

WI-Online Preprint

© 2000 Verlag Vieweg. Referenz: <http://www.wirtschaftsinformatik.de>

Dieser Beitrag ist für eine Veröffentlichung in der Zeitschrift WIRTSCHAFTSINFORMATIK vorgesehen. Redaktion und Verlag behalten sich Änderungen bis zur endgültigen Veröffentlichung vor.

Dieser Beitrag ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne schriftliche Zustimmung des Vieweg-Verlags unzulässig und strafbar. Das gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und für die Einspeicherung und Verarbeitung in elektronischen Systemen.

Es wird ausdrücklich darauf hingewiesen, dass die Bereitstellung dieses Aufsatzes im Internet durch Dritte in beliebiger Form unzulässig ist.

Einsatz von mobilen Agenten und XML zur Angebotsrecherche im Business-to-Consumer-Commerce

Frank Siegemund, Clemens H. Cap, Andreas Heuer

Dipl.-Inf. Frank Siegemund, Graduiertenkolleg „Infrastruktur für den elektronischen Markt“, Fachbereich Informatik, Technische Universität Darmstadt, Wilhelminenstr. 7, D-64283 Darmstadt, E-Mail: fsiegem@rbg.informatik.tu-darmstadt.de; Prof. Dr. Clemens H. Cap, Gefördert durch die Heinz-Nixdorf-Stiftung, Lehrstuhl für Informations- und Kommunikationsdienste, Prof. Dr. Andreas Heuer, Lehrstuhl für Datenbank- und Informationssysteme, Universität Rostock, Albert-Einstein-Str. 21, D-18051 Rostock, E-Mail: {cap|heuer}@informatik.uni-rostock.de

Kernpunkte für das Management:

Der Beitrag befasst sich mit Architekturen zur Angebotsrecherche im Business-to-Consumer-Commerce, die zu einer erhöhten Benutzerfreundlichkeit und Übersichtlichkeit beim virtuellen Einkauf beitragen. Die Kernaussagen sind:

- Der Einsatz von Agenten als Vermittler im elektronischen Handel zum Zwecke einer weitgehenden Nutzerunterstützung und der Vertretung unterschiedlichster Anwenderinteressen ist dringend erforderlich.
- Ein umfassender Angebotsvergleich ist ohne explizite Zustimmung der Anbieter möglich und wird durch die Verwendung von XML vereinfacht.
- Der Einsatz mobiler Agenten führt zu einer erhöhten Flexibilität und einer einfachen Erweiterbarkeit der Anwendungen zur Angebotsrecherche.
- Die Repräsentation von Angebotsinformationen in XML-Dokumenten erleichtert deren Transport, Austausch und Weiterverarbeitung und vermindert den Implementierungsaufwand.

1. Einführung

1.1. Motivation

Die Teilnehmer im E-Commerce sehen sich aufgrund des rasanten Wachstums und der zunehmenden Komplexität der Kommunikationsnetze einer gewaltigen Fülle von Informationen gegenüber, die ohne eine Kategorisierung bzw. automatische Verarbeitung nicht mehr durch den Einzelnen erschlossen werden kann. Die Anzahl der Firmen, die ihre Waren über Kommunikationsnetze vertreiben, stieg in den letzten Jahren deutlich an. So enthielt ein Verzeichnis von Online-Shops in Deutschland [Shop00] im September 2000 ca. 22500 Einträge in 500 Kategorien. Die hierdurch bedingte Intransparenz des elektronischen Marktes und eine oft unzureichende Nutzerunterstützung beim Erwerb von Produkten über Kommunikationsnetze behindern die ganzheitliche Durchsetzung und Akzeptanz des Business-to-Consumer-Commerce (B2C).

Der Einsatz von Softwareagenten vereinfacht die Interaktion auf elektronischen Märkten, weil diese die Interessen von Käufern gegenüber einer Vielzahl von Handelspartnern vertreten und die Komplexität elektronischer Märkte zumindest teilweise vor den Kunden verbergen können [MGM98]. Untersuchungen über das Verhalten von Kunden in der CBB-Forschung (Consumer Buying Behaviour) haben verschiedene Phasen des Kaufvorganges, wie beispielsweise die Bedarfsbestimmung oder die Produkt- und Händlerauswahl, identifiziert [RuSt87]. Erst durch die Verwendung elektronischer Agenten können Käufer einige dieser Phasen überhaupt umfassend durchführen [GMM99].

Die vorliegende Arbeit befasst sich mit Architekturen für Agenten zur Angebotsrecherche. Diese Agenten unterstützen die Produktauswahl, den seriösen Angebotsvergleich und das Bestellen von Produkten bei verschiedenen Anbietern. Die Architekturen werden auf der Grundlage definierter Anforderungskriterien verglichen und es wird aufgezeigt, inwiefern durch den Einsatz mobiler Agenten und XML eine erhöhte Flexibilität, leichtere Anpassbarkeit und eine einfachere Kommunikation mit den Anbietern erreicht werden kann. Eine konkrete Implementierung zeigt die Umsetzbarkeit einer der vorgestellten Ansätze.

1.2. Inhaltsübersicht

Im folgenden Abschnitt werden die für diese Arbeit grundlegenden Techniken sowie themenverwandte Arbeiten vorgestellt. Im Anschluss daran werden im Abschnitt 3 die Teilprobleme der Angebotsrecherche erörtert. Der Abschnitt 4 führt Entscheidungskriterien für die Verwendung mobiler Agenten auf. Die Architekturen für die Produktrecherche werden in Abschnitt 5 dargestellt. Der Zugriff auf die Angebotsinformationen und die Interaktion mit verschiedenen Händlern wird im darauffolgenden Abschnitt erläutert. Letztendlich befassen sich die abschließenden Abschnitte mit den Vorteilen von XML, der Umsetzung einer konkreten Architektur und einem Ausblick auf weitere Forschungsbemühungen.

2. Grundlegende Konzepte und Forschungsumfeld

2.1. Forschungsumfeld

Die Bedeutung von Agenten als Vermittler im elektronischen Handel wurde in einer Initiative am MIT untersucht [MGM98]. Dabei sind eine Reihe von Forschungsprototypen, wie z.B. Kashbah [ChMa96], Market Maker oder Tete-a-Tete [GMM99] entstanden. Der von der Firma Anderson Consulting entwickelte BargainFinder-Agent [Krul96] unterstützte Kunden bei der Händlerauswahl während des CD-Kaufs. Die rein serverbasierte Abfrage der Angebotsinformationen und die fehlende Bereitschaft einiger Händler, einen Preisvergleich zuzulassen, führte jedoch zu einer Blockade der von diesem Tool generierten Anfragen. Es zeigte sich aber auch, dass viele, vor allem kleinere Anbieter, an einer Aufnahme in den Vergleichsprozess interessiert und außerdem bereit waren, ihre Systeme zu öffnen. Das Tool Jango benutzt die Web-Browser der Anwender für die Generierung der Anfragen und umgeht so ähnliche Gegenmaßnahmen.

Der Pocket BargainFinder [BrGo99], ist ein Kleinstgerät, welches einige der Vorteile des elektronischen Handels auch beim traditionellen Einkauf anbietet. Anderson Consulting spricht in diesem Zusammenhang von „augmented commerce“. Der an der Universität von Washington entwickelte ShopBot-Agent [DEW97] zeigt auf, wie Methoden der KI bei der Zusammenarbeit mit verschiedenen Online-Shops verwendet werden können.

