

**Universität Kaiserslautern**

AG Grafische Datenverarbeitung und  
Computergeometrie

**Proseminar Visualizing the Semantic Web**

WS 2004/05

**Web Services und Recommender Systeme**

Einführung

**Ralf Kretschmann**

Kaiserslautern, den 18.09.2004

**Inhaltsverzeichnis**

<b>INHALTSVERZEICHNIS</b>	<b>2</b>
<b>0 VORWORT</b>	<b>4</b>
0.1 Inhaltsüberblick	4
0.2 Kurze Einführung in Semantic Web	5
<b>1 WEB SERVICES</b>	<b>6</b>
1.1 Einführung in Web Services	6
1.1.1 Einführung	6
1.1.2 SOAP, WSDL, UDDI	6
1.1.3 Schichtenmodell der Kommunikation	7
1.2 SOAP	7
1.3 WSDL	12
1.4 UDDI	13
1.5 Anwendungsbeispiel	15
1.6 Zusammenfassung und Ausblick	19
<b>2 RECOMMENDER SYSTEME</b>	<b>20</b>
2.1 Einführung in Recommender Systeme	20
2.2 Geschichtliche Entwicklung	21
2.3 Verschiedene Arten von Recommender Systemen	22
2.3.1 Content Based Filtering	22
2.3.2 Collaborative Based Filtering	22
2.3.3 Regelbasiertes Filtern	23
2.3.4 Wissensbasiertes Filtern	23
2.3.5 Demographisches Filtern	23
2.4 Input des Benutzers	24
2.5 Techniken für Content Based Filtering	25

2.5.1	Prinzip	25
2.5.2	Beispiel: Filme	25
2.5.3	Exact Match	26
2.5.4	Best Match	26
2.5.5	Bewertung	27
2.6	Techniken für Collaborative Based Filtering	27
2.6.1	Prinzip	27
2.6.2	Beispiel: Filme	27
2.6.3	Speicherbasierte Algorithmen	28
2.6.4	Modellbasierte Algorithmen	28
2.6.5	Bewertung	29
2.7	Beispiele	29
2.8	Zusammenfassung und Ausblick	30
3	LITERATURVERZEICHNIS	31

## 0 VORWORT

### 0.1 Inhaltsüberblick

Diese Arbeit wurde für das Proseminar „Visualizing the Semantic Web“ der AG von Prof. Hagen (Universität Kaiserslautern) im Wintersemester 2004/05 erstellt. Sie behandelt die Themen „Web Services“ und „Recommender Systems“.

In dieser Arbeit soll eine Einführung dieser beiden Themen im Gesamtkontext des Semantic Webs gegeben werden. Dazu wird auf der nächsten Seite grob umrissen, was Semantic Web bedeutet.

Web Services werden in Kapitel 1 behandelt. Es besteht aus einer kurzen Einführung, einer Erklärung der Technologien der Web Dienste und geht als Beispiel tiefer auf SOAP, WSDL und UDDI ein. Es hat zur Aufgabe, die Prinzipien von Web Services aufbauend auf dem Semantic Web zu verdeutlichen, und untersucht, wie Entwickler diese Technologien nutzen können und welche zukünftigen Einflüsse Web Services haben könnten.

Kapitel 2 enthält ein neues Thema und informiert über Recommender Systeme. Es sollen Methoden zur Erzeugung von Web Empfehlungen erläutert werden, die vor allem auf Collaborative Filtering also ein Filtern von Informationen aufbauen, und gezeigt werden, wie die Schnittstellen dieser Systeme und prinzipielle Algorithmen modelliert sind und wie sie mit dem Semantic Web zusammenarbeiten.

Als direkte Literaturvorlage dienen Kapitel 9 und 10 des Buches „*Visualizing the Semantic Web*“ von *Vladimir Geroimenko, Chaomei Chen*. Zusätzliche Literaturen in Buchform oder als HTML-Dokumente sind im Literaturverzeichnis angegeben.

## **0.2 Kurze Einführung in Semantic Web**

Da die Themen der folgenden Arbeit sich auf das Semantic Web stützen, wird hier vorab die Frage grob geklärt: Was ist Semantic Web überhaupt? Was ist die Motivation für das Semantic Web?

Das Semantic Web (deutsch semantisches Netz; auch als semantisches Web bezeichnet) ist eine Erweiterung des WWW um maschinenlesbare Daten, welche die Semantik der Inhalte formal festlegen.

Informationen sollen zusätzlich zu der für Menschen lesbaren Form auch formal, in einer für Maschinen verarbeitbaren Form repräsentiert werden, damit Programme darauf operieren können.

Zwar verweist die wortwörtliche Übersetzung "Semantisches Netz" auf eine Verwandtschaft mit der Theorie der semantischen Netzwerke (semantic nets), es handelt sich jedoch um eine konkrete, in der Entwicklung befindliche Implementierung eines verteilten semantischen Netzes. Die Annotation der HTML/XML-Seiten im Web geschieht z.B. mittels Wissens-/Ontologie-Repräsentationssprachen wie RDF oder dem darauf aufbauenden OWL.

Nach Berners-Lee et. al. Artikel im Scientific American (2001-05) ist das Semantic Web eine Erweiterung des herkömmlichen Webs in der Informationen mit eindeutigen Bedeutungen versehen werden, um die Arbeit zwischen Mensch und Maschine zu erleichtern: "The Semantic Web is an extension of the current web in which information is given well-defined meaning, better enabling computers and people to work in cooperation" (ebenda).

Zusammenfassend ausgedrückt ist das Ziel des Semantischen Webs die durch das Web übertragenen Daten mit Bedeutungsinformationen (Semantik) anzureichern für die Verarbeitung durch Maschinen und für die Nutzung von Menschen. Um dies zu erreichen kommen Techniken (Logik, Ontologien, ..) zum Einsatz, die bisher in dem Bereich der K.I. verwendet wurden.

## **1. WEB SERVICES**

### **1.1 Einführung in Web Services**

#### **1.1.1 Einführung**

Der Begriff Web Services bezeichnet den heute modernen Ansatz zur Realisierung verteilter Anwendungen. Web Services verfolgen das Ziel, Anwendungsfunktionalität mit Hilfe (mehr oder weniger) standardisierter Komponenten und Protokolle über das Internet verteilt zur Verfügung zu stellen.

So gelten auch von anderen Programmen wieder verwertbare Dienste, die über das Internet angeboten werden, als Web Services. Denkbar sind hierbei Such- und Übersetzungsfunktionen, Rechtschreibprüfungen und die Authentifizierung (z.B. Microsoft Passport), die in eigene Programme eingebaut werden können.

Von der Funktion gibt es solche Webanwendung schon seit langer Zeit, man braucht nur an die verschiedenen Suchmaschinen zu denken. Allerdings sind diese auf die Bedienung durch den Menschen zugeschnitten.

#### **1.1.2 SOAP, WSDL, UDDI**

Um die Verwendung auch für Programme möglich zu machen, sind folgende Funktionen erforderlich:

- eine kompatible Übermittlung der Informationen (SOAP)
- eine verwertbare Beschreibung des Dienstes (WSDL)
- ein Verzeichnis vorhandener Dienste (UDDI)

Unter Verwendung dieser drei Technologien könnte eine Suchanfrage eines Programms so aussehen:

- 1.) das Programm fragt in einem UDDI Verzeichnis nach Diensten, die die Suchanfrage beantworten
- 2.) von diesen Diensten wählt es einen aus und holt sich davon eine WSDL-Beschreibung, wie der Dienst zu verwenden ist
- 3.) mit diesen Informationen nimmt das Programm nun per SOAP-Nachricht Kontakt zu dem Dienst auf und erhält das Ergebnis der Suche zurück.

