

Matchmaking in Web Services

Seminararbeit
von
Stefan Graber
aus
Mannheim

vorgelegt am
Lehrstuhl für Praktische Informatik
Prof. Dr. W. Effelsberg
Fakultät für Mathematik und Informatik
Universität Mannheim

August 2004

Betreuer: Marcel Busse

Inhaltsverzeichnis

Abkürzungsverzeichnis.....	II
Abbildungsverzeichnis.....	III
Tabellenverzeichnis.....	IV
Vorwort.....	V
1. Einleitung.....	1
2. Definitionen von Matchmaking.....	2
3. Allgemeine Anforderungen.....	3
4. Matchmaking im Framework des W3C für Web Services.....	5
5. Weitere Anforderungen ans Matchmaking im Kontext von Web Services.....	7
6. Suchsysteme.....	11
6.1 Keywordbasierte Suchsysteme.....	11
6.2 Frameworkbasierte Suchsysteme.....	13
7. Zusammenfassung und Ausblick.....	15
Literaturverzeichnis.....	VI

II

Abkürzungsverzeichnis

DAML	DARPA Agent Markup Language
HTTP	Hypertext Transfer Protocol
OIL	Ontology Interface Language
RDF	Resource Description Framework
SOAP	Simple Object Access Protocol
TFDIF	Term Frequency Times Log of Inverse Document Frequency
WSDL	Web Services Description Language
UDDI	Universal Description, Discovery and Integration
URL	Uniform Resource Locator
WWW	World Wide Web
XML	Extensible Markup Language

III

Abbildungsverzeichnis

Abbildung 1	Match.com, eine Kontaktseite
Abbildung 2	Webserver, Kampfhubschrauber und Indianer
Abbildung 3	Schematischer Ablauf eines Web Services
Abbildung 4	Das Webportal von Google

IV

Tabellenverzeichnis

Tabelle 1	Überprüfung auf Teilmatches
Tabelle 2	Das Angebot der Bleicker GmbH dargestellt durch ein Frame
Tabelle 3	Die Nachfrage der Firma Elektro Lutz modelliert als Frame

Vorwort

Diese Ausarbeitung ist Teil des "Teleseminar Web Services", das in Kooperation der Lehrstühle von Prof. Dr. W. Effelsberg, Universität Mannheim sowie von Prof. Dr. H. Schmeck und Prof. Dr. H. Hartenstein, beide Universität Karlsruhe, abgehalten wurde und hat "Matchmaking in Web Services" zum Thema. Des Weiteren umfasst sie nur den ersten Teil, der sich hauptsächlich mit grundlegenden und weiteren Anforderungen an das Matchmaking beschäftigt. Anschließend werden Überlegungen über zwei Suchsysteme angestellt, die aber aus unterschiedlichen Gründen nicht vollends überzeugen können. Im zweiten Teil, für den ich auf die Seminararbeit meines Karlsruher Kommilitonen Joachim Jakobi verweise, wird das "Semantic Web" thematisiert und erläutert, warum es eine befriedigende Lösung bieten kann. Vertiefend wird hierbei auf RDF und DAML+OIL als "Ontology Languages" eingegangen.

Für einen groben Überblick über das Thema "Matchmaking in Web Services" empfiehlt sich der gemeinsame Seminarvortrag, der unter folgender URL zu finden ist:

<http://www.informatik.uni-mannheim.de/pi4/lectures/ss2004/seminar/vortraege/matchmaking.pdf>

1. Einleitung

Wenn man bei Google, der weltweit führenden Suchmaschine, "Matchmaking" als Suchbegriff eingibt, werden Unmengen von Partnerschaftsvermittlungsseiten als Treffer zurückgeliefert, auf denen jeder versuchen kann, den für sich am besten passenden Partner zu finden. Anhand persönlicher Kriterien wird sequentiell eine Liste von Steckbriefen anderer Benutzer, die potentielle Matches darstellen, durchlaufen. Wichtig ist hierbei, dass die Suche von vorneherein eingeschränkt wird, denn sonst ist es auf Grund der oft sehr großen Anzahl registrierter User einer Kontaktseite, die bei den führenden Seiten in die Million geht, nahezu unmöglich einen passenden Partner zu finden. Wie Abbildung 1 zu entnehmen ist, stellt beispielsweise Match.com ein Interface zur Verfügung, welches sowohl die Präferenzen des Suchenden als auch die seiner möglichen Zielpartner berücksichtigt. So stehen neben der Auswahl, ob eine Frau oder ein Mann gesucht werden soll, noch Einschränkungen wie Altersspanne oder Hobbys zur Verfügung. Dadurch kann von vornherein ausgeschlossen werden, dass in der Liste beispielsweise Partner des gleichen Geschlechts oder einer völlig anderen Generation auftauchen, die sowieso nicht in Frage kämen.



Abbildung 1: Match.com, eine Kontaktseite

Des Weiteren ist zu beachten, dass jeder Ersteller eines Steckbriefes ebenfalls seine Präferenzen mit angibt um so zu gewährleisten, dass er nur in Listen von Suchenden auftaucht, die mit seinen Vorstellungen eines potentiellen Matches vereinbar sind.