Diese Arbeit konzentriert sich auf die zugrundeliegenden Infrastrukturen für den Einsatz von Agenten im B2C-Commerce. Dazu werden verschiedene Architekturkonzepte, von denen einige in oben aufgeführten Systemen benutzt wurden, einander gegenübergestellt und der Vorteil neuerer Techniken, wie mobiler Agenten und XML, in diesem Anwendungskontext beschrieben. Es wird argumentiert, dass die zunehmende Durchsetzung von XML zu einfacheren Interaktionsmustern mit den Anbietern führt. Bei Letzteren kann es sich um einzelne Online-Shops, virtuelle Marktplätze oder Konglomerate von Händlern handeln, die über eine einzelne Plattform ihre Produkte vertreiben wollen. In diesem Fall wird auch die Bereitschaft größer sein, Angebotsinformationen preiszugeben.

2.2.Mobile Agenten

Mobile Agenten sind intelligente Softwareagenten mit der Eigenschaft, in einem heterogenen Rechnernetz zwischen verschiedenen Knotenpunkten migrieren zu können. Ein mobiler Agent ist also bei der Lösung einer Aufgabe nicht an einen einzelnen Rechner gebunden, sondern kann die Dienste unterschiedlichster Rechner direkt in Anspruch nehmen. Dazu wird der gesamte Zustand des Agenten, sein Code einschließlich aller lokalen Daten über das Netzwerk transferiert. Mobile Agenten können mit anderen Agenten auf Agentenplattformen interagieren und dabei Informationen austauschen [BZW98, Klus99, Matt98, RoPo97].

Grundlegende Eigenschaften intelligenter Softwareagenten sind Autonomie, Kommunikation, Reaktivität und Proaktivität [BZW98]. Ein Agent heißt intelligent, wenn er in der Lage ist, sich im Rahmen einer vorgegebenen Problemstellung sinnvoll zu verhalten. Er zieht durch die Beobachtung seiner Umgebung Schlüsse und leitet daraus ein Verhalten ab, das ihm der Lösung des zugrundeliegenden Problems näherbringt. Dabei kann er seine Umgebung durch sein Verhalten aktiv beeinflussen [Runo95, Weis00].

2.3. Methoden zur Darstellung und Verarbeitung von Angebotsinformationen

Die Hypertext Markup Language (HTML) ist die vorherrschende Sprache im World Wide Web (WWW). Sie eignet sich hervorragend für die Beschreibung der visuellen Darstellung von Informationen, nicht jedoch für deren automatische Verarbeitung.

Im Gegensatz dazu ermöglicht die Extensible Markup Language (XML) [XML00] eine Differenzierung zwischen Layout und Anwendungsdaten. Das führt zu einer vereinfachten Verarbeitung der im jeweiligen Anwendungskontext relevanten Daten. So können z.B. Angebotsinformationen in einem eigenständigen XML-Dokument dargestellt und deren visuelle Interpretation durch Extensible Style-sheet Language Transformations (XSLT) [XSLT00] beschrieben werden. Die Struktur von XML-Dokumenten kann durch Document Type Definitions (DTDs) oder XML Schemata [XMLS00] beschrieben werden. Hierdurch wird es möglich, unternehmensübergreifende Vokabularien zu definieren, welche die Interoperabilität beim Informationsaustausch zwischen verschiedenen Anwendungen sicherstellen [CXML00]. Bild 1 zeigt, wie Angebotsinformationen durch XML-Dokumente dargestellt werden können.

```
<?xml version="1.0"?>
<!-- Ein einfaches XML-Dokument mit Produktinformationen
für Prozessoren -->

<Cpu>
  <Produktbezeichnung>AMD K6-2 3D Now!</Produktbezeichnung>
  <Geschwindigkeit Einheit="MHz">450</Geschwindigkeit>
  <Typ>Sockel 7</Typ>
  <Preis Währung="EUR" Betrag="99.99" />
</Cpu>
```

Bild 1 XML-Dokument mit Angebotsinformationen für Prozessoren

Das Document Object Model (DOM) [DOM00] ermöglicht eine einfache Verarbeitung von HTML- und XML-Dokumenten und den gezielten Zugriff auf Informationen über definierte, sprachenübergreifende Schnittstellen. Das DOM ist ein Objektmodell. Es zeigt auf, wie Dokumente durch Objekte und logische Beziehungen zwischen den Objekten repräsentiert werden können. Die Manipulation der Dokumente erfolgt durch Methoden der Objekte, die Darstellung der Daten durch deren Attribute.

Neben HTML und XML könnte auch das Resource Description Format (RDF) bei der Darstellung von Angebotsinformationen durch die Anbieter Verwendung finden.

3. Angebotsrecherche

3.1. Identifizierbare Teilprobleme

Das Ziel der Angebotsrecherche ist die einheitliche Interaktion von Kunden mit einer Vielzahl von Händlern. Sie ermöglicht potenziellen Käufern die Produktauswahl, den Angebotsvergleich sowie die Produktbestellung über Händlergrenzen hinweg. Kunden werden dadurch Kaufentscheidungen erleichtert, Käufer beim Einkauf unterstützt und die Transparenz des elektronischen Marktes erhöht.

Folgende Teilprobleme müssen bei der Angebotsrecherche gelöst werden:

- Extraktion der Angebotsinformationen von Online-Shops, da diese im Allgemeinen nicht an einem Vergleich ihrer Angebote mit denen anderer Händler interessiert sind,
- Realisierung von Nutzungsoberflächen für die Spezifikation von Suchkriterien, den Vergleich und die Gegenüberstellung von Angeboten,
- Suche und Akkumulation von Angebotsinformationen, die den von den Benutzern spezifizierten Suchkriterien entsprechen,
- Schaffung einer Schnittstelle für das Bestellen in verschiedenen Online-Shops.

Die Architekturen zur Angebotsrecherche müssen die Lösung dieser Teilprobleme durch ihre Strukturierung widerspiegeln.

3.2.Anforderungen

Die Angebotsrecherche findet in einem äußerst heterogenen Umfeld statt. Online Shops unterscheiden sich hinsichtlich der angebotenen Produkte, der Präsentation des Produktkataloges und der gesamten Ablauforganisation. Sie selbst stellen komplexe Softwareprodukte dar, die ihrerseits eine gewisse Ausfallwahrscheinlichkeit und Fehleranfälligkeit besitzen. Anwendungen zur Angebotsrecherche müssen daher auf Unerreichbarkeit der Anbieter und auf Änderungen der Präsentations- und Seitenstruktur adäquat reagieren.

Architekturen zur Angebotsrecherche müssen ein Höchstmaß an Fehlertoleranz, Robustheit und Flexibilität gewährleisten. Eine gute Erweiterbarkeit, wie z.B. an neue Produktarten und Online-Shops, ist wünschenswert. Ein großer potenzieller Benutzerkreis kann nur dann erreicht werden, wenn sich die Applikationen zur Angebotsrecherche durch einfache Bedienung, intuitiven Umgang und einen nur minimalen Administrationsaufwand auszeichnen. Weiterhin müssen die Anwendungen Interaktivität und damit geringe Reaktionszeiten bei Anfragen gewährleisten. Die bei der Recherche ermittelten Angebote müssen konsistent sein, d.h. aktuell gültige Angebote der Online-Shops darstellen.

4.Entscheidungskriterien für den Einsatz mobiler Agenten bei der Angebotsrecherche

In diesem Abschnitt werden die Vor- und Nachteile mobiler Agenten beim Einsatz für die Angebotsrecherche aufgeführt. Die Mehrheit der hier besprochenen Punkte ist auch in anderen Anwendungskontexten gültig.

4.1.Nachteile

- Für die Abarbeitung mobiler Agenten muss eine ausreichende Infrastruktur von Agentenplattformen vorhanden sein.
- Der Einsatz mobiler Agenten auf nicht vertrauenswürdigen Rechnern bedeutet ein Sicherheitsproblem, da die Agenten sowohl ihren Code als auch ihre Daten vollständig diesen Rechnern übergeben [FGS96; Matt98].
- Der Code mobiler Agenten sollte nicht zu umfangreich sein.

