Semantikbasierte Ähnlichkeitssuche in Datenbanksystemen

Jürgen Palkoska

Institut Für Anwendungsorientierte Wissensverarbeitung (FAW)
Johannes Kepler Universität Linz
Hauptstr. 119
A-4232 Hagenberg
jp@faw.uni-linz.ac.at

Abstract: Semantikbasierte Ähnlichkeitssuchen repräsentieren eine fundamentale Funktion in vielen Anwendungsbereichen der Informatik. Allerdings existieren kaum Ansätze, die Konzepte für eine flexible Integration semantischer Metadaten in konventionelle Datenbankanwendungen zur Verfügung stellen. In der vorliegenden Dissertation wurden daher Methoden entwickelt, welche die Modellierung semantischer Metadaten innerhalb von Datenbanksystemen gestatten und dadurch flexible semantikbasierte Ähnlichkeitssuchen ermöglichen.

1 Einleitung

Ähnlichkeitssuchen stellen eine wichtige Funktion in vielen aktuellen Bereichen der Informatik dar. Abhängig von der jeweiligen Applikation kommen dabei die unterschiedlichsten Suchverfahren zur Anwendung. Allen gemein ist jedoch die Berücksichtigung unscharfer Aspekte der Abfragebearbeitung. D.h. anstatt die bloßen Fakten in Betracht zu ziehen, wird auch die Bedeutung (die Semantik) der Daten einbezogen. Die informationstechnologische Modellierung der semantischen Metadaten ist in den meisten Fällen sehr komplex und daher für die jeweiligen Anwendungen, wie z.B. Data-Mining-Verfahren, Multimediadatenbanken oder CAD-Systeme optimiert.

Kommerzielle Datenbanksysteme (DBS) repräsentieren hingegen in einer Vielzahl von Anwendungsbereichen den Standard für die Verwaltung großer Datenmengen. Die dabei eingesetzten Technologien sind für die effiziente Suche nach Datenelementen optimiert, die exakt den spezifizierten Abfragekriterien entsprechen. Mit dem Entstehen neuer Anwendungsbereiche führt die auf dem scharfen Vergleich der Daten basierende Arbeitsweise konventioneller DBS jedoch häufig auch zu Nachteilen.

Zur Veranschaulichung der Problematik soll folgendes Beispiel dienen: Ein Benutzer eines Tourismusinformationssystems sucht nach einem Hotelzimmer im Ort *Wagrain*, das zwischen 100,- und 120,- Euro kostet. Findet das DBS keine passenden Angebote, so könnte der Benutzer zwar den Preisbereich aufweichen und einen weiteren Suchlauf zwischen 90,- und 130,- Euro ausführen, ein passendes Angebot im Nachbarort *St. Johann* wird jedoch u.U. übersehen.

Wäre das DBS allerdings in der Lage, selbständig semantische Zusatzinformationen wie z.B. die geographische Nähe von Orten zu berücksichtigen, so könnte es dem Benutzer im Falle des Scheiterns einer konventionellen Abfrage selbständig sinnvolle Alternativen vorschlagen.

Ziel der Dissertation war daher, Verfahren für die Integration semantikbasierter Ähnlichkeitssuchen in konventionelle Datenbankanwendungen zu entwickeln. Dadurch werden konventionelle DBS in die Lage versetzt, beim Ausführen von Suchoperationen auch die Semantik der Daten und dadurch die semantische Ähnlichkeit von Objekten zu berücksichtigen. Voraussetzung für entsprechende Verfahren ist eine adäquate Modellierung semantischer Metadaten auf Datenbankebene. Grundlegende Forderung bei der Entwicklung der Methoden ist ihre vollkommen transparente Integration in konventionelle DBS. Nur so ist die einfache Erweiterung bereits existierender Datenbankanwendungen um semantikbasierte Abfragemechanismen möglich.

