Information Retrieval in vernetzten heterogenen Datenbanken*

Norbert Gövert†
Universität Dortmund
Lehrstuhl Informatik VI
D-44221 Dortmund

25. Juni 1996

Inhaltsverzeichnis

1 Einleitung **2**

2 Datenunabhängigkeit **2**

3 Physikalische Datenunabhängigkeit **3**

4 Logische Datenunabhängigkeit **4**

5 SFgate **7**

6 Zusammenfassung und Ausblick **9**

Literatur **9**

Zusammenfassung

Mit der Verwendung des Attributkonzepts von freeWAIS-sf ist eine Befragung mehrerer WAIS-Datenbanken nur dann möglich, wenn die einzelnen Datenbanken jeweils die gleichen Anfrageattribute anbieten. Basierend auf dem Konzept der Datenunabhängigkeit wird eine externe Sicht auf mehrere freeWAIS-sf-Datenbanken präsentiert, die es zuläßt, auch Datenbanken mit unterschiedlichen Anfrageattributen parallel zu befragen; die Heterogenität der zu befragenden Datenbanken wird vor dem Benutzer verborgen. SFgate ist ein Gateway zwischen dem World Wide Web und freeWAIS-sf, welches den hier vorgestellten Ansatz implementiert.

Summary

With the field capabilities of freeWAIS-sf it became difficult to query more than one database in parallel. The set of searchable fields can differ for different databases, they may have heterogeneous schemas. An important concept in database systems is data independence. Based on this concept a unifying view on multiple freeWAIS-sf databases is presented: the aspect of heterogeneity of different databases is hidden from the user. SFgate is a gateway between the World Wide Web and freeWAIS-sf which implements this approach.

*Diese Arbeit entstand im Rahmen des BMBF-Projektes *MeDoc* (Förderkennzeichen 08 C 7829 6)
†Email: goevert@ls6.informatik.uni-dortmund.de

1 Einleitung

Mit der zunehmenden Vernetzung und dem immer größer werdenden Informationsangebot im Internet wächst der Bedarf an modernen Werkzeugen, die es dem Netzbenutzer ermöglichen, die Informationsflut für sich auszunutzen. Ein früher Ansatz auf dem Weg zum netzbasierten Information-Retrieval-System (IR-System) ist das WAIS-System (*Wide Area Information Servers*). WAIS bietet dem Benutzer Freitextsuche in Dokumentkollektionen. Dabei können mehrere, beliebig im Internet verteilte Datenbanken parallel befragt werden. Eine weitere herausragende Eigenschaft von WAIS ist die Präsentation von Suchergebnissen in rangierter Form. Jedem Dokument wird ein Wert zugeordnet, der als Grad seiner Ähnlichkeit mit der Anfrage interpretiert werden kann. Da die Dokumente nach fallenden Ähnlichkeitsgraden sortiert ausgegeben werden, sind die relevanten Dokumente mit großer Wahrscheinlickeit bereits am Anfang des Suchergebnisses zu finden.

FreeWAIS-sf [Pfeifer et al. 95] ist eine Erweiterung von WAIS. Mit über 1500 Installationen weltweit ist es wohl eines der derzeit meist genutzten netzbasierten IR-Systeme im Internet. Eine wesentliche Erweiterung von freeWAIS-sf ist die Unterstützung der logischen Struktur von Dokumenten, so wie es z. B. in bibliographischen Datenbanken üblich ist: eine Dokumentrepräsentation besteht aus einer Menge von Attributen (z. B. *Autor*, *Titel*, *Erscheinungsjahr*) und deren Werten. Die Attribute können dabei auch mengenwertig sein. Anfragebedingungen können sich nun gezielt auf bestimmte Attribute von Dokumenten beziehen, z. B. *Titel=(information retrieval) und Erscheinungsjahr>1990*.

Das Attribut-Konzept führt jedoch zu dem Problem, daß das parallele Durchsuchen mehrerer verteilter Datenbanken nur noch mit Einschränkungen möglich ist. Die einzelnen Datenbanken bieten in der Regel ja auch unterschiedliche Anfrageattribute an, sie besitzen *heterogene Schemata*. Um eine gleichzeitige Befragung mehrerer Datenbanken mit heterogenen Schemata zu ermöglichen, wird eine höhere Abstraktionsebene benötigt, auf der Unterschiede in den Datenbankschemata nicht mehr sichtbar sind. Eine solche Form der Datenorganisation auf unterschiedlichen Abstraktionsebenen ist in der Datenbankwelt als Konzept der *Datenunabhängigkeit* bekannt.

Aufbauend auf [Fuhr 96] wird mit Hilfe des Datenunabhängigkeitskonzeptes ein Mechanismus vorgestellt, der die Heterogenität von freeWAIS-sf-Datenbanken ausgleicht und damit das parallele Befragen mehrerer Datenbanken ohne explizite Reformulierung der Anfrage ermöglicht. Für den Benutzer wird eine Ebene des Datenbankzugriffs hergestellt, auf der er sich um die in den einzelnen Datenbanken unterschiedlich vorhandenen Anfrageattribute nicht mehr kümmern muß.

