Konzepte der AI

Informationsagenten/systeme im Internet

Robert Baumgartner

Klassifikation nach Ausführungsort

- *Internetagenten:* Websuchagenten, Web-Serveragenten, Informationsfilteragenten, Informationsbeschaffungsagenten, Notification Agenten, Service-Agenten, Mobile Agenten
- *Intranetagenten:* Kollaborative Anpassungsagenten, Prozeßautomatisierungsagenten, Datenbankagenten, Resourcebrokeringagenten
- *Desktopagenten:* Interface-Agenten für OS, Applikationen etc.

Bots, Spiders und Agenten

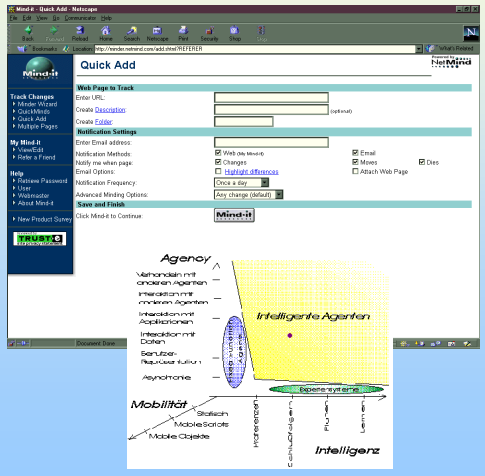
- Bots: unbeachtete Programme die auf das Internet zugreifen
- Spiders: Bots die bestimmte Webseiten durchsuchen; Spider scannen das Netz kontinuirlich und folgen den Links auf den Seiten (crawling); Verwendung in Suchprogrammen wie Altavista, Excite
- Intelligent Agents: Bots mit höherer Intelligenz und interaktiven Komponenten

Gute Agenten halten sich an die Robot Exclusion Standards (robots.txt und robot meta tags) und melden sich als passender User-Agent mit email-Adresse an.

Beispiele

- News Bot: beobachtet ohne Einwirken des Users diverse online News Items und verständigt den User bei Neuigkeiten; Interaktion mit dem User
- Projekt Softbot: verwendet Planen und maschinelles Lernen und entscheidet selbst wie und wo vom User gewünschte Informationen gewonnen werden
- Letizia: beobachtet User bei Verwendung eines Browsers und ahmt nach, versucht verauszuahnen welche Links den Benutzer am meisten interessieren
- Beispiel eines Nichtinternet-Informationsagenten: Erinnerungsagent, der User einmal pro Stunde an eine fünfminütige Pause erinnert; Richtlinien: einfaches Deaktivieren/Aktivieren und User sonst nicht stören

MindIt-Agent



- zur Überwachung von Änderungen auf Webseiten
  statischerNotification/Interface Agent, Interaktion mit Daten, Präferenzen und Schlußfolgern

- Umgang mit Updates
  htmldiff tools, Update Monitoring (z.B. Continual Queries Projekt: Trigger welche Änderungen relevant)

Funktionen und Komponenten

- Funktionen
  - Aufgabenausführung
  - Wissensverarbeitung
  - Kommunikationsfähigkeit (mit Nutzer)

- Komponenten
  - Persönliches Profil (Informationsbeschreibg.sprache XML,...)
  - Schnittstellen zu Diensten (CORBA,... und ACL)
  - Aufgabenbeschreibungen/Missionsskripts (Java, Tcl, Perl,...)