4.2.Vorteile

- Der Einsatz mobiler Agenten erfordert keine ständig aktive Netzwerkverbindung zwischen den beteiligten Rechnern, weil sie weitgehend autonom auf den entfernten Plattformen agieren können.
- Im Vergleich zum typischen Client-Server-Ansatz kann die Verwendung mobiler Agenten bei vorgegebenen Schnittstellen auf Seiten der Server zu einer deutlichen Verringerung der Netzwerklast führen. Server stellen im Allgemeinen eine vorgegebene Anzahl von Diensten bereit, die eine bestimmte Ergebnismenge zurückliefern. Typischerweise wird dieses Ergebnis in seiner Gesamtheit zum Client transferiert. Mobile Agenten haben die Möglichkeit, die Ergebnismenge noch vor der Übertragung zu evaluieren, einzuschränken oder umzuformen. Dafür muss die Funktionalität des Servers nicht verändert werden. Übertragen werden bei diesem Ansatz nur noch die eingeschränkte Ergebnismenge und der Agent selbst. Mobile Agenten haben die Möglichkeit, die Menge der relevanten Angebote schon auf den Agentenplattformen der Anbieter einzuschränken. Dadurch ergibt sich ein erhebliches Einsparpotenzial im Umfang der zu übertragenden Daten.
- Der Einsatz mobiler Agenten erhöht die Flexibilität von Anwendungen im Vergleich zu Techniken, bei denen fest vorgegebene Funktionen anderer Rechner über ein Netzwerk benutzt werden. Mobile Agenten können eine Vielzahl unterschiedlicher Aufgaben auf den entfernten Rechnern ausführen. Neuartige Agenten erfordern nicht zwingend Anpassungen des Servers [BZW98].
- Im Umfeld der Angebotsrecherche ermöglicht der Einsatz mobiler Agenten die Verwirklichung von Verhandlungsphasen mit den Anbietern. Sie können die Ergebnisse der Interaktion mit Anbietern direkt bei Verhandlungen mit anderen nutzen, wenn sie die Händler sequenziell nacheinander besuchen. Sie sind nicht an eine vorgegebene Reihenfolge beim Besuch von Anbietern gebunden. Die Fähigkeit der Agenten, autonom mit andern Agenten auf den unterschiedlichen Plattformen zu interagieren, ermöglicht dynamische Anpassungen im Vorgehen der Agenten und den Austausch von Angebotsinformationen unter den Agenten.

5.Architekturen zur Angebotsrecherche

5.1.Grobstruktur

Bei den im Folgenden dargestellten Architekturen wird stets zwischen Front- und Back-End differenziert. Das Front-End besteht aus einem Client für die Interaktion mit dem Benutzer und Schnittstellen zwischen Client und Server. Im Front-End wird die Benutzungsoberfläche realisiert, die den Anwendern die Spezifikation von Anforderungskriterien und den Vergleich von Ergebnissen ermöglicht. Die Verarbeitung der Anwenderdaten sowie der Zugriff auf Datenbanken und Online-Shops erfolgt im Back-End. Es besteht aus einem Server und architekturabhängig aus weiteren Bestandteilen, wie z.B. Datenbanken und mobilen Agenten.

5.2.Das Front-End

Für die Umsetzung der Interaktion zwischen Client und Anwender können HTML-Formulare und Java-Applets in Verknüpfung mit Web-Browsern oder eigenständige, von Web-Browsern unabhängige Applikationen benutzt werden. Bei der Verwendung von HTML-Formularen benötigt der Anwender

lediglich einen rudimentären Web-Browser für die Interaktion mit der Anwendung. Die Formulardaten werden über das Hypertext Transfer Protocol (HTTP) an den Server übertragen, der in diesem Fall ein Web-Server ist. Bei der Gestaltung von Oberflächen durch HTML bzw. HTML-Formulare bestehen starke Restriktionen bei der Auswahl von Oberflächenelementen.

Die Verwendung von Java ermöglicht die Gestaltung komplexerer Oberflächen. Die Benutzung von nicht vertrauenswürdigen Code – den Java-Applets ohne Zweifel darstellen – ist für die Verwendung in E-Commerce-Lösungen jedoch problematisch und reduziert das Vertrauen in solche Anwendungen. Von Web-Browsern unabhängige Applikationen bedeuten erhöhten Administrationsaufwand für die Anwender, weil die Programme explizit installiert werden müssen. Dies widerspricht der bereits weiter oben festgelegten Anforderung an einen geringen Administrationsaufwand.

5.3. Architekturen für das Back-End

In diesem Abschnitt wird davon ausgegangen, dass die Interaktion zwischen dem Anwender und der Applikation über einen Web-Client erfolgt. Der zentrale Server ist damit ein Web-Server, dessen Funktionalität durch anwendungsspezifische Skripte, wie Servlets, ASP- (Active Server Page) oder CGI-Skripte, erweitert wird. Der Benutzer legt durch Verwendung des Clients die Anforderungskriterien an zu berücksichtigende Angebote, wie z.B. eine Preisspanne oder Lieferbedingungen, fest. Daraufhin werden diese Daten an den Server übertragen und dort durch serverseitige Skripte verarbeitet. Diese ermitteln die zu den festgelegten Anforderungskriterien passenden Angebote und betten sie in ein HTML/XML-Dokument ein, welches an den Web-Client gesandt und dort dem Anwender dargestellt wird. Bei der Bestellung von Angeboten kontaktieren die Skripte direkt das Warenkorbsystem der Anbieter.

5.3.1. Architekturen ohne mobile Agenten

In dieser ersten Architekturvariante erfolgt die Extraktion der Angebotsinformationen von den Anbietern direkt nachdem ein Nutzer Suchparameter spezifiziert hat. Ein serverseitiges Skript kontaktiert sequenziell die zu durchsuchenden Online-Shops und erhält daraufhin XML- bzw. HTML-Dokumente, in denen die Daten zu aktuellen Angeboten eingebettet sind. Aus diesen Dokumenten müssen die Angebotsinformationen extrahiert und auf Übereinstimmung mit den Suchkriterien hin überprüft werden (siehe Kap. 6). Als Resultat generiert das Skript ein HTML- bzw. XML-Dokument mit den Suchergebnissen und sendet es zurück an den Web-Client. Dieser interpretiert das gesandte Dokument und stellt die Ergebnisse dem Benutzer dar.

Da die Angebotsdaten unmittelbar nach der Formulierung einer Suchanfrage durch direkten Zugriff auf die Online-Shops gewonnen werden, spiegeln diese Informationen stets die aktuell gültigen Angebote wider. Die Konsistenz der dem Nutzer präsentierten Daten ist dadurch gewährleistet. Es wird außerdem nur auf die Angebotsdaten zugegriffen, die tatsächlich benötigt werden und im gegenwärtigen Kontext relevant sind.

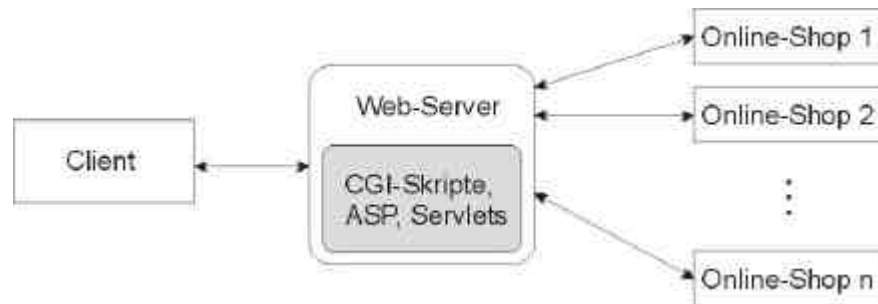


Bild 2 Architektur zur Angebotsrecherche ohne mobile Agenten

In der Tatsache, dass die Extraktion der Angebotsinformationen erst zu dem Zeitpunkt stattfindet, zu dem ein Anwender die Suchanfrage stellt, liegt jedoch auch ein wesentlicher Nachteil dieser Methode begründet. Da der Vorgang der Extraktion relativ viel Abarbeitungszeit in Anspruch nimmt, wird ein Anwender mit hohen Reaktionszeiten konfrontiert. Die Interaktivität kann deshalb bei diesem Ansatz nicht gewährleistet werden.