Dieser Artikel ist folgendermaßen strukturiert: Nach einem Überblick über existierende Konzepte für Ähnlichkeitssuchen in DBS und Möglichkeiten für das Messen von Ähnlichkeitsgraden erfolgt die Vorstellung des im Rahmen der Dissertation entwickelten Vague Query Systems (VQS). Das VQS umfasst Methodensammlungen für die Modellierung semantischer Metadaten in DBS, für mehrere häufig benötigte Suchverfahren, sowie für effizienzsteigernde Indizierungsverfahren. Abschließend wird auf die Evaluierung der Konzepte des VQS eingegangen, in deren Rahmen ein bestehendes E-Commerce-System um semantikbasierte Ähnlichkeitssuchen erweitert wurde.

2 Existierende Ansätze

Viele Ansätze, die Ähnlichkeitssuchen für Datenbanken zur Verfügung stellen, sind auf numerische Attribute beschränkt. Die SQL-Erweiterungen von Bosc et. al. [BGH88], die *Fuzzy Database-Query Language* von Wong et. al. [WL90] und *Fuzzy Base* von Gazotti et. al. [Gaz95] stellen einige Beispiele dafür dar. *ARES* ist ein Ansatz, der auch die Ähnlichkeit nicht-numerischer Attribute berücksichtigt, indem die Ähnlichkeitsgrade zwischen allen möglichen Attributausprägungen in sog. *Ähnlichkeitsrelationen* verwaltet werden [IH86]. Als Nachteil von ARES ist zu werten, dass Attributräume großer Kardinalität zu sehr großen Ähnlichkeitsrelationen führen. Auch das von Motro vorgestellte *VAGUE*–System versucht, die Abfragefunktionen konventioneller Datenbanksysteme in bezug auf Ähnlichkeitssuchen zu erweitern [Mot88]. Der Ansatz basiert auf sog. *Daten-Metriken*, welche die Semantik sowohl von numerischen, als auch von nicht-numerischen Attributwerten ausdrücken. Da VAGUE mehrere Typen von Metriken zur Verfügung stellt, erweist sich das System als sehr flexibel. Leider basiert VAGUE auf einer interaktiven Vorgehensweise, wodurch die Ergebnisse durch eine mehrmalige Interaktion mit dem Benutzer extrahiert werden müssen.

3 Modellierung von Ähnlichkeitsaspekten in Informationssystemen

3.1 Modellierung der Semantik

Um das menschliche Verständnis von Ähnlichkeitssuchen algorithmisch nachzubilden, muss insbesondere auch die Semantik, d.h. die Bedeutung der die Objekte beschreibenden Objekteigenschaften, explizit modelliert werden. In Spezialanwendungen existieren dafür unterschiedlichste Ansätze. Vektormodelle, neuronale Netze, selbstorganisierende Karten, semantische Netze und Topic Maps stellen entsprechende Beispiele dar. Ein Überblick über existierende Verfahren kann in [Pal02] nachgelesen werden. Meist trennen Spezialanwendungen nicht explizit zwischen der Semantik und der Programmlogik. Daher sind die semantischen Metadaten häufig inhärenter Bestandteil der Algorithmen.

3.2 Definition von Ähnlichkeitsmaßen

In nahezu allen Anwendungsbereichen für Ähnlichkeitssuchen kommen Maßzahlen zur Anwendung, um den Grad der Ähnlichkeit zweier Objekte auszudrücken. Die dafür eingesetzten Funktionen bilden Objektpaare meist auf Basis ihrer semantischen Metadaten in den nicht-negativen reellen Zahlenraum ab:

$$D : O \times O \rightarrow \Re_0^+$$

Im Rahmen der Dissertation kommt das Konzept der *semantischen Distanz* zur Anwendung. Dabei weist ein kleiner Wert der Funktion *D* auf eine große Ähnlichkeit der gegenübergestellten Objekte hin. Eine Distanz von *0* ist somit ein Indikator für die semantische Gleichheit zweier Elemente. Nach oben hin wird die semantische Distanz häufig durch den Wert *1* begrenzt, der auf die größte im betreffenden Szenario denkbare *Unähnlichkeit* hindeutet. Die Funktion D wird meist gemäß einer *Metrik* definiert. Liegen die semantischen Metadaten in Vektorräumen vor, so kommen für die Definition der Metrik z.B. *Taxicab-*, *Euklidische-* und *Maximum Coordinate- Distanz* in Frage.

