Informationsspeicherung in GETESS*
oder
Die Strukturierung der Semistrukturiertheit

Meike Klettke, Andreas Heuer
Universität Rostock
Fachbereich Informatik

Zusammenfassung

In diesem Artikel werden drei Möglichkeiten zur Behandlung semistrukturierter Daten dargestellt und es wird erläutert, wie diese im Projekt GETESS eingesetzt werden.

1 Einleitung

Die Anwendungsgebiete von Datenbanken nehmen zu. Nicht nur klassische Anwendungen, z.B. auf ökonomischem oder technischen Gebiet benötigen Datenbankunterstützung. In vielen Bereichen, die nicht einheitlich zu strukturieren sind, fallen große Datenmengen an, sodaß es wünschenswert wäre, die Vorteile von Datenbanken hier zu nutzen.
Diesen Aufgaben stellen sich "semistrukturierte Datenbanken". Semistrukturierte Datenbanken sind z.B. erforderlich, wenn Daten, die die gleiche Information beschreiben, eine häufig wechselnde Struktur haben, also differierende, fehlende oder zusätzliche Attribute aufweisen. Die Merkmale semistrukturierter Daten sind u.a. in [2] zusammengefaßt. Ein klassisches Beispiel für semistrukturierte Daten sind Anwendungen im WWW.

Das Projekt GETESS (vorgestellt in [4]) hat das Ziel, eine intelligente Suchmaschine für Informationen im WWW zu entwickeln. Innerhalb des Projektes sollen für Dokumente im WWW die wesentlichen Inhalte bestimmt und in Form von sogenannten Abstracts gespeichert werden. Dabei beschreibt eine umfangreiche Ontologie[1] die Struktur der Daten in den abzuleitenden Abstracts. Bei der Abstractbildung[2] werden konkrete Werte für die modellierten Strukturen ermittelt. Diese Informationen werden im GETESS-Projekt u.a. in objektrelationalen Datenbanken gespeichert[3]. Dabei wird die Struktur der Datenbank durch die Ontologie bestimmt, die Tupel der Datenbank kommen aus den Abstracts. Abbildung 1 zeigt das Zusammenwirken dieser Komponenten anhand eines Beispiels.

Austauschformat für die Abstracts wird innerhalb des Projektes XML sein, im folgenden Abschnitt wird dieses kurz vorgestellt. In Abschnitt 3 werden drei Varianten dargestellt, um die in XML dargestellten Abstracts zu speichern. Die gespeicherten Abstracts sollen die Basis für eine intelligente Suchmaschine bilden. In Abschnitt 3 wird deshalb ebenfalls erläutert, wie auf die gespeicherten Informationen zugegriffen werden kann, um Antworten auf Suchanfragen zu ermitteln.

2 XML

XML ist ein durch das World Wide Web Consortium (W3C) entwickeltes Dokumentenaustauschformat ([3],[1]). Beschreibungen in XML sind formal und präzise und trotzdem einfach lesbar. In XML-Dokumenten sind spezielle Anweisungen (tags) enthalten, die Dokumente sind also *selbstbeschreibend*, das heißt eine Beschreibung der Struktur der Dokumente ist in den Dokumenten enthalten. Aufgrund

*German Text Exploitation and Search System, gefördert vom BMBF, Fördernummer: 01 IN B02
[1] Teilprojekt Ontologiemodellierung: Universität Karlsruhe, AIFB, AG Prof. Studer
[2] Teilprojekt Abstractgenerierung: DFKI Saarbrücken, AG Prof. Uszkoreit
[3] Teilprojekt Datenbanklösung: Universität Rostock