2. Definitionen von Matchmaking

Wie am Beispiel der Kontaktseite zur erkennen war, ist das Matchmaking dafür verantwortlich, dass nur potentiell passende Partner in einer überschaubaren Zielmenge, das heißt idealerweise maximal 50 Kontakte, auftauchen. Diese Entscheidung trifft der Matcher, wie der Entscheidungsalgorithmus auch genannt wird, hauptsächlich auf Basis der angegebenen Präferenzen, aber unter Umständen werden auch weitere Einschränkungen vorgenommen, welche sich in der Anordnung der Treffer widerspiegeln.

Allerdings ist der Prozess des Matchmaking nicht immer einfach: Wenn ein Dienst, wie beispielsweise "Lieferung von drei Pizzen innerhalb zwei Stunden ins Mannheimer Schloss" in Anspruch genommen werden soll, dann muss das Matchmaking sehr viel mehr als nur einen einfachen Abgleich von Präferenzen, leisten können. Dies zeigt sich auch in den zahlreichen Definitionen von Matchmaking, die sich in verschiedenen Papern finden lassen:

Matchmaking is defined as a process that requires a repository host to take a query or advertisement as input, and to return all advertisements which may potentially satisfy the requirements specified in the input query or advertisement. [1]

Matchmaking is the process by which parties that are interested in having exchange of economic value are put in contact with potential counterparts. [2]

The process of searching the space of possible matches between demands and supplies can be defined as Matchmaking. Notice that this process is quite different from simply finding, given a demand a perfectly matching supply. [3]

Es fällt auf, dass alle Definitionen sehr abstrakt gehalten sind und nicht beschreiben, was genau unter "Input" zu verstehen ist. Ebenso wird nicht darauf eingegangen, wie es zu einer Entscheidungsfindung kommt. Es ist lediglich von einem Prozess die Rede, der dafür sorgt, dass in einer Ergebnismenge nur noch mögliche Partner (potential counterparts) enthalten sind. Auch wird nicht darauf eingegangen, nach was gesucht wird, was im Umkehrschluss bedeutet, dass es keinerlei Einschränkungen gibt.

Abstrakt gesehen handelt es sich beim Matchmaking also um einen Prozess, der eine Menge A mit einer sehr viel größeren Menge B vergleicht, auf Überschneidungen prüft und alle Erfüllungen von B, die A entsprechen, an den Anwender zurückliefert.

3. Allgemeine Anforderungen

Des Weiteren verfügt jeder Mensch über einen eigenen Matcher: Sein Gehirn. Es hilft dabei, die alltäglichen Probleme lösen, allerdings hat es den entscheidenden Nachteil, dass die Menge an Informationen, die es auf Abruf bereithalten kann, beschränkt und zudem noch von der Tageszeit abhängig ist. Für einen Bibliothekar ist es beispielsweise unmöglich, jederzeit zu wissen, wo alle seine Bücher und Werke sind. Deshalb kommen an dieser Stelle EDV-Systeme ins Spiel, welche auf Grund ihrer in der Praxis nahezu unbegrenzten großen Datenbanken fähig sind, Informationen über jedes Buch zu speichern, zu verwalten und jederzeit Auskunft geben zu können. Dies hat jedoch zur Folge, dass Überlegungen anzustellen sind, wie das Matchmaking konkret realisiert werden könnte, und welche zu bestimmten Anforderungen führen, die nun anhand eines Beispiels dargestellt werden:

Angenommen man möchte eine Diplomarbeit über IBMs erfolgreichen Webserver Apache schreiben und sich zur Hilfe passende Literatur aus einer Bibliothek heraussuchen, dann steht man vor dem Problem, das nicht bekannt ist, wo in der Bibliothek, die oft mehrere tausend Werke umfasst, sich nützliche Literatur befindet. Es ist jetzt zu entscheiden, ob entweder alle Regale abgelaufen werden sollten, um sich so Buch für Buch vorzunehmen, oder ob das EDV-basierte Suchsystem benutzt werden sollte, welches die Bibliothek zur Verfügung stellt.

Weil die erste Variante aus Zeitgründen nicht in Betracht kommt, gibt man, da das Suchsystem der Bibliothek ein schlüsselwortbasiertes Suchsystem ist, als Suchbegriff "Apache" ein und erhält alle Werke, die die Zeichenkette "Apache" im Titel beinhalten. Des Weiteren erfährt man, ob überhaupt Informationen über "Apache" verfügbar sind. Es wäre natürlich durchaus denkbar gewesen, dass über das gesuchte Thema in der Bibliothek nichts vorhanden gewesen wäre und man bei der ersten Variante, also beim Ablaufen der Regale, viel kostbare Zeit verloren hätte. Dies setzt allerdings voraus, dass das Ergebnis der Suche möglichst schnell geliefert wird, nicht das am Ende, falls das EDV-basiertes Suchsystem etliche Minuten brauchen würde, noch eine Indifferenz zwischen den Alternativen bestünde. Somit stellt das Einhalten einer zeitkritischen Schranke, die je nach Anwendung unterschiedlich sein kann, eine erste Anforderung an das Matchmaking dar, welche vor allem in der Praxis eine sehr wichtige Rolle spielt: Kein Nutzer von Amazon.de würde über 10min warten, nur um mitgeteilt zu bekommen, ob das Buch, das er gerne bestellen möchte, auch vorrätig auf Lager ist. Aus diesem Grund ist die Antwortzeit einer

Anfrage für Amazon.de, wie auch für viele andere Firmen, ein wichtiger unternehmensspezifischer Erfolgsfaktor.

