

Architektur und Realisierung eines persistenten XML-Speichers

Birger Hänsel

Universität Rostock

Fachbereich Informatik

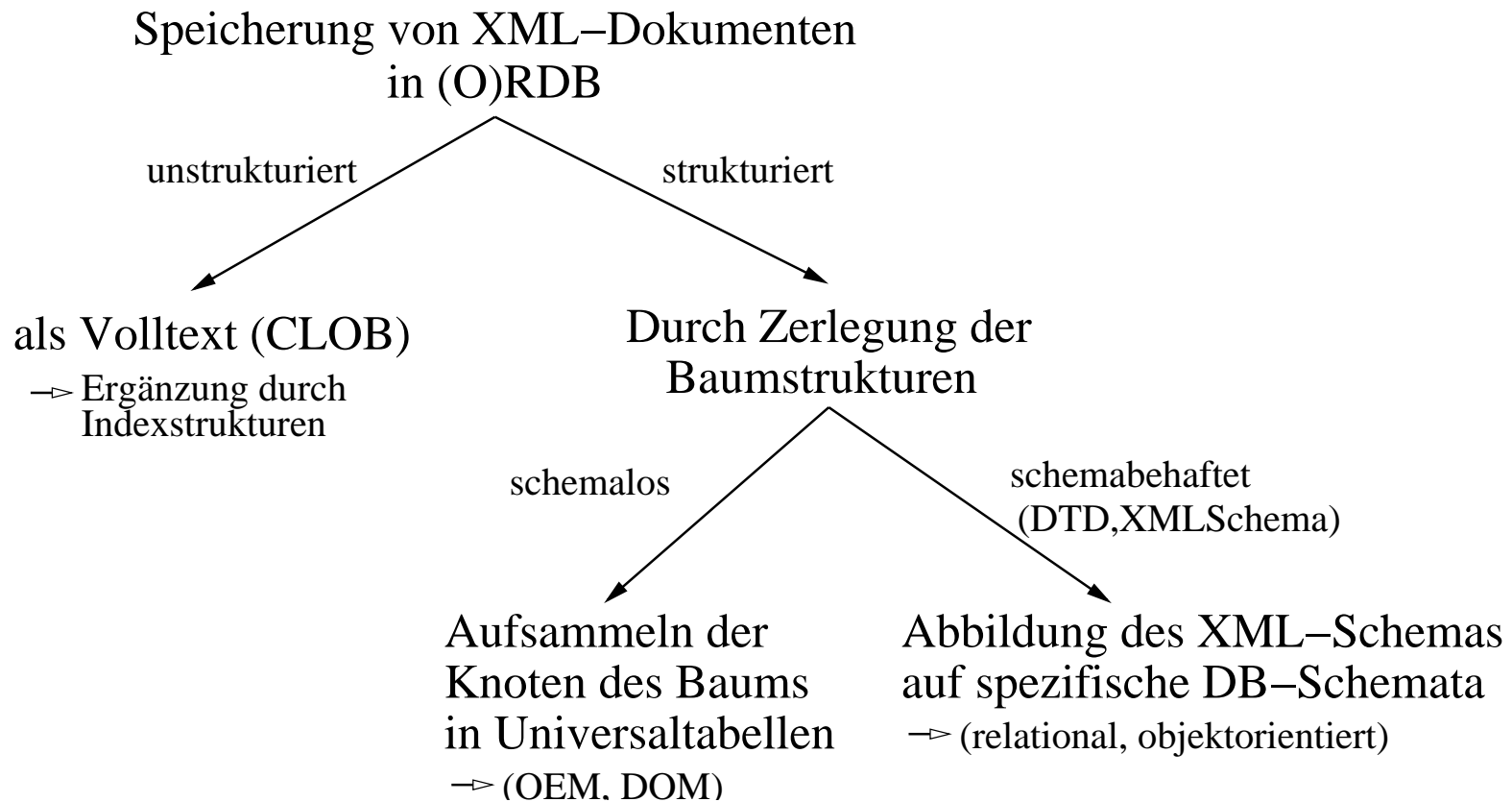
`Birger.Haensel@informatik.uni-rostock.de`