Intelligente Internetsysteme

- Usermodellierung
- Finden und Analyse
  - Auffinden
  - Extrahieren (parsen, wrappen)
  - Übersetzen (Semantik geben)
  - Auswerten
- Informationsintegration
- Webseitenmanagement

Wichtige Grundbegriffe

- IP-Adressen
  identifizieren jeden Rechner eindeutig im Netz; Multicast-Adressen
- TCP und UDP Protokoll
  Transport Control Protocol, User Datagram Protocol
- CGI
  Common Gateway Interface Scripts
- SQL,...
  Structured Query Language (Datenbankkommunikation)
- Parsen von Text,...
- CORBA
  Common Object Request Broker Architecture
  - Standard für komponentenbasierte Software-Entwicklung
  - Abstraktion über primitive Netzwerkdienste Middleware

Wichtige Grundbegriffe (2)

- HTTP, FTP, WAP,...
  Hypertexttransfer Protocol, File Transfer Protocol
- SGML (Standard Generalized Markup Language) ist ein internationaler Standard die Struktur und den Inhalt maschinell lesbarer Information zu beschreiben. SGML "Dokumente" bestehen aus Text, Grafik und Hypertextlinks.
- XML ist ein einfacher Dialekt von SGML der für WWW/Intranet entworfen wurde. XML ist eine Teilmenge von SGML.
- HMTL ist eine SGML-Applikation
- XSL(T): Extensible Stylesheet Language
  ist eine Sprache um XML zu transformieren und formatieren
- Rescource Description Framework RDF
  stellt eine allgemeine Methode zur Verfügung um Metadaten für XML Dokumente zu beschreiben (Ressourcen, Eigenschaften, Aussagen)

eXtensible Markup Language

- Informationsbeschreibungssprache für Agenten
- frei definierbare Tags
- selbstbeschreibend (wenn keine DTD: well-formed XML)
- layoutunabhängig (Trennung von Layout und Markup; Separater Mechanismus zur Visualisierung)
- Document Type Definition legt Grammatikregeln fest
  (wenn konform zu DTD und well-formed: valides XML)
- Beispiel (Elemente `title` etc., Attribute `currency`)

```
<book><title>The Lord of the Rings</title>
<author>
    <firstname>John Ronald Reuel</firstname>
    <lastname>Tolkien</lastname>
</author>
<price currency="USD">9.25</price></book>
```

DTD

```
<!ELEMENT book (title,author+,price)>
<!ELEMENT title (#PCDATA)>
<!ELEMENT author (#PCDATA | lastname | firstname | fullname)*>
<!ELEMENT price (#PCDATA)>
<!ATTLIST price        currency CDATA "USD"
                       source (list|regular|sale) "list"
                       taxed CDATA #FIXED "yes">
<!ELEMENT lastname (#PCDATA)>
<!ELEMENT firstname (#PCDATA)>
<!ELEMENT fullname (#PCDATA)>
```

*+ drückt Moglichkeit mehrerer Autoren aus. * null oder öfter*
PCDATA (parsed character data, i.e. daß Markup innerhalb)
nicht erwähnte Elemente sind durch die DTD nicht beschränkt.
Default Währung ist hier US-Dollar, source kann drei Werte haben und default list; und taxed hat hier den Fixwert ja. Attribute müssen nicht spezifiziert werden (außer bei #REQUIRED)
Spezielle Linkattribute: id, idref Entitäten: Textbausteine

Rund um XML

- XPath (Adressierung von XML Dokument-Teilen)
- XPointer, XLink (Verbinden von Objekten in XML Dokumenten)
- XML Schema (Erweiterung von DTDs)
- DOM (API für Navigation und Manipulation in Dokumentstruktur)
- Querysprachen (XML-QL, XQL, XML-GL, Lore, XQuery)

```
WHERE <book>
   <publisher><name>Addison-Wesley</></>
   <title> $t </>
   <author> $a </>
   </> IN www.text.edu/bib.xml
CONSTRUCT $a
```

XSL

- navigiert Dokument mittels XPath
- formatiert (z.B. HTML Output), strukturiert und führt Funktionen aus
- Strukturelle Rekursion
- auch als Querysprache geeignet
- ist programmiersprachenvollständig