Ausgehend davon, dass die Suche nach "Apache" drei Treffer zurückgeliefert hat, werden folgende Werke als Treffer angezeigt: Das erste Werk beschreibt die Architektur und Funktionen von IBM's Webserver Apache. Das Zweite stellt veröffentlichte technische Daten des US-Kampfhubschraubers Apache und dessen Einsatzzwecke dar und das Dritte handelt von Leben und Kultur der Apache-Indianer:



Abbildung 2: Webserver, Kampfhubschrauber und Indianer [4]

Es fällt sofort auf, dass zwar alle drei "Apache" im Titel tragen, inhaltlich aber völlig verschieden sind und es somit, da nur Informationen zu dem Webserver gewünscht waren, zu zwei falschen Treffern gekommen ist. Abstrakt gesehen handelt es sich hierbei um das Problem der falschen Semantik bei gleicher Syntax, welches natürlich vermieden werden sollte und weshalb vom Matcher gefordert werden muss, dass dieser gewährleistet, alle falschen Treffer herauszufiltern.

Ein sehr ähnliches Problem ist, dass Semantisch korrekte Treffer vergessen werden, weil sie über eine andere Syntax verfügen: Ein Werk mit dem Titel "IBM Webserver" führt somit, weil das Schlüsselwort nur "Apache" beinhaltet, bei der Suche zu keinem Treffer.

Zusammengefasst erhält man drei wesentlichen Anforderungen an das Matchmaking:

1. Erfüllung einer zeitkritischen Schranke beim Suchen
2. Minimierung von semantisch falschen Treffern
3. Keine semantisch korrekte Treffer vergessen

Entscheidend für ein gutes Matchmaking ist es nun, diese drei Anforderungen miteinander in Einklang zu bringen, was sich in der Praxis als nicht ganz unproblematisch herausstellen wird.

4. Matchmaking im Framework des W3C für Web Services

Bevor die Frage zu klären ist, wie das Matchmaking bei Web Services effizient ausgestaltet werden kann, gilt es zu definieren, was unter Web Services zu verstehen ist. Ebenso ist zu zeigen, wie das Framework des W3Cs aufgebaut ist und wo der Prozess des Matchmaking abläuft.

Definiert werden Web Services als

Dienst, der mit Hilfe von XML auf der Basis von Internet-Netzwerkprotokollen erbracht wird und drei wichtige Teile der Zusammenarbeit abbildet: Zusammenfinden, Binden und Datenaustausch.[5]

Oder auch:

Web Services im engeren technischen Sinn meint automatisierte Kommunikation zwischen Applikation übers Internet. Es werden also nicht HTML-Seiten zu einem Webbrowser geschickt, die von einem Menschen betrachtet werden, sondern Programme tauschen Daten und starten auf entfernten Rechnern Funktionen. [6]

Letztlich läuft es darauf hinaus, dass Web Services Methoden sind, die über ein Netzwerk auf entfernten Rechner zur Verfügung stehen. Wie dies im Framework des W3C aussehen sollte, veranschaulicht Abbildung 3: Ein Dienst, den ein Service Provider zur Verfügung stellen will, wird von einem Service Broker veröffentlicht und kann dort von einem Service Klienten gefunden und anschließend beim Service Provider ausgeführt werden.

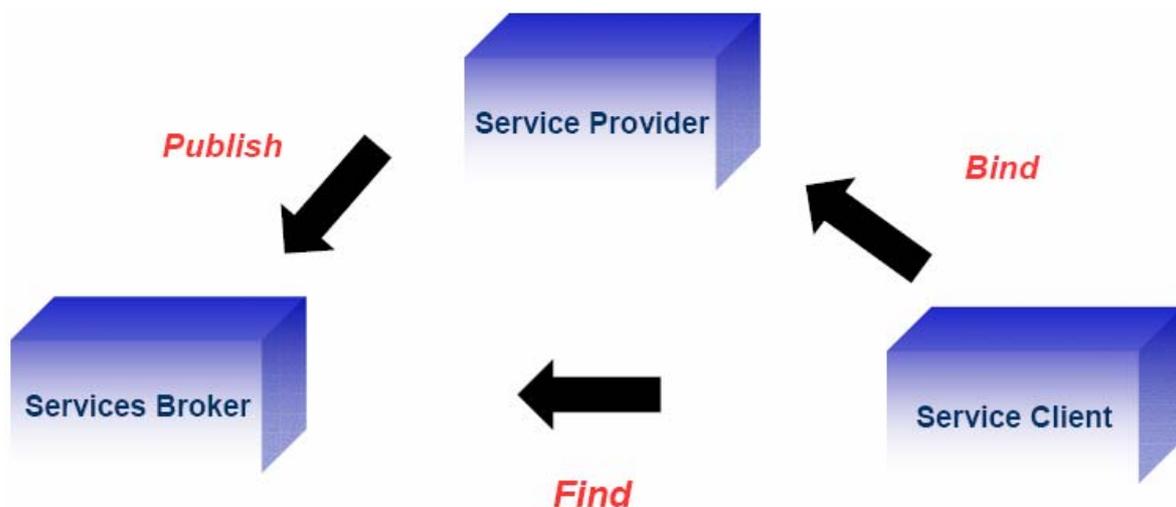


Abbildung 3: Schematischer Ablauf eines Web Services [7]

Damit dies realisiert werden kann bedarf es Standards, da sonst die Interoperabilität im

Zusammenhang mit Web Services zwischen den einzelnen Systemen der beteiligten Parteien nicht gewährleistet werden könnte. Aus diesem Grund definierte der W3C, in Zusammenarbeit mit den führenden Softwareherstellern (Microsoft, IBM, ...), folgende Standards für Prozeduren und Protokolle:

