Schema Integration und XQuery Core in Digitalen Bibliotheken

Ammar Balouch * Andreas Heuer † Holger Meyer ‡
Lehrstuhl Datenbank- und Informationssysteme
18059 Rostock

1 Einleitung

In der Vergangenheit wurde ein Vielzahl von digitalen Bibliotheken und elektronischen Katalogen durch Verlag, Bibliotheken und andere Einrichtungen bereitgestellt, um ublikationen bzw. die zugehörigen bibliographischen Metadaten zu Publikationen (zum Beispiel Autoren, Titel, Verlag) einer großen Anzahl von Benutzern über ntsprechende Dienste zur Recherche zur Verfügung zu stellen. Um die Recherche nach bibliographischen Metadaten zu Publikationen und inhaltsbasiert in den Dokument elbst zu erleichtern, ist eine Integration von vorhandenen digitalen Bibliotheken mit Ziel notwendig, einheitliche Zugriffschnittstellen auf die gespeicherten Informationen bereitzustellen. Zur Integration digitaler Bibliotheken werden derzeit bekannte Lösungen aus dem Bereich föderierter und mediatorbasierter Informationssysteme eingesetzt. Wir beschreiben in diesem Vortrag unsere Herangehensweise am Beispiel der Integration, die auf einer Sammlung von XML-Daten basiert. Dabei gehen wir insbesondere auf Probleme bei der Integration ein, sowie die notwendigen Transformationen und die Berücksichtigung der unterschiedlichen Anfragemöglichkeiten der Anfragesprache XQuery und speziell XQuery Core.

2 XML und Datenbanken

XML-Anwendungen sind vielfältig und reichen vom Zwischenformat zur Datenrepräsentation oder zum Datenaustausch bis hin zur Markup-Sprache für Volltextdokumente zur Speicherung und Anfrage von großen strukturierten Datenbeständen. Datenbanken werden eingesetzt, um große Datenmengen sicher zu speichern und effizient anfragen zu können. Sie dienen in erster Linie der Speicherung von strukturierten Daten. XML ermöglicht neben der Darstellung strukturierter Daten auch die Repräsentation von semistrukturierten Daten, also Daten, den Struktur unregelmäßig ist, wechseln kann oder gar nicht explizit vorhanden ist. Auch Dokumente, die Volltextinformationen beinhalten, können mit XML-Syntax dargestellt werden.Der Sprachvorschlag XQuery basiert auf XPath und ist durch Sprachen wie SQL, Lorel, XML-QL und YATL beeinflusst worden. XPath selbst wurde wiederum durch Anfragesprachen wie XQL mit geformt [FMM$^+$02, BCF$^+$02].XQuery Core [DFF$^+$02] ist eine Untermenge von XQuery, die zur formalen Definition der Semantik von XQuery herangezogen wird. Darüber hinaus ist sie gut geeignet als Sprache für Anfragetransformationen und die Basis einer logischen Optimierung zu bilden. Wir setzen hier XQuery Core ebenfalls als Sprache zur Schemaintegration ein und zeigen, dass sie gut geeignet ist, viele Schemaintegrationsprobleme zu lösen.

3 Schema Integration

Die Integration heterogener Datenbestände stellt nach wie vor ein aktuelles Problem dar. So existieren einerseits die über Jahre gewachsenen Bestände in den Unternehmen und die Vielzahl von öffentlich

*E-mail: Ammar.Balouch@informatik.uni-rostock.de
†E-mail: Andreas.Heuer@informatik.uni-rostock.de
‡E-mail: Holger.Meyer@informatik.uni-rostock.de

zugänglichen Quellen wie z.B. im Internet. Andererseits führt gerade die damit verbundene Informationsflut zum Wunsch nach einer Integration und Verdichtung dieser Daten [PLM+02]. In den vergangenen