Sprachen für Internetagenten

- *Smalltalk* - frühe AI-Sprache
- *Tcl/Tk* - interpretierte Skriptsprache. Tk ist graphisches User Interface; Agent-Tcl Erweiterung für mobile Agenten
- *Perl* - Scriptsprache mit C-artiger Syntax und einer reichhaltigen Ansammlung von Modulen
- *Telescript* - objektorientierte Sprache für mobile Agentensysteme; Nachfolger: Odyssee (beide nichtmehr verfügbar)
- *C,C++,* etc...
- *ECMA Script* (Java-Script, objektorientierte Skriptsprache, keine Parallelität)
- *Java*

Java

- objektorientiert
- interpretiert (Java Bytecode)
- Virtuelle Maschine ist maschinenunabhängig
- verfügbar für viele Platformen
- erlaubt multiple threads
- Remote Method Invocation
- Security Manager
- praktische Klassen: java.net, javax.swing.HTMLEditorKit,...
- Java Tutorial: http://java.sun.com
- Applets, Servlets

Java (2)

aufbauend auf Java:

- IBM Aglets (für mobile Agenten)
- ZEUS, Impact (für Multi-Agentensysteme)
- Jedi (Wrapping Semistructured Data)

Bsp.: IP-Adresse des lokalen Rechners eruiren

```
import java.net.InetAddress;
import java.net.UnknownHostException;

public class get {
   public static void main(String[] args) throws
   UnknownHostException {
        InetAddress local = InetAddress.getLocalHost();
        System.out.println(local.getHostAddress());
        System.out.println(local.getHostName());
 }
}
```

Beispiel

Bsp.: Browser

Ein kleiner Browser mit Swing (ohne Hyperlink folgen; ohne Scrollbars)

```
import java.io.*;
import java.net.*;
import javax.swing.*;
   public class view extends JPanel {
    public static void main (String args[]) throws Exception {
      JFrame f = new JFrame("Viewer");
      URL urli = new URL(args[0]);
      JEditorPane p = new JEditorPane();
      p.setPage(urli); p.setEditable(false);
      f.getContentPane().add(p); f.setSize(640,480); f.show();
    }
   }
```

Beispiel

Perl

- Skriptsprache
- legt keinerlei unnötigen Beschränkungen auf
- Regular Expressions, Pattern Matching
- Mächtige Manipulationsoperatoren (Substitution, Translation)

Beispiel:

```
print "What is your name?";
$name = <STDIN>;
chomp ($name) ;
print "Hello, $name!\n";
```

Semistrukturierte Daten

- Für Datenextraktion vom Web und Update Monitoring von Webseiten ist es nötig, daß Agenten mit semistrukturierten Daten umgehen, da HTML Seiten gleichen Types semistrukturierte Daten darstellen
- Semistrukturierte Daten haben gewisse Struktur, passen aber nicht in ein relationales oder objektorientiertes Datenbankschema (selbstbeschreibend)
- Ansätze:

  - eigenes Datenbankschema
    z.B. Lore (Lightweight Object Repository); aber auch XML ist semistrukturiert

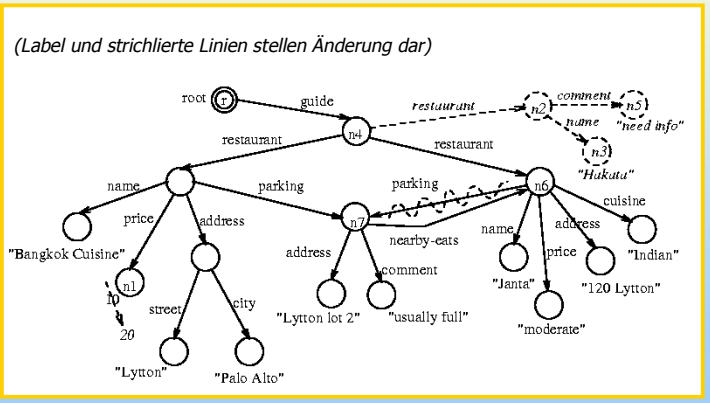
  - Wrapper um Daten herumbauen
    um sie in Struktur einzuhüllen die für Queries verwendet wird; für Wrappergenerierung gibt es große Anzahl an Tools/Papers, die für Agenten verwendet werden können (Jedi, Florid, Tsimmis,...)