4 Das Vague Query System (VQS)

Da sich bei der Analyse existierender Spezialanwendungen zeigte, dass deren Konzepte für Ähnlichkeitssuchen nicht ohne weiteres auf konventionelle Datenbanksysteme umgelegt werden können, wurde das sog. *Vague Query System* (*VQS*) entwickelt. Vorrangiges Ziel bei der Konzeption des Systems war der Entwurf eines Modells, das die flexible Darstellung der Semantik für eine möglichst große Klasse von Aufgabenstellungen ermöglicht. Dabei wurde der transparenten Integrierbarkeit des VQS in bereits existierende Datenbankanwendungen eine große Bedeutung beigemessen.

Das VQS wurde als applikationsbereichsunabhängiger Aufsatz für beliebige Datenbankanwendungen konzipiert und bildet eine zusätzliche Schicht zwischen dem Datenbankmanagementsystem und der Applikationslogik.

Dadurch muss bei der Integration semantikbasierter Ähnlichkeitssuchen in bereits im Einsatz befindliche Datenbankanwendungen deren Datenbankstruktur in keiner Weise verändert werden.

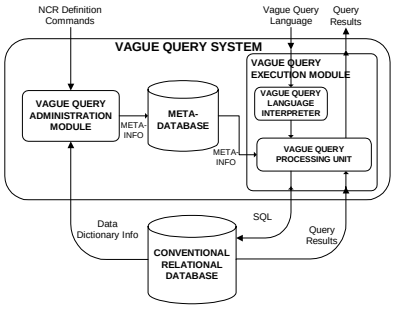


Abbildung 4.1 Systemarchitektur des Vague Query Systems

4.1 Verwaltung semantischer Metadaten in NCR-Tabellen

In bezug auf die Modellierung semantischer Metadaten in DBS wurde insbesondere das Vektorraummodell als optimale Lösung identifiziert. Vor allem seine universelle Einsetzbarkeit für eine Vielzahl von Anwendungsbereichen passt gut in das Konzept des VQS. Um allen für Datenbankanwendungen relevanten Anforderungen wie z.B. Mehrbenutzerzugriff und Recovery zu genügen, erfolgt im VQS die Abspeicherung der Vektordaten nicht in einem proprietären Dateiformat, sondern innerhalb der Datenbank. Beim Ausführen von Ähnlichkeitssuchen greift das VQS auf diese als *NCR-* (*Numeric-Coordinate-Representation-*) *Tabellen* bezeichneten Strukturen zurück, um die benötigten Vektordaten zu ermitteln. Jede NCR-Tabelle repräsentiert einen Vektorraum beliebiger Dimensionalität.

4.2 Mehrschichtige Berechnung semantischer Ähnlichkeitsmaßzahlen

Um auch die Semantik von Datenbankobjekten komplexer Struktur ausdrücken zu können, wurde im VQS folgende dreischichtige Architektur für die Berechnung von Ähnlichkeitsmaßzahlen definiert.

Schicht 1: Projektion von Attributen in Vektorräume

In der ersten Schicht erfolgt die Modellierung der Semantik der kleinsten Informationseinheiten der Datenbank, nämlich der Attribute. Zu diesem Zweck enthält die Metadatenbank des VQS Informationen über die Projektion von Attributwerten in beliebige multidimensionale Vektorräume (modelliert durch NCR-Tabellen). Die Definition der semantischen Metadaten bleibt vor der eigentlichen Applikationslogik verborgen. Der Administrator des VQS ist somit der einzige, der sich um die Definition der semantischen Metadaten kümmern muss. Dementsprechend wird er auch als *Domain-Expert* bezeichnet (vgl. z.B. [FL95]).