SOAP (Simple Object Access Protocol)
WSDL (Web Services Description Language)
UDDI (Universal Description, Discovery and Integration)

Alle drei basieren auf XML (Extensible Markup Language) und ermöglichen, das bei Web Services benötigte Zusammenspiel von "Datenaustausch", "Zusammenfinden" und "Binden". So wird ein erfolgreicher Datenaustausch zwischen den einzelnen Parteien mit Hilfe von SOAP über HTTP gewährleistet; man bezeichnet SOAP daher auch als "Umschlag für ein XML-Dokument". Für das Binden von Webservices wird WSDL verwendet: hierbei wird ein Service durch WSDL formal beschrieben; vor allem Schnittstellen, Nachrichtenstrom-Formate und Funktionsaufrufe. Das Zusammenfinden übernimmt UDDI, ein webbasiertes Informationssystem für Web Services. Es bietet zudem ein Verzeichnis von Adress- und Produktdaten sowie von Anwendungsprogrammierschnittstellen (WSDL-Dateien). Entscheidend ist, dass UDDI selbst ein Web Service ist, der dank seines Verzeichnisses zum Veröffentlichen und Auffinden anderer Web Services fungiert.

Zusammengefasst wird immer dann ein Matchmaking durchgeführt, wenn nach einem passenden Web Service beim Service Broker gesucht wird. Des Weiteren findet der Prozess des Matchmaking im UDDI Protokoll statt.

5. Weitere Anforderungen ans Matchmaking im Kontext von Web Services

Web Services sind also Dienste, die über ein Netzwerk auf entfernten Rechnern zu Verfügung stehen, aber vorher erst gefunden werden müssen. Es besteht dementsprechend gegenüber dem Bibliotheksbeispiel der vor allem Unterschied, dass Dienste nicht nur aus einem treffenden Begriff bestehen, sondern sich vor allem über seine Funktionen und Restriktionen definieren. Dies wird nun anhand eines Beispiel-Web Services, wie er im Bereich des E-Commerce durchaus denkbar wäre, dargestellt:

Die Bleicker Schrauben + Dübel GmbH bietet Schrauben zum Verkauf an und garantiert dem Käufer, dass die Schrauben bis spätestens 2 Tage nach Eingang der Bestellung deutschlandweit durch den UPS Transportservice ausgeliefert werden. Für den Fall, dass die Lieferung schadhaft oder nicht pünktlich sein sollte zahlt sie eine maximale Konventionalstrafe von 10.000 Euro. Weil die Bezahlung nicht per Vorkasse erfolgen muss, erwartet sie, dass der Käufer über eine Mindestkreditwürdigkeit von ‚AA‘ verfügt. Des Weiteren verlangt sie eine Mindestbestellmenge von 5000 Stück, einen Preis von mindestens 20 Euro je 1000 Schrauben und eine Mindestversandpauschale von 20 Euro. Anschließend platziert sie das Angebot bei einem Service Broker.

Im Kern besteht dieser Web Service also aus der Dienstleistung, Waren des Typs Schrauben innerhalb von zwei Tagen zu einem bestimmten Ort zu liefern. Kompliziert wird er allerdings durch die zahlreichen Restriktionen, die sowohl untere Schranken wie "mindestens 20 Euro für je 1000 Schrauben" als auch obere Schranken "maximale Konventionalstrafe in Höhe von 10.000 Euro" haben können. Problematisch ist vor allem, dass zwar einige dieser Restriktionen beim Käufer gerade gespiegelt sind, andere wiederum nicht und für manche existiert überhaupt kein Gegenpart. Dies wird anhand einer Anfrage eines potentiellen Käufers verdeutlicht:

Die Firma Elektro Lutz möchte innerhalb von 3 Tagen 10.000 Schrauben nach Mannheim geliefert bekommen. Sie ist bereit, bis zu 23 Euro je 1000 Schrauben zu bezahlen. Des Weiteren möchte sie gerne eine Liefergarantie haben. Im Falle einer verspäteten oder gar ausgefallenen Lieferung verlangt sie eine Mindestkonventionalstrafe von 8.000 Euro.

Erwartungsgemäß setzt die Firma Elektro Lutz als Käufer beim Preis eine maximale obere Schranke, welche in diesem Fall mit der Unteren des Verkäufers vereinbar ist. Gleiches gilt für die maximale Frist von bis zu drei Tagen sowie das Ziel der Auslieferung. Schwieriger ist die Berücksichtigung der Versandpauschale von 20 Euro, da der Käufer hierzu keine Angabe gemacht hat. Es ist nun Aufgabe des Matcher dieses Problem zu lösen; denkbar wäre beispielsweise Folgendes: Es wird geprüft, ob ein potentieller Käufer eine Angabe hierzu

gemacht hat, falls nicht, dann wird die Bleicker Schrauben + Dübel GmbH nicht weiter berücksichtigt. Dass diese Lösung nicht befriedigend sein kann, lässt sich schnell erkennen: Der Käufer ist bereit 23 Euro für je 1000 Schrauben zu zahlen, seine Bestellmenge liegt bei 10.000 Schrauben. Das heißt er wäre bereit maximal 230 Euro zu bezahlen. Der Verkäufer verlangt mind. 20 Euro je 1000 Schrauben und zusätzlich noch die Versandpauschale; also mind. 220 Euro. Das bedeutet, dass es zu einem (Teil-)Match kommen müsste, da sich beide Seiten über den Preis einig sind.