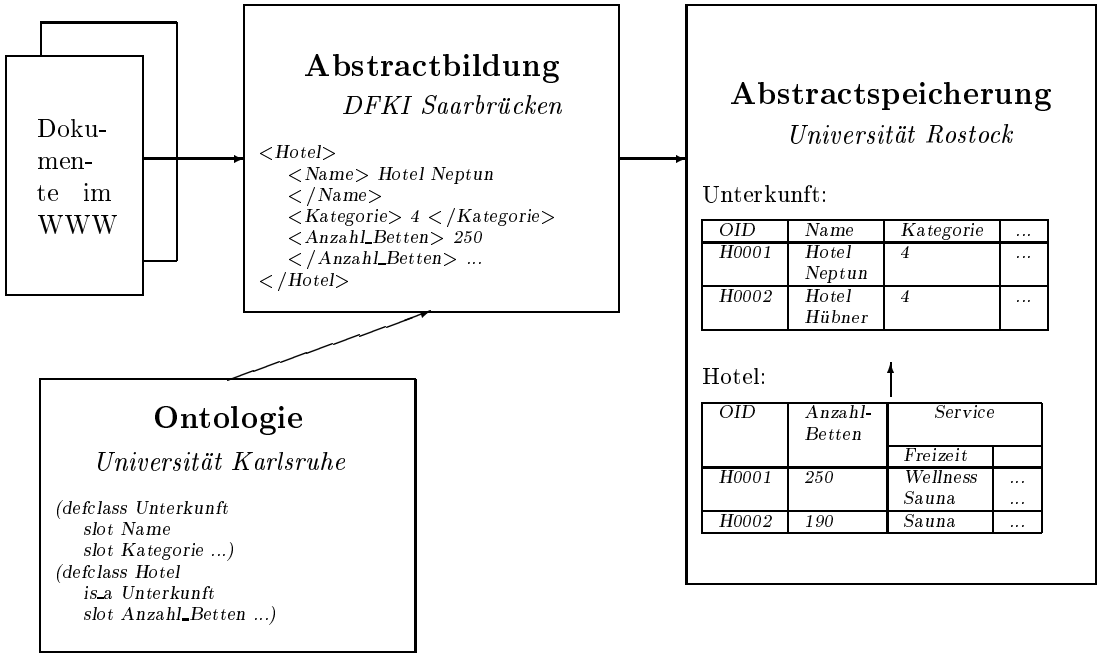

Abbildung 1: Zusammenwirken der Ontologie, Abstractbildung und Abstractspeicherung

dessen ist XML für strukturierte Dokumente und besonders auch für *semistrukturierte Anwendungen* geeignet. Die gegenwärtige Popularität von XML erklärt sich auch daraus, daß der Austausch von (meist strukturierten) Dokumenten mit XML gut realisierbar ist, da die Struktur der ausgetauschten Daten im Dokument dargestellt werden kann.
Dokumente können eine DTD haben, die die erlaubten Elemente der XML-Struktur und ihre logische Schachtelung definiert.

Im Rahmen des GETESS-Projektes wird eine Beschreibung von semistrukturierten Dokumenten (den Abstracts) zum Austausch und zur Speicherung benötigt. Für diese Zwecke ist XML eine geeignete Variante. Ein Beispiel aus der Anwendungsdomäne Tourismus zeigt das mögliche Aussehen solcher Abstracts:

```
<Hotel>
  <Name>Strand Hotel Hübner</Name>
  <Kategorie>4</Kategorie>
  <Adresse> <Ort>Warnemünde</Ort> ... </Adresse>
  <Preise> ... </Preise>
  <Service> ... </Service>
  <Umgebung> ... </Umgebung>
</Hotel>
```

Im GETESS-Projekt werden die DTD's inhaltlich durch die Ontologie bestimmt.

3 Drei Varianten zur Umsetzung von XML-Dokumenten

Ziel des GETESS-Projektes ist die Entwicklung einer Suchmaschine, die verteilt, also auf verschiedenen Servern eingesetzt werden soll. Aufgrund dessen müssen unterschiedliche Voraussetzungen berücksichtigt werden, für die Speicherung von Informationen muß dabei beachtet werden, daß nicht in jedem Fall ein Datenbanksystem vorhanden sein wird. Neben einer Lösung unter Verwendung eines Datenbanksystems muß deshalb auch eine alternative Methode zur Speicherung entwickelt werden.
Eine zweite Unterscheidung bei der Speicherung von Informationen ergibt sich durch die unterschiedliche Strukturiertheit der Abstracts. Abstracts, die meist gleich strukturiert sind, können dabei anders behandelt werden als Abstracts, die sich strukturell stark voneinander unterscheiden. Aufgrund dieser beiden Kriterien werden in Abbildung 2 drei verschiedene Methoden zur Speicherung von Abstract-Informationen dargestellt und im folgenden erläutert.

