*-Objekt für die Datenbankverbindung instanziiert. Der XML-String des *Negotiation Receipt* wird an die Methode *parseNegotiationRequest()* übergeben und eingelesen. Die Werte der `<item>`-Elemente des übertragenen Dokuments werden dabei ausgelesen und in Objekte des Typs *ProductEntity* gespeichert. Diese wiederum werden in ein *Bean*-Objekt vom Typ *NegotiationRequest* gehüllt und an die Methode *writeRequestInDatabase()* übergeben, die sämtliche Datenbankinitialisierungen, wie etwa das Anlegen der Verhandlungspartner, der Verknüpfung der Partner und das Speichern der Produkte mit ihren Verhandlungsattributen, steuert.

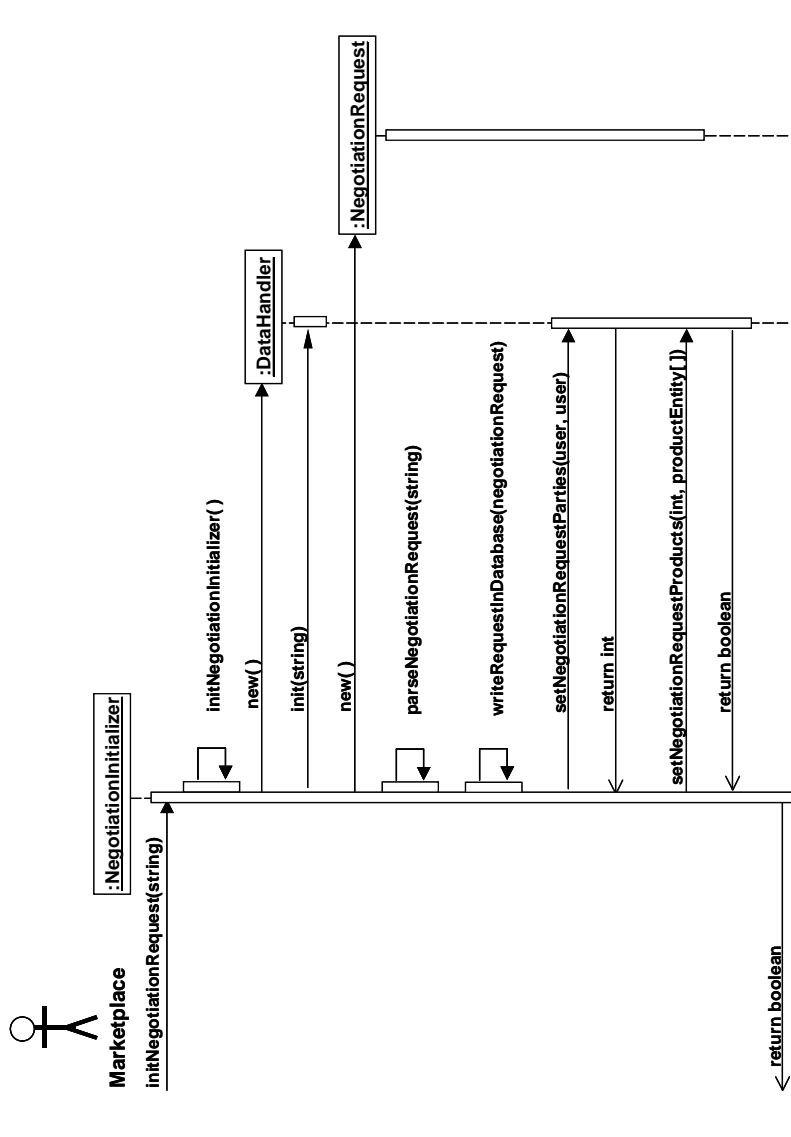


Abbildung 4: Sequenzdiagramm – Init Negotiation Service und Initialisierungslogik

Die Methoden *setNegotiationRequestParties()* und *setNegotiationRequestProducts()* der Klasse *DataHandler* sind so realisiert, dass sie nach gezielten Abfragen der bestehenden Daten entweder neue Datensätze anlegen oder bereits vorhandene Datensätze aktualisieren. Nach Abschluss der Datenbankinitialisierung

gibt die Methode *writeRequestIndatabase()* einen booleschen Wert an die ursprüngliche Web-Service-Methode *initNegotiationRequest()* zurück.