Denkbar wäre aber auch, dass bei einem Angebot nur variable Kosten angegeben werden dürften. Das hätte zur Folge, dass die Bleicker Schrauben + Dübel GmbH ihren Mindestpreis für Schrauben auf 24 Euro erhöhen müsste, um bei 5000 bestellten Schrauben den gleichen Gewinn zu erzielen:

$$5.000 \times 20 \text{ Euro} / 1000 \text{ Stk} + 20 \text{ Euro} = 5.000 \times 24 \text{ Euro} / 1000 \text{ Stk}$$

Problematisch ist allerdings, dass der Käufer bei einem variablen Schraubenpreis von 24 Euro wiederum nicht bestellen würde, da die maximale Schranke von 23 Euro überschritten worden wäre. Als Konsequenz hierauf prüft die GmbH, ob sie den variablen Preis auf 22 Euro senken und dafür die Mindestbestellmenge auf 10.000 Schrauben anheben sollte: Zwar würde nun die Anfrage des Käufers ein Match ergeben und zustande kommen, aber dies hätte zur Konsequenz, dass viele kleinere Kunden, deren Bestellmenge kleiner als 10.000 wäre, als potentielle Kunden der Bleicker Schrauben + Dübel nicht mehr in Frage kommen würden.

Zusammenfassend lässt sich festhalten, dass es beim Matchmaking nicht nur darauf ankommt, dass mittels Direktvergleich zu jeder Angabe das passende Match gefunden wird, sondern dass auch die Logik, die dahinter steckt, verstanden werden muss. Man spricht an dieser Stelle davon, dass ein Matcher die Semantik verstehen muss, um entscheiden zu können, ob Käufer und Verkäufer miteinander in Einklang zu bringen sind. So wäre es hier sinnvoll wenn ein Matcher auf Gesamtkostenbasis seine Entscheidung trifft: (Pseudo-Code)

```

if (Bestellmenge >= Mindestbestellmenge)
    if (Bestellmenge x MaxPreisKäufer >= Bestellmenge x varPreis + FixBetrag)
        matched();
    else
        notmatched();
else
    notmached();

```

Dies bedeutet jedoch, dass der Matcher den Zusammenhang zwischen der Versandpauschale und

dem Preis pro 1000 Schrauben erkennen können müsste. Wie sich das konkret realisieren lässt, wird aber erst in späteren Kapiteln thematisiert, denn zunächst einmal muss noch auf ein weiteres wichtiges Problem eingegangen werden, das das Matchmaking bei Web Services erschwert: Es handelt sich hierbei um die Garantiewünsche, die beide Seiten formuliert haben:

Verkäufers	Käufers	Match ja/nein
Lieferung nach Deutschland	Mannheim	ja
2 Tage Lieferzeit	Maximal 3 Tage	ja
Mindestmenge 5.000	Bestellmenge 10.000	ja
MinKreditwürdigkeit. AA	Keine Angabe	???
Min. 20€/ 1000 Schrauben	Max. 23€/ 1000 Schrauben	ja
20€Mind.Versandpauschale	Keine Angabe	???
Max. 10.000 Euro	Min. 8.000 Euro	ja
Konventionalstrafe	Konventionalstrafe	
Keine Angabe	Garantie für Lieferung	???

Tabelle 1: Überprüfung auf Teilmatches

Auf den ersten Blick scheint dieses Problem äquivalent zum ersten, da auch hier eine Restriktionsanforderung gestellt wurde, dem keine Angabe gegenüber steht. Dies soll hier aber nicht betrachtet werden, da es ähnlich dem Obigen gelöst werden kann.

Die Problematik die sich hier ergibt ist, dass bei einer Transaktion über Web Services Risiken auftreten, die jeweils eine Seite zu tragen hat. Beispielsweise liefert der Verkäufer seine Ware fristgerecht aus, aber erhält erst einige Zeit später die vereinbarte Entlohnung. Ebenso vereinbart der Käufer mit einem ihm normalerweise völlig unbekanntem Unternehmen, dass die Lieferung innerhalb einer bestimmten Frist zu erfolgen hat und unterliegt ebenfalls dem Risiko der Nichteinhaltung der getroffenen Vereinbarungen. Im Beispiel verlangt die "Bleicker Schrauben + Dübel GmbH" deswegen von potentiellen Käufern ein Mindestkreditrating von AA, allerdings ist die Firma Elektro Lutz nur ein kleines Mittelständiges Unternehmen, das, wie es bei einem Grossteil vergleichbarer Unternehmen der Fall sein dürfte, über kein Rating verfügt. Dies ist insofern kritisch zu betrachten, da sich Web Service vor allem an genau diese Kundengruppen richten. Da davon auszugehen ist, dass ein Service Provider nicht auf seine Mindestanforderung an das Rating möglicher Kunden verzichten möchte, bedeutet dies für einen Service Broker, der letztlich der Eigentümer des eingesetzten Matchmakers ist, dass er diese Risiken übernehmen müsste, falls er keine Umsatzeinbußen erleiden möchte. Dies wiederum hat zur Folge, dass er die

Kosten, die bei seiner Risikoübernahme entstehen, auf die Kunden umlegen müsste, was wiederum auf Grund des Prinzips der Wirtschaftlichkeit nicht unbegrenzt machbar ist. Ergo muss der Matcher so konfiguriert werden, das dieser ein verträgliches Maß findet: Hierbei es denkbar, für Kunden eine Art internes Rating anzulegen und anhand dessen zu entscheiden.