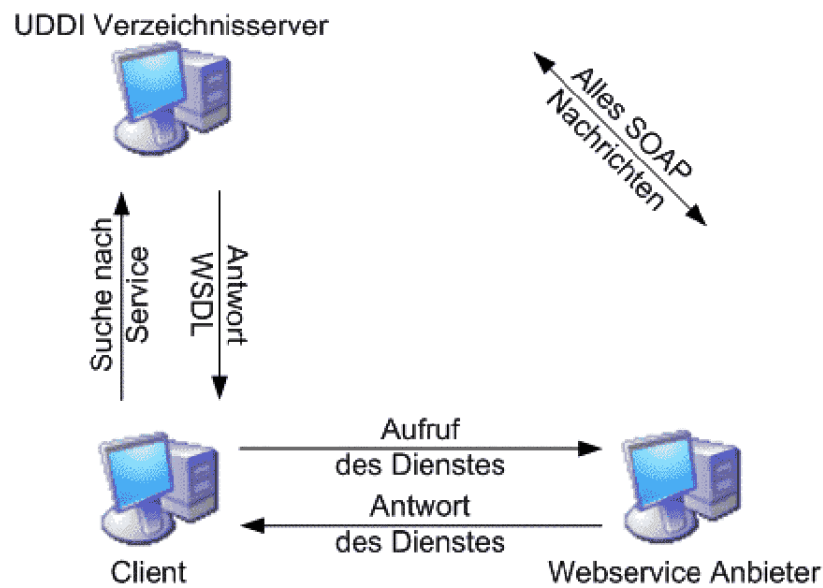


Abb. 1 Zusammenspiel SOAP-WSDL-UDDI

Das W3C definiert Web Services als Softwareanwendungen die durch einen URI (Uniform Resource Identifier) eindeutig benannt sind und deren Schnittstellen mit Hilfe von XML-Techniken definiert und genutzt werden können. Web Services interagieren durch XML-basierte Nachrichten, die über Internetprotokolle ausgetauscht werden.

Die Grundidee von Web Diensten besteht in der anwendungsübergreifenden und organisationsübergreifenden Kommunikation. Ein wichtiger Teil dessen sind entfernte Methodenaufrufe, bei denen die Datenübermittlung über Internet- oder Webtechniken geschieht; meist HTML oder auch die tiefer liegende Schicht TCP. Dabei entsteht die Forderung nach einer Möglichkeit für die Spezifikation von Sicherheitsaspekten, Routing, Management und anderen höherwertigen Kommunikationssemantiken.

### 1.1.3 Schichtenmodell der Kommunikation

Die Technik des Aufbaus der Kommunikation von Web Diensten lässt sich genauer mit folgendem Schichtenmodell beschreiben:

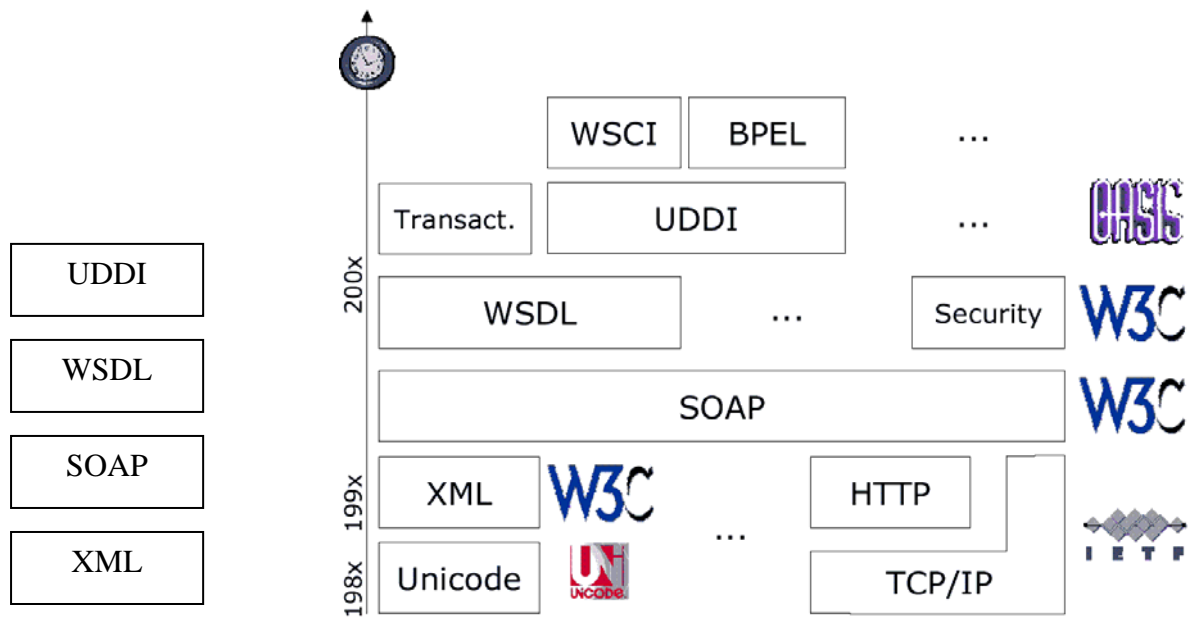


Abb.2 Schichtenmodell für Webdienste und zeitliche Entwicklung der Standards

In diesem Modell dient als Grundlage XML zur Inhaltsdarstellung, das XML-Schema zur Inhaltsbeschreibung, SOAP zur Verpackung und Übertragung des Inhaltes, WSDL zur technischen Beschreibung der Schnittstelle und UDDI zur Ablage der Schnittstellen.

Eine Methode zur Konzeption und Realisierung von verteilten Anwendungen, die unabhängig von den zugrunde liegenden Betriebssystemen oder Implementierungssprachen einen Datenaustausch erlaubt, ist die Service Orientierte Architektur (SOA):



Abb. 3 Schichtenmodell für Prozesse basierend auf Dienste



Zur Realisierung werden vollständige Anwendungen oder Teile daraus als Dienste angeboten, die ohne Kodierungsaufwand genutzt werden können. Diese Anwendungen basieren auf den Techniken des Semantic Webs: Prozesse greifen auf die durch WSDL beschriebenen Dienste mit entfernten Methodenaufrufe (RPCs) zu, benutzen zur Übertragung des Inhalts SOAP, die SOAP-Erweiterungen stellen höherwertigen Kommunikationssemantiken wie Sicherheit, korrekte Übertragung, .. sicher und letztendlich geschieht die Kommunikation auf der untersten Ebene durch bekannte Protokolle wie HTTP.

