

Ein Agentensystem für digitale Bibliotheken im WWW

Norbert Gövert Norbert Fuhr
Universität Dortmund

{goevert,fuhr}@ls6.cs.uni-dortmund.de

Die Zahl digitaler Bibliotheken im WWW wächst. Literaturrecherchen lassen sich immer öfter ohne den Gang in die „reale“ Bibliothek durchführen. Wir beschreiben den Entwurf eines Agentensystems, welches dem Benutzer / der Benutzerin Funktionalität zur Literaturrecherche auf einer höheren Ebene anbietet, als sie von einzelnen digitalen Bibliotheken erbracht werden kann. Dazu wird die von digitalen Bibliotheken angebotene Funktionalität ausgenutzt und kombiniert; die Basiseigenschaften von Agenten, *Adaptivität*, *Kommunikationsfähigkeit* und *Autonomie*, ermöglichen das auf flexible Weise.

1 Einleitung

Das WWW wird mit steigender Zahl der dort angebotenen digitalen Bibliotheken immer mehr zum unverzichtbaren Hilfsmittel bei der Literaturrecherche. Verlage, wie z. B. der *Springer-Verlag*¹ oder die *Association for Computing Machinery* (ACM)² bieten ihr Programm in digitaler Form und online an. Zudem gibt es für alle Fachgebiete zahlreiche verlagsunabhängige Angebote, z. B. für die Informatik den DBLP-Server³, eine Datenbank mit bibliographischen Informationen zu Informatik-relevanten Zeitschriften und Konferenzbänden, oder die *Networked Computer Science Technical Reference Library* (NCSTRL)⁴, eine verteilte Sammlung von technischen Berichte aus zur Zeit etwa 160 Informatikinstitutionen und -einrichtungen weltweit.

Dennoch stößt man bei der Nutzung der digitalen Angebote schnell auf Probleme: Zur Literaturrecherche benötigt man meist mehrere digitale Bibliotheken. Dazu muss man die entsprechenden Zugangspunkte kennen oder suchen. Jede benutzte digitale Bibliothek muss einzeln angesprochen werden, jeweils über eine eigene Benutzerschnittstelle mit jeweils individueller Funktionalität. Untersucht man die angebotenen Funktionalitäten, stellt man schnell fest, dass digitale Bibliotheken nur Basisfunktionen anbieten; Berghel bemerkt dazu, dass in vielen digitalen Bibliotheken, abgesehen vom Ausliefern von Information in elektronischer Form, keine zusätzliche Funktionalität geliefert wird im Vergleich zu herkömmlicher Publizierung. Berghel plädiert daher für *Value-added Publishing* [Berghel 99].

¹<http://link.springer.de/>

²<http://www.acm.org/dl/>

³<http://dblp.uni-trier.de/>

⁴<http://www.ncstrl.org/>

Im vorliegenden Artikel wird der Entwurf eines Agentensystems vorgestellt, welches diese Mängel überwindet. Das System bietet dem Benutzer / der Benutzerin höherwertige, *zielgerichtete* Funktionalität an, als sie von einzelnen digitalen Bibliotheken erbracht werden kann. Zunächst wird dazu in Abschnitt 2 die zu realisierende Funktionalität beschrieben und charakterisiert. In Abschnitt 3 wird die Gesamtarchitektur des Agentensystems erläutert. Abschnitt 4 beschreibt, wie Basiseigenschaften von Agenten ausgenutzt werden, um die höherwertige Funktionalität zu implementieren; insbesondere wird auf *Kommunikationsfähigkeit*, *Adaptivität*, und *Autonomie* der verwendeten Agenten eingegangen. Im Ausblick in Abschnitt 5 werden einige Aspekte der Implementierung des Systems herausgegriffen.