MyCoRe Workshop Bonn, 9.-12. Dezember 2002

Übersicht

1. XML-Speicherungstechniken
2. XQuery-Umsetzung
3. Konzeption einer XML-Update-Sprache
4. Zusammenfassung und Ausblick
5. Fragen und Diskussion

XML-Speicherungstechniken

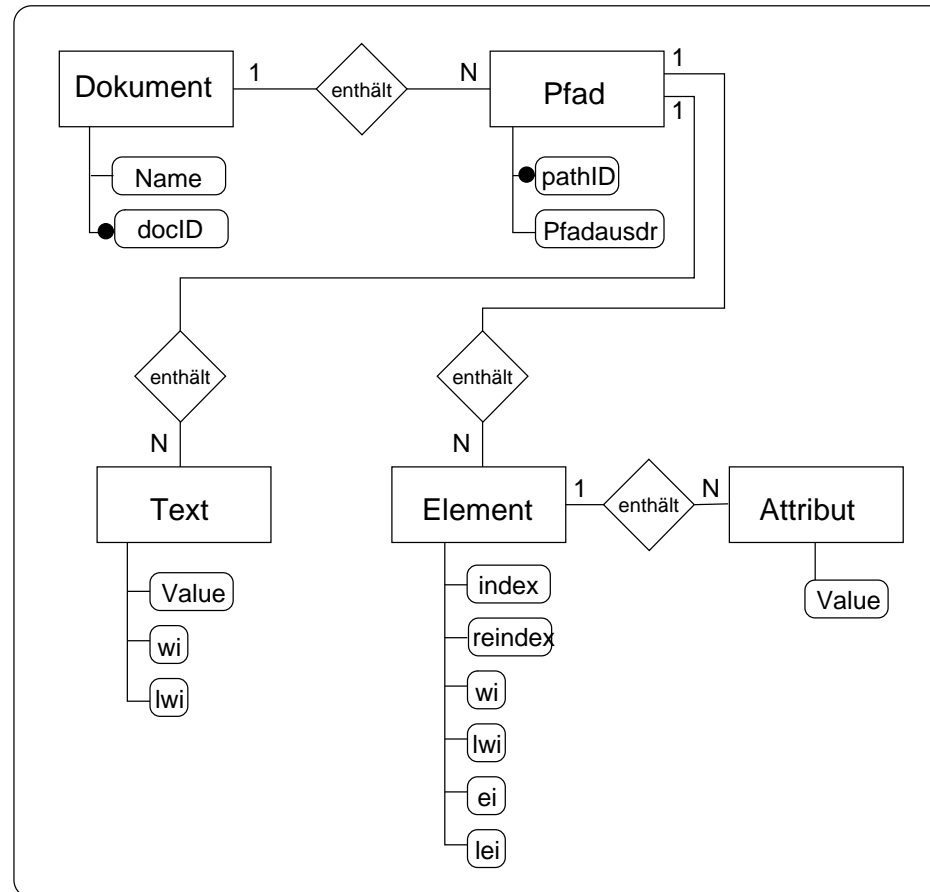


Generisches Speicherungsmodell

- Speicherung von Baumstrukturen in ein fest definiertes Datenbankschema
- Nummerierungsschema: **word index**, **element index**

```
0<books>1  
  <book style="textbook">2  
    <title>3Designing1 XML2 applications3</title>1  
    <editor>2  
      <family>3Bob4</family>1 <given>2Kraft5</given>1  
      ...
```

- Region für Element family (**3.3**, **4.1**)