Abschnitt 2 definiert das Konzept der Datenunabhängigkeit anhand des ANSI/SPARC-Modells für Datenorganisation. Darauf aufbauend beschreibt Abschnitt 3 den Aspekt der *physikalischen Datenunabhängigkeit*, der Voraussetzung für die zu schaffende *logische Datenunabhängigkeit* ist. Die Schritte von der physikalischen zur logischen Datenunabhängigkeit sind das zentrale Thema dieses Artikels und werden in Abschnitt 4 dargestellt. SFgate ist ein Gateway zwischen dem WWW und freeWAIS-sf. Anfragen des Benutzers werden über HTML-Formulare entgegengenommen und auf eine oder mehrere WAIS-Datenbanken abgebildet. In der Version 5.0 implementiert SFgate das Konzept der Datenunabhängigkeit im Zusammenhang mit freeWAIS-sf-Datenbanken. SFgate wird im Abschnitt 5 vorgestellt. Abschließend wird ein Ausblick gegeben auf mögliche Erweiterungen des hier vorgestellten Konzeptes der logischen Datenunabhängigkeit sowie auf weiterführende Entwicklungen im Bereich des netzbasierten Information Retrieval.

2 Datenunabhängigkeit

Das Konzept der Datenunabhängigkeit ist typischerweise in herkömmlichen Datenbanksystemen implementiert, während es in der Welt der IR-Systeme bislang kaum berücksichtigt wurde. Das ANSI/SPARC-Modell [Tsichritzis & Klug 78] beschreibt drei Ebenen der Datenorganisation:

- Die **interne Ebene** befaßt sich mit Datenstrukturen und Zugriffsmethoden auf Dateiebene.

Sie ist am nächsten zur physikalischen Speicherung von Daten angesiedelt. Hier sind interne Datenstrukturen, Datensatzformate, sowie Zugriffspfade anzusiedeln.

- Auf der **konzeptionellen Ebene** ist das Schema einer Datenbank mit den Attributen und (Anfrage-)Prädikaten sichtbar. Diese Ebene sorgt für die *physikalische Datenunabhängigkeit*: Änderungen auf der internen Ebene (z. B. die Implementierung effizienterer Zugriffspfade) sind hier transparent.
- Die **externe Ebene** ist die dem Benutzer einer Datenbank sichtbare Ebene. Hier werden spezielle Sichten auf die konzeptionelle Ebene einer Datenbank realisiert. Die externe Ebene implementiert die *logische Datenunabhängigkeit*: Änderungen auf der konzeptionellen Ebene (z. B. das Einfügen neuer Attribute oder auch das Löschen von Attributen) bleiben ihr bis zu einem gewissen Grad verborgen.

Datenunabhängigkeit wird also durch die Einführung verschiedener Abstraktionsebenen erreicht, wobei Änderungen auf einer niederen Ebene niemals die jeweils höher liegenden Ebenen berühren.

3 Physikalische Datenunabhängigkeit

Wie im vorhergehenden Abschnitt angesprochen, gibt es in den meisten IR-Systemen das Konzept der Datenunabhängigkeit nicht. Anfragen an solche Systeme beziehen sich immer auf die interne Ebene. Ein Vorteil einer solchen Implementierung liegt darin, daß Anfragen effizient prozessiert werden können, da das System auf die Prozessierung von festgelegten Anfragekonzepten ausgelegt werden kann. Nachteil ist jedoch, daß die Anfragemöglichkeiten des Benutzers direkt von der Implementierung des Systems abhängen. Zwei Beispiele sollen das verdeutlichen:

- Die Suche nach Nominalphrasen ist in vielen IR-Systemen durch *Proximity*-Operatoren realisiert. Der Benutzer spezifiziert in seiner Anfrage zusätzlich zu den Bestandteilen einer Phrase einen maximalen Wortabstand. In diesem Abstand müssen Dokumente die Bestandteile der Phrase enthalten, um als Treffer bezüglich der Anfrage durchzugehen. Der Benutzer selbst muß also die genauen Randbedingungen für die Identifikation einer Phrase festlegen. Das System entscheidet dann binär nach diesen Bedingungen, ob in einem Dokument die angefragte Phrase existiert. Sinnvoll wäre es, statt der Proximity-Operatoren auf der konzeptionellen Ebene nur ein *enthaltensein*-Prädikat für Phrasen anzubieten. Die Implementierung dieses Operators auf der internen Ebene kann dann z. B. mittels einer Funktion in Abhängigkeit von Wortabständen geschehen, aber auch durch linguistische Methoden, wie die Verwendung eines Parsers zur Identifikation von Nominalphrasen. Mit einem solchen Prädikat kann das System nun eine *vage* Entscheidung treffen, mit welcher *Wahrscheinlichkeit* eine Phrase in einem Dokument vorkommt.
- IR-Systeme bieten in der Regel entweder die Suche nach Vollformen oder nach Stammformen an. Viele Systeme besitzen nur einen Index über die Vollformen. Benötigt der Benutzer eine Stammformsuche, so steht ihm als Annäherung oftmals nur die Rechtstrunkierung zur Verfügung. Sinnvoller wären jedoch zwei *enthaltensein*-Prädikate, das eine für die Suche nach Stammformen, das andere für die Suche nach Vollformen. Die Implementierung dieser Prädikate sollte hier ebenfalls auf der internen Ebene verborgen werden.