Analoges gilt auch für die Provider: Es macht keinen Sinn, einem Käufer, der auf eine Frist von 3 Tagen Lieferzeit angewiesen ist, einen Provider als positives Match anzubieten, wenn dieser zwar laut Angabe in 2 Tagen liefern kann, sich in der Vergangenheit aber gezeigt hat, dass er diese Frist hin und wieder nicht einhalten kann.

Zusammengefasst bedeutet das, dass ein Matchmaking bei Web Services nicht nur die Matches zurückzuliefern hat, die vollständig übereinstimmen, sondern das auch Teilmatches notwendig sind.

Eine letzte Anforderung an das Matchmaking im Kontext von Web Services ergibt sich dadurch, dass es erwünscht ist, Anfragen maschinell zu stellen. Man denke beispielsweise an eine Lagerhaltungssoftware, die, sobald der Bestand an Schrauben unter eine vorgegebene Mindestmenge fällt, neue Schrauben bestellen soll. Damit eine solche Transaktion reibungslos funktionieren kann, muss die Software alle für ein Matchmaking benötigten Daten zur Verfügung stellen. Dies hat zur Konsequenz, dass eine Standardisierung benötigt wird, die durch den Einsatz von WSDL, SOAP und UDDI, geleistet werden kann. Für das Matchmaking ist hierbei vor allem wichtig, dass nur ein einziges Ergebnis zurückgeliefert wird, welches so strukturiert sein muss, das die Software den Service des Providers in Anspruch nehmen kann.

Des Weiteren stellt die letzte Anforderung gleichzeitig auch den wesentlichen Vorteil von Web Services gegenüber traditionellen Verfahren dar: Zwar ist es schon seit einiger Zeit machbar, dass eine Lagerhaltungssoftware automatische Bestellungen aufgeben kann, jedoch ist immer ein fester Vertragspartner vorgegeben, der die gewünscht Ware nur noch ausliefert. Es wird vor allem nicht überprüft, ob die gewünschten Leistungen nicht günstiger am Markt zu erwerben sind. Und eben diese Überprüfung findet bei vergleichbaren Web Services dadurch statt, dass sie bei einem Service Broker veröffentlicht werden und sich somit im Konkurrenzkampf behaupten müssen.

6. Suchsysteme

In den beiden folgenden Unterkapiteln wird jetzt vorgestellt, inwiefern die keywordbasierten bzw. die framebasierten Suchsysteme die zusammengetragenen Anforderungen an das Matchmaking bei Web Services erfüllen können.

6.1 Keywordbasierte Suchsysteme

Der bekannteste Vertreter der keywordbasierten Suchsysteme ist wohl die Suchmaschine Google, die sich weltweit durchgesetzt hat. Nach Eingabe von einem oder mehreren Schlüsselbegriffen, auch "keywords" genannt, durchsucht sie alle von ihr erfassten Seiten des WWW und prüft, ob diese die gesuchten Schlüsselwörter beinhalten und liefert sie gegebenenfalls als Treffer zurück. Allerdings weist sie einige wichtige Unterschiede gegenüber der oben vorgestellten Suchmaschine der Bibliothek auf: Erstens sucht sie für eingegebene Schlüsselwörter entsprechende Synonyme (Antonyme, Hyperonyme und Hyponyme) heraus und bezieht diese in die Suche mit ein. So werden z.B. für das Schlüsselwort "HTML" die Begriffe "Script", "Netscape", "Tutorial" und "Tag" als Synonyme verwendet, was zur Folge hat, dass zwar die Forderung, möglichst keine semantisch korrekten Treffer zu vergessen, weitestgehend erfüllt wird, dies jedoch auf Kosten des Kriteriums nach Minimierung der Zurücklieferung falscher Treffer geschieht. Dieses Problem versucht Google dadurch in den Griff zu bekommen, dass überprüft wird, ob die gesuchten Keywords auch auf verlinkten Seiten vorkommen, allerdings werden trotzdem noch viele falsche Treffer zurückgeliefert. Suchen man beispielsweise einen Pizza Service in Mannheim und gibt deshalb "Pizza + Service + Mannheim" ein, so liefert Google in 0.61 Sekunden 14.000 Treffer zurück. Eine von Altavista.com, einer ebenfalls keywordbasierten Suchmaschine, verwendete Technik namens TFIDF, bei der berechnet wird, wie häufig ein Schlüsselwort vorkommt und dies beim zurückliefern berücksichtigt, führt ebenfalls zu keinem wesentlichen besseren Ergebnis; im Gegenteil, dieselbe Suche führt hier sogar zu 20.900 Treffern. Interessant ist hierbei vor allem, dass die Reihenfolge, in der die Treffer beider Suchmaschinen angezeigt werden, nicht nur unterschiedlich ist, sondern dass es bei Google zu korrekten Treffern kommt, die Altavista nicht anzeigt, was wahrscheinlich auf die größere

Datenbasis zurückzuführen ist. Auch Rankings und Sponsoring spielen hierbei eine Rolle.