ER-Diagramm der Speicherstruktur

XQuery-Umsetzung

- Ergebnis der Diplomarbeit von Guido Rost
- Erweiterung des relationalen Datenbankschemas um Typen (DB2-UDT's) und zugehörigen DB2-Transform-Functions
- Vorteil: Direkte Abbildung von binären Operationen zwischen Pfadausdrücken und Konstanten auf SQL
- Auswertung von Pfadausdrücken liefert schrittweise SQL-Anfragen

XQuery-Umsetzung (Beispiel)

- Selektiere alle Kindelemente des DocumentNode mit dem Namen "nodeName".

```
document(...)/nodeName
```

```
SELECT * FROM element e, path p
WHERE e.pathID=p.pathID AND e.docID=node.getDOCID()
      AND p.docID=e.docID AND e.wi=0 AND e.ei<=2
      AND p.pathexpr LIKE '/' + nodeName
ORDER BY e.wi, e.ei;
```

XML-Updatesprache (Motivation)

- *XML*: Entwicklung vom universellen Datenformat zur Technologiefamilie
- Aus Sicht der Datenbanken vor allem *XML Schema* und *XQuery* interessant
- Verschiedene Ansätze für XML-Anfragesprachen in *XQuery* vereinigt, damit Chance auf zukünftiges *SQL4XML*
- *ABER*: Im Unterschied zu SQL bisher keine Update-Operationen in *XQuery* vorgesehen
- Ersatz durch prozedurale Änderungen über DOM-API (OP's nur auf einzelnen Komponenten)

Warum Änderungssprache?

- Soll knappe Beschreibung von Änderungen auch auf Mengen ermöglichen
- Damit Abkopplung von der Speicherungsebene und Unterstützung der logischen/physischen Datenunabhängigkeit
- These: Änderungsoperationen sollten integraler Bestandteil einer vollständigen XML-Anfragesprache sein
- Ziel der DA: Entwurf einer XML-Update-Sprache als Erweiterung zur XQuery Language

Anforderungen an eine XML-Updatesprache

- Allgemeine Anforderungen (deskriptiv, abgeschlossen, adäquat, . . .)
- XML-spezifische Anforderungen
 - Datenmodell (Infoset als gemeinsame Basis)
 - Beachtung eines Schemas (DTD, XML Schema)
 - Ordnungserhaltung (Reihenfolge und Nestung)
 - Spracherweiterung/eigenständige Sprache
 - Transaktionen (Hierarchische Sperrprotokolle)
 - Transformationen vs. Basis-Änderungen

Existierende Ansätze (1)

- Document Object Model
 - Navigation über die Baumstruktur (`parentNode`, `firstChild`, ...)
 - Manipulation von Knoten (`appendChild`, `insertBefore`, `removeNode`, `replaceNode`)
- XML:DB XUpdate und SiXDML
 - Sprachvorschlag in XML-Syntax (Einbettung von XPath ähnlich XSLT)
 - SiXDML basiert auf XUpdate (syntaktisch an SQL und DL/I angelehnt), zusätzlich Dokument-Kollektionen, Schemas und Indexe

Existierende Ansätze (2)

- Logikbasierender Ansatz
 - Ausgehend vom Relationenkalkül Entwicklung eines Pfad-Prädikaten-Kalküls
 - Unterstützung von Typinformationen, speziellen Multi-Media-Operatoren
 - MMDOC-QL: Anfrage- und Änderungssprache mit eingebettetem PPK
- Andere mögliche Ausgangspunkte
 - OODM
 - Lorel
 - Hierarchische Modelle