OEM – Object Exchange Model

- Datenmodell für semistrukturierte Daten
- Daten als gerichteter Graph; Objekte als Knoten
- von Objekt ausgehende Kanten stellen Attribute des Objekts dar
- Atomare und komplexe Objekte
  Atomare Objekte sind vom Typ String, Integer etc. und stellen Blätter dar
- Der Graph hat mindestens eine Wurzel
- flache und tiefe Gleichheit
- Änderungen darstellbar (Delta OEM): Operationen wie Create Node; annotated nodes; *Chorel* als Erweiterung zu *Lorel*

OEM - Beispiel

(Label und strichlierte Linien stellen Änderung dar)



Lore

- verwendet OEM
- leicht für XML adaptierbar
- verwendet Data Guides
  kurze und exakte Übersicht der Struktur einer Datenbank, z.B. ein abstrahierter OEM Graph. exakte und approximierte Dataguides; zur Ähnlichkeitsabschätzung wird Object Matching und Role Matching verwendet (wenn keine DTD vorliegt)
- Anfragesprache Lorel

```
select M.Büro
from Abteilung.Mitarbeiter M
where M.Alter>40
```

Notebook Kauf

Benutzer möchte sich über Notebooks informieren
- die weniger als öS 20000 kosten
- nicht von Firma X
- mindestens 128 MB RAM

Problem: lange mühselige manuelle Suche
- bestenfalls beschränkte Querymöglichkeiten
- keine site-übergreifenden Queries

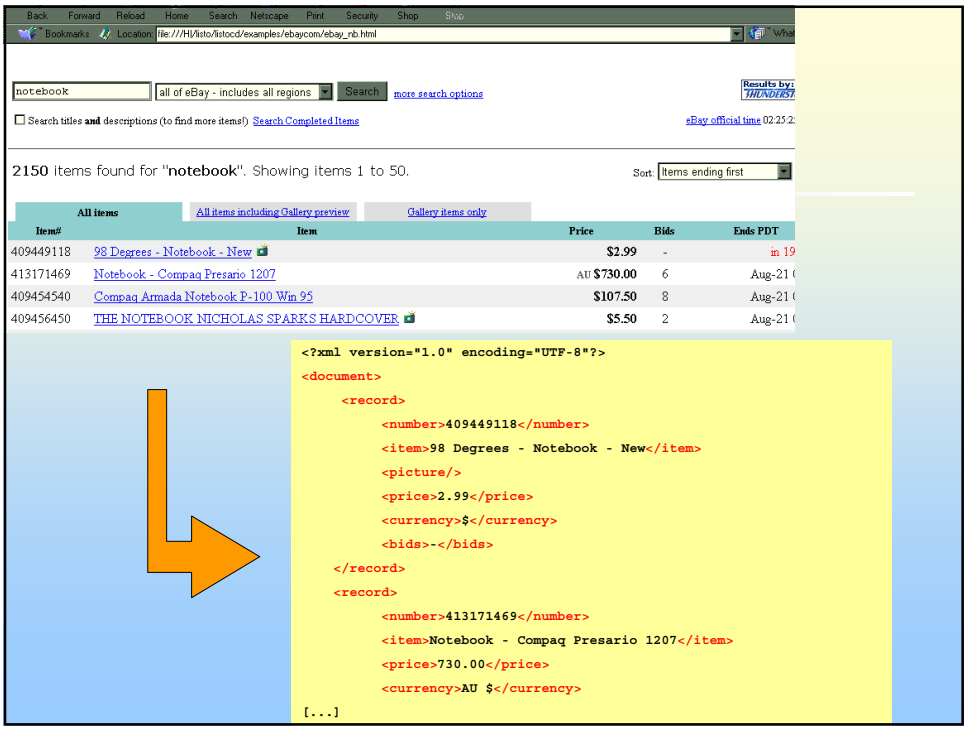
Lösungsansatz:
- Wrappertechnologie für jede Art von Webseite
- Extrahierte Daten in selbes XML Schema abbilden
- Informationsmediator für flexible Queries nun leicht realisierbar
- Automatische Queries: z.B. Benutzerinfo via email