5.3.2. Zugriff auf Angebotsinformationen mit mobilen Agenten

Diese Architektur hat Ähnlichkeit mit dem zuvor dargestellten Ansatz. Die Akkumulation der Angebotsinformationen erfolgt hierbei jedoch durch mobile Agenten, die von den im Web-Server residierenden Skripten generiert werden. Eine Voraussetzung dabei ist, dass die Online-Shops eine Agentenplattform für die Abarbeitung dieser Agenten bereitstellen, in denen die Agenten z.B. durch Kommunikation mit Informationsagenten der Anbieter aktuelle Angebote ermitteln können. Anbieter stellen zurzeit solche Agentenplattformen nicht zur Verfügung. Im heutigen Umfeld ist diese Architektur daher nicht direkt umsetzbar. In der Zukunft könnten solche Ansätze jedoch verwendet werden, wenn die Systeme der Anbieter z.B. für Verhandlungsphasen um Kaufkonditionen geöffnet werden. Solche Funktionen sind am ehesten durch Verhandlungen zwischen Agenten umsetzbar, die ihrerseits unterschiedliche Parteien vertreten.

5.3.3. Ablage von Angebotsinformationen in lokalen Datenbanken

Anwendungen zur Angebotsrecherche beschränken sich im Allgemeinen auf eine festgelegte Anzahl von Produktarten. Für diese Produktgruppen existieren Formulare für die Spezifikation von Anforderungskriterien an Angebote und die Generierung von Suchanfragen. Nun ist es möglich, in regelmäßigen, möglichst kurzen Abständen, die Anbieter zu kontaktieren und sämtliche Angebote der betrachteten Produktkategorien zu ermitteln und in einer zum Server lokalen Datenbank abzulegen. Die Abfrage der Angebotsinformationen erfolgt im Hintergrund unabhängig von konkreten Anfragen. Dadurch vereinfacht sich die Wirkungsweise des serverseitigen Skriptes für die Behandlung von Nutzeranfragen, weil es lediglich die Anforderungen des Benutzers in eine Datenbankanfrage transformieren muss. Diese Vorgehensweise impliziert kürzere Antwortzeiten und sichert damit die Interaktivität beim Umgang mit der Anwendung.

Bild 3 zeigt schematisch den Aufbau einer solchen Architektur, wobei hier die Akkumulation der Angebotsinformationen durch mobile Agenten erfolgt. Dabei handelt es sich um eine klassische 3-tier-

Architektur, erweitert um den Einsatz mobiler Agenten. [EdVo97] beschreibt, wie N-tier-Architekturen im Zusammenhang mit großen, skalierbaren, webbasierten Systemen Verwendung finden können.

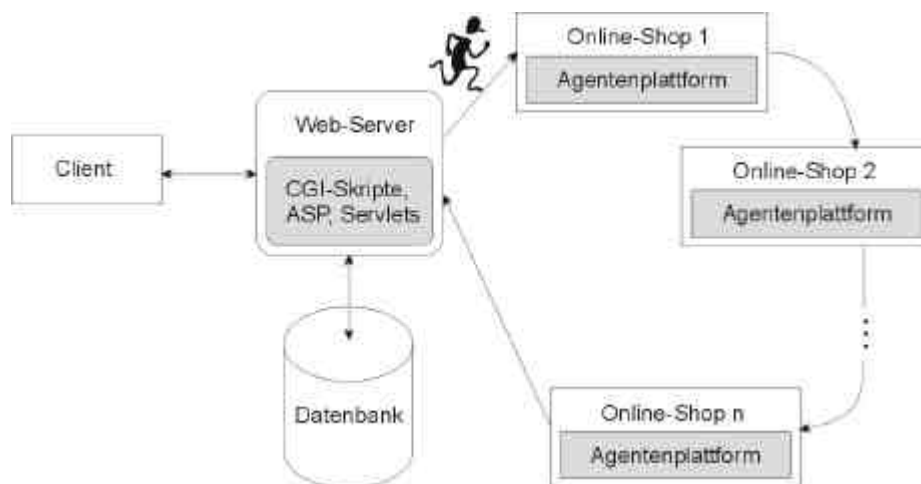


Bild 3 Einsatz mobiler Agenten für die Interaktion mit den Anbietern

Ein Problem bei diesem Ansatz besteht darin, die Konsistenz zwischen den in der Datenbank abgelegten und den im Online-Shop dargestellten Angeboten sicherzustellen. Finden sich in der Datenbank veraltete Daten, dann kann das zu Fehlersituationen bei der Zusammenarbeit mit den Online-Shops führen. Der Online-Shop Alternare beispielsweise reagiert beim Versuch, einen nicht mehr in der Produktpalette enthaltenen Artikel zu bestellen, mit einem Überlauf beim Wert des Warenkorbs. Die Angebotsdaten der Händler müssen daher in regelmäßigen, verhältnismäßig kurzen Abständen abgefragt werden.

Für die Ermittlung der Angebote von den Händlern können mobile Agenten eingesetzt werden. Ebenso ist es möglich, die Online-Shops direkt zu kontaktieren.

5.3.4. Akkumulation der Angebote von mehreren Servern

Die bisher vorgestellten, auf mobilen Agenten basierenden Architekturen sind für den Einsatz im heutigen E-Commerce-Umfeld wenig praktikabel, da die Online-Shops keine Plattformen für die Abarbeitung mobiler Agenten bereitstellen. Die nun einzuführende Architektur integriert Agentenplattformen in die Anwendung zur Angebotsrecherche selbst. Mobile Agenten werden dabei für die flexible Akkumulation und den Austausch der Angebotsdaten eingesetzt. Die Architektur realisiert einen verteilten Ansatz.

Die Grundidee ist, dass verschiedene im Internet verteilte Server in regelmäßigen Abständen die aktuellen Angebote unterschiedlicher Anbieter ermitteln. Diese Anbieter können Online-Shops sein; prinzipiell kann es sich bei ihnen aber auch um Marktplätze für Auktionen o.ä. handeln. So könnte beispielsweise ein Rechner in New York die Angebote New Yorker Online-Shops ermitteln, während ein zweiter in Rostock die Angebote Rostocker Tauschbörsen untersucht. Die Ergebnisse des Extraktionsprozesses werden dann in relativ zu den jeweiligen Rechnern lokalen Datenbanken abgespeichert. Die Rechner stellen Agentenplattformen für die Abarbeitung mobiler Agenten bereit. Der Zugriff auf die

Angebotsinformationen der Händler erfolgt durch direkte Kontaktierung der Online-Shops (siehe Kap. 6).