Als erstes Zwischenergebnis lässt sich festhalten, dass Google immerhin zwei der drei allgemeinen Anforderungen an das Matchmaking erfüllt. Die Minimierung falscher Treffer auf Grund gleicher Syntax bei falscher Semantik, lässt sich dadurch in Griff bekommen, dass nur die ersten Treffer betrachtet werden, was allerdings zur Folge hat, dass die Korrektheit dieser Treffer gewährleistet sein muss. Dies lässt sich in der Praxis allerdings genauso wenig realisieren wie die Web Service spezifischen Anforderungen: Zum einen ist es, um auf das Pizza Beispiel zurückzukommen, nicht möglich einer Software eine URL zurückzuliefern mit Hilfe derer sie auf der entsprechenden Webseite bestellen kann. Denn zum einen ist jede Webseite anders aufgebaut und strukturiert und zum anderen ist es nicht möglich nach komplexeren Diensten zu suchen, da, wie auf Abbildung 4 zu erkennen ist, nur eine Zeile für die Eingabe zur Verfügung steht. Selbst eine einfache Suche wie "Liefere innerhalb von zwei Stunden drei Pizzen ins Mannheimer Schloss" scheitert daran, dass alle Dienste in Schlagwörter gefasst werden müssten, was schlichtweg nicht möglich ist. Denn schon die Eingabe von "Pizza + Auslieferung + Mannheim" liefert vollkommen falsche Treffer, obwohl nur auf eine Restriktion eingegangen wurde.



Abbildung 4: Das Webportal von Google

An dem zuletzt genannten Problem scheitern letztlich alle keywordbasierten Suchsysteme, was auf deren Design zurückzuführen ist. Eine Suche mittels Schlüsselwörtern ist, wie der Name schon besagt, darauf ausgelegt, durch Schlagwörter nach einfachen Dateien, Webseiten oder ähnlichen Items zu suchen. Da dieses System aber nicht die Semantik einer Eingabe versteht, also nicht interpretieren kann, welche Bedeutung die Schlüsselbegriffe haben, scheidet es für ein Matchmaking im Kontext von Web Services aus.

6.2 Framebasierte Suchsysteme

Eine erste Möglichkeit Semantik aufzunehmen besteht darin, einen Web Services durch einen Frame zu modellieren. Hierbei werden Eigenschaften, wie beispielsweise Restriktionen, durch vorher festgelegte Attribute beschrieben, weshalb in diesem Zusammenhang auch von einem so genannten "pre-enumerated" Vokabular[8] gesprochen wird. Dieses vordefinierte Vokabular stellt den größten Fortschritt gegenüber der keywordbasierten Suche dar: Dadurch, dass die Syntax der Attribute festgelegt ist, kann der Matcher relativ einfach überprüfen, ob Nachfrage und Angebot zueinander passen. Dies wollen wir nun wiederum anhand des Schraubenbeispiels verdeutlichen.

Bevor ein Service Provider ein Angebot bei einem Service Broker, der ein framebasiertes Suchsystem einsetzt, platzieren kann, ist es notwendig, dieses als Frame zu modellieren:

Beschreibung	Schraubenverkauf+Lieferung
Anbieter	Bleicker Schrauben + Dübel GmbH
MinBestellMenge	5000
MinPreisPro1000	20,00
MinVersandpauschale	20,00
MinKreditWürdigkeit	AA
MaxZeitBisAuslieferung	48
Max. Konventionalstrafe	10.000
LieferungNach	Deutschland

Tabelle 2: Das Angebot der Bleicker GmbH dargestellt durch ein Frame

Hierbei werden die einzelnen Restriktionen als Tupel aus Attribut und Ausprägung formuliert und anschließend in eine Liste eingefügt. Entscheidend ist, dass alle Attribute sowohl für die Service Angebote als auch für die späteren Anfragen vordefiniert sind. So müsste die Bleicker Schrauben + Dübel GmbH, bei der Modellierung ihres Angebot als Frame, sowohl darauf achten, dass sie als Bezeichnung für "Mindestpreis je Schraube" das Attribut "MinPreisPro1000" verwendet, als auch, dass die Angabe des Preises für 1000 Schrauben und in Euro erfolgen muss. Des Weiteren werden die genaue Syntax eines Schraubenverkaufs und Lieferungsservices und das Format der anderen Ausprägungen vorgegeben.

Wenn jetzt die Firma Elektro Lutz ihre Anfrage, die in Tabelle 3 als Frame formuliert ist, beim Service Broker stellen würde, so würde der Matcher wie folgt überprüfen, ob es zu einem Match

kommt:

Es wird das erste Tupel (Beschreibung, Schraubenverkauf+Lieferung) der Anfrage ausgelesen; anschließend wird geprüft, ob bei den abgelegten Angeboten ein Frame dabei ist, bei dem dieses Tupel ebenfalls vorkommt, was in diesem Fall zu einem Match führt.

Beschreibung	Schraubenverkauf+Lieferung
Käufer	Elektro Lutz
BestellMenge	5000
MaxPreisPro1000	23,00
MaxVersandpauschale	-
KreditWürdigkeit	-
MaxZeitBisAuslieferung	72
Min. Konventionalstrafe	8.0000
LieferungNach	Mannheim

Tabelle 3: Die Nachfrage der Firma Elektro Lutz modelliert als Frame

In der Praxis werden zwar jetzt die meisten Web Service aussortiert worden sein, aber für die übrig gebliebenen, grundsätzlich richtigen Dienste, muss nun überprüft werden, ob sie zu einem vollständigen Match führen. Das wesentliche Vorgehen ist hierbei analog: Dadurch, dass die Attribute vordefiniert sind, erkennt der Matcher welche Tupel zu vergleichen sind. So wird beispielsweise bei der Bestellmenge überprüft, ob der Ausdruck ($BestellMenge \geq MinBestellMenge$) ein "true" ergibt.