Wünschenswert ist eine Art des Datenbankzugriffs, die von der Implementation und somit von der internen Ebene abstrahiert. Dazu werden auf der konzeptionellen Ebene Attribute und Datentypen mit darauf definierten (vagen) Anfrageprädikaten [Fuhr & Rölleke 97] benötigt, die ein effektives Suchen ermöglichen. Eine Anfragebedingung besteht dann aus einem Attribut, einem möglicherweise vagen Prädikat und einem Vergleichswert. Durch die Einführung von vagen Prädikaten wird

die Unsicherheit des Benutzers bei der Anfrageformulierung berücksichtigt. Statt eine binäre Entscheidung bezüglich der Erfüllung einer Anfragebedingung zu treffen, schätzt ein vages Prädikat die Wahrscheinlichkeit, mit der die Anfragebedingung in einem Dokument erfüllt ist.

FreeWAIS-sf verfolgt bis zu einem bestimmten Grad das Konzept einer solchen physikalischen Datenunabhängigkeit. Über die logische Struktur der zu verwaltenden Dokumente werden den Dokumenten Attribute zugeordnet, die zusätzlich unterschiedliche Datentypen besitzen können. FreeWAIS-sf bietet die in Tabelle 1 aufgeführten Datentypen und Prädikate an.

Datentyp	**Prädikate**
text	enthaltensein (Gleichheit)
numerisch	kleiner, größer, gleich
name	Gleichheit, phonetische Ähnlichkeit (soundex, phonix)

Tabelle 1: Datentypen in freeWAIS-sf

Für die aus Attributen und Datentypen bestehende konzeptionelle Ebene kann nun ein einfaches Datenmodell angegeben werden, wie es in vielen IR-Systemen, insbesondere auch in freeWAIS-sf verwendet wird. Einem Dokument ist eine Menge von Attributen zugeordnet, wobei ein Attribut A_i einen Attributnamen N_i sowie einen Datentyp D_i besitzt. Das Schema S einer Datenbank ergibt sich aus der Gesamtmenge der den Dokumenten zugeordneten Attribute:

$$S = \{N_1 : D_1, \ldots, N_n : D_n\}$$

Aufbauend auf dieses Modell wird im folgenden Abschnitt eine logische Datenunabhängigkeit für freeWAIS-sf-Datenbanken hergeleitet.

4 Logische Datenunabhängigkeit

Nachdem nun eine physikalische Datenunabhängigkeit erreicht ist, wo für jede Datenbank eine Menge von Attributen und Prädikaten zur Formulierung einer Anfrage zur Verfügung steht, stellt sich die Frage, wie mit der parallelen Befragung mehrerer Datenbanken umgegangen wird. Besitzen alle Datenbanken das gleiche Schema, liegt der Fall klar auf der Hand. In einer Anfragemaske kann ein Eingabefeld zu jedem Attribut angegeben werden. In der Regel besitzen unterschiedliche Datenbanken jedoch heterogene Schemata; die Datenbanken bieten jeweils unterschiedliche Attribute zur Formulierung von Anfragen an. Das ergibt sich im wesentlichen aus den unterschiedlichen Typen der in den Datenbanken verwalteten Dokumente. Ein Datenbankschema zur Verwaltung von technischen Berichten enthält z. B. die Attribute *Autor*, *Titel*, *Jahr* und *Institution* während ein Schema zur Verwaltung von Zeitschriftenaufsätzen die Attribute *Autor*, *Artikeltitel*, *Zeitschriftentitel*, *Verlag* und *Jahr* anbietet.

Ziel ist es nun, dem Benutzer eine externe Sicht auf Datenbanken mit unterschiedlichen Schemata zu geben. Das heißt, der Benutzer soll die Möglichkeit erhalten, Anfragen zu stellen, ohne Rücksicht auf die Schemata der zu befragenden Datenbanken nehmen zu müssen, oder gar die Anfrage für jede zu befragende Datenbank reformulieren zu müssen. Es muß ein Mechanismus gefunden werden, der Anfragen von dieser externen Ebene auf die konzeptionelle Ebene abbildet. Insbesondere müssen die in der Anfrage der externen Ebene vorkommenden Attribute auf die in den Schemata der zu befragenden Datenbanken vorkommenden Attribute der konzeptionellen Ebene abgebildet werden.

