Ein hybrider Ansatz zur flexiblen Speicherung von XML-Daten in RDBMS am Beispiel von PubMed

Michael Fiedler
fiedler@informatik.uni-leipzig.de

Betreuer der Diplomarbeit: Dr. Andreas Thor, Prof. Dr. Erhard Rahm
Abteilung Datenbanken, Institut für Informatik, Universität Leipzig
{thor, rahm}@informatik.uni-leipzig.de

Abstract: In diesem Beitrag wird ein hybrider Ansatz zur flexiblen Speicherung von XML-Daten in relationalen Datenbanken vorgestellt. In Abhängigkeit von der jeweiligen Häufigkeit des Auftretens der einzelnen XML-Elemente werden die Daten dabei im domänenspezifischen oder generischen Schema abgelegt. Dadurch kann die Notwendigkeit einer sofortigen Anpassung des relationalen Schemas bei inkrementellen Veränderungen des XML-Schemas vermieden werden.

1 Einleitung

Die U.S. National Libray of Medicine (NLM) veröffentlicht bibliografische Daten zu ca. 18 Millionen Artikeln im Bereich der Lebenswissenschaften (Stand: Dezember 2008). Über das Web werden diese Daten durch die Suchmaschine PubMed [http://www.ncbi.nlm.nih.gov/pubmed/] zugänglich gemacht. Der komplette Datenbestand (ca. 60 GB, ca. 16 Mio. Publikationen, Stand: September 2007) kann zusätzlich im XML-Format heruntergeladen werden und wird regelmäßig (mindestens einmal pro Jahr) aktualisiert. Die Daten können im Rahmen von Forschungsarbeiten verwendet werden, z.B. für Data Mining, Data Cleaning oder Zitierungsanalysen.

Eine vorherige Speicherung der XML-Daten in einer relationalen Datenbank ist dabei in vielen Anwendungsfällen vorteilhaft, da z.B. viele Data-Cleaning-Algorithmen auf relationale Daten ausgelegt sind. Die Speicherung von XML-Daten in relationalen Datenbanksystemen ist ein viel beachtetes Forschungsgebiet (siehe z.B. [FK99] und zitierende Arbeiten). Ziel ist es, die XML-Daten verlustfrei in ein relationales Modell zu überführen, so dass auf sie anschließend effizient zugegriffen werden kann (z.B. mittels SQL-Anfragen). Es lassen sich dabei zwei Arten unterscheiden. Bei der domänenspezifischen Speicherung wird ein relationales Modell erstellt, welches in seinem Aufbau (Relationen, Attribute) der Domäne angepasst ist. Die generische Speicherung verwendet - unabhängig von den XML-Daten - ein festes Schema, mit welchem die XML-Daten gespeichert werden können. So kann z.B. in einer Relation die Vater-Kind-Beziehung der XML-Elemente abgelegt werden.

Die in dieser Arbeit betrachteten PubMed-Daten werden regelmäßig von der NLM aktualisiert. Dabei werden nicht nur neue Publikationen (d.h. Datensätze) aufgenommen, sondern ggf. auch neue XML-Elemente eingeführt − dies jedoch i. Allg. nur für einen Teil der Publikationen. So ist z.B. das *Abstract* nur für ca. 52% der Publikationen erfasst. Allgemein variiert die Anzahl der XML-Elemente pro erfasster Publikation zwischen 48 und 148. Somit führt die Speicherung mit domänenspezifischem Schema zu einem sehr großen (und ggf. unübersichtlichen) Schema, wobei eine Vielzahl von Attributwerten mit NULL belegt sind. Gleichzeitig muss das relationale Schema bei jeder Aktualisierung ggf. angepasst werden, was wiederum Anpassungen in den zugreifenden Anwendungen erfordern kann (Schemaevolution; siehe z.B. [RB05] für eine Übersicht). Auf der anderen Seite führt die generische Speicherung bei der Verwendung zu unübersichtlichen Anfragen,

welche auf den (wenigen) generischen Relationen eine Vielzahl von JOIN-Operationen ausführen. Insbesondere vor dem Hintergrund der Größe der PubMed-Daten (ca. 60 GB) können dabei lange Ausführungszeiten entstehen.