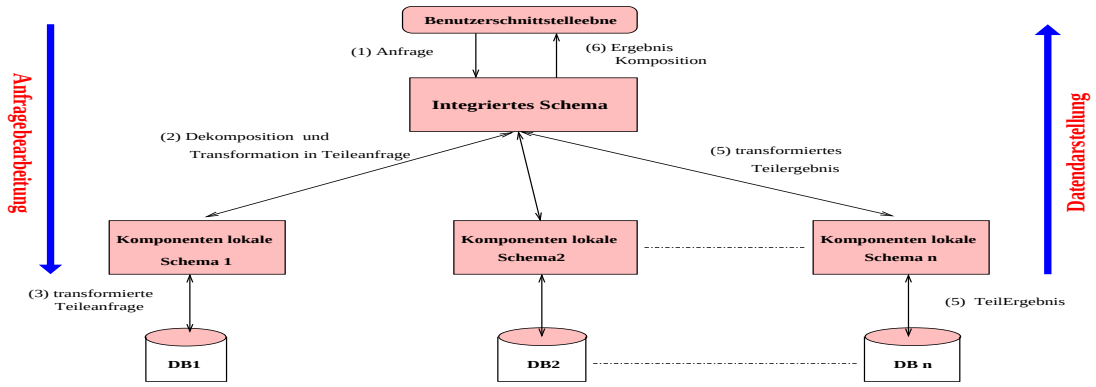

Abbildung 1: Schema Architektur

Jahren wurde eine Vielzahl von Integrationsansätze entwickelt. Stellvertretend seinen hier, Mediatoren [Mil99] und föderierte Datenbanksysteme genannt [EHS+00]. Alle diese Ansätze basieren mehr oder weniger auf der Idee einer integrierten Sicht auf die verschiedenen Datenbestände, wobei diese Sichtweise für föderierte Datenbanken nach der Sheth-Larson-Architektur besonders ausgeprägt ist. Eine abgewandelte Architektur, wie sie hier verwendet wird, ist in Abbildung 1 dargestellt.

Die Ableitung dieser integrierten Sicht ist Gegenstand der Schemaintegration. Im Verlauf des Integrationsprozesses sind die Schemata der einzelnen Quellen zu analysieren, ein globales Schema zu definieren, und die Abbildung zwischen den lokalen Schemata und dem globalen Schemata festzulegen. Mit Hilfe dieser Abbildungsinformationen können Anfrage auf globaler Ebene in Teilanfragen für die einzelnen Datenquellen zerlegt und übersetzt sowie die ermittelten Teilergebnisse auf globaler Ebene wieder zusammengesetzt werden. Der Ablauf der Integration am Beispiel der im weiteren beschriebenen lokalen Schemata ist in Abbildung 2 schematisch dargestellt.

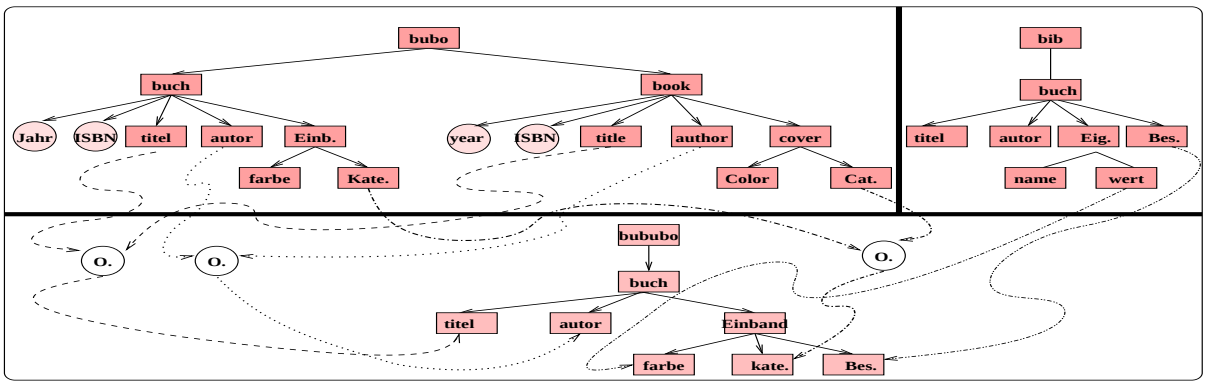

Abbildung 2: Ablauf der Integration

Es gibt eine Vielzahl von Projekten [ASD+91, GMPQ+, CGMJ+94], die sich mit dem Thema Schemaintegration speziell auch auf semistrukturierten Daten beschäftigt haben. Keines der Projekte verwendet, so wie wir es hier tun, die Möglichkeiten der Sprache XQuery bzw. XQuery Core zur Integration. Wir benutzen hier XQuery gleichzeitig als Anfragesprache und mit XQuery Core ein Subset zur Beschreibung der Abbildungsvorgänge und zur Konfliktauflösung.