Die ausschließliche Verwendung von Attributen aus der Schnittmenge der beteiligten Schemata zur Formulierung einer Anfrage ist eine unzureichende Lösung, da eine solche Schnittmenge wohl

häufig nicht viel größer als die leere Menge ist. Ebenfalls unzureichend wäre die einfache Umbenennung der Attribute in den einzelnen Schemata, so daß letztendlich doch wieder einheitliche Schemata entstehen. Das wäre nur zulässig, wenn man — abgesehen von den unterschiedlichen Attributnamen — doch homogene Schemata vorliegen hätte.

Die hier vorgestellte Lösung basiert auf einer vordefinierten Menge von Attributen, die auf der externen Ebene sichtbar sind, d. h. die der Benutzer in seinen Anfragen benutzen kann. Die Auswahl dieser Attribute wurde anhand des *Scientific and Technical Attribute Set (STAS)* getroffen[1]. STAS definiert mehr als 500 Standardbezeichner für Attribute in wissenschaftlichen und technischen Datenbanken. Weitere Informationen zu STAS sind im Internet unter der WWW-Adresse `<http://vinca.cnidr.org/STAS.html>` erhältlich.

Als weitere Voraussetzung muß gelten, daß die Schemata der zu befragenden Datenbanken — abgesehen von Umbenennungen — ausschließlich Anfrageattribute aus dieser Menge anbieten.

Innerhalb der Attributmenge werden nun Ähnlichkeiten der Attribute untereinander gesucht. Durch Generalisierung bzw. Spezialisierung werden Beziehungen zwischen diesen Attributen definiert. Seien z. B. die Attribute *Autor*, *Herausgeber* und *Urheber* in der Menge der vordefinierten Attribute, dann sind *Herausgeber* und *Autor* Spezialisierungen des Attributes *Urheber*. Durch derartige Beziehungen entsteht eine *Attributhierarchie* (Abbildung 1). An der Wurzel dieser Hierarchie steht das ausgezeichnete Attribut *TOP*, welches alle anderen Attribute verallgemeinert. Die Befragung einer Datenbank mit diesem Attribut kommt einer Befragung mit allen im Schema vorhandenen Attributen gleich.

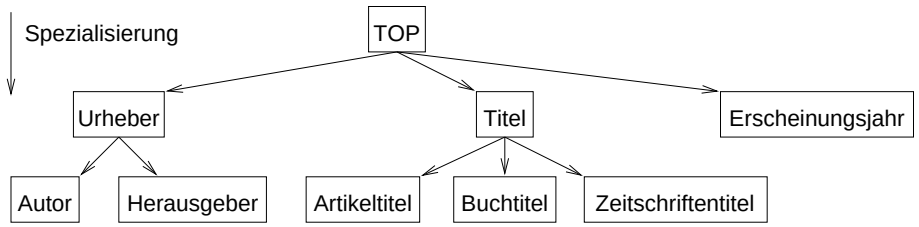

Abbildung 1: Beispiel einer Attributhierarchie

Die Attributhierarchie wird zusammen mit den folgenden Operationen verwendet, um eine Anfragebedingung q_A mit einem Attribut A aus der Hierarchie auf eine oder mehrere neue Anfragebedingungen für eine Datenbank mit Schema S abzubilden:

1. **Gleichheit**: Existiert das Attribut A in S, so findet keine Änderung der Anfragebedingung statt. Die Anfragebedingung bezieht sich ja unmittelbar auf das durch das Datenbankattribut A abgedeckte Datenbankwissen.

2. **Spezialisierung**: Existiert A nicht in S, werden die Attribute $A_1, \ldots, A_n$ in S gesucht, die gemäß der Attributhierarchie spezieller sind als A (d. h. alle Nachkommen von A). Durch die Eigenschaft der Attribute $A_i \in A_1, \ldots, A_n$, Spezialisierungen von A zu sein, ist gewährleistet, daß das in diesen Attributen in der Datenbank repräsentierte Wissen zumindest eine Teilmenge dessen ist, auf was sich das Anfrageattribut A bezog. Gibt es solche Attribute, so wird die ursprüngliche Anfragebedingung ersetzt: Für jedes der spezielleren Attribute A_i wird disjunktiv die ursprüngliche Anfragebedingung, in der der Attributname A durch den Namen des spezielleren Attributs A_i ersetzt wird, an die neue Anfragebedingung angehängt. Sei z. B. die Anfragebedingung *Urheber=Fuhr* gegeben. Soll nun eine Datenbank befragt werden, die nicht das Attribut *Urheber*, wohl aber die Attribute *Autor* und *Herausgeber* besitzt, so würde sich für diese Datenbank als neue Anfrage ergeben: *Autor=Fuhr oder Herausgeber=Fuhr*.