5 Diskussion und Ausblick

Die im Rahmen des Gesamtprojekts entwickelten Anwendungskomponenten zur Unterstützung elektronischer Verhandlungen können als funktionale Erweiterung elektronischer Märkte dienen, sind jedoch konzeptionell und administrativ von diesen unabhängig und ebenso ohne diese nutzbar. Die implementierten Web Services übernehmen die Kommunikation zwischen den einzelnen Anwendungen bzw. Komponenten der beteiligten Parteien. Innerhalb des integrierten Verhandlungsprozesses fungiert die Verhandlungsanwendung als Datenschnittpunkt, an dem die Informationsflüsse der einzelnen Web Services dynamisch zusammenlaufen. Das Konzept der Web Services hat sich dabei als sehr leistungsfähig herausgestellt. Der Projektverlauf zeigt deutlich, dass die Web-Service-Integration von Anwendungen auch in einem schlanken Entwicklungsprozess möglich ist, und demonstriert die Mächtigkeit des Web-Service-Paradigmas im Bereich elektronischer Markttransaktionen. Alle beschriebenen Services und die daran angeschlossene Prozesslogik der Verhandlungsplattform und des Marktplatzes konnten grundsätzlich praxistauglich implementiert werden.

Die Erfahrungen aus der Implementierung bestätigen die Kompatibilität von Web-Service-Paradigma und der Entwicklungsplattform Java. Insgesamt sind deutliche Bestrebungen erkennbar, das Web-Service-Paradigma zukünftig vollständig in die Java-Entwicklungsplattform zu integrieren. Besonders positiv wirkte sich der geringe Implementierungsaufwand für die eigentliche Web-Service-Kommunikation aus. Die zusätzliche Entwicklungszeit konnte somit für die Implementierung der Logikkomponenten und die Steigerung der Anwendungsstabilität verwendet werden. Weitere kostensenkende Faktoren für Integrationsvorhaben bestehen in der möglichen Verwendung standardisierter Werkzeuge (XML) sowie in der relativ einfachen Umwandlung bereits implementierter Funktionalitäten in Web Services.

In der Modellierungsphase wurden noch vorhandene Schwächen der UML erkennbar. Für eine detaillierte Darstellung der Dienste unter Berücksichtigung der jeweilig ausgetauschten Datenstrukturen fehlen derzeit standardisierte Beschreibungsformen. Hier bleibt Entwicklungsbedarf im Hinblick auf entsprechende UML-Diagrammtypen bestehen. Das Defizit konnte jedoch durch die in der Konzeptionsphase entwickelten WSDL-Schnittstellenbeschreibungen teilweise aufgefangen werden. Diese dienten im Rahmen eines Top-Down-Entwicklungsprozesses als Ausgangsbasis für die Dienst- und Logikimplementierungen.

Die interne Datenhaltung wurde zunächst über relationale Schemata gelöst. Dies erfordert das Einlesen und Abbilden der ausgetauschten XML-Dokumente in die

Tabellenstrukturen der Verhandlungsplattform. Zumindest für die von der Anwendung generierten und signierten Kontraktdateien bietet sich allerdings im Rahmen der nächsten Schritte eine native XML-Datenhaltung an, die durch den Einsatz spezifischer Datenbankserver oder relationaler Datenbankprodukte mit XML-Erweiterungen erreicht werden kann. Für einen produktiven Einsatz müsste die Fehlertoleranz der XML-verarbeitenden Komponenten noch erweitert werden. Der Prototyp geht von größtenteils fehlerfrei übertragenen XML-Dokumenten aus, die zudem Konformität mit den jeweiligen XML-Schemata aufweisen. Darüber hinaus ist zur Gewährleistung der Rechtsverbindlichkeit des elektronischen Kontrakts noch ein sicherer Signaturmechanismus zu implementieren.