3.1 Konfliktbehandlung

Ein Hauptprobleme bei der Integration heterogener Datenbestände bilden Konflikte, die durch unterschiedliche Datenmodelle oder auch durch verschiedene Repräsentation der Real-Welt-Objekte entstehen. Die Konflikte müssen im Rahmen der Integration erkannt und bei der Definition der Abbildung zwischen globalen und lokalen Schemata aufgelöst werden. Diese Abbildung ist die Grundlage für die Bearbeitung von Anfragen. Eine einfache und für die Praxis gut verwendbare Klassifikation von Konflikten unterscheidet vier Arten.Im folgenden erläutern wir diese vier Konfliktarten und geben typische, häufig auftretende Beispiele an.

Semantische Konflikte: liegen vor, wenn zwei unabhängig voneinander entstandene Schemata den gleichen Weltausschnitt beschreiben. Für die Anwendungen, die auf dem einen Komponentensystem laufen, ist vielleicht nur eine Teilmenge der Objekte relevant, die in einem anderen Komponentensystem verwaltet werden. Dieses Problem tritt häufig in Digitalen Bibliotheken auf, lokale Bibliotheken können sowohl gleiche wie auch völlig verschiedene Bestände enthalten.

Beschreibungskonflikte: Wenn in den zu integrierenden Schemata Objektmengen gefunden wurden, die gleiche Objekte der realen Welt repräsentieren, können sie sich doch in der Beschreibung der Eigenschaften dieser Objekte unterscheiden. Auch die Problematik *homonymer* und *synonymer* Bezeichnungen gehört zu den Beschreibungskonflikten. Dies kann sich auf Namen von Eigenschaften (Attribute) genauso beziehen wie auf Objektbezeichnungen und Namen von Beziehungen. Unterschiedliche Formate für die beschreibenden (Meta-)Daten sind typisch für Digitale Bibliothekssysteme. Dies gilt nicht nur auf Schema- sondern auch auf Werteebene, etwa bei verschieden verwendeten Klassifikationen etwa.

Heterognitätskonflikte: Bei der Integration enstehen viele Probleme, durch die Verwendung unterschiedlicher Datenmodelle für die zu integrierenden Schemata. Unterschiedliche Datenmodelle bedingen, daß unterschiedlich viel Modellierungskonzepte zur Verfügung stehen. Diese Probleme müssen bei der Schematransformation in das globale Datenmodell beachtet werden. Die Heterogenität der Datenmodelle bedingt in der Regel auch strukturelle Konflikte, die durch die Schematransformation nicht beseitigt werden könne. Lokale Bibliothekssysteme benutzen verschiedenste Modelle, wir versuchen über XML sie einheitlich zu repräsentieren.

Strukturelle Konflikte: liegen vor, wenn in den lokalen Systemen unterschiedliche Modellierungskonzepte für ein und den selben Sachverhalt benutzt werden. So kann eine Eigenschaft einmal als Attribut und in anderen Fällen als Subelement eines anderen Elementes modelliert werden. Eine besondere Art struktureller Konflikte sind *Meta-Konflikte*, bei denen Werte von Eigenschaften in einem Schema als konkrete Werte gespeichert werden, während dieser Wert in einem anderen Schema als Information auf Schemaebene enthalten ist.

Konkrete Konfikte lassen sich nicht immer genau einer dieser vier Klassen zuordnen. Dies liegt darin begründet, daß ein Konflikt durchaus mehrere verschiedene Ursachen und damit verschiedene Erscheinungsformen haben kann. Wir zeigen im folgenden ein Beispiel, das die erwähnten Konflikte enthält und deuten an, wie man diese Konflikte durch XQuery/XQuery Core auflösen kann.