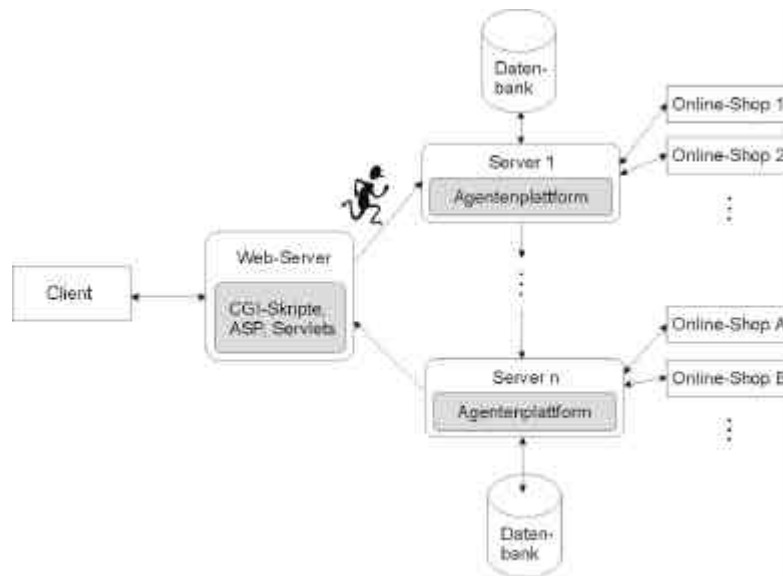


Bild 4 Akkumulation der Produktdaten durch mobile Agenten

Als Reaktion auf eine Anfrage generieren die Skripte des Web-Servers mobile Agenten, die zu den entfernten Rechnern migrieren. Dort ermitteln sie zu den spezifizierten Anforderungskriterien passende Angebote, indem sie mit anderen Agenten Daten austauschen und Anfragen an die Datenbanken stellen. Dies kann auf unterschiedliche Weisen erfolgen. Mobilen Agenten selbst sollte der Zugriff auf die Datenbanken nicht gestattet sein, da sie für die entfernten Rechner nicht vertrauenswürdigen Code darstellen. Sinnvoll ist in diesem Zusammenhang die Verwendung stationärer Agenten mit weitreichenderen Rechten in den jeweiligen Plattformen zum Zugriff auf die Datenbanken. Damit erfolgt die Akkumulation der Daten auf einheitliche Weise: durch Interaktion mit anderen Agenten. Letztendlich kehrt der mobile Agent zu dem Server zurück, auf dem er erzeugt wurde.

Flexibilität, gute Erweiterbarkeit und Skalierbarkeit zeichnen diesen Ansatz aus. Hohe Flexibilität ist dadurch gewährleistet, dass das Verhalten der mobilen Agenten dynamisch in Abhängigkeit von den Ergebnissen der Kommunikation mit anderen Agenten angepasst werden kann. Befinden sich beispielsweise von verschiedenen Benutzern generierte Agenten in einer Plattform, so können die Agenten untereinander Angebotsinformationen der Händler austauschen. Unter Umständen muss ein Agent bestimmte Server dann gar nicht besuchen, weil die von ihm gesuchten Angebote in der Ergebnismenge anderer Agenten bereits enthalten waren. Zusätzliche Rechner, die weitere Anbieter untersuchen, können problemlos in eine solche Architektur integriert werden. Da die Interaktion mit den Anbietern unabhängig von konkreten Anfragen erfolgt, wird interaktives Arbeiten unterstützt.

5.4. Minderung der Fehleranfälligkeit beim Einsatz mobiler Agenten

Bei den bisher vorgestellten Architekturen führte die Verwendung mobiler Agenten zu einer erhöhten Fehleranfälligkeit. Bei einem Ausfall einzelner Agentenplattformen eines Anbieters oder bei Netzwerkfehlern, die zu einem Verlust des mobilen Agenten führten, konnte kein einziges Ergebnis zum Client transferiert werden. D.h., ein einziger Fehlerpunkt führte zum Fehlschlagen gesamter Anfragen.

Ein erster Lösungsvorschlag für dieses Problem besteht darin, Zwischenergebnisse des Agenten nach dem Besuch einer festgelegten Anzahl von Anbietern an den Server zu übertragen. Geht der Agent im Nachhinein verloren, so können zumindest die bis dahin erhaltenen Ergebnisse dem Anwender dargestellt werden. Ebenso ist es möglich, durch das serverseitige Skript mehrere Agenten zu generieren, die verschiedene Agentenplattformen besuchen.

Der Vorgang der Migration mobiler Agenten erfordert, dass deren gesamter Code und Zustand in ein Format transformiert werden kann, welches in einem Netzwerk übertragbar ist. Dieses Format kann auch verwendet werden, um die Agenten und die mit ihnen assoziierten Informationen auf einem Datenträger abzulegen. Damit kann der Agent über eine festgelegte Zeitspanne zwischengespeichert und im Ausnahmefall reaktiviert werden.

Eine weitere Architekturvariation besteht in der Erzeugung zweier Agenten, die einander mit der Verzögerung von einer Agentenplattform folgen. Verlässt der erste Agent eine Plattform, so kontaktiert er den zweiten Agenten, der daraufhin ebenfalls die nächste Agentenplattform besucht. Das ist die, die der erste Agent gerade verlässt. Trifft kein Signal zum Verlassen der Plattform in einer vordefinierten Zeitspanne ein, wird der zweite Agent dupliziert und der neu erzeugte Agent übernimmt die Rolle des ersten. Dabei wird vom Verlust des ersten Agenten und von einer Fehlersituation in der entsprechenden Plattform ausgegangen. Diese Plattform wird durch den zweiten Agenten nicht noch einmal besucht. Analog verhält sich der erste Agent bei Verlust des zweiten.

6. Interaktion mit den Online-Shops

6.1. Zugriff auf die Funktionalität der Anbieter

Bei der Beschreibung der Architekturen zur Angebotsrecherche wurde schon mehrfach auf die notwendige Zusammenarbeit mit den Online-Shops hingewiesen. Stellen Anbieter Agentenplattformen bereit, dann kann die Kommunikation mit ihnen direkt über die von der Agentenplattform bereitgestellten Schnittstellen erfolgen. Ein solcher Ansatz ist z.B. im Zusammenhang mit auf Agenten basierenden Marktplätzen denkbar, bei der Struktur heutiger Online-Shops jedoch nicht praktikabel. Wie das Beispiel BargainFinder [Krul96] zeigt, sind einige Anbieter an einem Vergleich ihrer Angebote nicht interessiert und werden Anwendungen zur Angebotsrecherche keine erweiterten Zugriffsmöglichkeiten auf ihre Daten gewähren.

Anwendungen zur Angebotsrecherche müssen sich daher wie normale Kunden verhalten, um an Angebotsinformationen zu gelangen und Produkte zu bestellen. Hierbei wird der prinzipielle Aufbau von Online-Shops ausgenutzt, zu deren wesentlichen Komponenten die Präsentationskomponente sowie Datenbanken mit Angebots- und Kundeninformationen zählen [Merz99]. Zum Online-Shop gehörende Skripte verarbeiten die von den Benutzern eingegebenen Daten und nehmen die Umsetzung der in den Datenbanken abgelegten Informationen in HTML- bzw. XML-Dokumente vor (siehe Bild 5). Beim Anbieter Avitos verrichten Active Server Pages (ASP) diese Tätigkeiten. Online-Shops wie Alternate und Mix verwenden in Perl realisierte CGI-Skripte. Kontaktiert ein Kunde mit seinem Web-Browser

einen Online-Shop, blättert durch den angebotenen Produktkatalog oder fügt Produkte einem Warenkorb hinzu, so werden bei fast jedem dieser Schritte Skripte des Anbieters kontaktiert.