## 1.2 SOAP

### SOAP (Simple Object Access Protocol)

Das Simple Object Access Protocol ist mittlerweile in der Version 1.2 spezifiziert und wird wahrscheinlich mit nur geringen Änderungen zum vom W3C standardisierten XML Protocol Version 1.0 übernommen werden. In Verwendung befindet sich zurzeit auch noch die Version 1.1, die im Mai 2000 Ariba, CommerceOne, Compaq, HP, IBM, Microsoft und SAP beim W3C eingereicht wurde. Die anfängliche Entwicklung startete Microsoft schon 1997. Bei SOAP geht es generell um den Nachrichtenaustausch von XML-Dokumenten, meist zum entfernten Methoden-, Prozeduraufruf auf anderen Rechnern. Für Typdeklarationen bietet es sich an, XML Schema zu nutzen. Der eigentliche Transport findet meist über HTTP statt. Das hat den Vorteil, dass der Verkehr nicht von Firewalls geblockt wird. Aber auch der Austausch über FTP, SMTP oder andere Protokolle ist machbar.

```
<s:Envelope xmlns:s=http://www.w3.org/2001/06/soap-envelope>
  <s:Header>
    ....
  </s:Header>
  <s:Body>
    ...
  </s:Body>
</s:Envelope>
```

In dem Beispiel sieht man den generellen Aufbau einer SOAP-Nachricht. An dem Namespace kann man erkennen, dass es sich im Beispiel um eine SOAP Version 1.2 Nachricht handelt. Bei Version 1.1 wäre der Namespace `http://schemas.xmlsoap.org/soap/envelope/`. Header und Body dürfen nur wohlgeformten namespace qualifizierten XML Code enthalten. Im Body stehen die eigentlich zu übertragenden Nutzdaten. Der Header ist optional und kann somit auch entfallen. Bei der Verwendung von http als Transportprotokoll würde der ganze Envelope in eine entsprechende POST Nachricht gesetzt.

```
<n: quadriere xmlns:n="urn:Rechnen">
<zahl xsi:type="xsd:int">
  2
</zahl>
</n: quadriere>
```

Um z.B. eine Java Methode `public int quadriere (int zahl)` korrekt aufzurufen, bedürfte es nur einem Standardrumpf - wie oben angegeben - mit dem Body, der die aufzurufende Methode und den Variablennamen und Wert übergibt.

An Alternativen gibt es zwar die schon länger etablierten Standards für Remote Procedure Calls wie CORBA (Common Object Request Broker Architecture), DCOM (Distributed Component Object Modell) auf Windows Systemen und RMI (Remote Method Invocation) in Java. Aus einer früheren SOAP Version hat sich der XML-RPC abgespalten. Diese haben aber den Nachteil keine wirklich plattform- und sprachübergreifende Kommunikation anzubieten. Entweder man ist auf Windows oder Java festgelegt, oder es fehlt an Implementierung. Wenn es nicht um den RPC-Aufruf sondern um Dokumentenaustausch geht, gibt es zwar auch bereits einen Standard namens EDI (Electronic Document Interchange). Dieser wird aber aufgrund seiner Komplexität nur von ein paar großen Firmen genutzt.

SOAP ist für alle relevanten Programmiersprachen erhältlich, Implementierungsbeispiele wären: Apache SOAP for Java, SOAP::Lite for Perl und Microsoft .NET. Damit ist eine Plattform- und sprachübergreifende Kommunikation möglich.

## 1.3 WSDL

### WSDL (Web Service Description Language)

Die Web Service Description Language wurde von IBM und Microsoft entwickelt und beschreibt, wie der Zugriff auf einen Web Service funktioniert. Der Client holt sich also die Anweisungen über das WSDL-Dokument, statt dass man ihm explizit mitteilen muss, wie er den Service aufrufen soll. Das Schreiben von Clients wird also dadurch wesentlich erleichtert.

So müssen auch für Änderungen am Server die Clients nicht mehr angepasst werden, sondern nur noch die entsprechende WSDL-Beschreibung.

WSDL entspricht damit der Interface Definition Language (IDL) von CORBA und COM.

```
Data Types
<wsdl:types/>
Messages
<wsdl:message/>
Interfaces
<wsdl:portType/>
Services
<wsdl:binding/>
<wsdl:service/>
```

Eine WSDL-Beschreibung ist ein XML Dokument. Die verschiedenen Teile der Beschreibung bauen von oben nach unten aufeinander auf. Zuerst können neue Datentypen deklariert werden, die später beim Austausch Verwendung finden. Es werden Nachrichten deklariert, die aus oben definierten oder Standarddatentypen bestehen. In einem Interface werden die Nachrichten zu ihrer Verwendung wie Ein-/Ausgabe zusammengefasst. Das Binding setzt die logischen Operationen zu physischen zusammen und definiert eine Kodierung für sie. Schließlich wird in Service eine oder auch mehrere Adressen, auf denen der Service verfügbar ist, definiert.

An Implementierungen gibt es u.a. das IBM Web Services Toolkit als Ergänzung zu Apache SOAP und das Apache Axis Projekt. Auch in Microsoft .NET und SOAP::Lite for Perl ist WSDL Unterstützung integriert.

Nachteil von WSDL ist, dass es nur eine technische Beschreibung des Dienstes liefert, nicht aber Informationen zur Bedeutung der Funktionen und Parameter. Hierfür gäbe es zwar eine Alternative in Form von DAML-S (DARPA Agent Markup Language – Services), die auch solche Informationen unterstützt. Ihr fehlt aber wie den meisten Technologien des Semantic Webs die Firmenunterstützung.

## 1.4 UDDI

### UDDI (Universal Description, Discovery and Integration)

Der UDDI-Standard wurde von Ariba, IBM und Microsoft entwickelt. Ein UDDI-Verzeichnis bietet eine Auflistung von Web Services und wo man mehr über sie erfährt. Man kann sich das Verzeichnis wie die Gelben Seiten vorstellen, nur anstelle der Telefonnummer/ Anschrift findet man hier einen Zeiger auf die WSDL-Beschreibung des Dienstes.

Außer einer Suche über den Namen der Firma, ist es natürlich auch möglich, nach einer bestimmten Dienstart zu suchen. Alle verwendeten Datenstrukturen werden als TMODEL bezeichnet. Die Identifizierung findet über eindeutig vergebene Schlüssel statt. So hat jede Firma ihren eigenen Businesskey und jeder Service einen ServiceKey. Diese Nummer heißen Universally Unique Identifiers (UUID). Der Zugriff auf das UDDI-Verzeichnis kann selbst wieder per SOAP und über die WSDL-Beschreibung des Verzeichnisses erfolgen. Diese sollte jedem Programm gleich mitgegeben werden.

●Inquiry API	●Publishers API
<ul style="list-style-type: none"> <li>&gt;Find things</li> <li>•find_business</li> <li>•find_service</li> <li>•find_binding</li> <li>•find_tModel</li> <li>&gt;Get Details</li> <li>•get_businessDetail</li> <li>•get_serviceDetail</li> <li>•get_bindingDetail</li> <li>•get_tModelDetail</li> </ul>	<ul style="list-style-type: none"> <li>&gt;Save things</li> <li>•save_business</li> <li>•save_service</li> <li>•save_binding</li> <li>•save_tModel</li> <li>&gt;Delete things</li> <li>•delete_business</li> <li>•delete_service</li> <li>•delete_binding</li> <li>•delete_tModel</li> <li>&gt;security...</li> <li>•get_authToken</li> <li>•discard_authToken</li> </ul>

Abb 4

API für den UDDI Zugriff

Zur Zeit ist noch UDDI Version 1 vom September 2000 in Benutzung. Öffentliche Verzeichnisse werden von IBM und Microsoft angeboten. Seit Juni 2001 ist bereits Version 2 verabschiedet und es sind mit HP und SAP bereits zwei weitere Anbieter hinzugekommen. Eine Standardisierung durch das W3C ist nach Version 3 geplant. Aber auch so sind in der UDDI-Organisation ([www.uddi.org](http://www.uddi.org)) bereits fast alle wichtigen Firmen vertreten und haben ihre Unterstützung angekündigt.