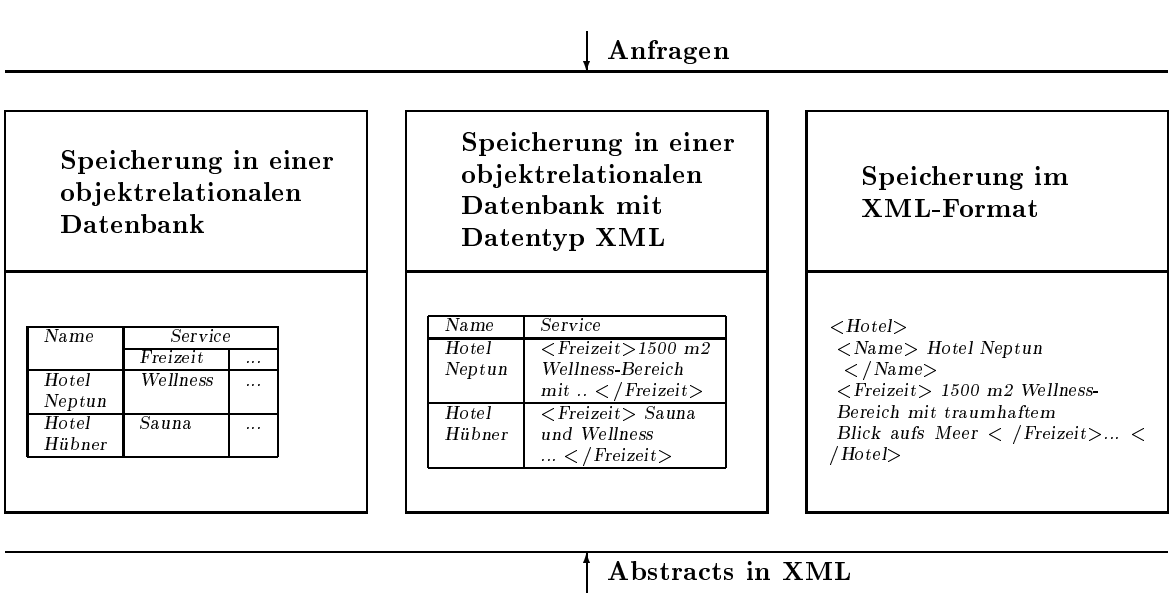

Abbildung 2: Drei Methoden zur Speicherung von Abstract-Informationen

3.1 Übersetzung von Informationen im XML-Format in eine objektrelationale Datenbank

In diesem Abschnitt wird der Versuch dargestellt, die als semistrukturierte Daten dargestellten Abstract strukturiert zu speichern. Das klingt zunächst paradox, da man damit die Besonderheiten der semistrukturierten Daten nicht berücksichtigt. Wenn häufig gleiche Strukturen in den Abstracts auftreten, bietet diese Methode jedoch alle Vorteile von Datenbankmanagementsystemen, insbesondere eine erweiterte Anfragefunktionalität.

Speicherung. Die Umsetzung einer XML-Struktur in eine objektrelationale Datenbank soll hier anhand eines Beispiels kurz erläutert werden. Dabei läßt sich eine Sequenz von Elementen auf Attribute einer Relation abbilden, hierarchische Schachtelung von Elementen lassen sich auf NF^2-Relationen abbilden und optionale Elemente sind als Attribute mit erlaubten Nullwerten darstellbar. Das Beispiel aus Abschnitt 2 würde durch folgende Relation repräsentiert werden:

Name	Kategorie	Adresse		Preise	Service	Umgebung
		Ort	...			

Anfragemöglichkeiten. Der wesentliche Vorteil der Verwendung von Datenbanken sind erweiterte Anfragemöglichkeiten. Einige davon, die für die Verwendung innerhalb einer Suchmaschine besonders relevant sind, seien hier aufgezählt:

- *Typabhängige Vergleiche.* Es sind Vergleiche über Integerwerten möglich, in der Domäne Tourismus z.B. über Preis- und Entfernungsangaben, weiterhin kann nach solchen Werte die Ergebnismenge sortiert werden.
- *Aggregatfunktionen.* Man kann Minimum-, Maximum-, Durchschnittswerte usw. ermitteln.
- *Joins.* Es können aus den Abstracts verschiedener WWW-Dokumente Antworten abgeleitet werden, auch wenn die Originaldokumente nicht durch Links verbunden sind. Die Speicherung in Datenbanken bietet die Möglichkeit, nicht nur Ergebnisse auf Suchanfragen zu finden, sondern auch Antworten auf komplexere Anfragen abzuleiten.