Die ausgetauschten XML-Strukturen sind generisch und damit branchenunabhängig angelegt. Zwar können auch standardisierte Produkte verhandelt werden, interessanter ist jedoch die Verhandlung von Dienstleistungen oder komplexeren Produkten, die mit den entwickelten Diensten und Datenstrukturen möglich ist. In diesen komplexeren Anwendungssituationen zeigt sich die Leistungsfähigkeit der flexiblen, und doch semantisch präzisen Aushandlung der Vertragsattribute. Auch die weitergehende Option, zusätzliche Produkteigenschaften während einer Verhandlung als neue Attribute aufzunehmen, ist in der Datenstruktur und implementierten Logik des Prototypen bereits berücksichtigt, kann jedoch mit dem vorliegenden Funktionsumfang noch nicht verwaltet werden. Diese Komponenten sind im nächsten Schritt zu realisieren.

In der erreichten Flexibilität der Verhandlungen bei gleichzeitiger Unterstützung semantischer Präzision und Weiterverarbeitbarkeit der ausgehandelten Attribute liegt der besondere Innovationsgehalt der hier konzipierten und implementierten Lösung. Andere Lösungen definieren dagegen entweder feste Datenstrukturen, um die Eindeutigkeit und Weiterverarbeitbarkeit der verhandelten Daten zu gewährleisten, oder sie geben die semantische Eindeutigkeit und damit automatisierte Weiterverarbeitbarkeit der Daten zugunsten größerer Flexibilität der Kontraktstrukturen auf. Die Stärken der realisierten Anwendungskomponenten werden in den nächsten Schritten noch weiter auszubauen sein. Besonderer Augenmerk wird dabei auf der Verwaltung der Semantik der ausgetauschten Daten sowie dem Management der entsprechenden semantischen Bezugssysteme liegen.

Literatur

- [Add+00] Addis, M.; Allen, P.; Surrige, M.: Negotiating for Software Services. In: Tjoa, A.M. et al. (Hrsg.): Proceedings of the 11th International Workshop on Database and Expert Systems Applications. IEEE Computer Society, Los Alamitos, CA 2000, S. 1039-1043.
- [BaLo01] Barbuceanu, M.; Lo, W.-K.: Multi-attribute Utility Theoretic Negotiation for Electronic Commerce. In: Dignum, F. et al. (Hrsg.): Agent-Mediated Electronic Com-