Außer den oben genannten öffentlichen Verzeichnissen steht es Firmen natürlich frei für ihr Intranet private UDDI-Verzeichnisse aufzubauen. Bei öffentlichen Verzeichnissen spielt es keine Rolle, welchen der Anbieter man wählt, da die Daten von allen auf die anderen repliziert werden. Unterstützt werden wieder alle wichtigen Programmiersprachen. So ist es in .Net und SOAP::Lite for Perl gleich enthalten oder es gibt einen Zusatz wie z.B. für Java mit IBM UDDI4J.

Nachteil des Verfahrens ist seine geringe Sicherheit. So kann sich jeder als noch nicht vertretene Firma ausgeben und dort Einträge anlegen.

Auch wird die Pflege der Datenbank mit der Zeit ein großes Problem werden, da der Nutzen eines Verzeichnisses mit längst verschwundenen Firmen eingeschränkt ist.

So waren nach einer Studie im November 2001 schon 48% aller Links fehlerhaft. Hier spielt natürlich noch der experimentelle Status der Technologie mit. So sind zwar bis jetzt sehr viele Firmen vertreten, aber meist nur mit ihren Kontaktinformationen. Sie bieten meist jedoch noch keinerlei Web Services an.

Als Alternative haben Microsoft und IBM selber die Web Service Inspection Language (WS-IL) entwickelt. Hier kennt man schon den Service Provider und sucht nur bei diesem nach angebotenen Diensten. Hier befindet sich einfach eine Datei namens `inspection.wsil` im root-Verzeichnis des Webservers.

## 1.5 Anwendungsbeispiel

Nachdem in Abschnitt 1.1 die Funktionsweise von Web Services mit ihren Bestandteilen grob umrissen und in 1.2 bis 1.4 detaillierter auf die einzelnen Protokolle eingegangen wurde, ist dieser Abschnitt einem Anwendungsbeispiel gewidmet, das das Zusammenspiel zwischen SOAP, WSDL, UDDI und weiteren Diensten, die zu einem lauffähigen Web Service notwendig sind, demonstriert.

Aufgabenstellung:

Implementiere einen einfachen Web Service, der zählt, wie oft in einer gegebenen Zeichenkette ein gegebenes Muster auftritt.

Letztendlich muss ein Programm das Problem lösen, dies tut folgendes Fragment eines JAVA Kodes:

```
public class CountPat {
    public int count(String string, String pat) {
        int cnt = 0, pos = string.indexOf(pat);
        while(pos >= 0) {
            ++cnt;
            pos = string.indexOf(pat, pos+pat.length());
        }
        return cnt;
    }
}
```

Listing 1 Java-Code CountPat.java

Die Methode count bekommt als Argumente die beiden Zeichenkette vom Typ String und liefert die Anzahl des Auftretens des Musters als Integer-Wert zurück.

Nun muss dieser Dienst mit einer SOAP-Anfrage angesprochen werden (hier in Version SOAP 1.1):

```
<?xml version = "1.0" encoding = "UTF-8"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <count>
      <arg0 xsi:type = "xsd:string">
        alphabetagammaalpha
      </arg0>
    </count>
  </SOAP-ENV:Body>
</SOAP-ENV:Envelope>
```

```

    <arg1 xsi:type = "xsd:string">
      alph
    </arg1>
  </count>
</SOAP-ENV:Body>
</SOAP-ENV:Envelope>

```

Listing 2 SOAP-Anfrage

Im <SOAP-ENV:Envelope .. > Header wird die Kodierung der Nachricht festgelegt, das Wichtige ist im Body der Nachricht: Er wird mit den Tags <count>..</count> (= Name der JAVA Methode) umschlossen und enthält die beiden Argumente für den Aufruf der Methode, die mit bezeichnenden und typzuweisenden Tags umschlossen werden.

Die SOAP-Nachricht wird über HTTP vom Sender zum Empfänger transportiert:

```

POST /axis/CountPat.jws http/1.0
Content-Length: 467
Host: localhost
Content-Type: text/xml; charset=utf-8
SOAPAction: "/count"

```

**SOAP-Anfrage Listing 2**

Listing 3 HTTP-Anfrage

An dieser http-Anfrage ist der SOAPAction Header hervorzuheben, der ebenfalls den Namen der auszuführenden Aktion des SOAP-Servers enthält. Die erste Zeile mit /axis/CountPat.jws adressiert obige JAVA Methode, die als Java Web Service in den HTTP-Server eingebunden ist. Näheres dazu kommt weiter unten.

Der SOAP-Server erhält die Anfrage, führt den Dienst aus und liefert als Ergebnis eine HTTP-Antwort:

```

http/1.1 200 OK
Content-Type: text/xml; charset=utf-8
Content-Length: 427
Date: Sun, 06 Jan 2002 16:00.00 GMT
Server: Apache Tomcat/4.0.1-b1 (HTTP/1.1 Connector)
<?xml version = "1.0" encoding = "UTF-8"?>
<SOAP-ENV:Envelope
  SOAP-ENV:encodingStyle = "http://schemas.xmlsoap.org/soap/encoding/"
  xmlns:SOAP-ENV = "http://schemas.xmlsoap.org/soap/envelope/"
  xmlns:xsd = "http://www.w3.org/2001/XMLSchema"
  xmlns:xsi = "http://www.w3.org/2001/XMLSchema-instance">
  <SOAP-ENV:Body>
    <countResponse>
      <countResult xsi:type = "xsd:int">3</countResult>
    </countResponse>
  </SOAP-ENV:Body>

```



```
</SOAP-ENV:Envelope>
```

Listing 4 HTTP-Antwort

Der SOAP-Client kann nun das Ergebnis der Anfrage entsprechend interpretieren und erhält die Zahl 3.

Ein SOAP-Server ist ein HTTP-Server, der zusätzlich eine Erweiterung zur Ausführung von Web Services hat in Form von z.B. Java Servlets oder JavaServer Pages. Apache Tomcat kann als solch ein Server benutzt werden. Dazu wird CountPat.java aus Listing 1 als CountPat.jws dem Server hinzugefügt. Der Speicherpfad dieses Java Web Services beim HTTP-Server muss wie in Listing 3 mit dem Pfad in der HTTP-Anfrage übereinstimmen.