Auf Basis dieses Konzepts kann die strukturelle Modellierung des Informationssystems vollkommen von der Modellierung der Semantik entkoppelt erfolgen. Um die semantische Ähnlichkeit zweier Attributwerte zu errechnen, zieht das VQS auf Basis der den Attributen zugeordneten NCR-Tabellen eine normierte metrische Distanz (z.B. die Euklidische Distanz) zwischen den Vektorrepräsentationen zweier Attributwerte als Maßzahl für ihre Ähnlichkeit heran.

Schicht 2: Parallele Projektion von Attributen in mehrere Vektorräume

Häufig erscheint es sinnvoll, beim Vergleich von Attributwerten parallel mehrere Typen semantischer Metadaten zu berücksichtigen. Bei der Gegenüberstellung der Größe von Orten ist beispielsweise häufig sowohl die Berücksichtigung der Einwohnerzahl als auch der Grundfläche des Ortes (beide darstellbar durch einen 1-dimensionalen Vektorraum) von Relevanz. Schicht 2 des VQS trägt Anforderungen dieser Art Rechnung, indem es die parallele Projektion von Attributen in mehrere Vektorräume und somit die Zuordnung zu mehreren NCR-Tabellen erlaubt.

Um die Ähnlichkeit solcher Attributwerte auszudrücken, werden die in Schicht 1 errechneten metrischen Distanzen mathematisch zu einer einzelnen repräsentativen Maßzahl kombiniert. Bei dieser Berechnung kann den in Schicht 1 referenzierten Vektorräumen sogar ein unterschiedliches Gewicht beigemessen werden.

Schicht 3: Semantik komplexer Objekte

Die Modellierung semantischer Metadaten erfolgt im VQS wie bereits erläutert auf Attribut-Ebene (Schicht 1 und 2). Die dritte Schicht ermöglicht jedoch auch das Messen der Ähnlichkeit komplexer Objekte, die aus beliebig vielen Attributen zusammengesetzt sind. Da auch Datenbankabfragen dynamisch virtuelle Objekte definieren, deren Semantik mit den restlichen Datenbankobjekten verglichen werden muss, wurde das VQS in die Lage versetzt, die in den Schichten 1 und 2 definierte Semantik dynamisch zu kombinieren und auf ihrer Basis die semantische Ähnlichkeit komplexer Objekte zu errechnen. Um in Schicht 3 die Ähnlichkeit komplexer Objekte durch eine einzelne Maßzahl auszudrücken, kommt erneut eine spezielle Berechnungsvorschrift zur Anwendung. Bei der Berechnung kann den betreffenden Attributen wiederum ein unterschiedliches Gewicht beigemessen werden. Auch das als *Totale Distanz* (*TD*) bezeichnete Resultat der Berechnung erfüllt gemäß [Pat99] die Eigenschaften einer Metrik und repräsentiert somit die *semantische Distanz* zwischen komplexen Objekten.

4.3 Suchverfahren des VQS

Um eine möglichst große Klasse von Aufgabenstellungen in einer Vielzahl von Anwendungsbereichen semantikbasierter Ähnlichkeitssuchen abdecken zu können, werden vom VQS alle für Ähnlichkeitssuchen relevanten Suchmechanismen unterstützt. Beispiele dafür sind *Ranking-Verfahren*, *Good-Match-Suchen*, *(k-) Nearest-Neighbor-Suchen* und *Unscharfe Joins*. Als repräsentativer Vertreter wird im Folgenden die Methode für das effiziente Ausführen von *Nearest-Neighbor-Suchen* vorgestellt.

4.4 Das inkrementelle Hyperkubus-Verfahren

Optimal ist eine Nearest-Neighbor-Suche erst dann, wenn der beste Treffer (d.h. das Objekt mit der kleinsten Totalen Distanz zum Abfrageobjekt) gefunden wird, ohne die zeitaufwendige Berechnung des Wertes TD für jedes einzelne Objekt der Datenbank durchführen zu müssen. Ziel ist somit, die Objekte der Datenbasis durch Einsatz eines adäquaten Indizierungsmechanismus so zu organisieren, dass optimale Treffer für semantikbasierte Ähnlichkeitssuchen auf Basis ihrer semantischen Metadaten möglichst effizient abgerufen werden können.