- merce III. Current Issues in Agent-Based Electronic Commerce Systems (LNCS Vol. 2003). Springer, Berlin 2001, S. 15-30.
- [BeSe99] Beam, C.; Segev, A.; Bichler, M. et al.: On Negotiations and Deal Making in Electronic Markets. In: Information Systems Frontier 1 (1999) 3, S. 241- 258.
- [BeRo01] Béjar, J.; Rodríguez-Aguilar, J. A.: To Bid or Not To Bid - Agent Strategies in Electronic Auction Games. In: Dignum, Frank et al. (Hrsg.): Agent-Mediated Electronic Commerce III. Current Issues in Agent-Based Electronic Commerce Systems (LNCS Vol. 2003). Springer, Berlin 2001, S. 173-191.
- [BeKe00] Benyoucef, M.; Keller, R. K.: A Conceptual Architecture for a Combined Negotiation Support System. In: Tjoa, A.M. et al. (Hrsg.): Proceedings of the 11th International Workshop on Database and Expert Systems Applications. IEEE Computer Society, Los Alamitos, CA 2000, S. 1015-1019.
- [BiKI00] Bichler, M.; Klimesch, R.: Simulation multivariater Auktionen - Eine Analyse des OTC-Handels mit Finanzderivaten. In: Wirtschaftsinformatik 42 (2000) 3, S. 244-252.
- [Bic+98] Bichler, M.; Segev, A.; Beam, C.: An Electronic Broker for Business-to-Business Electronic Commerce on the Internet. In: International Journal of Cooperative Information Systems 7 (1998) 4, S. 315-330.
- [CaOl01] Cardoso, H. L.; Oliveira, E.: A Platform for Electronic Commerce with Adaptive Agents. In: Dignum, Frank et al. (Hrsg.): Agent-Mediated Electronic Commerce III. Current Issues in Agent-Based Electronic Commerce Systems (LNCS Vol. 2003). Springer, Berlin 2001, S. 96-107.
- [Cha+97] Chavez, A.; Dreilinger, D.; Guttman, R. H.: A Real-Life Experiment in Creating an Agent Marketplace. <http://guttman.www.media.mit.edu/people/guttman/research/pubs/paam97.pdf>, Abruf 2001-05-23.
- [Cha+99] Chan, N. T.; LeBaron, B.; Lo, Andrew W. et al.: Agent-Based Models of Financial Markets: A Comparison with Experimental Markets. <http://ebusiness.mit.edu/research/agents.pdf>, Abruf 2001-05-23.
- [Cha+96] Chavez, A.; Maes, P.: Kasbah: An Agent Marketplace for Buying and Selling Goods. In: Crabtree, B.; Jennings, N. (Hrsg.): Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology (PAAM'96). The Practical Application Company Ltd, Blackpool 1996, S. 75-90.
- [Chr+01] Christensen, E.; Curbera, F.; Meredith, G.; Weerawarana, S.: Web Services Description Language (WSDL) 1.1, W3C Note 15 March 2001, <http://www.w3.org/TR/wsdl>, Abruf 2002-09-11.
- [Col+98] Collins, J.; Tsvetovatyy, M.; Mobasher, B.: MAGNET: A Multi-Agent Contracting System for Plan Execution. In: Proceedings of Workshop on Artificial Intelligence and Manufacturing: State of the Art and State of Practice, August 1998, S. 63-68.
- [Comm02] CommerceOne, Inc.: Solutions. <http://www.commerceone.com/solutions/>, Abruf 2002-10-01.
- [DaSm83] Davis, R.; Smith, R. G.: Negotiation as a Metaphor for Distributed Problem Solving. In: Artificial Intelligence 20 (1983) 1, S. 63-109.