Der SOAP-Client kann als JavaServer Page modelliert werden. Er sendet die HTTP-Anfrage, die die SOAP Nachricht enthält, und empfängt die Antwort des SOAP-Servers:

```
<%
String method = request.getParameter("method");
String arg1 = request.getParameter("arg1");
String arg2 = request.getParameter("arg2");
String endpoint = "http://localhost:8080/axis/CountPat.jws";
org.apache.axis.client.ServiceClient client =
    new org.apache.axis.client.ServiceClient(endpoint);
try {
    Object[] args = new Object[] {arg1, arg2};
    Object obResult = client.invoke("", method, args);
%>
    <html><body><h3>
        The result is: <% = obResult.toString() %>
    </h3></body></html>
<%
} catch(Exception e) {}
%>
```

Listing 5 SOAP Client CountPat.jsp

Die Funktionalität des SOAP Clients steckt in der Zeile `Object obResult = client.invoke("", method, args)`: hiermit wird eine Anfrage an den Server geschickt und das Ergebnis in `obResult` abgelegt. Der SOAP Client als JavaServer Page gehört zu einem Web Server und bekommt die Argumente `method`, `arg1`, `arg2` z.B. von einem Web Browser, in dem vom Benutzer ein Formular ausgefüllt wurde, mitgeteilt. Der SOAP Client stellt dann über den Web Server das Ergebnis in

Form einer Webseite dar. Folgende Abbildung verdeutlicht den gesamten Informationsfluss:

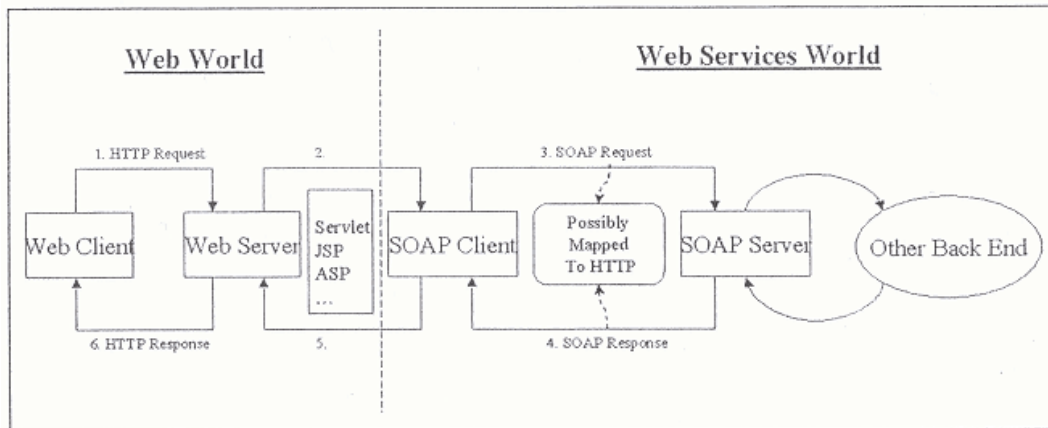


Abb. 5 Kommunikation zum Ausführen eines Web Services

Auf WSDL und UDDI wird hier nicht weiter eingegangen. Toolkits und Frameworks zur Entwicklung von Web Services enthalten automatisierte Dienste zur Erzeugung einer WSDL Beschreibung auf Basis der Implementierung des Services in einer Programmiersprache. Aus dieser Beschreibung kann ein SOAP Service generiert werden und aus ihm ein SOAP Client. Diese automatisch generierten Quellcodes sind zwar nicht komplett lauffähig, aber sie bilden einen nützlichen Startpunkt für die Fertigstellung eines Web Services.

## ***1.6 Zusammenfassung und Ausblick***

Die verschiedenen Protokolle werden sich wohl zumindest auf der Ebene ohne semantische Zusatzinformationen durchsetzen. So haben praktisch alle wichtigen Software-Firmen ihre Unterstützung der Standards angekündigt. Und mit der Standardisierung durch das W3C wird die Verbreitung wahrscheinlich noch einen großen Schub erfahren.

Letztendlich wird aber die Qualität der angebotenen Dienste über die Verbreitung entscheiden. Ein großes Problem stellen hierbei aber die noch nicht vorhandenen Standards zur Abrechnung dar. Denn ohne die Aussicht auf finanziellen Gewinn wird die Entwicklung leistungsfähiger Dienste auf sich warten lassen. Aber es wird schon an Erweiterungen gearbeitet. So gibt es u.a. WS-Security um mehr Sicherheit zu gewährleisten und das Web Service Choreography Interface (WSCI) um WSDL um komplexere Ablaufbeschreibungen zu erweitern.

Hierbei könnte es aber auch zu Problemen kommen, falls die Hersteller sich für unterschiedliche Technologien entscheiden, und so die Kompatibilität verloren geht.

## 2 Recommender Systeme

### 2.1 Einführung in Recommender Systeme

Unter Recommender Systemen, oder auch Empfehlensystemen, versteht man Systeme, die automatisch Empfehlungen generieren. Die Motivation dafür ist eine effizientere Nutzung der Informationsmenge des WWW, denn Internetrecherchen sind in vielen Fällen ohne fremde Hilfe nicht mehr machbar. Da Suchmaschinen Informationen nicht personalisiert finden und darstellen können, gibt es einen Bedarf an individuell gestaltete und zusammenhängende Informationen. Der Einsatz von Empfehlungssystemen ist besonders sinnvoll, wenn die Nutzerzahl und die Informationsmenge besonders groß sind. Diese Systeme sollen insbesondere Kunden bei der Informations- oder Produktauswahl unterstützen. Anwendungsgebiete sind Informations- und Nachrichtenportale, E-Commerce und E-Learning. Hierbei kann man unterschiedliche Recommender Systeme unterscheiden: Nicht-personalisierte und personalisierte Recommender Systeme.

Nicht-personalisierte Empfehlensysteme gelten für alle Benutzer die selben Empfehlungen aus. Beispiele hierfür sind Bestsellerlisten. Im Gegensatz dazu geben personalisierte Recommender Systeme Empfehlungen ab, die sich an einzelne Benutzer richten.

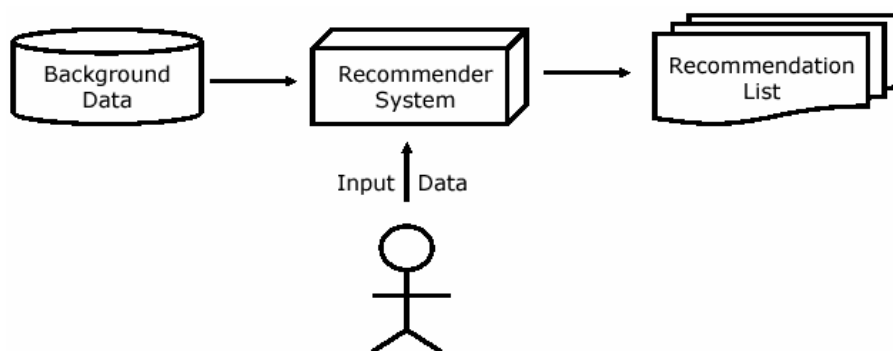


Abb. 6 Recommender Systeme schließen automatisch von vorhandenen Informationen auf neue Daten

## **2.2 Geschichtliche Entwicklung**

Recommender Systeme sind noch relativ jung. Sie haben sich aus den Decision Support Systems (DSS) entwickelt. Die Grundlagen der DDS oder auch Entscheidungs Unterstützung Systeme wurden schon Ende der 50er/Anfang der 60er Jahre entwickelt. Decision Support Systems sind Werkzeuge, die bei der Entscheidungsfindung unterstützen sollen. Durch sie können in Daten, in der Regel mit Hilfe von Data Mining, Strukturen gefunden werden und verschiedene Alternativen einer Problemlösung dargestellt werden. Im Gegensatz zu den Recommender Systemen wird aber keine Bewertung der einzelnen Möglichkeiten vorgenommen. DDS werden vor allem im Managementbereich eingesetzt.