Diese Anfragemöglichkeiten erweitern herkömmliche Suchanfragen in erheblicher Weise.

3.2 Übersetzung von Informationen im XML-Format in objektrelationale Datenbanken mit dem Datentyp XML

Obwohl durch die Ontologiemodellierung eine Struktur für die Datenbanken zur Abstractspeicherung entworfen wird, haben die Abstractdaten die klassischen Züge von semistrukturierten Daten. Die aus dem WWW analysierten Abstracts sind - ebenso wie die Originaldokumenten - in starkem Maße unterschiedlich strukturiert. Bei einem Versuch, diese vollständig in strukturierte Datenbanken abzubilden, würden sehr große Datenbank-Schemata entstehen, in denen sehr viele Nullwerte auftauchen. In der Anwendungsdomäne mit mehreren hundert Hotelbeschreibungen gibt es verschiedene Angaben (z.B. Preisstrukturen), die nur in einer Beschreibung vorkommen [6]. Für solche Fälle erweist sich die anschließend vorgestellte zweite Variante der Speicherung als sinnvoll.

Speicherung. Häufig vorkommende Strukturelemente werden aufgelöst und als Attribute in einer Relation gespeichert (dieses erfolgt wie in Abschnitt 3.1 beschrieben), seltener vorkommende Teile werden in der XML-Struktur belassen und als Attribut mit Typ XML aufgenommen.

Dazu wurde in [5] ein Prototyp vorgeschlagen, der aufbauend auf einem Text Extender eine XML-Erweiterung für objektrelationale Datenbanken (Informix und DB2) vorschlägt und für DB2 erstellt.

Der *Vorteil* dieses Zuganges ist, daß der Text Extender von DB2 bzw. das Excalibur Text Data Blade unter Informix als Basis verwendet werden kann, um solche Funktionen wie Synonymsuche, Wortstammreduktion, Fuzzy-Suche, usw. nutzen zu können. Als *Hauptproblem* erwies sich bei dieser Lösung, daß der Index nur auf den gesamten XML-Dokumenten gebildet werden kann. Die XML-Dokumente können also nur als Volltext aufgefaßt werden, sodaß nur eine Vorselektion für Anfragen getroffen werden kann. Es ist nach der Verwendung des TextExtenders ein zweiter Durchlauf durch das Dokument notwendig, bei dem die XML-Struktur analysiert wird, um Anfragen zu beantworten.

Beispiel:

<Name>Strand Hotel Hübner</Name>
<Adresse> <Ort>Warnemünde</Ort> ... </Adresse> ...
<Umgebung>10 km nach Rostock</Umgebung>

Durch Verwendung des Text Extenders kann man zum Beispiel nicht feststellen, ob das Element *Ort=Rostock* erfüllt ist. Man kann aufgrund des Indexes feststellen, daß sowohl *Ort* als auch *Rostock* im Dokument vorkommen, da aber das Dokument nicht strukturiert indexiert wird, muß anschließend durch eine Analyse der XML-Struktur überprüft werden, in welchem Zusammenhang die gefundenen Terme stehen.

Anmerkung: Bei der Verwendung des Text Extenders von DB2 besteht weiterhin das Problem, daß kein Rückgabevektor existiert, sodaß viele Funktionen nicht verwendbar sind. Man kann dabei zum Beispiel feststellen, daß ein Synonym eines Suchbegriffes vorhanden ist, weiß aber nicht welches Synonym und ebenfalls nicht, wo es gefunden wurde.

Trotz dieser Nachteile ist diese Methode eine relativ schnell zu realisierende Möglichkeit, um einen einfachen Zugriff auf XML-Dokumente innerhalb einer Datenbanksystems zu realisieren. Eine kommerzielle Erweiterung von DB2 zur Unterstützung von XML-Datentypen ist angekündigt und wird diesen Prototyp ablösen.