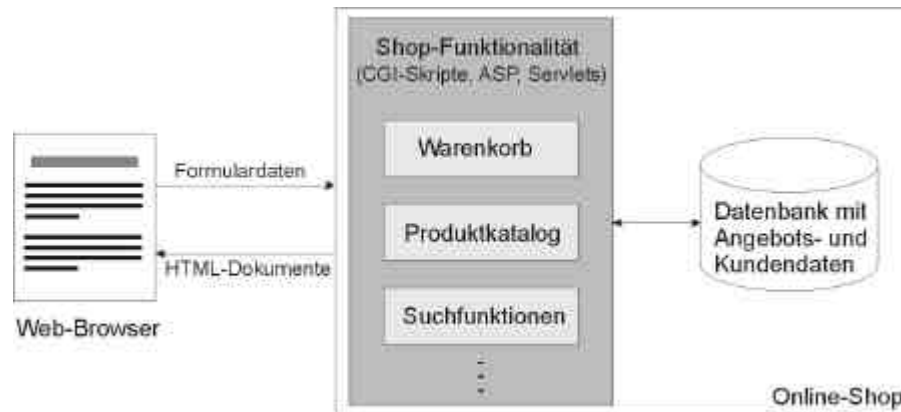


Bild 5 Typischer Ablauf einer Anfrage bei Online-Shops

Die Kommunikation mit den Skripten der Online-Shops, die im Normalfall über HTML-Formulare erfolgt, muss in der Anwendung zur Produktrecherche nachgebildet werden. Dazu müssen folgende Punkte untersucht werden:

- *Funktionalität der Skripte.* Zunächst stellt sich die Frage, welche Programme ein Online-Shop überhaupt zur Verfügung stellt und was diese Skripte letztendlich bewirken. Dazu muss zumeist der Quellcode der von dem Online-Shop generierten HTML-/XML-Dokumente untersucht werden. Insbesondere die dort beinhalteten Formulare sollten beachtet werden, da sie z.B. zulässige Parameter für die Skripte festlegen. Die Funktionalität im Einzelfall hängt von den spezifizierten Parametern ab.
- *Übertragungsart der Formulare.* Der *form*-Tag von HTML-Dokumenten gibt Aufschluss darüber, welche Übertragungsart im Hypertext Transfer Protocol für die Parameter verwendet wird.
- *Identifikation zulässiger Parameter und Parameterwerte.* Letztendlich müssen die von den Skripten verarbeitbaren Parameternamen und -werte bestimmt werden. Die Parameternamen sind die Namen der Elemente eines HTML-Formulars. Bei Auswahlboxen und versteckten Elementen sind die zulässigen Parameterwerte sofort klar, weil sie im HTML-/XML-Dokument angegeben sind. Bei Eingabefeldern ergeben sie sich aus dem Kontext.

Es existieren Werkzeuge, welche die Untersuchung solcher Formulare vereinfachen [W4F00].

6.2.Extraktion von Angebotsinformationen

Erfolgt die Interaktion mit den Anbietern durch den externen Zugriff auf deren Skripte, dann erzeugt das Katalogsystem der Anbieter im Allgemeinen HTML-/XML-Dokumente, in denen die Angebotsinformationen eingebettet sind. Für die automatische Weiterverarbeitung müssen die relevanten Daten aus diesen semistrukturierten Dokumenten extrahiert werden. Werkzeuge wie der Wrappergenerator W4F [W4F00] oder JEDI [Huck00] vereinfachen diesen Vorgang. Als Ergebnis des Extraktionsprozesses werden XML-Dokumente mit aktuellen Angebotsinformationen erzeugt.

Bei jedem Anbieter besitzen die Dokumente mit den Angebotsinformationen einer Produktkategorie implizit eine gemeinsame Struktur, weil sie automatisch durch das gleiche Skript erzeugt werden. Dies wird im Extraktionsprozess ausgenutzt.

Folgende Ansätze können bei der Extraktion aus HTML-/XML-Dokumenten unterschieden werden:

- Durch einen Übersetzer werden die Dokumente in ein internes Modell überführt, über dessen Methoden der Zugriff auf relevante Bestandteile erfolgt. Das kann beispielsweise durch eine Transformation in eine DOM-Repräsentation vollzogen werden. Die Überführung des gesamten Dokuments in eine interne Darstellung ist aufwendig, wenn nur einzelne Teile des Dokuments betrachtet werden.
- Für bestimmte Klassen von Dokumenten werden kontextfreie Regeln und diesen Regeln zugeordnete semantische Aktionen formuliert, welche den Aufbau des Ergebnisdokuments beschreiben. Dieser Ansatz wurde in der Referenzimplementierung realisiert.
- Stellen die Anbieter XML-Dokumente mit Angebotsinformationen bereit, so können XSLT (Extensible Stylesheet Language Transformations) [XSLT00] für die Überführung in das Zielformat benutzt werden, das ja wiederum XML ist. Einigen sich die Anbieter auf ein einheitliches Format zur Darstellung der Angebotsinformationen, dann kann auf die Transformation ganz verzichtet werden!

7. Verwendung von XML

Die Verwendung von XML ist in verschiedenen Phasen der Angebotsrecherche sinnvoll. Es wurde bereits beschrieben, wie XML als Ergebnisformat bei der Angebotsdatenextraktion benutzt wird. Der wesentliche Vorteil ist dabei eine vereinfachte Verarbeitung und Übertragung von Angebotsinformationen. In der Referenzimplementierung (siehe Kap. 8) werden die Angebotsdaten in XML-Dokumenten dargestellt. Das DOM, von dem es viele freie Implementierungen gibt, wird daraufhin verwendet, um auf einzelne XML-Elemente zuzugreifen, diese in einer relationalen Datenbank abzulagern oder in andere Formate zu transferieren. Ebenso wäre es möglich, die Daten der DOM-Repräsentation in einer objektorientierten Datenbank abzuspeichern; das DOM selbst ist ja ein Objektmodell. Da die Datenextraktion in diesem Fall formal als eine Abbildung semistrukturierter Daten auf XML-Dokumente angesehen werden kann, ergeben sich einfachere Transformationen besonders dann, wenn die Online-Shops XML- statt HTML-Dokumente erzeugen. Eine solche Entwicklung hin zur Verwendung von XML auf Seiten der Anbieter ist zu erwarten [BeMi98]. In diesem Fall können XSLT für die Transformation zwischen verschiedenen Formaten verwendet werden. Der Einsatz von XML verbessert damit die Zusammenarbeit mit den Anbietern, weil die Datenextraktion einfacher und bei der Verwendung gleicher Vokabularien sogar überflüssig wird. Das impliziert eine vereinfachte Erweiterbarkeit der Anwendungen zu Angebotsrecherche. XML-Dokumente können sehr flexibel in andere Formate transformiert oder in Datenbanken abgelegt werden [KIMe99].

Der Einsatz von XML ist jedoch nicht nur auf die Extraktionsphase beschränkt. Auch die mobilen Agenten können XML als Format für den Informationsaustausch benutzen.

8. Schnäppchenjäger – ein webbasiertes System zur Angebotsrecherche auf der Basis mobiler Agenten

8.1. Funktionsumfang

Die Anwendung Schnäppchenjäger realisiert die in Abschnitt 5.3.4 eingeführte Architektur. Durch die Applikation werden die Angebote der Online-Shops Alternate, Mix und Avitos miteinander verglichen. Exemplarisch wurden die Produktgruppen Motherboards, Prozessoren und Monitore für eine Gegenüberstellung ausgewählt. Auf die aktuellste Version der Anwendung kann über folgende URL zugegriffen werden: <http://gkpc14.rbg.informatik.tu-darmstadt.de/schnaepchen>.

In einem typischen Anwendungsszenario entscheidet sich der Anwender für die Recherche in einer Produktgruppe und hat daraufhin die Möglichkeit, Anforderungen an Angebote in dieser Kategorie zu spezifizieren. Dazu steht ihm jeweils ein Formular zu Verfügung. Die Anforderungskriterien sind vielfältig: in der Kategorie Prozessoren können u.a. der Prozessortyp, eine Produktbezeichnung, der gewünschte Hersteller und eine Preisspanne angegeben werden. Wie in Abschnitt 5.3.4 beschrieben, werden auf der Grundlage dieser Daten zugehörige Angebote der betrachteten Händler ermittelt und dem Benutzer in einer Ergebnisliste dargestellt.