Als den Erfinder des Collaborative Filtering kann man John Hey bezeichnen. Dieser meldete in den Jahren 1987 und 1989 zwei Patente an. Im ersten Patent beschrieb er grob das Verfahren von Collaborative Filtering; in dem zweiten beschrieb er den Zusammenhang dieses Algorithmus mit der automatischen Generierung von Empfehlungen. Allerdings nannte er das noch nicht Collaborative Filtering. Dieser Begriff wurde zum ersten Mal in einem Aufsatz über Tapestry, einem Mailfiltersystem der XEROX Palo Alto Research Group erwähnt. Bei Tapestry handelt es sich um ein Mailfiltersystem, das es jedem Benutzer ermöglicht, nach dem Lesen einer Email diese zu bewerten. Die Bewertungen fließen in das Filtersystem ein und ermöglichen dann das Aufstellen von Regeln, wie zum Beispiel „zeige nur Emails an, die Herr Maier als relevant bewertet hat“. Der Nachteil dieses Systems war, dass man die Vorlieben der anderen Benutzer kennen musste um vernünftige Ergebnisse zu erzielen. Durch das Forschungsprojekt Group Lens, das an der University of Minnesota 1992 gestartet wurde, war es erstmals möglich automatisch ähnliche Benutzer zu finden, wodurch das Recommender System nicht mehr auf Netzwerke begrenzt war, in denen jeder jeden kennt.

Im Jahr 1993 wurde ein weiteres Forschungsprojekt namens OpenSesame! entwickelt. Hierbei handelt es sich um einen Lernenden Agenten, der zunächst nur auf dem Mac das Benutzerverhalten aufzeichnete und nach wiederkehrenden Mustern suchte. Ein Beispiel für solch ein Muster wäre, dass der Browser jeden Tag um 9 Uhr gestartet wird. Dieses Konzept wurde später zur Anwendung in Netzwerken und für mehrere Benutzer erweitert und wird heute unter dem Namen LearnSesame! als Produkt angeboten.

Ein letzter Meilenstein war die Portierung von GroupLens auf das Usenet zur Bildung von Communities im Jahre 1996.

## **2.3 Verschiedene Arten von Recommender Systemen**

### **2.3.1 Content Based Filtering**

Beim Content Based Filtering, oder auch Eigenschaftsbasierten Filtern, hat seinen Ursprung im Information Retrieval und es wird nach Ähnlichkeiten zwischen Objekten gesucht. Das heißt, es werden Metadaten von Objekten mit einander verglichen. Um ein Objekt zu beurteilen wird es häufig in Beziehung zu anderen Objekten gesetzt, bei welchen der Benutzer/Kunde Interesse gezeigt hat. Diese Beziehung basiert auf Co-Kaufen Daten, Vorlieben des Kunden oder einer anderen Gruppe. Die Beziehung kann entweder automatisch hergestellt werden wegen eines konstanten Verhaltens des Kunden oder sie wird manuell hergestellt, wenn der Kunde explizit sein Interesse angibt. In beiden Fällen kann die Empfehlung auf kurzlebigen Daten basieren, da die Vergangenheit des Kunden nicht einbezogen wird.

Basiert ein Recommender System auf Content Based Filtering und interessiert sich ein Benutzer beispielsweise für einen Audi mit Allradantrieb, so empfiehlt das System auch Autos anderer Hersteller mit Allradantrieb. Content Based Filtering kommt vor allem für textbasierte Objekte in den Gebieten Informationsportale, Webseiten und Newsgroups zur Anwendung.

### **2.3.2 Collaborative Filtering**

Im Gegensatz zum Content-Based-Filtering wird beim Collaborative Filtering nicht direkt nach ähnlichen Objekten gesucht, sondern es wird nach Personen gesucht, die ähnliche Präferenzen haben, um dann Objekte zu empfehlen, die diese Person für gut bewertet hat. Also steht nicht die Klassifizierung von Objekten im Vordergrund sondern die Beziehung von Nutzern zu Objekten. Diese Beziehung entsteht durch Bewertung der Inhalte (implizit oder explizit) und Dokumente werden als ähnlich

eingestuft, wenn sie gleich bewertet wurden. Die Bewertung kann einerseits direkt durch den Benutzer erfolgen oder durch das System generiert werden.

Beispielsweise kann der Kauf eines Buches als eine positive Wertung interpretiert werden. Durch dieses Verfahren können auch subjektive Empfindungen in die Empfehlungen einfließen. Der Vorteil ist, dass diese Methode auch einsetzbar ist, wenn keine Beschreibungen der Objekte vorliegen. Die Anwendungsgebiete sind: E-Commerce, E-Learning und Informationen, Filme, Restaurants...

### **2.3.3 Regelbasiertes Filtern**

Eine andere Herangehensweise stellen regelbasierte Filterverfahren dar. Hier werden Regeln aufgestellt nach denen dann Empfehlungen generiert werden. Zum Beispiel könnte man eine Regel definieren, wonach Männern, die mindestens 60 Jahre alt sind und ein Haus besitzen, die Alarmanlage SecurityR5 empfohlen wird. Die Empfehlungen werden in Wenn-Dann-Regeln kodiert. Der Vorteil bei dieser Art ist ein leichtes Verständnis beim Erstellen des Systems, der Nachteil ist ein hoher Wartungsaufwand.

### **2.3.4 Wissensbasiertes Filtern**

Bei dem Wissensbasierten Filtern steht das funktionale Wissen, wie Objekte mit Bedürfnissen des Nutzers übereinstimmen, im Vordergrund.

### **2.3.5 Demographisches Filtern**

Demographische Recommender Systeme basieren auf Daten, die grundlegende Attribute des Kunden sind: Herkunft, Lebensalter, Region, Einnahmen, ... . Das System benutzt eine typische Benutzer-zu-Benutzer Korrelation, die sich von Collaborative Recommender System darin unterscheidet, dass es andere Informationen benutzt. Wenn ein Kunde z.B. einen Schnaps bei einer demographisch Recommendation orientierten E-Webseite kaufen möchte, nachdem

er im Formular seine demographischen Daten eingegeben hat, System kann ihm verschiedene Schnaps nach seinen Daten empfehlen. Wenn der Benutzer in Canada lebt, wird ein hochprozentiger Schnaps wegen Kälte möglicherweise eher empfohlen.

## **2.4 Der Input des Benutzers**

Der Input über den Zielbenutzer wird in den Recommendation Prozess eingezogen, um personalisierte Empfehlungen zur Verfügung zu stellen. Eine Anwendung, die keinen Input über den Ziel Kunden benutzt, kann nur unpersonalisierte Empfehlungen produzieren. Eine Implizite Navigation auf einer Webseite bezieht sich auf eine unbewusste Benutzung eines Recommendation Prozesses; während sich Kunden in Amazon.com die Produkte anschauen, empfiehlt ihnen das System die wahrscheinlich interessantesten Bücher.

Gegensatz zur impliziten Navigation geht es bei der expliziten Navigation um direkte Empfehlung an die Kunden gemäß ihrem Interesse. Z.B. in Drugstore.com bietet das System jedem Kunden eine Advisor an. Rating ist hier üblicherweise die Form von Recommendation. In der Regel gibt es 2 Rating Arten:

Die eine benutzt als Darstellung Ziffern. Webseiten werden mit 1 bis 5 bewertet, um z.B. die Intensität von Interesse zu zeigen: 1 ist sehr schlecht, 5 ist sehr gut. Eine andere Art benutzt Wahrheitswerte: 0, „mag ich nicht“; 1, „mag ich“.