Obwohl sich Schicht 1 des VQS multidimensionaler Vektordaten bedient, um die Semantik der Objekte abzubilden, hat es keinen Sinn, bereits auf dieser Ebene nach Attributwerten zu suchen, die im Vektorraum die kleinste semantische Distanz zum betreffenden Vektor des Abfrageobjekts aufweisen. Durch die Kombination der metrischen Distanzen in den Schichten 2 und 3 kann nämlich durchaus ein Objekt die kleinste *Totale Distanz* zur Abfrage aufweisen, dessen Attributwerte die Abfragekriterien in Schicht 1 nur bedingt erfüllen. Somit ist kein direkter Einsatz konventioneller multidimensionaler Indexstrukturen in den Vektorräumen der Schicht 1 möglich. Auch in Schicht 3 ansetzende, NCR-Tabellen-übergreifende Indexstrukturen können nicht zur Anwendung kommen, da aufgrund der hohen Flexibilität des VQS nicht vorhersehbar ist, welche Vektorräume zur Abfragezeit in den Schichten 2 und 3 kombiniert werden.

Aus diesem Grund wurde im Rahmen der Dissertation das sog. *inkrementelle Hyperkubus-Verfahren* entwickelt. Dieses tastet sich schrittweise an den optimalen Treffer heran und versucht, die Totale Distanz für so wenige Objekte wie möglich zu berechnen. Grundlage des Hyperkubus-Verfahrens ist die Annahme, dass die Vektorkoordinaten des besten Treffers in Schicht 1 zumindest in relativer Nähe zu den Vektorkoordinaten der Abfragekriterien liegen. Ziel ist daher, in den durch die Abfrage referenzierten Vektorräumen vorrangig in der Nähe der Abfragekriterien nach Treffern zu suchen.

Zu diesem Zweck werden in allen durch die Abfrage referenzierten Vektorräumen Suchintervalle um die mit den Abfragekriterien korrespondierenden Vektorkoordinaten gebildet. Um die Menge der Vektorkoordinaten innerhalb des Suchintervalls zu bestimmen, müsste jedoch für sämtliche Elemente des jeweiligen Vektorraums die metrische Distanz zu den Vektorkoordinaten des Suchwerts berechnet werden. Da jedoch genau diese zeitaufwendige Berechnung vermieden werden soll, kommt eine Approximation der äquidistanten Suchintervalle durch sog. *multidimensionale Hyperkuben* zur Anwendung. Als Grundlage für die Selektion der betreffenden Vektordaten können dadurch multidimensionale Range-Suchen herangezogen werden, deren Ausführung auch in konventionellen Datenbanksystemen relativ effizient möglich ist. Falsche Treffer, die durch die Approximation in die Ergebnismenge gelangen, werden in einem zweiten Schritt einfach weggefiltert.

Die Hyperkuben in allen die Abfrage betreffenden Vektorräumen werden dann so weit vergrößert, bis mindestens ein Objekt *o* der Datenbasis gefunden wurde, dessen Attribute in bezug auf alle Vektorräume mit Vektordaten innerhalb der Hyperkuben korrespondieren.

Bei diesem Objekt muss es sich jedoch keineswegs um den formell besten Treffer der gesamten Datenbasis handeln. Beispielsweise könnte die Datenbasis ein Objekt enthalten, das lediglich in einem einzigen Vektorraum einen Vektorwert außerhalb des korrespondierenden Hyperkubus besitzt. Die *Totale Distanz* dieses Objekts könnte somit durchaus kleiner als jene des Objekts *o* sein. Deshalb werden auf Basis der Totalen Distanz des Objekts *o* die Hyperkuben aller Vektorräume einer mathematischen Berechnung folgend ein weiteres Mal vergrößert. Das eingesetzte Verfahren garantiert, dass unter jenen Objekten, deren Vektordaten in den erweiterten Hyperkuben liegen, der beste Treffer der gesamten Datenbasis enthalten ist. Aus dieser relativ kleinen Objektmenge kann nun sehr leicht der tatsächliche beste Treffer ermittelt werden.