- [deP+01] de Paula, G. E.; Ramos, F. S.; Ramalho, Geber L.: Bilateral Negotiation Model for Agent-Mediated Electronic Commerce. In: Dignum, Frank et al. (Hrsg.): Agent-Mediated Electronic Commerce III. Current Issues in Agent-Based Electronic Commerce Systems (LNCS Vol. 2003). Springer, Berlin 2001, S. 1-14.
- [Fre+02] Fremantle, P.; Weerawarana, S.; Khalaf, R.: Enterprise Services. In: Communications of the ACM 45 (2002) 10, S. 77-82.
- [Glas00] Glass, G.: The Web Services (R)evolution, Part 1. Applying Web Services to Applications. <http://www-106.ibm.com/developerworks/library/ws-peer1.html>, November 2000, Abruf 2002-10-01.
- [GuMa98] Guttman, R. H.; Maes, P.: Cooperative vs. Competitive Multi-Agent Negotiations in Retail Electronic Commerce. In: Proceedings of the Second International Workshop on Cooperative Information Agents (CIA'98), Paris, France, July 3-8, 1998
- [IBM02] IBM: SilkRoad project. <http://www.zurich.ibm.com/csc/ebizz/silkroad.html>, Abruf 2002-10-01.
- [Ito+01] Ito, T.; Fukuta, N.; Shintani, T.: BiddingBot: A Multiagent Support System for Cooperative Bidding in Multiple Auctions. <http://www.cs.cmu.edu/~softagents/papers/itota-icmas00-poster.pdf>, Abruf 2001-05-23.
- [Ker+99] Kersten, G. E.; Noronha, S. J.: Negotiations in Electronic Commerce: Methodological Misconceptions and a Resolution. INR02/99. <http://interneg.org/interneg/research/papers/1999/02.pdf>, Abruf 2001-05-23.
- [Koe+00] Koetsier, M.; Grefen, P.; Vonk, J.: Contracts for Cross-Organizational Workflow Management. In: Bauknecht, Kurt et al. (Hrsg.): Electronic Commerce and Web Technologies. Springer, Berlin 2000, S. 110-121.
- [Kreg01] Kreger, H.: Web Services Conceptual Architecture (WSCA 1.0). <http://www.alphaworks.ibm.com/tech/webservicestoolkit>, May 2001. Abruf 2002-10-01.
- [Lee+97] Lee, J. G.; Kang, J. Y.; Lee, E. S.: ICOMA: An Open Infrastructure for Agent-Based Intelligent Electronic Commerce on the Internet. In: Proceedings of the International Conference on Parallel and Distributed Systems (CPADS 1997). IEEE Computer Society, Los Alamitos, CA 1997, S. 648-655.
- [Mult03] MultiNeg Consortium: MultiNeg Project Web Site. <http://www.fbw.fh-darmstadt.de/multineg>, Abruf 2003-05-01.
- [Orac02] Oracle Corp.: Exchanges. <http://www.oracle.com/appsnet/products/exchanges/content.html>, Abruf 2002-10-01.
- [PrCo01] Pradella, M.; Colombetti, M.: A Formal Description of a Practical Agent for E-Commerce. In: Dignum, Frank et al. (Hrsg.): Agent-Mediated Electronic Commerce III. Current Issues in Agent-Based Electronic Commerce Systems (LNCS Vol. 2003). Springer, Berlin 2001, S. 84-95.
- [Rebs00] Rebstock, M.: Elektronische Geschäftsabwicklung, Märkte und Transaktionen - eine methodische Analyse. In: HMD Praxis der Wirtschaftsinformatik 37 (2000) 215, S. 5-15.