Der Input von Kunden kann auch anderer Information sein, z.B. in CDNow.com kann der Kunde auch Attribute der Produkte eingeben, z.B.: „Ich brauche die CD mit einem Preis unter 10 EUR“

Die Kaufgewohnheit ist auch ein wichtiger Bestandteil für den Input an Informationen. Hier wird je Kunde ein Konto eröffnet und alle Einkaufsinformationen werden von der Webseite verzeichnet, um eine Datenbank dieses Kunden aufzubauen.



## 2.5 Techniken für Content-Based Filtering

### 2.5.1 Prinzip

Im Folgenden wird die grundlegende Vorgehensweise zur Erzeugung einer Empfehlung für Textdokumente geschildert.

1. Dazu müssen alle Inhalte, mit denen der Benutzer Kontakt hat, klassifiziert werden: Der Inhalt eines Textdokumentes, das sich der Benutzer anzeigen lässt, lässt sich durch seine Schlüsselwörter beschreiben.
2. Aus der Sammlung aus gewichteten Schlüsselwörtern wird ein Benutzerprofil erstellt, die sein Interessensgebiet beschreibt.
3. Weiterhin wird das Interessensprofil durch weiteren Kundeninput bestärkt: explizit durch z.B. Fragebögen oder Bewertungen, implizit durch z.B. die Verweildauer auf der Seite.
4. Nun wird der Inhalt aller Dokumente mit dem Benutzerprofil verglichen und anschließend das ähnlichste Dokument dem Benutzer vorgeschlagen.

### 2.5.2 Beispiel: Filme

Als Beispiel dient folgende Fragestellung: Der Benutzer hat den Film „Sieben“, welcher Film soll ihm empfohlen werden?

Film\Eigenschaft	Drama	Liebe	Gewalt	Humor	Action
Sieben	5	1	10	1	5
Hannibal	4	0	9	2	3
Titanic	8	10	1	0	2

Abb 7 Filme und ihre gewichteten Schlüsselwörter

Das Content-Based Filtering System wird aufgrund der Ähnlichkeiten der Beschreibung von „Hannibal“ mit dem Benutzerprofil diesen Film vorschlagen.

Die Ähnlichkeit zwischen einer Beschreibung und dem Benutzprofil kann auf zwei Wegen festgestellt werden:

- Exact-match: Es liefert nur Dokumente, die Anfrage vollständig erfüllen (Bsp. Boolean Matching)
- Best-match: Dokumente werden nach Relevanz geordnet (Bsp. Vektorraum Modell)

### 2.5.3 Exact Match

Exact-Match verwendet z.B. boolesche Operationen (AND, OR, NOT,..), bewertet jedes Dokument mit 0 oder 1 und schränkt somit die Auswahl der in Frage kommenden Dokumente stark ein.

### 2.5.4 Best-Match

Best-Match verwendet z.B. das Vektorraum Modell: Alle Dokumente und das Benutzerprofil werden im n-dimensionalen Raum als Vektoren dargestellt, wobei jede Dimension eine Eigenschaft/ Schlüsselwort darstellt. Nun wird der Abstand des Profilvektores und den Dokumentvektoren bestimmt, dieser Abstand gibt Auskunft über die Ähnlichkeit: Je geringer, desto ähnlicher.

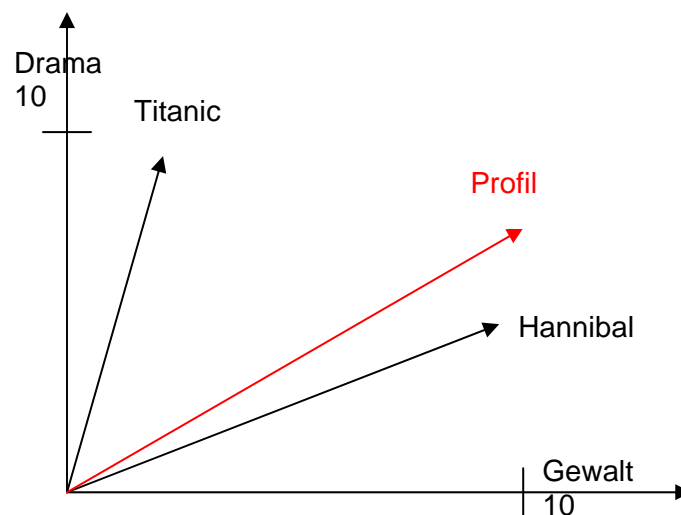


Abb 8 Vektorraum-Modell mit den Schlüsselwörtern Gewalt, Drama

Der Abstand Profil-Hannibal ist geringer als Profil-Titanic, deshalb wird bezüglich der beiden Schlüsselwörter „Hannibal“ empfohlen.

Im Vergleich zu Exact Match gibt es beim Best-Match eine größere Auswahl an Ergebnissen, die zusätzlich nach ihrer Ähnlichkeit zum Profil geordnet werden.

### **2.5.5 Bewertung**

Die Vorteile von Content-Based Filtering sind, dass Empfehlungen für Betreiber nachvollziehbar sind, in vielen Bereichen Kategorien schon gegeben sind und das System auch bei geringen Nutzerzahlen funktioniert.

Die Beschränkung auf textbasierte Objekte ist nachteilig, ebenso wie eine Überspezialisierung und dass keine subjektive Bewertungen erfasst werden.

## **2.6 Techniken für Collaborate Based Filtering**

### **2.6.1 Prinzip**

Im Folgenden wird die grundlegende Vorgehensweise zur Erzeugung einer Empfehlung für Objekte geschildert.

1. Es wird die Ähnlichkeit des aktiven Nutzers zu anderen Nutzern bestimmt.
2. Es werden Nutzer bestimmt, die sowohl mit Profil des aktiven Nutzers überlappen als auch mindestens ein weiteres Objekt bewertet haben.
3. Die Empfehlung wird berechnet, indem die Bewertungen der Nutzer gewichtet und aufsummiert werden.

Die Algorithmen zur Erstellung der Empfehlung können hier auch in zwei Klassen eingeteilt werden: Bei der **speicherbasierten** Vorgehensweise werden alle Berechnungen auf einer Datenmatrix durchgeführt, bei der **modellbasierten** Methode wird offline ein Modell erlernt und online nur dieses Modell anstatt der Datenmatrix verwendet.

### **2.6.2 Beispiel: Filme**

Fragestellung: Nutzer 1 bis Nutzer 4 haben Vorlieben für bestimmte Filme, welche Nutzer sind sich ähnlich?