Mediator

- Ein Informationsmediator stellt ein **query-only intermediate layer** zwischen den Clients und einer großen Anzahl von heterogenen Sourcen dar. Darunter finden sich etwa verschiedene Arten von strukturierten Datenbanken als auch semistrukturierte websourcen bzw. Textdateien.
- Clients von Informations Mediatoren können Informationssourcen abfragen und die Resultate **integrieren** ohne über Implementationsdetails wie Adressen, Formate, Sprachen, Platformen Bescheid zu wissen.
- Das Informationsmediationssystem entscheidet bei einer gegebenen Query **welche Informationsquellen** verwendet werden, wie die gewünschte Information beschafft wird, wie und wo sie zwischengespeichert wird und wie Daten manipuliert werden.
- Informations Mediatoren sind **flexibler und erweiterbarer** als traditionelle Multidatenbankansätze, weil sie dynamische Integration der relevanten Informationsquellen als Antwort einer Query durchführen können.

Wrapper

- Informations-Mediatoren benötigen Wrapper um Daten aus dem Web zu extrahieren.
- Hauptaufgabe: Information aus einer gegebenen Menge von Webseiten extrahieren und die Resultate als *(semi)strukturierte Datentupel* wiedergeben (z.B. in XML)
- Informations Extraktions Problem; ein Wrapper kann die Regularität des Erscheinungsbildes anstelle linguistischer Information verwenden
- Ein Wrapper für eine Klasse von Sourcen



Wrapper Ansätze

- manuell
  - geeignete Skriptsprache die die Extraktionssprache implementiert
- automatisch
  - machine und pattern learning
  - Lernalgorithmen basierend auf Beispielen
  - Generalisierungs-/Spezialisierungs-Algorithmen
- überwacht
  - interaktive geleitete Erstellung
  - angeben von Kriterien

Ansätze zur Extraktion

Wrapper - einige Seiten gleichen Aufbaus (wie eBay Seiten) werden analysiert und die Struktur z.B. mit kontextfreier oder regulärer Sprache beschrieben oder Automaten, versch. Ansätze im Detail:

- reguläre Grammatiken verwenden
- logische Programmierung
- mehrdeutige kontextfreie Grammatiken
- delimiter: Start und Ende Tags
- skipto-Sequenzen
- Finite State Transducer, endliche Automaten
- hierarchisch (HTML Parsebaum navigieren)
- Ontologien

Jedi

- erlaubt Wrappergenerierung für einen Typus von HTML-Dokument (Erstellung einer Skriptdatei)
- besteht aus handcodierten *Wrapper* (sammelt Daten durch Navigation durch verschiedene Dokumente unter Berücksichtigung ihrer logischen Struktur) und *Mediator* der die heterogenen Informationen integriert.
- verwendet als Regelsprache mehrdeutige kontextfreie Grammatiken und Parsing-Strategie die diese Mehrdeutigkeiten in sinnvoller Art und Weise auflöst.
- Mediator verwendet ein allgemeines Objekt-Modell, das auch erlaubt mit strukturellen Abweichungen umzugehen

Stalker

- benutzermarkierte Beispiele
- semiautomatische Wrappergenerierung
- embedded catalogs Hierarchie
- in einer Sequenz von Tokens mit skipto Sequenzen von links und rechts arbeiten um Element zu finden
- z.B. SkipTo(<b>)SkipTo(,)
  (bzw. als endlicher Automat darstellen)
- Stalker-Algorithmus erstellt und optimiert Disjunktion von SkipTo Sequenzen aus den gegebenen Beispielen mit diversen Verfeinerungen bis bei möglichst allen Beispielen positiv
- beginnt mit kurzer SkipTo Sequenz und verfeinert (umgekehrte Richtung wäre ebenso möglich, Maximierung)