Anfragemöglichkeiten. Bei einer solchen Realisierung stehen die *erweiterten Anfragemöglichkeiten* über den in der Datenbank *strukturierten Attributen*, die in Abschnitt 3.1 beschrieben wurden, ebenfalls zur Verfügung. Über den XML-Attributen müssen zusätzliche Anfragen realisiert werden [5], wie z.B.

- Suche nach Termen
- Suche nach Attributnamen
- Suche nach Elementnamen
- Vergleich Attribut-Wert
- Vergleich Element-Wert
- Suche nach verschiedenen Termen im gleichen Element
- Suche nach einem Wert in einem Element und in allen Child-Elementen
- Realisierung von Wildcards in Pfadausdrücken
- ...

3.3 Speicherung von Informationen im XML-Format

Speicherung. Wenn kein Datenbanksystem zur Verfügung steht, sollen die Informationen in der XML-Struktur die Basis für die Suchanfragen sein. Die Abstracts werden dann in dem Austauschformat belassen und gespeichert.

Anfragemöglichkeiten. Die Anfragerealisierung bei dieser Variante erfolgt analog zu der 3.2 beschriebenen Variante, sie unterscheidet sich nur dadurch, daß keine Vorselektion durch einen Text Extender möglich ist. Es müssen dort ebenfalls die oben beschriebenen Anfragen realisiert werden, können aber die gleichen Methoden zur Analyse und Auswertung der XML-Strukturen eingesetzt werden.

Diese drei Methoden zur Speicherung sollen in der Suchmaschine eingesetzt werden, sie bedingen sehr unterschiedliche Anfragemöglichkeiten. Es ist jedoch eine einheitliche Schnittstelle erforderlich. Für diese wird innerhalb des Projektes GETESS eine Sprache IRQL entwickelt ([4]) und eingesetzt, die aufwärtskompatibel zu SQL3 und IR-Anfragesprachen ist. Durch die IRQL werden Suchanfragen beschrieben und auf die verschiedenen Anfragemöglichkeiten umgesetzt.

4 Zusammenfassung und Ausblick

Die Verwendung von Datenbanken zur strukturierten Speicherung von Informationen bereichert Suchmaschinen, da dadurch qualitativ neue Suchanfragen realisierbar werden.
Abstract-Informationen aus einer eingeschränkten Anwendungsdomäne werden in GETESS aus WWW-Dokumenten abgeleitet. Diese werden auf verschiedene Weise gespeichert und stehen für komplexe Anfragen zur Verfügung. Dabei werden - soweit möglich - die Vorteile strukturierter Datenbanken für eine semistrukturierte Anwendung genutzt.

Momentan werden XML-Strukturen innerhalb von GETESS als Austauschformat für Abstracts verwendet. Wenn Originaldokumente im WWW in XML dargestellt sind - diese Entwicklung kann man nicht vorhersagen, momentan erscheint diese Annahme jedoch realistisch - kann man den hier kurz beschriebenen Ansatz auch auf Originaldokumente erweitern. Sofern die Dokumente aussagekräftige beschreibende Elemente enthalten, kann man versuchen, Teile aus Originaldokumenten aufgrund der modellierten Ontologien strukturiert zu speichern und so - ebenso wie bei Abstracts - eine Speicherung und Auswertung von strukturierten und in XML belassenen Strukturen zu kombinieren.

Literatur

[1] Extensible Markup Language. http://www.heise.de/ix/raven/Web/xml/.

[2] Serge Abiteboul. Querying Semi-Structured Data. In Foto N. Afrati and Phokion Kolaitis, editors, *Database Theory - ICDT '97, 6th International Conference*, volume 1186 of *Lecture Notes in Computer Science*, pages 1–18, Delphi, Greece, January 1997. Springer Verlag.

[3] Neil Bradley. *The XML companion.* Addison Wesley, 1998.

[4] Antje Düsterhöft, Andreas Heuer, Meike Klettke, and Denny Priebe. Getess: Der Text-orientierte Anfrage- und Suchdienst im Internet. In *11. Workshop Grundlagen von Datenbanken*, Luisenthal, May 1999.

[5] Beate Porst. Untersuchungen zu Datentyperweiterungen für XML-Dokumente und ihre Anfragemethoden am Beispiel von DB2 und Informix. Master's thesis, Universität Rostock, 1999.

[6] Melanie Siegel. Hotel-Template. Internes Papier, October 1998.