Beispiel Wir setzen jetzt voraus, daß wir zwei XML-Dokumente (Beispiel 1, Beispiel 2) haben. **Beispiel 1** beschreibt Bücher entweder in Deutsch oder in English. Dieses Dokument sieht wie folgt aus:

```
<bubo>
    <buch jahr="2003" isbn="13-89864-148-1"><titel>XML und Datenbanken</titel>
        <autor>Meike Klettke</autor><autor>Holger Meyer</autor>
        <einband><farbe>blau</farbe><kategorie>sehr gut</kategorie></einband>
    </buch>
    <book year="2002" isbn="3-89721-269-X"><title>Java & XML</title>
        <author>Brett McLaughlin</author>
        <cover><color>white</color><category>very good</category></cover>
    </book>
</bubo>
```

Beispiel 2 beschreibt Bücher nur in Deutsch. Es enthält zum Teil gleiche Bücher wie Beispiel 1. Es sieht folgendermassen aus:

```
<bib>
    <buch><titel>XML und Datenbanken</titel>
        <autor>Meike Klettke</autor> <autor>Holger Meyer</autor>
        <eigenschaften><name>Farbe</name><wert>Blau</wert></eigenschaften>
        <beschreibung>Lehrbuch an der Uni Rostock</beschreibung>
    </buch>
    <buch><titel>Java & XML</titel>
        <autor>Brett Mclaughlin</autor>
        <eigenschaften><name>Farbe</name><wert>Weis</wert></eigenschaften>
        <beschreibung>Buch zu Behandlung</beschreibung>
    </buch>
</bib>
```

Wir haben sowohl Namenskonflikte (englisch, deutsche Bezeichungen) als auch Heterogenitätskonfligte bei der Darstellung von Eigenschaften (hier Farbe), einmal als Tag-Name, ein ander mal als Wert eines Elementes.

3.2 Konfliktbehandlung und XQuery Core

Im folgenden stellen wir die XQuery Core Anfrage dar, mit der wir die Integration beider Schemata vornehmen. Die Konfliktauflösung erfolgt auf Anfrageebene u.a. durch die Möglichkeit in XQuery Konstruktoren einzusetzen, um aus Werten Schemainformation (hier Elementnamen) und umgekehrt zu erzeugen.

```
FOR $a IN children($bubo0/bubo/buch) RETURN
FOR $b IN children($bubo0/bubo/book) RETURN
FOR $c IN children($bib0/bib/buch) RETURN
LET $ta := string-value($a/titel) RETURN
LET $tb := string-value($b/title) RETURN
LET $tc := string-value($c/titel) RETURN
IF eq($tc, $ta) THEN <bibubo>
    <buch><titel>$a/titel</titel>, <autor>$a/autor</autor>,
        <einband><farbe>$c/eigenschaften/wert</farbe>
            <kategorie>$a/einband/kategorie</kategorie>,
            <beschreibung>$c/beschreibung</beschreibung>
        </einband>
    </buch>
</bibubo>
ELSE IF eq($tc,$tb) THEN <bibubo>
    <buch><titel>$b/title</titel>, <autor>$b/author</autor>,
         <einband><Farbe>$c/eigenschaften/wert</farbe>
              <kategorie>$a/einband/kategorie</kategorie>,
              <beschreibung>$c/beschreibung</beschreibung>
         </einband>
    </buch>
</bibubo> ELSE ( )
```

Wir benutzen XQuery Formal Semantics, um das Schema des integrierten Ergebnis klar zu beschreiben:

```
TYPE Bibubo = ELEMENT bibubo (Buch+)
TYPE Buch = ELEMENT buch (
     ELEMENT titel (xs:string), ELEMENT autor (xs:dtring),
     ELEMENT einband (
          ELEMENT farbe (xs:string),  ELEMENT kategorie (xs:string),
          ELEMENT beschreibung (xs:string)))
```

Weitere Konflikte können durch die Verwendung nutzerdefinierte Funktion oder von Standardfunktionen gelöst werden.