2 Suchfunktionalität

Bates hat 1990 eine Studie [Bates 90] vorgelegt, in der Suchaktivitäten professioneller Rechercheure und Bibliotheken untersucht werden. Suchaktivitäten lassen sich danach in vier Abstraktionsebenen kategorisieren:

Elementare Aktionen sind identifizierbare Gedanken oder Aktionen, die Teil einer Informationssuche sind. Sie entsprechen typischerweise den Kommandos, die von einem Retrieval-System zur Verfügung gestellt werden, also z. B. das Hinzufügen eines Terms oder einer Bedingung zu einer Anfrage oder das Verfolgen eines Verweises.

Taktiken umfassen mehrere elementare Aktionen, die eine Suche beschleunigen und das Suchergebnis verbessern. Dazu gehört z. B. das Verfeinern von Anfragen wie die Generalisierung von Suchbegriffen, um Recall zu erhöhen, oder etwa das Spezialisieren von Suchbegriffen, um Precision zu erhöhen.

Strategeme umfassen meist mehrere Taktiken, um Informationsstrukturen in einer Suchdomäne auszunutzen. Dazu gehören z. B. der Journal-Run (Navigation in den Ausgaben einer Zeitschrift, die als zentral für ein gegebenes Informationsbedürfnis identifiziert wurde), die Footnote-Chase, bei der die Referenzen in einem relevanten Dokument verfolgt werden, oder die Autorensuche mit dem Ziel, weitere Dokumente eines Autors zu finden, der bereits in einem oder mehreren relevanten Dokumenten aufgetaucht ist.

Strategien sind Pläne, die Aktionen, Taktiken und Strategeme beinhalten, um eine Informationssuche vollständig durchzuführen und somit ein gegebenes Informationsbedürfnis komplett abzudecken. Eine einfache Strategie zur Literaturrecherche über ein neues Gebiet wäre etwa eine themenorientierte Suche im ersten Schritt, wonach man mit den in den relevanten Dokumenten gefundenen Autorennamen eine Autorensuche anschließt.

Bates stellt fest, dass sich Retrievalsysteme meist auf der Ebene der elementaren Aktionen bewegen. Die anderen Funktionalitätskategorien grenzen sich hiervon insbesondere durch ihre Orientierung an einem oder mehreren vorgegebenen Zielen ab, sowie durch ihr strategisches Element: Taktiken, Strategeme und Strategien beschreiben stets Pläne mittels derer ein vorgegebenes Ziel erreicht werden soll. Ziel des hier vorgestellte Agentensystems ist demzufolge die Implementierung einer *strategischen* Unterstützung für den Benutzer / die Benutzerin bei der Informationssuche. Neben elementaren Aktionen wird Funktionalität auf der Ebene von Taktiken und Strategemen angeboten.

3 Gesamtarchitektur

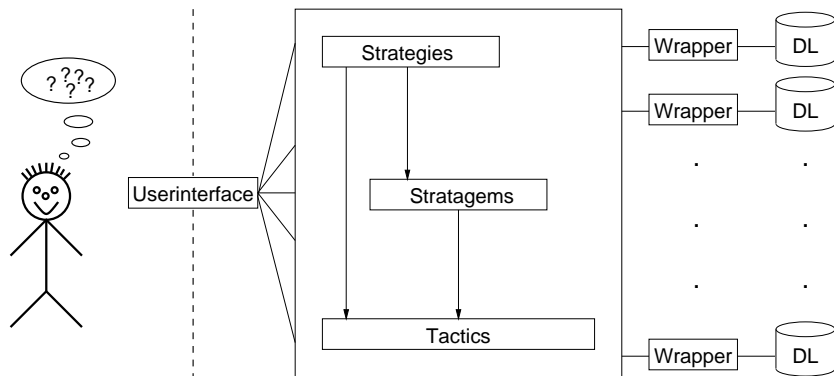


Abbildung 1: Architektur des Agentensystems

Benutzerinteraktion mit digitalen Bibliotheken über das Agentensystem läßt sich in vier Schichten modellieren (Abbildung 1).

Auf unterster Ebene befinden sich die im WWW angebotenen digitalen Bibliotheken. Transparenter Zugriff auf deren Funktionen und Daten ist durch die Schicht der Wrapper-Agenten gewährleistet. Aufgabe der Wrapper-Agenten ist es, die von den digitalen Bibliotheken erbrachten Funktionen und Daten so zu kapseln, dass sie für das Gesamtsystem nutzbar sind.