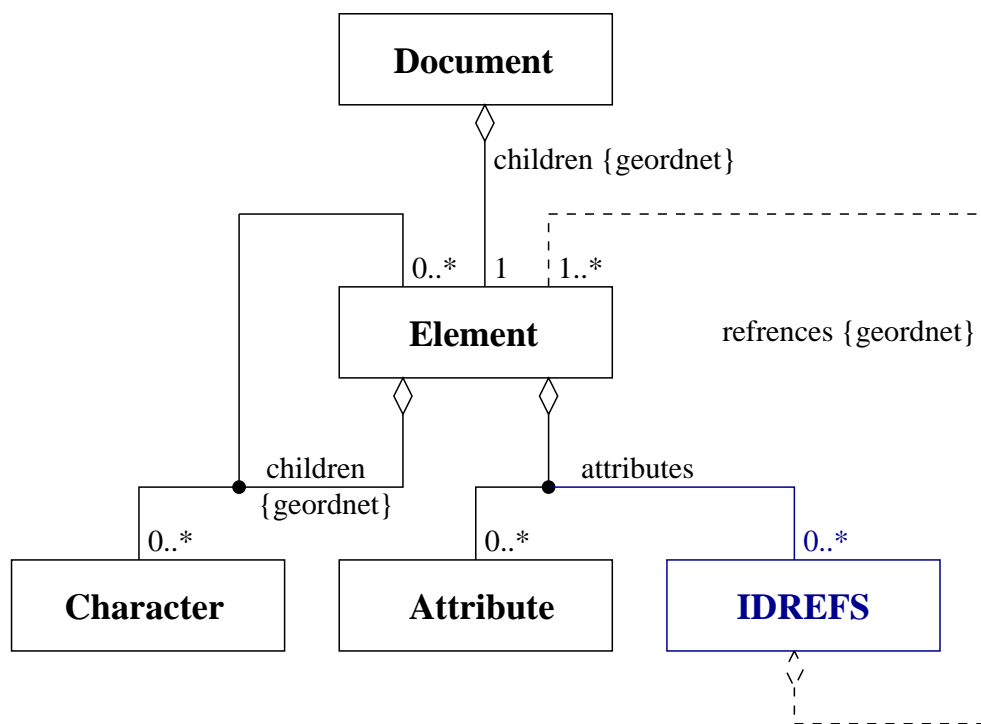
Existierende Ansätze (Vergleich)

Kriterium	DOM	XUpdate	MMDOC-QL	XQuery
Deskriptiv	-	+	+	+
Adäquat	+	-	-	+
Optimierbar	-	+	+	+
Orthogonal	-	-	+	+
Datenmodell	Infoset	Infoset-	Infoset-	erw. Infoset
Schema	-	-	-	+
Sprache	Interface/PL	XML,XPath+	eigene	XPath+
Transaktionen	-	-	-	-

Konzeption einer XML-Updatesprache (Grundlagen)

- Relationale Änderungsoperationen
 - Datenbankzustand als Kollektion von Mengen $d(\mathcal{S})$, $\mathcal{S} := \{S, \Gamma\}$
 - Änderung dann Zustandsübergang $\mathbf{DAT}(\mathcal{S}) \longrightarrow \mathbf{DAT}(\mathcal{S})$
 - lokale Mengenoperationen (insert, delete, replace) als Änderungsmodell
- Änderungen in XML
 - analoger Ansatz, jedoch unterschiedliche Datenmodelle
 - Tupelmengen vs. hierarchische Sequenzen von Knoten
 - außerdem implizite Struktur- und Schemainformationen
 - Klassifizierung: Änderung von Schemas, Strukturen und Werten

Konzeption einer XML-Updatesprache (Datenmodell)



- Vereinfachung des XQuery/XPath Data Model's
- geordneter Baum von benannten Knoten erweitert um Referenzen und Sequenzen
- Vorhandensein eines Schemas wird nicht gefordert
- Explizite IDREFS (Strukturelle Informationen vs. Attributwerte)
- Konstruktoren
 - *createElem*
 - *createRefs*
 - ...