Weitere induktive Ansätze

- NoDOSE
  automatische Datenextraktion aus Textquellen mit interaktivemInterface zur Angabe von negativen und positiven Beispielen
- RoadRunner
  Automatische Informationsextraktion ohne markierte Beispiele Datenfelder und Struktur werden erkannt
- MIA
  Multiagentenarchitektur, basierend auf Logikprogrammierung; durchsucht webseiten nach zB Restaurants in Umgebung; neurales Netzwerk für Textklassifizierung; verwendet Unifikation

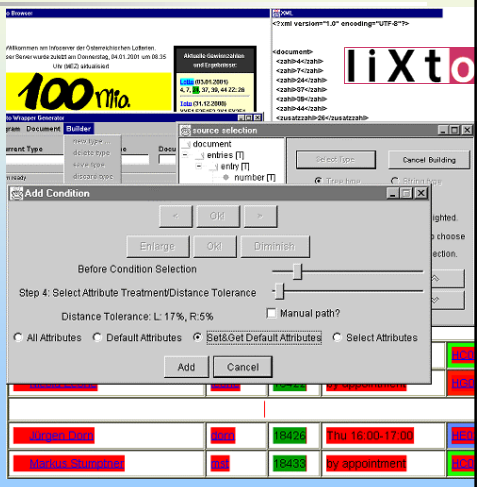
Vorteile: stark automatisiert
Nachteile: weniger flexibel
viele Beispiele nötig
ist Beispielmenge ausreichend?

Überwachte Wrappergenerierung

- XWrap (Liu et al.)
  - einige Erkennungsheuristiken
  - semi-automatisch, und nur tw. visuell
  - prozedurales Wrapperprogramm
  - "Templates" statt Vielzahl von Bedingungen
  - Variableniteration; Baumnavigation; keine Stringextraktion
  - semantische Tokens
- W4F (Azavant, Sahuguet)
  - deklarative Sprache HEL ("SQL-artige Aussagen")
  - Baum- und Stringnavigation
  - beschränktes UI
  - Hierarchische Extraktion, Bedingungen nicht strikt hierarchisch
  - verwendet XML-QL für Weiterverarbeitung

Lixto Suite

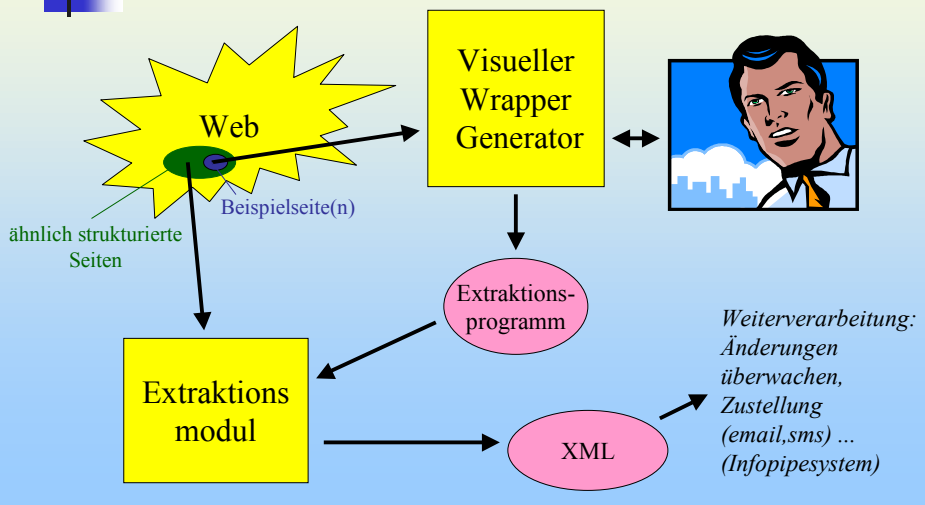
- Lixto Visual Wrapper
  www.dbai.tuwien.ac.at/proj/lixto
- Infopipes (Lixto TS)
  personalisierbare Informationskanäle; source;wrapper;merger;transformer; deliverer
- Möglichkeit zu Praktika und Diplomarbeiten