Auf der anderen Seite kommuniziert der Benutzer / die Benutzerin mit dem Gesamtsystem über einen Userinterface-Agenten. Die Aufgabe des Userinterface-Agenten ist zweifältig. Zum einen muss er die vom Gesamtsystem erbrachte Funktionalität dem Benutzer / der Benutzerin anbieten. Ausgehend von den Aktionen der Benutzer macht er dazu entsprechende Angebote bzw. nimmt die Funktionalität des Gesamtsystems in Anspruch. Zum anderen muss der Userinterface-Agent die von den Service-Agenten erbrachten Ergebnisse visualisieren.

Zwischen Userinterface- und Wrapper-Agenten befindet sich nun das Herzstück des Agentensystems, eine kritische Masse von Service-Agenten, die die zusätzliche Funktionalität implementieren. Jeder angebotene Dienst, der sich wiederum auf den Ebenen der Suchfunktionalität nach Bates einordnen läßt, wird von einem Service-Agenten implementiert. Zur Realisierung eines Dienstes greift ein Service-Agent auf Dienste von anderen Service-Agenten, sowie der Wrapper-Agenten zurück.

4 Das Agentensystem

Warum Agenten zur Realisierung des Gesamtsystems? Die sich ständig ändernde Landschaft digitaler Bibliotheken im Internet erfordert flexible, einfach erweiterbare Systeme, um eine möglichst mächtige Funktionalität anbieten zu können. Sowohl müssen neue digitale Bibliotheken als auch zusätzliche Funktionalität in das System integrierbar sein. Die von Nwana [Nwana 96] genannten Basiseigenschaften von Agenten, *Kommunikationsfähigkeit*, *Adaptivität* und *Autonomie*, erlauben den Entwurf eines derart flexiblen Systems.

4.1 Kommunikationsfähigkeit

Agenten erfüllen ihre Aufgaben in Zusammenarbeit mit anderen Agenten. Agenten sind „teamfähig“. Services können nur durch Kommunikation erbracht werden. Sollen Services auf mehreren digitalen Bibliotheken basieren, muss eine Kommunikation mit diesen Bibliotheken gewährleistet sein. Service-Agenten, die einen Dienst basierend auf Diensten anderer Service-Agenten erbringen, müssen mit diesen kommunizieren.

Für die Kommunikation der Service-Agenten wurde ein Bussystem konzipiert, das Broadcasting, Groupcasting sowie Peer-to-Peer-Kommunikation zwischen einzelnen Agenten ermöglicht. Die Objekte, über die kommuniziert wird, orientieren sich an den Objekten, die von den digitalen Bibliotheken in das System eingebracht werden. Diese lassen sich über die von Fuhr genannten Ebenen von Informationsstrukturen [Fuhr 99] in Datenbanken identifizieren:

Schema: Auf Schemaebene werden die möglichen Attribute von Dokumenten und von deren Metadaten modelliert. Beispiele für standardisierte Schemata sind der *Dublin Core* oder bibliographische Datenformate wie MAB und MARC.

Attributwert: Der Wertebereich zu einem Attribut aus dem Datenbankschema kann zum Navigieren und zur Anfrageerstellung verwendet werden. Häufig sind Attributwerte selbst strukturiert, z. B. in Thesauri oder Klassifikationen.

Metadatensatz: Ein Metadatensatz faßt die Attributwerte bezüglich eines Schemas zu einem gegebenen Dokument zusammen, z. B. bibliographische Attribute, wie Titel, Autor, Erscheinungsjahr und Quelle.

Volltext: Das Dokument selbst, mit Inhalt und Struktur.