Als Ergebnis werden anschließend alle Treffer in Abhängigkeit ihrer Übereinstimmung angezeigt, wobei ein Matcher eine untere Schranke, wie z.B. mind. 90%, vorgeben kann.

Es lässt sich also festhalten, dass die framebasierten Suchsysteme theoretisch durchaus geeignet sind, um nach Web Services zu suchen. Zwar bringt das vordefinieren der Attribute das Problem mit sich, dass nur Web Services bei einem Broker hinterlegt werden können, die auch unterstützt werden, jedoch überwiegt hier der Vorteil, das sich sowohl Anbieter als auch Kunde anschauen können, wie ein Web Service "Schraubenverkauf+Lieferung" auszusehen hat. Dadurch wird die Problematik, dass bei einer Anfrage nicht auf alle Attribute eingegangen wird, merklich eingeschränkt. Außerdem spricht der weltweite Erfolg der Relationalen Datenbank Management Systeme (RDMS) für die Verwendung dieses Suchsystems, da sich die Suche dort recht einfach implementieren lässt. Wohl auch aus diesen Gründen basierten die ersten Versionen von UDDI auf dieser Suchtechnik, jedoch traten in der Praxis einige Probleme auf, die letztendlich dazu

geführt haben, dass es sich nicht durchsetzen konnte.

Sehr problematisch ist beispielsweise, dass es durch die Zerlegung von teilweise sehr komplizierten Funktionen von Web Services in Attribute und Ausprägungen zu dem "Impedance Mismatch" ähnlichen Problem kommt: Ein Frame müsste so unter Umständen über einige 100 Tupel verfügen, um einen komplexen Web Service vollständig darstellen zu können, was zwar theoretisch machbar, aber praktisch nicht zu realisieren ist: Man stelle sich hier vor, die Komplexität einer Software wie SAP's R3 durch Tupel darzustellen.

Aber auch bei der Suche nach einfachen Web Services stellt sich das Problem, dass jeder Broker seine Attribute selbst definieren kann und ein Kunde deshalb seinen Anfrageframe umschreiben müsste, um bei einem Broker eine Anfrage stellen zu können. Dies führt dazu, dass eine Firma oft nur die Dienste eines oder nur wenigen Brokern in Anspruch nimmt und es deshalb durchaus passieren kann, dass das günstigste Angebot teurer ist, als es bei einem festen Hauslieferanten der Fall wäre, was den Sinn von Web Services ad absurdum führen würde.

Anzumerken ist allerdings, dass diese "Kundenunfreundlichkeit" durchaus gewollt ist, da die Service Broker hierdurch versuchen ihre Kunden zu binden. Verstärkt wird dies dadurch, dass zusätzliche Funktionen, in Form von Attributen, angeboten werden, über die die Konkurrenz nicht verfügt und die einen späteren Brokerwechsel deutlich erschweren. Letztendlich endet es in starkem Wildwuchs, der auf Kosten der Nutzer von Web Services geht, da nicht sichergestellt werden kann, dass zum Einen der gesuchte Web Service wirklich der günstigste ist, noch das die Gebühren ihres Brokers minimal sind.

7. Zusammenfassung und Ausblick

Als Fazit lässt sich ziehen, dass sowohl die keywordbasierten als auch die framebasierten Suchsysteme nicht vollends geeignet sind, die Suche nach Web Services zu gewährleisten.

Als mögliche Lösung kommt die Einführung eines so genannten "Semantic Web"[9] in Betracht, welches auf Standards wie RDF bzw. DAML+OIL basiert. Hier kann das Matchmaking, unter Zuhilfenahme so genannter "Process Ontology's", derart konfiguriert werden, dass alle Anforderungen zufrieden stellend erfüllt werden können.

Für weitere Details verweise ich an dieser Stelle auf die Seminararbeit meines Karlsruher Kommilitonen Joachim Jakobi.

Literaturverzeichnis

- [1] Lei Li, Ian Horrocks. *A Software Framework For Matchmaking Based on Semantic Web Technology*. Department of Computer Science, University of Manchester. WWW2003, May 20-24, 2003, Budapest, Hungary.
- [2] Javier González-Castillo, David Trastour and Claudio Bartolini. *Description Logics for Matchmaking of Services*. HP Laboratories Bristol, UK 2001.
- [3] Tommaso Di Noia, Eugenio Di Sciascio, Francesco M. Donini, Marina Mongiello. *A System for Principled Matchmaking in an Electronic Marketplace*. WWW2003, May 20-24, 2003, Budapest, Hungary.
- [4] Uwe Krüger. *Semantic Web, Eine Wegbeschreibung zum WWW 2.0*. Seminararbeit im Seminar "Neue Technologien im Internet und WWW, Wintersemester 2003/04, Universität Jena
- [5-7] Can Okutan, Sascha Schnauffer. *Einführung in Web Services (Seminarvortrag)*. Universität Karlsruhe (TH), Universität Mannheim, 2004
- [8] Prieto-Diaz, R., *Implementing faceted classifications for software reuse*. 12th International Conference on Software Engineering, 1990
- [9] Mark Klein, Abraham Bernstein. *Searching for Services on the Semantic Web Using Process Ontologies*. Center for Coordination Science, Massachusetts.