[1] für die Beispiele in diesem Artikel wurden zur Vereinfachung daraus nur 8 Attribute ausgewählt

3. **Generalisierung**: Wurden auch keine spezielleren Attribute in S gefunden, so wird das nächste Attribut A' in S gesucht, welches eine Generalisierung des ursprünglichen Attributes darstellt (also bezüglich der Attributhierarchie der nächste Vorfahre von A). A' repräsentiert das durch das Attribut A angefragte Wissen, es ist ja eine Generalisierung von A. Problematisch ist dabei jedoch, daß durch das allgemeinere Attribut darüberhinaus weiteres Datenbankwissen abgedeckt wird (es gibt neben dem Attribut A weitere Attribute, die Spezialisierungen von A' sind; das zugehörige Datenbankwissen wird ebenfalls von A' abgedeckt). Das Attribut A' ersetzt dann das Attribut aus der ursprünglichen Anfrage. Sei hier nun die Anfragebedingung *Artikeltitel=(Information Retrieval)* gegeben. Soll nun eine Datenbank befragt werden, die weder das Attribut *Artikeltitel* noch eine Spezialisierung davon enthält, wohl aber das Attribut *Titel*, so ergibt sich durch die Generalisierung die neue Anfragebedingung *Titel=(Information Retrieval)*.

4. **Ignorieren**: Scheitern die drei vorhergenannten Operationen, so wird die Anfragebedingung für die Anfrage an die zu S gehörende Datenbank ignoriert, da kein Attribut im Schema der zu befragenden Datenbank gefunden werden konnte, welches dem Anfrageattribut A semantisch ähnlich ist. Es muß also davon ausgegangen werden, das im Schema der Datenbank kein Attribut existiert, welches das durch das Attribut A angefragte Wissen in der Datenbank abdeckt.

Durch die Attributabbildung bei der Transformation von Anfragen wird zusätzliche Unsicherheit in den Retrievalprozeß hineingetragen. Insbesondere die Anwendung der beiden letztgenannten Operationen *Generalisierung* und *Ignorieren* lassen eine Verschlechterung des Suchergebnisses erwarten; es besteht die Gefahr, daß zusätzlich zu den für den Benutzer relevanten Dokumenten weitere irrelevante Dokumente im Suchergebnis auftauchen, da sich die transformierte Anfrage auf ein größeres Datenbankwissen bezieht, als durch die ursprüngliche Anfrage beabsichtigt wurde. Diesem Phänomen sollte durch eine geeignete Gewichtung der Anfragebedingungen bzw. des Suchergebnisses entgegengewirkt werden.

Im Falle der *Generalisierung* sollten die entsprechenden Anfragebedingungen herabgewichtet werden, da diesen bei der Prozessierung der Anfrage eine geringere Bedeutung als anderen Anfragebedingungen zugemessen werden sollte. Als Maß für diese Herabgewichtung kann die Ähnlichkeit der bei der Generalisierung einbezogenen Attribute herangezogen werden; diese könnten für die in der Attributhierarchie dargestellten Spezialisierungbeziehungen als Kantenbeschriftung eingeführt werden. Der Grad der Ähnlichkeit der Attribute könnte heuristisch ermittelt werden, bei folgender Interpretation: Seien A und A' zwei Attribute. Für diese gelte, daß A gemäß der Attributhierarchie spezieller als A' ist. Die Ähnlichkeit von A und A' ist dann die Wahrscheinlichkeit dafür, daß ein Benutzer ein Dokument als relevant einstuft, welches aus einer Anfragebedingung resultiert, in der das ursprüngliche Attribut A durch *Generalisierung* durch das Attribut A' ersetzt wurde.

Im Fall des *Ignorierens* von Anfragebedingungen sollten die resultierenden Dokumente im Gesamtsuchergebnis mit einem verringerten Gewicht erscheinen; die Wahrscheinlichkeit, daß eines dieser Dokumente relevant in bezug auf die Anfrage des Benutzers ist, ist ja bezogen auf die ursprüngliche Anfrage als geringer zu bewerten. Hierbei ließe sich das Maß der Herabgewichtung aus folgender Betrachtung heuristisch ermitteln: Sei q_A eine Anfragebedingung, die für die Befragung einer Datenbank ignoriert wird und E_A das zugehörige Suchergebnis. Man erzeuge das Ergebnis E'_A, indem man die Dokumente aus E_A eleminiert, die vom System durch das Ignorieren der Anfragebedingung q_A in das Ergebnis aufgenommen wurden. Setzt man nun die Anzahlen der Dokumente in E_A und in E'_A ins Verhältnis, so erhält man das gesuchte Maß für die Herabgewichtung der Dokumente im Gesamtergebnis. Dazu folgendes Beispiel: Die Anfrage eines Benutzers enthalte die Bedingung *Erscheinungsjahr>1990*. Das Suchergebnis einer Datenbank enthalte nach Ignorieren dieser Anfragebedingung 10 Dokumente. Sind nun hiervon 5 Dokumente tatsächlich nach 1990 erschienen, so ergibt sich die Wahrscheinlichkeit, daß eines der gefundenen Dokumente relevant in bezug auf die ignorierte Anfragebedingung ist, zu $0,5$.