Anfrageobjekte innerhalb des Agentensystems beziehen sich nun entweder auf Objekte innerhalb einer Ebene der Informationsstrukturen, oder sie schaffen Übergänge zwischen zwei Ebenen. Es ergibt sich die in Abbildung 2 gezeigte Klassenhierarchie für Anfrageobjekte:

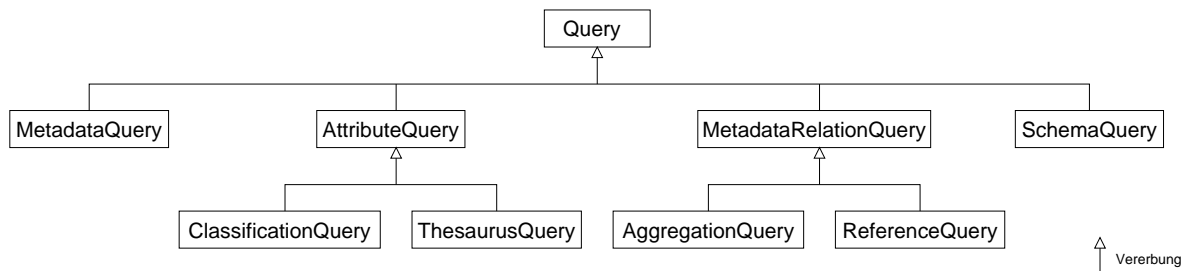


Abbildung 2: Anfrageklassen

- Hinter der `MetadataQuery` verbirgt sich die Standardfunktionalität für viele IR-Systeme. Gesucht wird in der Metadatenebene, Anfragen beziehen sich auf Attribute der Metadatensätze. Als Resultat erhält man ein Ranking mit Metadatensätzen von Dokumenten, die die Anfragebedingungen am besten erfüllen.
- Die `AttributeQuery` bezieht sich auf die Attributwertebene. Eine Anfrage mit einem Attributnamen, einem Vergleichswert und einem Vergleichsoperator resultiert in einer

Liste von Attributwerten, die die Anfragebedingung erfüllen. Da Attributwertemengen strukturiert sein können, ergeben sich als Spezialisierungen Anfragetypen für Taxonomien (*ClassificationQuery* bzw. *ThesaurusQuery*), mit denen eine Navigation in solchen Strukturen möglich ist. Als Beispiele seien hier Vergleichsoperatoren wie *Subclasses_of* und *Superclass_of* bzw. *Broderterm_of* und *Narrowerterms_of* genannt.

- Eine *MetadataRelationQuery* bildet Metadatenätze aufeinander ab. Beispiele sind die Suche nach den in einem Dokument referenzierten Dokumenten oder das Identifizieren von Komponenten eines aggregierten Dokumentes (aggregierte Dokumente sind z. B. Zeitschriftenausgaben, die Artikel aggregieren, oder Konferenzbände, die entsprechend Konferenzbeiträge aggregieren).
- Mit einer *SchemaQuery* kann nach Attributen in einem Datenbankschema gesucht werden. Resultat einer solchen Anfrage sind die Attributnamen eines Schemas.

Betrachtet man die zu den Anfrageobjekten möglichen Resultatobjekte, ergibt sich das in Abbildung 3 gezeigte Klassenmodell für Ergebnisobjekte. Die Resultate werden in Form von Listen zurückgegeben, die jeweils beliebig viele Ergebnisobjekte enthalten können. Für Attributanfragen sind die Attributwerte in *AttributeValueList* gespeichert; Metadatenätze sind Resultat von Metadatenanfragen und Anfragen auf Beziehungen zwischen Metadatenätzen und werden als *MetadataList* zurückgegeben; die *AttributeNameList* schließlich enthält Attributnamen als Resultat von Schemaanfragen.