Produktbezeichnung	Taktfrequenz	Preis	Online-Shop	Info	Auf die Preisliste
Pentium III (FC-PGA) 600E MHz	600 MHz	589 DM	www.alternate.de	Info	<input type="checkbox"/>
Intel Pentium III 600E MHz 32/256 KB Sockel 370 100 MHz (FC-PGA)(Bustakt 100 MHz)	600 MHz	560 DM	www.mix-computer.de		<input type="checkbox"/>
Pentium III (FC-PGA) 600EB MHz	600 MHz	579 DM	www.alternate.de	Info	<input type="checkbox"/>
Pentium III (FC-PGA) 600 MHz	600 MHz	579 DM	www.alternate.de	Info	<input type="checkbox"/>
Intel Pentium III 600EB MHz 32/256 KB Sockel 370 133 MHz (FC-PGA)(Bustakt 133 MHz)	600 MHz	580 DM	www.mix-computer.de		<input type="checkbox"/>
Pentium III (SECC2) 600EB MHz	600 MHz	599 DM	www.alternate.de	Info	<input type="checkbox"/>
Pentium III (SECC2) 650 MHz	600 MHz	599 DM	www.alternate.de	Info	<input type="checkbox"/>

Bild 6 Ergebnisliste mit Angeboten verschiedener Händler

Da diese bei einer großen Anzahl von Elementen unübersichtlich wirkt, hat der Benutzer die Möglichkeit, interessante Angebote auf eine Preisliste zu setzen. Diese Liste kann jederzeit aufgerufen werden. Auf ihr werden die wichtigsten Informationen zu den Angeboten zusammengefasst. Durch direkte Kontaktierung der Anbieter ist es möglich, nähere Informationen zu ausgewählten Produkten zu bekommen. Weiterhin hat der Anwender die Option, ein Produkt aus der Anwendung heraus zu bestellen. Dabei wird das ausgewählte Produkt dem Einkaufskorb beim jeweiligen Anbieters hinzugefügt.

8.2. Implementierungsdetails

Der Client für die Interaktion mit dem Anwender wurde als HTML-Oberfläche realisiert, wobei die Nutzereingaben in HTML-Formularen verarbeitet werden. Damit ist für die Benutzung des Programmes lediglich ein Web-Browser erforderlich. Eine Installation oder Administration der Anwendung durch den Benutzer entfällt dadurch.

Bei der Implementierung des Back-Ends wurde darauf Wert gelegt, die Programmiersprache Java möglichst durchgängig zu verwenden, weil sich dadurch die Kommunikation zwischen den einzelnen Teilkomponenten vereinfacht. Als zentraler Server wird ein Tomcat-Webserver verwendet. Servlets erweitern die Funktionalität dieses Servers und verarbeiten die Nutzereingaben. Sie wurden unter Verwendung des *Java Servlet Development Kit* (JSDK) implementiert. Tomcat ist direkt in der Lage, Servlets zu verarbeiten.

Zusätzlich zum zentralen Web-Server existieren weitere im Internet verteilte Server. Sie stellen jeweils eine Plattform für mobile Agenten bereit, die unter Benutzung des *Aglet Software Development Kit* (ASDK) von IBM erstellt wurden. Jedem dieser Server sind bestimmte Online-Shops zugeordnet, deren Angebote alle zwei Stunden abgefragt werden. Die Extraktionsregeln wurden mit der Skriptsprache von JEDI [Huck00] formuliert. Für jeden Shop gibt es ein übergeordnetes Skript, das allgemeine Regeln für die Verarbeitung von Preisangaben, Tabellenspalten usw. enthält. Die Skripte für die Produktgruppen werden dadurch sehr einfach. Als Ergebnis des Extraktionsprozesses entstehen XML-Dokumente mit den aktuellen Angebotsinformationen auf die über das DOM zugegriffen wird. JAXP (Java API for XML Parsing) ermöglicht diese Art der Verarbeitung. Die Angebotsinformationen werden in einer zu den Servern lokalen Datenbank abgespeichert. Dazu dient zurzeit eine PostgreSQL-Datenbank, auf die über JDBC (Java Database Connectivity) zugegriffen wird. In jedem der Server residieren stationäre Agenten, die den Datenbankzugriff vornehmen. Mobile Agenten können nicht direkt auf die Datenbank zugreifen, sondern müssen dafür mit den stationären Agenten kommunizieren. Bei der Angebotsrecherche werden seitens der Servlets mobile Agenten generiert, welche die von den Servern bereitgestellten Agentenplattformen besuchen. Dort akkumulieren sie durch Interaktion mit den anderen Agenten aktuelle Angebotsinformationen und kehren letztendlich zum Web-Server zurück.

9. Schlussfolgerungen aus der Implementierung

In der Praxis treten teilweise unvorhersehbare Fehlerbedingungen auf. Der Online-Shop Mix bot beispielsweise gleiche Produkte zu unterschiedlichen Preisen an; die Skripte des Anbieters Alternate generierten, indem sie den abschließenden `<html>`-Tag unterschlugen, keine vollständigen HTML-Dokumente. Über die von den Händlern generierten semistrukturierten Dokumente sollten daher so wenig Annahmen wie möglich getroffen werden. In diesem Zusammenhang hat sich der Einsatz von JEDI für die Angebotsextraktion bewährt. Selbst wenn unvorhersehbar Regeln nicht angewandt werden können, wird im Nachhinein weiterhin versucht, Ersetzungsregeln anzuwenden. Dies ist ein sehr fehlertolerantes Vorgehen [Huck00].

Flexibilität und der einfache Austausch von Informationen mit anderen Anwendungen ist in einem hohen Maße durch die Verwendung von XML gewährleistet. Es war überraschend, welche Vielfalt an größtenteils frei verfügbaren Werkzeugen die Verarbeitung von XML vereinfachen. Mit diesen Tools können XML-Dokumente einfach in eine DOM-Repräsentation überführt, in andere Formate transformiert oder in Datenbanken abgelegt werden. Dies reduziert signifikant den Implementierungsauf-

wand. Sollten die Händler in der Zukunft XML-Dokumente mit Angebotsinformationen erzeugen und deren visuelle Darstellung z.B. durch Stylesheets beschreiben, dann wird die Kommunikation mit den Anbietern sehr einfach. Damit unterstützt XML die Durchsetzung von Agenten im E-Commerce.

Das ASDK ermöglichte die komfortable Realisierung der mobilen Agenten und der zugehörigen Plattformen. Durch den verteilten Ansatz und den Einsatz mobiler Agenten ist insbesondere die Integration weiterer Anbieter einfach möglich.

10. Zusammenfassung und Ausblick

Diese Arbeit befasste sich mit der Angebotsrecherche im B2C-Commerce als eine Möglichkeit, die Transparenz des elektronischen Marktes zu erhöhen und potenziellen Käufern weitgehende Unterstützung beim Erwerb von Produkten über Kommunikationsnetze zu geben. Es wurde beschrieben, wie XML und mobile Agenten in Architekturen zur Angebotsrecherche eingesetzt werden können und welche Vorteile sich daraus gegenüber anderen Ansätzen ergeben.