4 Zusammenfassung und Ausblick

Generell besitzt sowohl das XQuery-Datenmodell als auch die Sprache XQuery Core gute Vorraussetzungen um als Integrations-Middleware eingesetzt zu werden. Wir haben an Beispielen aus dem Umfeld Digitaler Bibliotheken gezeigt, wie XQuery Core zur Schemaintegration eingesetzt werden kann. Auch wenn nicht alle Quellen als Kollektion von XML-Dokumenten vorliegen, so lassen sie sich jedoch als XML-Dokumente repräsentieren oder exportieren. Gerade die Konzepte semistrukturierte Datenmodelle wie optionale Elemente, Mixed Content Type oder Alternativen als auch Möglichkeiten zur Erzeugung neuer Strukturen und Schemata über Element- und Attribut-Konstruktoren in der Sprache XQuery selbst, erlauben den Einsatz von XQuery und XQuery Core nicht nur zur verteilten Anfragebearbeitung sondern auch als Sprache zur Schemaintegration auf der Ebene der Föderation verteilter Bibliothekssysteme.

Weitere Arbeiten sollen sich auf die Optimierung der globalen Anfragebearbeitung auf Grundlage von Query rewriting-Techniken, einer speziellen XQuery-Algeba und kostenbasierter Auswahl konzentrieren.

Literatur

[ASD+91] Rafi Ahmed, Philippe De Smedt, Weimin Du, William Kent, Mohammad A. Ketabchi, Witold A. Litwin, Abbas Rafii, and Ming-Chien Shan. The Pegasus Heterogeneous Multidatabase System. Technical report, Hewlett-Packard Laboratories December, 1991.

[BCF+02] Scott Boag, Don Chamberlin, Mary F. Fernandez, Daniela Florescu, Jonathan Robie, and Jerome Simeon. XQuery 1.0: An XML Query Language . Technical report, W3C Working Draft 15 November, 2002.

[CGMJ+94] Sudarshan Chawathe, Hector Garcia-Molina, Joachim Jammer, Kelly Ireland, Yannis Papakonstantinou, Jeffrey Ullman, and Jennifer Widom. The TSIMMIS Project: Integeration of Heterogeneous Information Sources. Technical report, IPSJ Stanford, CA 94305-2140, 1994.

[DFF+02] Denise Draper, Peter Fankhauser, Mary Femandez, Ashok Malhotra, Kristoffer Rose, Micheal Rys, Jerome Simeon, and Philip Wadler. XQuery 1.0 and XPath 2.0 Formal Semantics . *W3C Working Draft*, November 2002.

[EHS+00] Martin Endig, Michael Höding, Gunter Saake, Kai-Uwe Sattler, and Eike Schallehn. Federation Services for Heterogeneous Didital Libraries Acessing Cooerative and Non Cooperative Sources . *Kyoto International Conference on Digital Libraries 2000*, pages 314–321, 2000.

[FMM+02] Mary Fernandez, Ashok Malhotra, Jonathan Marsh, Marton Nagy, and Norman Walsh. XQuery 1.0 and XPath 2.0 Data Model . Technical report, W3C, Working Draft 15 November, 2002.

[GMPQ+] Hector Garcia-Molina, Yannis Papakonstantinou, Dallan Quass, Anand Rajaraman, Yehoshua Sagiv, Jeffrey Ullman, Vasilis Vassalos, and Jennifer Widom. The TSIMMIS Approach to Mediation: Data Models and Languages.

[Mil99] Michael Milk. Schemaintegration und Anfragetransformation für heterogene Digitale Bibliotheken . Technical report, Mai - Oktober , Dortmund, Germany, 1999.

[PLM+02] Kalpdrum Passi, Louise Lane, Sanjy Madria, Bipin C. Sakamuri, Mukesch Mohania, and Sourav Bhowmick. A Model for -XML Schema Integration. Technical report, EC-Web2002: P3E2C6 Canda, MO65401 USA, 110016 India, Singapore, 2002.