5 Optimierung des Hyperkubus-Verfahrens durch Einsatz einer multidimensionalen Indizierungsmethode

Beim Hyperkubus-Verfahren erfolgt die Extraktion der Vektordaten durch multidimensionale Range-Suchen. Um den Zugriff auf die in den NCR-Tabellen abgelegten Vektordaten zu optimieren, kommen im VQS multidimensionale Indizierungsmechanismen zur Anwendung. Dazu wurde im Rahmen der Dissertation eine Indizierungsmethode entwickelt, die auf der *Pyramidentechnik* von Berchtold et. al. [BBK98] beruht, im Unterschied zum originalen Verfahren jedoch für Vektorräume beliebiger Ausdehnung anwendbar ist. Die Pyramidentechnik beruht auf einer Transformation numerischer Vektoren der Länge *d* in eindimensionale numerische Indexwerte und kann somit der Gruppe der *raumfüllenden Kurven* zugerechnet werden. Da die eindimensionalen Indexwerte in konventionellen Datenbankrelationen abgelegt werden können, braucht nicht auf externe Baumstrukturen ausgewichen werden.

Die Partitionierung des Vektorraums durch die Pyramidentechnik erfolgt in zwei Schritten. Im ersten Schritt unterteilt die Pyramidentechnik den *d*-dimensionalen Vektorraum in eine Anzahl von *2* x *d* multidimensionale Pyramiden, die nach einem festgelegten Schema durchnummeriert werden und deren Spitzen sich im Koordinatenursprung berühren. Im zweiten Schritt wird jede der Pyramiden in mehrere Ebenen unterteilt, die parallel zur Grundfläche der Pyramide ausgerichtet sind. Im Unterschied zur konventionellen Pyramidentechnik weisen die einzelnen Pyramiden im erweiterten Verfahren keine konstante Höhe auf, sondern dehnen sich ausgehend von ihren Spitzen in die Unendlichkeit aus. Der Indexwert eines Vektors $\vec{v}$ setzt sich aus der Nummer der das Element beinhaltenden Pyramide und der Höhenangabe für $\vec{v}$ (gemessen von der Pyramidenspitze) zusammen. Aufgrund eines speziell entwickelten Indexformats können die Indexwerte in bestehenden Datenbankrelationen abgelegt werden.

Mit Hilfe der adaptierten Pyramidentechnik ist die Transformation multidimensionaler Range-Suchen in mehrere eindimensionale Range-Suchen möglich. Diese werden dann auf den Pyramidenindizes der entsprechenden Vektorräume ausgeführt. Somit kann die Pyramidentechnik in allen Verarbeitungsschritten des Hyperkubus-Verfahrens angewendet werden, in denen innerhalb von NCR-Tabellen multidimensionale Range-Suchen ausgeführt werden.

6 Evaluierung des VQS im E-Commerce-System Tiscover

Das Tourismusinformationssystem Tiscover (www.tiscover.com) leidet wie viele andere E-Commerce-Systeme unter dem sog. *Empty Answer Problem*, d.h. sofern ein Benutzer in einem Suchmodul die Abfragekriterien zu eng fasst, ist das System nicht in der Lage, ein entsprechendes Suchergebnis zu ermitteln und liefert lediglich eine leere Ergebnismenge. Um sein Potential für die Umgehung des Empty Answer Problems zu demonstrieren, wurde das VQS im Rahmen der Dissertation in das Veranstaltungs-Modul von Tiscover integriert.

6.1 Definition der semantischen Metadaten

Nachdem die wichtigsten für die Veranstaltungssuche relevanten Suchkriterien identifiziert wurden, war es erforderlich, den betreffenden Attributen adäquate semantische Metadaten zuzuordnen. Dazu mussten für die Attribute Abbildungen ihrer Semantik in Vektorräume gefunden werden.

Um beispielsweise die geographische Nähe von Veranstaltungsorten auszudrücken, wurde auf die geographischen Koordinaten sämtlicher österreichischer Gemeinden zurückgegriffen, die in einer GIS-Datenbank zur Verfügung standen.