Durch die Verwendung mobiler Agenten kann neuartige Funktionalität in E-Commerce-Lösungen integriert werden, die für die weitere Entwicklung in diesem Bereich von großer Bedeutung sein wird. Dazu zählen beispielsweise die Realisierung von Verhandlungen beim Online-Shopping. In anschließenden Forschungsarbeiten soll ein Online-Shop entwickelt werden, der eine Agentenplattform bereitstellt und so Kunden die Möglichkeit zu direkter Interaktion mit dem Anbieter ermöglicht. Es lohnt sich zu untersuchen, wie z.B. Verhandlungen oder Beratungen zwischen Kunden und Verkäufern hierbei beschrieben und realisiert werden können. Weiterhin soll betrachtet werden, ob ein Verbund solcher Shops, z.B. in einem virtuellen Kaufhaus, den Komfort beim Einkauf erhöht.

Literatur

- [BeMi98] *Behme, Henning; Mintert, Stefan: XML in der Praxis, Professionelles Web-Publishing mit der Extensible Markup Language.* Addison-Wesley, Bonn 1998.
- [BrGo99] *Brody, Adam B.; Gottsmann, Edward J.: Pocket BargainFinder: A Handheld Device for Augmented Commerce.* In: Gellersen, Hans-W. (Hrsg.): *Handheld and Ubiquitous Computing.* Springer-Verlag, Berlin 1999, S. 44-51.
- [BZW98] *Brenner, Walter; Zarnekow, Rüdiger; Wittig, Harmut: Intelligente Softwareagenten, Grundlagen und Anwendungen.* Springer-Verlag, Berlin 1998.
- [ChMa96] *Chavez, A.; Maes, P.: Kashbah: An Agent Marketplace for Buying and Selling Goods.* In: *Proceedings of the First International Conference on the Practical Application of Intelligent Agents and Multi-Agent Technology.* London 1996.
- [CXML00] *cXML.org: Commerce XML Resources.* <http://www.cxml.org>, Abruf am 2000-07-10.
- [DEW97] *Doorenbos, Robert; Etzioni, Oren; Weld, Daniel: A scalable comparison-shopping agent for the world wide web.* In: *Proceedings of the First International Conference on Autonomous Agents.* Marina del Rey 1997. S. 39-48.
- [DOM00] *World Wide Web Consortium: Document Object Model Level 1 Specification.* <http://www.w3.org/TR/1998/REC-DOM-Level-1-19981001/cover.html>, Abruf am: 2000-08-10.
- [EdVo97] *Edwards, Jeri; de Voe, Deborah: 3-Tier Client/Server At Work.* Wiley Computer Publishing, New York 1997.

- [FGS96] *Farmer, W.M.; Guttman, J.D.; Sarup, V.*: Security for Mobile Agents: Issues and Requirements. In: Proc. 19th National Information Systems Security Conf. 1996, S. 591-591.
- [GMM99] *Guttman, Robert; Moukas, Alexandros; Maes, Pattie*: Agents as Mediators in Electronic Commerce. In: Klusch, Matthias (Hrsg.): Intelligent Information Agents. Springer-Verlag, Berlin 1999, S. 131-152.
- [Huck00] *Huck, Gerald*: The Jedi Tutorial. <http://www.darmstadt.gmd.de/~huck/jedi/tutorial/tutorial.html>, Abruf am: 2000-07-03.
- [KlMe99] *Klettke, Meike; Meyer, Holger*: Managing XML documents in object-relational databases. Universität Rostock 1999.
- [Klus99] *Klusch, Matthias (Hrsg.)*: Intelligent Information Agents: Agent-Based Information Discovery and Management on the Internet. 1. Aufl., Springer-Verlag, Berlin 1999.
- [Krul96] *Krulwich, B.*: The BargainFinder agent: Comparison price shopping on the internet. In: Williams, J. (Hrsg.): Bots and Other Internet Beasts. SAMS.NET, 1996.
- [Merz99] *Merz, Michael*: Electronic Commerce, Marktmodelle, Anwendungen und Technologien. dpunkt-Verlag, Berlin 1999.
- [Matt98] *Mattern, Friedmann*: Mobile Agenten. In: it+ti - Informationstechnik und Technische Informatik. 1998, S. 12-17.
- [MGM98] *Moukas, Alexandros; Guttman, Robert; Maes, Pattie*: Agent-mediated Electronic Commerce: A MIT Media Laboratory Perspective. MIT Media Laboratory 1998.
- [RuNo95] *Russel, Stuart J.; Norvig, Peter*: Artificial Intelligence: A Modern Approach. Prentice Hall, London 1995.
- [RuSt87] *Runyon, K.; Stewart, D.*: Consumer Behavior. 3. Aufl., Merrill Publishing Company, 1987.
- [RoPo97] *Rothermel, Kurt; Popescu-Zeletin (Hrsg.)*: Mobile Agents, First International Workshop, MA 97. Berlin 1997.
- [Shop00] *Shop.de*: Verzeichnis von Online-Shops. <http://www.shop.de>, Abruf am: 2000-08-15.
- [W4F00] *World Wide Web Wrapper Factory Website*: <http://db.cis.upenn.edu/W4F>, Abruf am: 2000-07-16.
- [Weis00] *Weiss, Gerhard (Hrsg.)*: Multiagent Systems: A Modern Approach to Distributed Artificial Intelligence. 2. Aufl., MIT Press, Cambridge (Massachusetts) 2000.
- [XML00] *World Wide Web Consortium*: Extensible Markup Language 1.0. <http://www.w3.org/TR/1998/REC-xml-19980210>, Abruf am: 2000-08-10.
- [XMLS00] *World Wide Web Consortium*: W3C Architecture Domain, XML Schema. <http://www.w3.org/XML/Schema.html>, Abruf am: 2000-08-10.
- [XSLT00] *World Wide Web Consortium*: XSL Transformations 1.0. <http://www.w3.org/TR/xslt/>, Abruf am: 2000-08-10.

Frank Siegemund, Clemens H. Cap, Andreas Heuer

On the employment of mobile agents and XML for the comparison of offers in B2C-applications

Abstract: Electronic markets are not transparent. Applications for product investigation help customers to compare and select offers and to order chosen products. This paper introduces architectures for such applications; they are compared on the basis of selected requirements. Particular effort is being made to investigate how the use of mobile agents and XML can guarantee higher flexibility, novel functionality, and easier adaptability to new stores and product groups. Approaches for the extraction and comparison of offers from online stores, and their ordering are analyzed. An application that was developed within the scope of this paper shows the feasibility of selected concepts. It uses mobile agents and XML to compare the offers of three German online shops.

Keywords: comparison-shopping agents, mobile agents, XML, online stores, electronic commerce

Zusammenfassung (nur für's WWW):

Der elektronische Markt ist intransparent. Anwendungen zur Angebotsrecherche unterstützen potenzielle Käufer bei der Gegenüberstellung und Auswahl von Angeboten und der Bestellung selektierter Produkte. Diese Arbeit schlägt Architekturen für solche Anwendungen vor, die auf der Basis ausgewählter Anforderungskriterien miteinander verglichen werden. Insbesondere wird darauf eingegangen, wie mobile Agenten und XML in einem solchen Anwendungsumfeld eingesetzt werden können, um erhöhte Flexibilität, neuartige Funktionalität und leichtere Anpassbarkeit der Anwendungen an neue Anbieter und Produktgruppen zu gewährleisten. Es werden Ansätze sowohl für die Angebotsextraktion und die Gegenüberstellung der Angebote als auch für das Ordern der Waren untersucht. Eine Implementierung stellt exemplarisch dar, wie ausgewählte Konzepte der Arbeit realisiert werden können. Sie zeigt, wie sich durch die Umsetzung der Untersuchungsergebnisse die Nutzerfreundlichkeit und Transparenz im E-Commerce erhöhen lässt.

Stichworte: Angebotsrecherche, mobile Agenten, XML, Online-Shops, Electronic Commerce