In dieser Arbeit wird daher ein hybrider Ansatz zur Speicherung der XML-Daten vorgestellt. Die Daten der einzelnen XML-Elemente werden entsprechend der Häufigkeit ihres Auftretens im domänenspezifischen oder generischen Schema abgelegt. Daraus resultieren die folgenden Eigenschaften:

- Die meisten Daten (sozusagen der "Kern") stehen im kompakten, domänenspezifischen Schema zu Verfügung, so dass viele Anfragen bzgl. diesem Schema gestellt werden können.
- Weniger häufig auftretende Informationen werden im generischen Schema gespeichert, so dass sie bei Bedarf durch entsprechende Anfragen mit dem domänenspezifischen Schema kombiniert werden können. Im Gegensatz zu einer komplett generischen Speicherung enthalten die Relationen wesentlich weniger Datensätze.
- Inkrementelle Veränderungen der XML-Daten (z.B. das Hinzufügen eines neuen XML-Elements) werden zunächst durch das generische Schema "abgefedert". Erst wenn ein neues Element z.B. signifikant oft auftritt, wird es in das domänenspezifische Schema übernommen. Dadurch werden XML-Schemaänderungen vom Nutzer kontrolliert und führen erst nach Bedarf zu einer Änderung des relationalen Schemas. Damit müssen abhängige Anwendungen seltener angepasst werden.

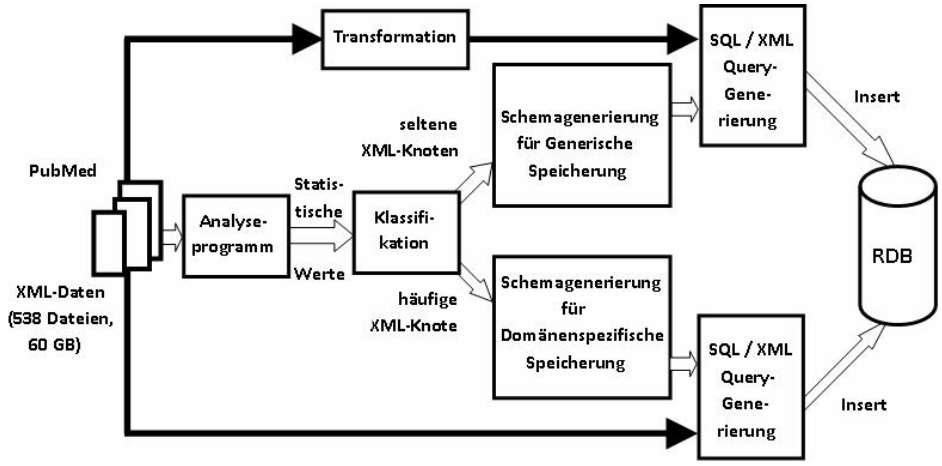

Abbildung 1. Gesamtarchitektur der hybriden XML-Speicherung

2 Architektur

Abbildung 1 zeigt die Gesamtarchitektur des Ansatzes. Die Eingabe ist eine Menge von XML-Dokumenten, deren Inhalt abschließend in einer relationalen Datenbank gespeichert wird.

Der hybride Speicheransatz erfordert vorab eine statistische Analyse der XML-Daten, d.h. welche XML-Elemente wie oft vorkommen. Dies wird mit einem Analyseprogramm realisiert, dass für alle XML-Dokumente die Häufigkeit des Auftretens der XML-Elemente ermittelt. Das Analyseprogramm stellt für alle XML-Elemente u.a. folgende Werte bereit: Name inkl. Pfad zum Wurzelelement der Publikation, Tiefe in der XML-Hierarchie, minimales / maximales Auftreten im Datensatz, absolute Anzahl in allen XML-Dokumenten, relative Häufigkeit des Auftretens innerhalb einer Publikation

und relative Häufigkeit der Publikationen mit mindestens einer Instanz des XML-Elements. Tabelle 1 zeigt einen Ausschnitt des Analyseergebnisses für die Elemente *PMID* (interne PubMed-ID der Publikation), *ArticleTitle*, *Year* und *Author*.

Die ermittelten statistischen Werte dienen anschließend zur quantitativen Bestimmung der "Relevanz" für die jeweiligen XML-Elemente und damit zur Entscheidung, in welchem Schema (domänenspezifisch oder generisch) die zugehörigen Daten gespeichert werden. Diese Klassifikation kann u.a. einfach durch nutzerdefinierte Schwellwerte realisiert werden, z.B. um alle Elemente, die in mindestens 50% der Publikationen vorkommen, im domänenspezifischen Schema und die übrigen im generischen abzubilden. Zusätzlich können Elemente vom Nutzer direkt (d.h. unabhängig von den statistischen Werten) einem der beiden Schema zugeordnet werden, wenn dieses z.B. in späteren Anwendungen von besonderem Interesse ist.