Eine Voraussetzung für die Schaffung logischer Datenunabhängigkeit nach dem hier vorgestellten Konzept ist das Vorhandensein einer Attributhierarchie. Dieses kann sich allerdings als Problem erweisen. Weisen nämlich die zu befragenden Datenbanken Dokumente aus sehr unterschiedlichen Anwendungsgebieten auf, so sind wohl Spezialisierungs- bzw. Generalisierungsbeziehungen zwischen den in solchen Datenbanken vorhandenen Attributen nicht mehr herstellbar. So dürfte es schwierig sein, eine externe Sicht über die Schemata der Datenbanken A und B herzustellen, wenn Datenbank A z. B. Produktinformationen beinhaltet, während Datenbank B Literaturreferenzen verwaltet. Auf der anderen Seite stellt sich hier natürlich die Frage, inwiefern es überhaupt sinnvoll ist, zwei derart unterschiedliche Datenbanken zugleich zu befragen.

5 SFgate

SFgate [Gövert & Pfeifer 96] ist ein Gateway zwischen dem World-Wide Web (WWW) und freeWAIS-sf. Die Schnittstelle zum Benutzer stellen HTML-Formulare dar, in denen der Benutzer Anfragen sowie die zu befragenden Datenbanken spezifizieren kann. Die Suchergebnisse werden dem Benutzer wiederum in Form von HTML-Seiten dargestellt. Zunächst werden die von SFgate zu einer Anfrage ermittelten Ergebnisdokumente in einer Kurzübersicht dem Benutzer präsentiert, aus der der Benutzer die ihm interessant erscheinenden Überschriften selektieren kann. Zu diesen werden die entsprechenden Dokumente dann komplett angezeigt. Über diese Basisfunktionalität hinaus bietet SFgate eine größere Anzahl von Optionen, über die die Prozessierung der Anfragen bzw. das Erscheinungsbild der Ausgabe beeinflußt werden kann. Abbildung 2 zeigt eine Beispielanwendung.