	Terminator	Rambo	Titanic	Dirty Dancing	Star Wars
Nutzer 1	5	-	1	2	5
Nutzer 2	3	5	-	4	5
Nutzer 3	3	2	4	4	3
Nutzer 4	5	4	1	2	4

Abb. 9 Datenmatrix – Nutzer und ihre Filmwertungen

### 2.6.3 Speicherbasierte Algorithmen

Der 1. Schritt aus dem Abschnitt 2.5.1, die Bestimmung der Ähnlichkeit zu anderen Nutzern wird über die Distanz der Nutzer in der Matrix bestimmt, z.B. mit dem euklidischen Abstand. Das Ergebnis ist eine Abstandstabelle:

	Nutzer 2	Nutzer 3	Nutzer 4
Nutzer 1	0	-0,47	0,97
Nutzer 2		0,85	-0,32
Nutzer 3			-0,65

Abb 10 Abstandstabelle der Nutzer aus Datenmatrix

Aus der Abstandstabelle folgt: Nutzer 3 und Nutzer 4 sind sich sehr unähnlich, Nutzer 1 und Nutzer 4 sind sich sehr ähnlich. Nur Nutzer, dessen Abstand eine vorgegebene Schwelle überschreitet, werden für Schritt 2 benutzt. In Schritt 3 werden nun diese Nutzer in der Summenberechnung benutzt; als Gewicht für ihre Bewertung geht der Abstand mit ein.

### 2.6.4 Modellbasierte Algorithmen

Speicherbasierte Algorithmen sind zwar einfach, neue Daten können einfach hinzugefügt werden, aber die Datenbanken werden groß und die Algorithmen sind speicher- und zeitaufwändig.

Bei Modellbasierten Algorithmen wird die Datenmatrix genutzt, um offline ein Modell zu schätzen und zu erlernen. Dann wird online nur noch auf das Modell

zugegriffen. Es gibt verschiedene Verfahren zum Erstellen des Modells (Clusteranalyse, Bayessche Netze, Neuronale Netze,..) Der Vorteil dieser Algorithmen ist, dass Online-Berechnung deutlicher schneller ablaufen, der Nachteil, dass Informationsverlust durch die Modellbildung auftreten und das Hinzufügen neuer Daten ist aufwendig.

### **2.6.5 Bewertung**

Die Vorteile des Collaborate Filterings sind der geringer Initialaufwand, dass subjektive Eigenschaften berücksichtigt werden, es recht flexibel einsetzbar ist und die hochwertigen Empfehlungen (Cross-Genre).

Zum Nachteile ist, dass ein Minimum an Benutzerprofilen notwendig ist (New-User-Problem). Außerdem müssen Objekte bewertet sein (New-Item-Problem). Die Inhalte der Objekte werden beim Prozess nicht berücksichtigt und der Empfehlungsprozess ist nicht transparent.

## **2.7 Beispiele**

### **CDNOW**

Als Beispiel für die Umsetzung von verschiedenen Recommender Systemen dient der Onlineshop CDNOW. Er bietet dem Kunden folgende Empfehler:

**Buyer 's Guides:** In diesem Bereich kann der Kunde unter verschiedenen Genres wählen und sich die zugehörigen bedeutendsten Werke anzeigen lassen.

**Album Advisor:** Wählt der Kunde einen Artikel aus, so bekommt er eine Liste mit Artikeln, die andere Kunden gekauft haben, die ebenfalls den gewählten Artikel gekauft haben. Ferner kann man drei Interpreten eingeben, und bekommt dann anhand dieser Eingaben CDs empfohlen.

**My CDNOW:** My CDNOW ermöglicht es dem Kunden seinen eigenen Musikladen zu kreieren. Die Kunden können angeben welche Alben sie besitzen und können Alben bewerten. Anhand dieser Bewertungen werden dann Artikel empfohlen, die Kunden mit ähnlichem Geschmack gut bewertet haben.

Beim Buyer's Guide handelt es sich um eine nicht-personalisierte Empfehlung und der Album Advisor, sowie MY CDNOW sind Beispiele für Collaborative Filtering.

## Movielens

### Beispiel für Collaborative Filtering

Predictions for you ↕	Your Ratings	Movie Information	Wish List
★★★★	Not seen	<b>Butterfly Effect, The (2004)</b> <a href="#">info</a>   <a href="#">imdb</a> Drama, Sci-Fi, Thriller	<input type="checkbox"/>
★★★★	Hide this	<b>Identity (2003)</b> <a href="#">DVD</a> , <a href="#">VHS</a> , <a href="#">info</a>   <a href="#">imdb</a> Crime, Horror, Mystery, Thriller	<input type="checkbox"/>
★★★★	0.5 stars	<b>Salton Sea, The (2002)</b> <a href="#">DVD</a> , <a href="#">VHS</a> , <a href="#">info</a>   <a href="#">imdb</a> Crime, Drama, Thriller	<input type="checkbox"/>
★★★★	1.0 stars	<b>Red Dragon (2002)</b> <a href="#">DVD</a> , <a href="#">VHS</a> , <a href="#">info</a>   <a href="#">imdb</a> Crime, Mystery, Thriller	<input type="checkbox"/>
★★★★	1.5 stars	<b>Phone Booth (2002)</b> <a href="#">DVD</a> , <a href="#">VHS</a> , <a href="#">info</a>   <a href="#">imdb</a> Drama, Thriller	<input type="checkbox"/>
★★★★	2.0 stars	<b>One Hour Photo (2002)</b> <a href="#">DVD</a> , <a href="#">VHS</a> , <a href="#">info</a>   <a href="#">imdb</a> Thriller	<input type="checkbox"/>
★★★★	2.5 stars	<b>Spy Game (2001)</b> <a href="#">DVD</a> , <a href="#">VHS</a> , <a href="#">info</a>   <a href="#">imdb</a> Action, Crime, Drama, Thriller	<input type="checkbox"/>
★★★★	3.0 stars		
★★★★	3.5 stars		
★★★★	4.0 stars		
★★★★	4.5 stars		
★★★★	5.0 stars		

Abb 11 Collaborative Filtering auf MovieLens Webseite  
<http://movielens.umn.edu>

## 2.8 Zusammenfassung und Ausblick

Die verschiedenen Filter Methoden haben alle ihr Vor- und Nachteile. Auch bei schwierigen Anforderungen an das Recommender System z.B. bei riesige Nutzdaten (z.B. Amazon: 29 Mio. Nutzer, Millionen Produkte) sollen hochwertige Empfehlungen in Echtzeit, gute Empfehlungen für neue Nutzer und schnelle Empfehlungen trotz umfangreicher Profile erzeugt werden. Um die Nachteile zu beseitigen, werden kombinierte, hybride Filtermethoden eingesetzt, um die Leistung zu erhöhen, z.B. Item Based Collaborative Filtering (bei Amazon verwendet). Allerdings bleiben einige Probleme, wie das Arbeiten eines Recommender Systems auf einem leeren Datenbestand (Coldstart) bestehen. Trotz der Vorteile und der vielen existierenden Systeme und Algorithmen nutzen nur wenige personalisierte Webseiten Recommender Systeme.

#### 4 Literaturverzeichnis

- [1] „Visualizing the Semantic Web“ von Vladimir Geroimenko, Chaomei Chen
- [2] "Einsatz von Collaborative Filtering zur Datenprognose" von Bohnert F. (2003)
- [3] <http://www.uddi.org> (UDDI.org)  
<http://www.w3c.org> (World Wide Web Consortium)  
<http://citeseer.ist.psu.edu/499652.html> (Using Content-Based Filtering for Recommendation (ResearchIndex))  
<http://www.cs.umn.edu/Research/GroupLens/papers/pdf/ECRA.pdf>  
(E-Commerce Recommendation Applications)  
<http://www.wikipedia.org> (Wikipedia, the free encyclopedia)  
<http://www.jeckle.de/files/swtour.pdf> (Web Services Einführung)