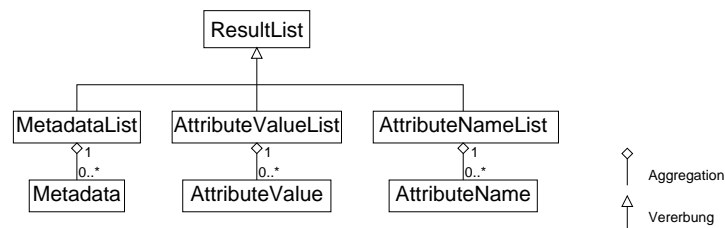


Abbildung 3: Ergebnisklassen

4.2 Adaptivität

Agenten passen sich ihrer Umwelt an. Sie reagieren auf Veränderungen in ihrer Umwelt, sie sind in der Lage zu lernen. Das Lernen hat zum Ziel, dass Agenten ihre Aufgaben mit der Zeit effektiver lösen können. Somit wird ein Mechanismus gebraucht, der es den Agenten erlaubt, ihre eigene Leistung zu bewerten.

Die Bewertung von Agentenleistungen erfolgt durch die Benutzer des Systems. Der typische Fall ist die Beurteilung der Relevanz von Metadatenätzen in Bezug auf ein Informationsbedürfnis. Diese Beurteilung muss nun zu den Agenten, die an der Erstellung des bewerteten Ergebnisses beteiligt waren, propagiert werden. Service-Agenten sind meist nur mittelbar an der Erstellung von Ergebnissen, die den Benutzern präsentiert werden, beteiligt. Daher kennzeichnet jeder Agent die von ihm produzierten Teilergebnisse mit einer Marke, die dem Agenten später eindeutig wieder zugeordnet werden kann. Diese Marke wird überall dort weitergereicht, wo das gekennzeichnete Teilergebnis einfließt. Kommt nun eine (mittelbare) Beurteilung für einen Agenten zurück, kann diese statistisch verwertet werden, zusammen mit den aus der

Sicht des Agenten relevanten Parametern, die zur Erstellung des beurteilten Ergebnisses geführt haben.

Diese Selbstbewertung kann dann von anderen Agenten verwendet werden, um eine bestimmte Güte eines von ihm genutzten Dienstes zu verlangen. Ein häufig verwendetes Kooperationsmodell hierfür sind Kontraktnetze [Davis & Smith 83]: Ein nachfragender Agent formuliert auf dem Kommunikationsbus ein Ziel. Anbieter-Agenten, die für die Ausführung des Requests in Frage kommen, machen ein Angebot mit der Güte, mit der sie das Ziel erfüllen können. Aus den Angeboten kann der nachfragende Agent nun das beste oder mehrere gute auswählen.

Ein Beispiel ist die Anfrageerweiterung. Ein nachfragender Agent legt die zu erweiternde Anfrage auf den Bus. Anfrageerweiterung kann von unterschiedlichen Agenten mit jeweils unterschiedlichen Taktiken durchgeführt werden, z. B. mittels Relevanz-Feedback-Methoden oder etwa durch Spezialisierung von Anfragetermen mit Hilfe eines Thesaurus. Agenten, die nun Anfrageerweiterung anbieten, machen entsprechend der in ihrer Statistik ermittelten Güte ein Angebot, aus denen der nachfragende Agent auswählt.

Wesentliches Merkmal von Kontraktnetzen ist die Lokalität der getroffenen Entscheidungen. Dieses läßt sich am Beispiel der Datenbankauswahl einem zentralistischem Vorgehen gegenüberstellen. In der hier vorgeschlagenen Kontraktnetz-Architektur entscheidet jeder Wrapper-Agent für die digitale Bibliothek, die er kapselt, ob diese in der Lage ist, eine gegebene Anfrage zu beantworten. Dabei können – sofern die Implementation des Wrappers das zuläßt – alle lokalen Eigenschaften der angeschlossenen digitalen Bibliothek bei der Entscheidungsfindung einbezogen werden. Anders verhält sich dies z. B. in dem im Projekt MeDoc [Dreger et al. 98] verfolgten Ansatz: Für die Datenbankauswahl ist dort eine zentrale Instanz, der *Broker* zuständig. Anhand von Metadaten, die der Broker von den angeschlossenen Datenbanken sammelt, wird eine globale Entscheidung zur Datenbankauswahl bei gegebener Anfrage getroffen. Somit können nur solche Eigenschaften der angeschlossenen Datenbanken berücksichtigt werden, die allen gemeinsam sind.