Lixto Visual Wrapper

- An unserem Institut entwickelte Methode zur überwachten Wrappergenerierung
- visuelles interaktives UI
- Vielzahl von Bedingungen visuell festlegbar
  inner und outer conditions, range, types, concepts
- multiple area matching, single area matching
- hierarchische Baumnavigation und Textnavigation
- "unsichtbare" deklarative Extraktionssprache
- implementiert in Java mit Swing-Klassen
- kontinuierliche Extraktion: "XML Companion"
- eingebettet in IP System

LiXto VW Architektur



Lixto Wrappergenerierung

- Patterngenerierungsalgorithmus
  - Ein Pattern charakterisiert eine Art von Information
  - Benutzer erstellt interaktiv Pattern aus einer Anzahl von Filtern. Jeder Filter kann aus mehreren Bedingungen bestehen.
  - Filter werden disjunktiv, Bedingungen konjunktiv interpretiert. So kann gewünschte Information eindeutig charakterisiert werden.
- datalog-ähnliche Regeln
  Programm ist Menge von Patterns

Lixto: Einfaches Beispiel

- Was ist der momentane Bestseller?
- Einmal pro Woche abfragen
- Ein Programmaufruf, einfache Programmerstellung
- Programm soll weiter funktionieren trotz kleinerer struktureller Änderungen
- Ziel: Automatisierung, Informationsweiterleitung (Handy, email, ...)

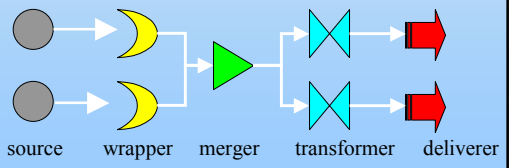
- Tägliches Wetter
- Rekursives Programm (eBay)
  verfolgen von Next Links

Beispiele ⇦

Now Playing!

- Extrahiert regelmäßig Daten von Radiostationen und Charts
- Verwendet Lixto und Infopipes Technologie
- In Zusammenarbeit mit T-Mobile

Beispiel ⇦



Einige Weblinks

- AgentWeb: http://agents.umbc.edu/
- Über Mobile Agenten: http://www.infosys.tuwien.ac.at/Research/Agents/
- MindIt: http://minder.netmind.com/
- Continual Queries: http://www.cse.ogi.edu/DISC/CQ/
- Ariadne (Knoblock et al.): http://www.isi.edu/ariadne/
- Florid: http://www.informatik.uni-freiburg.de/%7Edbis/florid/online.html
- JEDI: http://www.darmstadt.gmd.de/oasys/projects/jedi/jedie.html
- XWrap (Liu et al.): http://www.cse.ogi.edu/DISC/XWRAP/
- World Wide Web Wrapper Factory: http://db.cis.upenn.edu/W4F/
- Lixto: http://www.dbai.tuwien.ac.at/proj/lixto

Ausgewählte Literatur

Agenten allgemein:

- Brenner, Zarnekow, Wittig: Intelligent Software Agents – Foundations and Applications
- Müller: The Design of Intelligent Agents – A Layered Approach

Theoretische Grundlagen:

- Russell, Norvig: Artificial Intelligence – A Modern Approach

Internet-/Informationsagenten:

- Caglayan, Harrison: Agent Sourcebook – A complete guide to Desktop, Internet and Intranet Agents
- Cheong: Internet Agents – Spiders, Webs, Brokers and Bots
- Klusch (ed.): Intelligent Information Agents

Semistrukturierte Daten:

- Abiteboul, Buneman, Suciu: Data on the Web – From Relations to Semistructured Data and XML