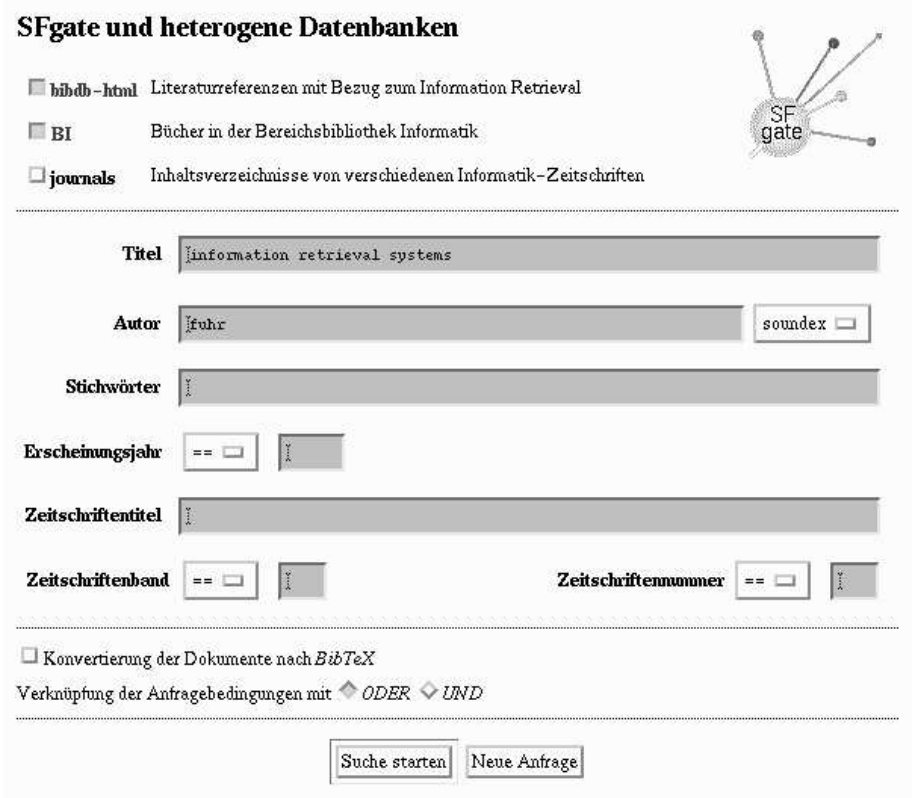

Abbildung 2: Eine Beispielanwendung mit SFgate und drei Datenbanken

Abgesehen von der Umgewichtung von Anfragebedingungen und Retrievalergebnissen bei den Operationen *Generalisierung* bzw. *Ignorieren* implementiert SFgate logische Datenunabhängigkeit nach dem im vorherigen Abschnitt beschriebenen Verfahren [Gövert 96]. Dazu gibt es eine vordefinierte Attributhierarchie, die für die Recherche in heterogenen Literaturdatenbanken ausgelegt ist. Für die in Abbildung 2 dargestellte Anwendung wurde die Attributhierarchie aus Abbildung 1 verwendet. Es ist jedoch möglich, die Attributhierarchie beliebig zu ändern oder komplett auszutauschen. Zusätzlich zur Attributhierarchie muß zu jeder der zu befragenden Datenbanken das Datenbankschema, wie es in Abschnitt 3 beschrieben wurde, angegeben werden. Dazu wird die Abbildung der in den Datenbanken existierenden Attribute auf die jeweils entsprechenden Attribute aus der Attributhierarchie in einer von SFgate lesbaren Konfigurationsdatei abgelegt. Die Attributkonfiguration für die in der Beispielanwendung befragte *bibdb-html*-Datenbank sieht folgendermaßen aus:

```
$attributes = {
  'py:numeric'      => 'Erscheinungsjahr',
  'au:soundex,text' => 'Autor',
  'ti:text'         => 'Buchtitel,Artikeltitel',
  'jt:text'         => 'Zeitschriftentitel',
  };
```

Hierbei wird beispielsweise das Datenbankattribut „py“ (*publication year*) mit dem Datentyp *numeric* auf das Attribut *Erscheinungsjahr* aus der Hierarchie abgebildet. Ebenso wird das Datenbankattribut „ti“ (*title*) mit dem Datentyp *text* auf die Attribute *Buchtitel* und *Artikeltitel* abgebildet.

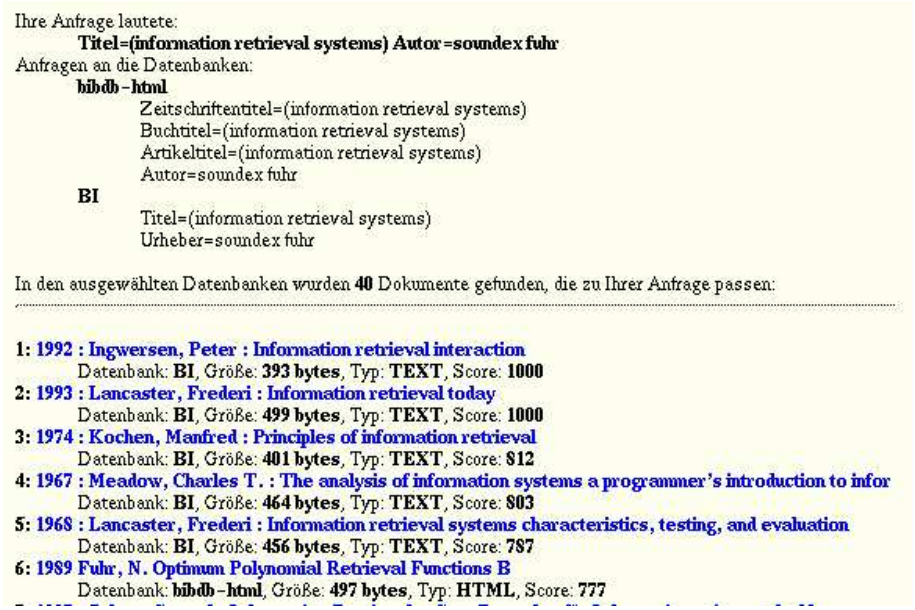
Ihre Anfrage lautete:
Titel=(information retrieval systems) Autor=soundex fuhr
Anfragen an die Datenbanken:
bibdb-html
Zeitschriftentitel=(information retrieval systems)
Buchtitel=(information retrieval systems)
Artikeltitel=(information retrieval systems)
Autor=soundex fuhr
BI
Titel=(information retrieval systems)
Urheber=soundex fuhr
In den ausgewählten Datenbanken wurden **40** Dokumente gefunden, die zu Ihrer Anfrage passen:
1: 1992 : Ingwersen, Peter : Information retrieval interaction
Datenbank: **BI**, Größe: **393 bytes**, Typ: **TEXT**, Score: **1000**
2: 1993 : Lancaster, Frederi : Information retrieval today
Datenbank: **BI**, Größe: **499 bytes**, Typ: **TEXT**, Score: **1000**
3: 1974 : Kochen, Manfred : Principles of information retrieval
Datenbank: **BI**, Größe: **401 bytes**, Typ: **TEXT**, Score: **812**
4: 1967 : Meadow, Charles T. : The analysis of information systems a programmer's introduction to infor
Datenbank: **BI**, Größe: **464 bytes**, Typ: **TEXT**, Score: **803**
5: 1968 : Lancaster, Frederi : Information retrieval systems characteristics, testing, and evaluation
Datenbank: **BI**, Größe: **456 bytes**, Typ: **TEXT**, Score: **787**
6: 1989 Fuhr, N. Optimum Polynomial Retrieval Functions B
Datenbank: **bibdb-html**, Größe: **497 bytes**, Typ: **HTML**, Score: **777**

Abbildung 3: Suchergebnis zur Anfrage aus Abbildung 2

Abbildung 3 zeigt das Suchergebnis aus der Anfrage in Abbildung 2, sowie die aus der Attributabbildung hervorgehenden Anfragen für die beiden befragten Datenbanken *bibdb-html* und *BI*. Hier ist bei der *bibdb-html*-Datenbank z. B. die Spezialisierung des Attributes *Titel* auf die Attribute *Zeitschriftentitel*, *Buchtitel* und *Artikeltitel*, sowie bei der *BI*-Datenbank die Generalisierung des Attributes *Autor* auf das Attribut *Verfasser* zu beobachten.