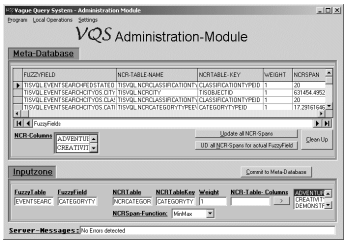
Die Semantik der Veranstaltungskategorien wurde in einem künstlichen vierdimensionalen Vektorraum dargestellt. Grundlage für die Konstruktion des Vektorraums bildete eine empirische Umfrage, welche ähnlich wie im Ansatz von [TB78] zu einer Klassifikation der Veranstaltungskategorien führte. Anschließend wurden diese Klassifikationen in einen künstlich aufgespannten Vektorraum projiziert. Die euklidische Distanz in diesem Vektorraum stellt ein direktes Maß für die subjektiv empfundene Ähnlichkeit der betreffenden Veranstaltungskategorien dar.

6.2 Integration des VQS in das Veranstaltungs-Modul von Tiscover

Um die Funktionalität von Tiscover so wenig wie möglich zu beeinflussen, wird im Fallbeispiel die Suchfunktion des VQS erst dann angestoßen, wenn eine zuvor ausgeführte konventionelle Veranstaltungssuche keine Resultate ermitteln konnte (vgl. *Empty Answer Problem*). Die notwendigen Anpassungen des Tiscover-Systems konnten daher auf sehr kleine Eingriffe in die Applikation beschränkt werden.

Auf Basis der definierten Metadaten und modifizierten Tiscover-Module können im E-Commerce-System beliebige semantikbasierte Suchverfahren angeboten werden. Abbildung 6.1 zeigt z.B. das Ergebnis einer k-Nearest-Neighbor-Suche. Obwohl eine exakte durch den Benutzer spezifizierte Veranstaltungssuche keine Ergebnisse ermitteln würde, werden dem Benutzer zumindest die *k* (z.B. *10*) semantisch besten Treffer für die Anfrage angeboten.

Durch das Fallbeispiel konnte gezeigt werden, dass beliebige E-Commerce-Anwendungen sehr einfach um Mechanismen für semantikbasierte Ähnlichkeitssuchen erweitert werden können. Auch die empirische Laufzeituntersuchung des VQS zeigte bereits eine hohe Reife für den Realeinsatz. Details zu den Laufzeituntersuchungen können in [Pal02] nachgelesen werden.



a.) Administrationsoberfläche des VQS

b.) Resultat einer semantikbasierten Veranstaltungssuche

Abbildung 6.1 Demonstration der Arbeitsweise des VQS im Tiscover-System

7 Zusammenfassung und Ausblick

Obwohl zahlreiche wissenschaftliche Arbeiten Modelle für Ähnlichkeitssuchen in Spezialanwendungen vorschlagen, existieren kaum Ansätze, die Konzepte für eine flexible Integration semantischer Metadaten in konventionelle Datenbankanwendungen zur Verfügung stellen. Vorrangiges Ziel der Arbeit war daher die Entwicklung von Verfahren, die auch in konventionellen Datenbankanwendungen flexible semantikbasierte Ähnlichkeitssuchen ermöglichen.

Da sich bereits bei den ersten Analysen zeigte, dass die meisten existierenden Modelle nicht ohne weiteres auf den Bereich konventioneller Datenbanksysteme umgelegt werden können, wurden speziell für Datenbanksysteme abgestimmte Verfahren für die Modellierung semantischer Metadaten und für das Ausführen semantikbasierter Ähnlichkeitssuchen entwickelt. Dabei waren u.a. auch effizienzsteigernde Indizierungsmethoden von hoher Relevanz. Die Evaluierung der Konzepte erfolgte durch einen Forschungsprototypen mit dem Namen VQS.

Die Konzepte des VQS wurden auf Basis eines Fallbeispiels in zweifacher Hinsicht einer Prüfung unterzogen. Erstens zeigte sich das 3-Schichten-Konzept gut geeignet, bestehende Applikationen um semantikbasierte Ähnlichkeitssuchen zu erweitern. Zweitens weist der Prototyp des VQS ein sehr effizientes Laufzeitverhalten auf. Das VQS ist somit bereits in der aktuellen Ausbaustufe durchaus für reale Anwendungsbereiche geeignet.