- [Rebs01a] Rebstock, M.: An Application Architecture for Supporting Interactive Bilateral Electronic Negotiations. In: Bauknecht, K.; Madria, S.K.; Pernul, G. (Hrsg.): Electronic Commerce and Web Technologies. Proceedings of the EC-Web 2001. Springer, Berlin 2001, S. 196-205.
- [Rebs01b] Rebstock, M.: Efficiency and Flexibility of Multi-Attribute Negotiations - The Role of Business Object Frameworks. In: Tjoa, A.M.; Wagner, R.R.; Al-Zobaidie, A. (Hrsg.): Proceedings of the 12th International Workshop on Database and Expert Systems Applications. IEEE Computer Society, Los Alamitos 2001, S. 742-746.
- [Rebs01c] Rebstock, M.: Elektronische Unterstützung und Automatisierung von Verhandlungen. In: Wirtschaftsinformatik 43 (2001) 6, S. 609-617.
- [ReAm02] Rebstock, M.; Amirhamzeh Tafreschi, O.: Secure Interactive Electronic Negotiations in Business-to-Business Marketplaces. In: Wrycza, S. (Hrsg.): Proceedings of the 10th European Conference on Information Systems (ECIS2002). Universität Gdansk, Gdansk 2002, S. 564-572.
- [Rung00] Runge, A.: Die Rolle des Electronic Contracting im elektronischen Handel. Diss., Universität St. Gallen 2000.
- [Schm93] Schmid, B.: Elektronische Märkte. In: Wirtschaftsinformatik 35 (1993) 5, S. 465-480.
- [Schm99] Schmid, B.: Elektronische Märkte - Merkmale, Organisation und Potentiale. In: Hermanns, A.; Sauter, M. (Hrsg.): Management-Handbuch Electronic Commerce. Vahlen, München 1999, S. 31-48.
- [ScKe01] Schmitz, V.; Kelkar, O.; Pastoors, T.: Spezifikation BMEcat Version 1.2. <http://www.bmecat.org>, Abruf 2001-05-26.
- [Stal02] Stal, M.: Web Services: Beyond Component-Based Computing. In: Communications of the ACM 45 (2002) 10, S. 71-76.
- [Ste+99] Steinmetz, E.; Collins, J.; Jamison, S.: Bid Evaluation and Selection in the MAGNET Automated Contracting System. In: Agent Mediated Electronic Trading, Lecture Notes in Artificial Intelligence (LNAI) 1571. Springer, Berlin 1999, S. 105-125.
- [Sun02a] Sun Microsystems, Inc.: Java 2 Platform Standard Edition, <http://java.sun.com/>, Abruf 2002-11-04.
- [Sun02b] Sun Microsystems Inc.: Java Blue Prints, Model-View-Controller, <http://java.sun.com/blueprints/patterns/MVC-detailed.html>, Abruf 2002-11-04.
- [Tei+99] Teich, J.; Wallenius, H.; Wallenius, J.: A Multiple Unit Auction Algorithm: Some Theory and a Web Implementation. In: EM - Electronic Markets 9 (1999) 3, S. 199-205.
- [TeKu02] Teuteberg, F.; Kurbel, K.: Simulation des Agentenverhaltens auf einem elektronischen Marktplatz zur Personalakquisition. In: Weinhardt, C.; Holtmann, C. (Hrsg.): E-Commerce. Netze - Märkte - Technologien. Physica, Heidelberg 2002, S. 253-271.
- [Tsv+97] Tsvetovatyy, M.; Gini, M.; Mobasher, B.: MAGMA: An Agent-Based Virtual Market for Electronic Commerce. In: Journal of Applied Artificial Intelligence 6 (1997)

- [UDDI02] UDDI: UDDI Version 3.0 Published Specification, 19 July 2002. http://uddi.org/pubs/uddi_v3.htm, Abruf 2002-10-10.
- [vaRi98] van Heck, E.; Ribbers, P. M.: Introducing electronic auction systems in the dutch flower industry - a comparison of two initiatives. In: *Wirtschaftsinformatik* 40 (1998) 3, S. 223-231.
- [W3C02a] W3C World Wide Web Consortium, <http://www.w3c.org>, Abruf 2002-10-01.
- [W3C02b] W3C World Wide Web Consortium: Extensible Markup Language (XML). <http://www.w3c.org/XML>, Abruf 2002-10-01.
- [W3C02c] W3C Web Services Description Working Group: Web Services Description Language (WSDL) Version 1.2. W3C Working Draft 9 July 2002. <http://www.w3.org/TR/wsd112>, Abruf 2002-10-01.
- [W3C02d] W3C XML Protocol Working Group: SOAP Version 1.2 Part 0: Primer. W3C Working Draft 26 June 2002. <http://www.w3.org/TR/soap12-part0/>, Abruf 2002-10-01.
- [Wur+98] Wurman, P. R.; Wellman, M. P.; Walsh, W. E.: The Michigan Internet Auction-Bot: A Configurable Auction Server for Human and Software Agents. In: Sycara, K. P.; Wooldridge, M. (Hrsg.): *Proceedings of the 2nd International Conference on Autonomous Agents (Agents'98)*. ACM Press, New York 1998, S. 301-308.
- [Zarn99] Zarnekow, R.: *Softwareagenten und elektronische Kaufprozesse*. Gabler, Wiesbaden 1999.