6 Zusammenfassung und Ausblick

Mit SFgate existiert nun die Möglichkeit, auf Basis einer logischen Datenunabhängigkeit in ihren Schemata heterogene freeWAIS-sf-Datenbanken parallel zu befragen. Das hier vorgestellte Konzept ist jedoch in vielerlei Hinsicht ausbaufähig:

Datenbankauswahl Derzeit findet die Datenbankauswahl manuell statt. Der Benutzer muß genau wissen, wo er die Datenbanken findet, die für ihn interessant erscheinen. Anstrebenswert ist ein System, welches aufgrund einer vom Benutzer gestellten Anfrage ermittelt, welche Datenbanken relevante Dokumente enthalten und daher befragt werden sollten.

Subsumierendes Schema Werden Datenbanken automatisch aufgrund einer Benutzer-Anfrage ausgewählt, so sollte vom System ebenfalls eine adäquate Benutzungsschnittstelle generiert werden, z. B. in Form eines HTML-Formulars. Dazu müssen die (heterogenen) Schemata der zu befragenden Datenbanken ausgewertet werden, um aus den dort vorhandenen Attributen eine Anzahl von Eingabefeldern zu erzeugen. Die Menge der darin vorkommenden Attribute sollte möglichst genau den Schemata der einzelnen Datenbanken entsprechen.

Integration von IR-Systemen In Abschnitt 3 wurde physikalische Datenunabhängigkeit für freeWAIS-sf-Datenbanken beschrieben. Wünschenswert ist jedoch, auf dieser Ebene auch andere IR-Systeme zu integrieren, z. B. Systeme, die über Z39.50 [ANSI 95] (Protokoll-Standard zur Vernetzung von Datenbank- und Information-Retrieval-Anwendungen) zugreifbar sind. Dazu müssen insbesondere auch die von den einzelnen Systemen zurückgelieferten Suchergebnisse interpretiert werden, um aus den Teilergebnissen der Einzelsysteme ein aussagekräftiges Gesamtergebnis erstellen zu können.

Die hier vorgestellten Erweiterungen sind unter anderem Gegenstand des *MeDoc*-Projektes, ein Fachinformationsprojekt im Bereich der Informatik. Ziel dieses Projektes ist eine verteilte, heterogene, digitale Informatikbibliothek. Studenten und Wissenschaftlern soll eine „kritische Masse“ von Informatik-Literatur über das Internet zur Verfügung gestellt werden. Insbesondere sollen aber auch Werkzeuge geschaffen und erprobt werden, die eine effektive und effiziente Erschließung und Nutzung der neuen Informationsquellen erlauben. Weitere Informationen zum MeDoc-Projekt sind im Internet unter der WWW-Adresse `<http://medoc.informatik.tu-muenchen.de/>` erhältlich.

Literatur

ANSI. (1995). *Information Retrieval (Z39.50): Application Service Definition and Protocol Specification (ANSI/NISO Z39.50-1995).* Technischer Bericht, NISO Press, Bethesda, MD.

Fuhr, N.; Rölleke, T. (1997). A Probabilistic Relational Algebra for the Integration of Information Retrieval and Database Systems. *ACM Transactions on Information Systems 14(1)*, S. 32–66.

Fuhr, N. (1996). Object-Oriented and Database Concepts for the Design of Networked Information Retrieval Systems. In: Barker, K.; Özsu, M. (Hrsg.): *Proceedings of the Fifth International Conference on Information and Knowledge Management*, S. 164–172.

Gövert, N.; Pfeifer, U. (1996). *SFgate. The WWW Gateway for freeWAIS-sf.* Universität Dortmund. `<http://ls6-www.informatik.uni-dortmund.de/ir/projects/SFgate/SFgate.\ html>`.

Gövert, N. (1996). *SFgate and Heterogeneous Databases.* `<http://ls6-www.informatik.uni\ -dortmund.de/ir/projects/SFgate/heterogeneous.html>`.

Pfeifer, U.; Fuhr, N.; Huynh, T. (1995). Searching Structured Documents with the Enhanced Retrieval Functionality of freeWAIS-sf and SFgate. In: D. Kroemker (Hrsg.): *Computer Networks and ISDN Systems; Proceedings of the third International World-Wide Web Conference*, S. 1027–1036. Elsevier, Amsterdam - Lausanne - New York - Oxford - Shannon - Tokyo.

Tsichritzis, D.; Klug, A. (1978). The ANSI/X3/SPARC DBMS Framework Report of the Study Group on Database Management Systems. *Information Systems 3*, S. 173–191.