Aufgrund der zufriedenstellenden Resultate der Evaluierung werden die Konzepte des VQS in jedem Fall weiter ausgebaut. Darunter fallen z.B. Maßnahmen, welche die Effizienz der Suchoperationen noch weiter erhöhen und eine Definition komplexerer Suchszenarien ermöglichen. Auch die Erweiterung des VQS um zusätzliche Metriken stellt eine große Herausforderung dar.

Literaturverzeichnis

[BBK98] S. Berchtold, C. Böhm, H.-P. Kriegel, „The Pyramid-Technique: Towards Breaking the Curse of Dimensionality“, Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 1998), 1998, 142 – 153

[BGH88] P. Bosc, M. Galibourg, G. Hamon, "Fuzzy Querying with SQL: Extension and Implementation Aspects", *Fuzzy Sets and Systems*, Vol. 28, No.3, 1988, pp. 333-349

[FL95] Ch. Faloutsos, K.-I. Lin, „FastMap: A Fast Algorithm for Indexing, Data Mining and Visualization on Traditional and Multimedia Datasets“, *Proceedings of the ACM SIGMOD International Conference on Management of Data (SIGMOD 1995)*, 1995, 163 – 174

[Gaz95] D. Gazotti, L. Piancastelli, C. Sartori, D. Beneventano, "Fuzzy Base: A Fuzzy Logic Aid for Relational Database Queries", *Database and Expert Systems Applications, 6th International Conference DEXA 1995*, Ed. Norman Revell, A Min Tjoa, Springer-Verlag Berlin Heidelberg New York Barcelona Budapest Hong Kong London Milan Paris Tokyo, London 1995, pp. 385-394

[IH86] T. Ichikawa, M. Hirakawa, "ARES: A Relational Database with the Capability of Performing Flexible Interpretation of Queries", *IEEE Transactions on Software Engineering*, Vol. 12, No. 5, 1986, pp. 624-634

[Mot88] A. Motro, "VAGUE: A User Interface to Relational Databases that Permits Vague Queries", *ACM Transactions on Office Information Systems*, Vol. 6, No. 3, 1988, pp. 187-214

[Pal02] J. Palkoska, „Semantikbasierte Ähnlichkeitssuche in Datenbanksystemen“, *Dissertation, Johannes Kepler Universität Linz, Österreich*, 2002

[Pat99] M. Patella, „Similarity Search in Multimedia Databases”, *Dissertation, Dipartimento di Elettronica Informatica e Sistemistica, Università degli Studi di Bologna, Italy*, 1999

[TB78] M. Toglia, W. Battig, „Handbook of Semantic Word Norms“, *Hillsdale, N. J., Lawrence Erlbaum Associates Inc.*, ISBN 0470263784, 1978

[WL90] M.H. Wong, K.S. Leung, "A Fuzzy Database-Query Language", *Information Systems*, Vol. 15, No. 5, 1990, pp. 583-590

Werdegang

Nach seiner Ausbildung im naturwissenschaftlichen Oberstufengymnasium Perg studierte Jürgen Palkoska von 1991 bis 1997 an der Johannes Kepler Universität Linz (Österreich) Informatik. Begleitend zum Studium war er vor allem mit der Softwareentwicklung für Zivilgeometer beschäftigt.

Seit dem Abschluss seines Diplomstudiums ist er Assistent am Institut Für Anwendungsorientierte Wissensverarbeitung (FAW) der Johannes Kepler Universität Linz (Österreich). Seine Tätigkeiten reichen von der Projektleitung/Abwicklung von Forschungs- und Entwicklungsprojekten für Wirtschaftspartner bis zur universitären Lehre. Im Bereich der universitären Forschung konzentriert sich seine Arbeit auf semantikbasierte Ähnlichkeitssuchen in Informationssystemen.

Seine Dissertation, die er ebenfalls in diesem Themenbereich ansiedelte, schloss er im September 2002 ab.