4.3 Autonomie

Autonomie bedeutet, dass Agenten ohne Interaktion mit einem Benutzer operieren können. Agenten handeln selbstständig, um die ihnen zugeteilten Ziele zu erreichen. Um einen Dienst anbieten zu können, greifen die Service-Agenten meist auf die Funktionalität eines oder mehrerer anderer Service- und Wrapper-Agenten zurück. Dabei entscheiden sie im Kontraktnetz selbstständig darüber, mit welchen dieser Agenten sie zusammenarbeiten. Ein Entscheidungskriterium ist dabei die im Abschnitt 4.2 besprochene Leistungsfähigkeit von Agenten.

Weiterer Aspekt der Autonomie ist *Proaktivität*. Agenten reagieren nicht nur, sondern sie werden auch von sich aus aktiv, um ihre Aufgaben zu erfüllen. Proaktivität wird verwendet, um dem Benutzer/der Benutzerin ein höheres Maß an Systemunterstützung zukommen zu lassen. Ein nach Bates anstrebenswertes Maß von Systemunterstützung ist das Beobachten von Suchaktivitäten durch das System, um daraus weitere Aktionen abzuleiten, die die Suche vorantreiben. Diese werden dann vorgeschlagen und/oder durchgeführt. Zum Beispiel kann ein Service-Agent, der eine Autorensuche realisiert, proaktiv dann tätig werden, wenn er realisiert, dass innerhalb eines Suchergebnisses Dokumente eines Autors mehrfach als relevant beurteilt werden. Das Ergebnis einer proaktiv ausgeführten Aktion wird dem Benutzer/der Benutzerin über den Userinterface-Agenten angezeigt; diesem muss dazu wiederum einen Push-Mechanismus zur Verfügung stehen.

Der Grad der Systemunterstützung wird letztlich von den Nutzern in ihren Profilen einge-

stellt. Sie sollen selbst entscheiden, ob das System nur Vorschläge für Aktionen machen soll, oder ob es diese auch selbstständig in die Tat umsetzen soll.

5 Ausblick

Basiseigenschaften von Agenten wie Kommunikationsfähigkeit, Adaptivität und Autonomie erlauben in flexibler Weise den Entwurf von hochwertiger, zielgerichteter Suchfunktionalität für digitale Bibliotheken. Durch solche Suchfunktionalität gelingt es, den Benutzer bei seinen Suchaktivitäten *strategisch* zu unterstützen; gleichzeitig ist eine Erhöhung des Grades der System-selbstständigkeit möglich.

6 Danksagung

Wir danken den zwölf Mitgliedern unserer studentischen Projektgruppe *PIANO*⁵ für die zahlreichen Diskussionen, durch die diese Arbeit initiiert wurde.

Literatur

- Bates, M.** (1990). Where Should the Person Stop and the Information Search Interface Start? *Information Processing and Management* 26(5), S. 575–591.
- Berghel, H.** (1999). Value-Added Publishing. *Communications of the ACM* 42(1), S. 19–23.
- Davis, R.; Smith, R. G.** (1983). Negotiation as a Methaphor for Distributed Problem solving. *Artificial Intelligence* 20(1), S. 63–109.
- Dreger, M.; Fuhr, N.; Großjohann, K.; Lohrum, S.** (1998). *Provider Selection—Design and Implementation of the Medoc Broker*, Band 1392 von *Lecture Notes in Computer Science*, chapter II: Design Considerations of the MeDoc System, S. 67–78. Springer, Heidelberg.
- Fuhr, N.** (1999). Information Retrieval in Digitalen Bibliotheken. In: *21. DGI-Online-Tagung – Aufbruch ins Wissensmanagement*. DGI, Frankfurt.
- Nwana, H. S.** (1996). Software Agents: An Overview. *The Knowledge Engineering Review* 11(3). <http://www.cs.umbc.edu/agents/introduction/ao/>.

⁵Proactive Intelligent Agents for Networked Objects,
<http://ls6-www.cs.uni-dortmund.de/ir/lehre/pg341/>