XML-Element (XPath-Ausdruck)	Tiefe	Auftreten pro Publikation		Anteil der Publikationen mit Element-Vorkommen
		Min.	Max.	
/PMID/	2	1	1	100%
/Article/ArticleTitle/	3	1	1	100%
/Article/AuthorList/Author/	4	0	744	97,8%
/Article/Journal/JournalIssue/PubDate/Year/	6	0	1	91,1%

Tabelle 1. Auszug des XML-Analyseergebnisses der PubMed-Daten

Im nächsten Schritt werden die beiden Schemata erzeugt. Das generische Schema ist fest vorgegeben und wird unabhängig von der vorherigen Klassifikation erstellt. Demgegenüber orientiert sich das domänenspezifische Schema an den zugeordneten XML-Elementen. Auf Grund der vorherigen statistischen Analyse können XML-Elemente, die maximal einmal pro Vater-Element auftreten, erkannt und somit als einfache Attribute abgebildet werden. Häufiger vorkommende Elemente werden in einer gesonderten Relation abgebildet, welche die 1:N-Beziehung zwischen Vater- und Kind-Element (inklusive Position des Kindelementes) repräsentiert. Die Benennung der Relationen und Attribute orientiert sich an den Namen der XML-Elemente. Sie kann zusätzlich im Rahmen der vorher benannten Zuordnung der Elemente zum Schematyp auch vom Nutzer verändert werden.

Der Import der XML-Daten setzt auf den im SQL:2003-Standard definierten XML-Erweiterungen [SQL] relationaler Datenbanksysteme auf. Dazu werden die XML-Dokumente zunächst "pur" in der Datenbank unter Verwendung des SQL-Datentyps XML gespeichert. Anschließend werden entsprechende SQL/XML-Anweisungen generiert und ausgeführt, welche die XML-Daten in die

```
INSERT INTO PubMedPublication (PMID, TITLE, YEAR)
SELECT X.PMID, X.TITLE, X.YEAR
FROM   PUREXMLTABLE AS XT,
       XMLTABLE ('$XTD//MedlineCitation'
         PASSING XT.XMLDOK AS XTD
      COLUMNS
        "PMID"  INTEGER PATH './PMID',
        "TITLE" VARCHAR(500) PATH './Article/ArticleTitle',
        "YEAR" INTEGER PATH
              './Article/Journal/JournalIssue/PubDate/Year'
       ) AS X
```

Abbildung 2. Beispiel für eine SQL/XML-Anweisung zur Befüllung der Tabellen

einzelnen Relationen einfügen. Abbildung 2 zeigt den grundsätzlichen Aufbau an Hand eines einfachen Beispiels, welches die ID, das Publikationsjahr sowie den Titel einer Publikation in die Tabelle PubMedPublication einspielt. Durch die Verwendung der XMLTABLE-Funktion, welche in der bei der Implementation verwendeten DB2v9.5 zur Verfügung steht, gelingt eine elegante Konvertierung der XML-Daten in eine relationale Struktur. Die SQL/XML-Anweisungen können dabei automatisch generiert werden, da die zu speichernden XML-Elemente und ihre XPath-Audrücke aus den vorherigen Schritten (Analyse und Klassifikation) bekannt sind. Die Speicherung der Daten im generischen Schema verläuft analog. Um die Anzahl der SQL/XML-Anweisungen zu reduzieren, werden die XML-Dokumente dazu zuvor transformiert (d.h. annotiert), so dass Elementname und Elementinhalt mit fest definierten XPath-Ausdrücken erreichbar sind.

3 Zusammenfassung

In diesem Beitrag wurde ein generischer, hybrider Ansatz zur Speicherung von XML-Daten in eine relationale Datenbank am Beispiel von PubMed vorgestellt. Wichtiges Kennzeichen der Arbeit ist, dass die Daten der XML-Elemente entsprechend der Häufigkeit ihres Vorkommens entweder im domänenspezifischen oder generischen Schema abgelegt werden. Damit können die meisten Daten im kompakten und übersichtlichen domänenspezifischen Schema gespeichert werden, wohingegen selten auftretende Informationen generisch abgelegt werden. Dies ermöglicht u.a. eine kontrollierte Schemaevolution, da inkrementelle Erweiterungen im XML-Schema zunächst im generischen relationalen Schema abgefedert werden können. Erst zu einem späteren Zeitpunkt, z.B. wenn die neu hinzugekommenen Elemente signifikant oft auftreten, ist eine Änderung im domänenspezifischen Schema nötig.

Literaturverzeichnis

[FK99] Florescu, D.; Kossmann, D.: Storing and Querying XML Data using an RDBMS. In IEEE Data Engineering Bulletin 22(3), 1999

[RB06] Rahm, E.; Bernstein, P.A.: An Online Bibliography on Schema Evolution. In Sigmod Record 35(4), 2006

[SQL] ISO/IEC 9075-14:2003 Information Technology - Database Languages - SQL - Part 14: XML-Related Specifications