Konzeption einer XML-Updatesprache (Basisoperationen)

- Ausgangsbasis: Bindung von Objekten an Variablen (Semantik XQuery)
- *target*-Bindung als impliziter Kontext für Sequenzen von Basisänderungen:
 - `Insert(content)`
 - `{InsertBefore | InsertAfter} (child, content)`
 - `Delete(child)`
 - `Rename(child, name)`
 - `Replace(child, content)`
 - `Move(source) (...{Before | After}...)`
 - `SubUpdate(xpathOP, predicate, updateOP+)`

Konzeption Basisoperationen (Problembetrachtungen)

- Restriktionen für SubUpdate
 - Bindungs-/Auswertungsreihenfolge beachten
 - Löschung von Bindungen in einer Update-Sequenz
- Löschen von Elementen mit ID-Attribut (dangling references)
 1. Zurückweisen der Löschung
 2. Eingehende Referenzen zurückverfolgen und kaskadierend IDREFS löschen
- Mehrdeutige Updates (Graphenmodell)

Konzeption einer XML-Updatesprache (Syntaxdefinition)

- *SUNFLWR* - Simple Update Notation for XQuery-FLWR-Expressions
- Abbildung der Basisoperatoren auf die XQuery-Syntax

```
FOR $i IN xpath-expr1, ...  
LET $j := xpath-expr2, ...  
WHERE predicate, ...  
UPDATE $i { updateOp {, updateOp}* }
```

Konzeption Syntaxdefinition (2)

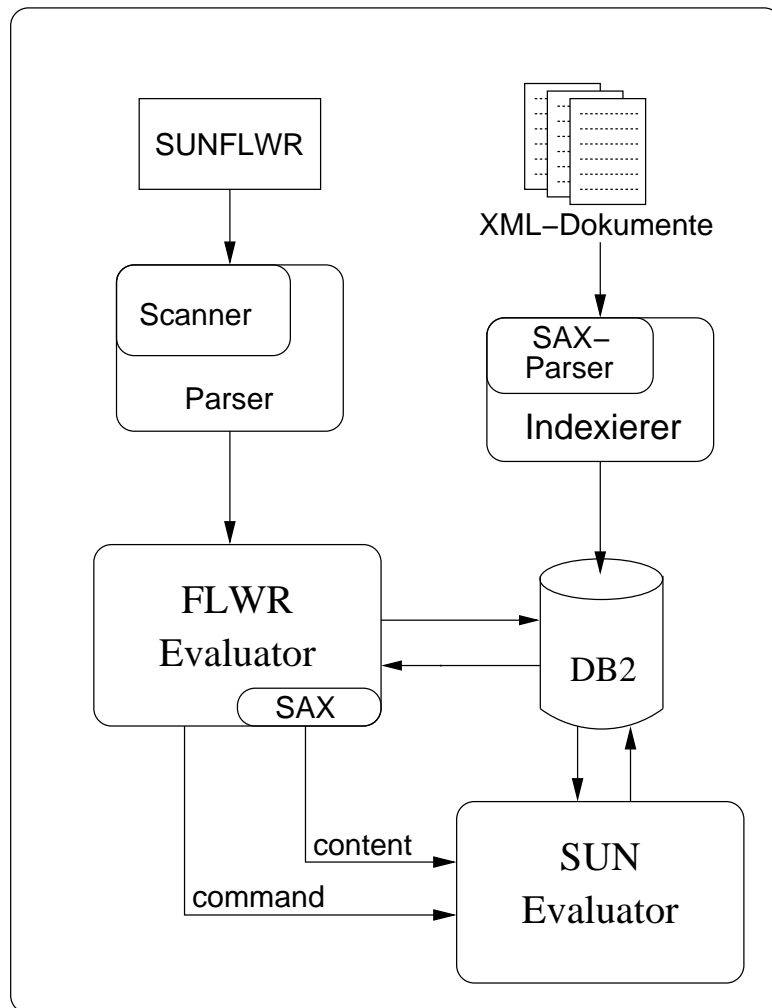
- Mögliche Konstrukte für `updateOp`

```
INSERT content [BEFORE | AFTER $child] |
DELETE $child |
RENAME $child TO name |
REPLACE $child WITH content |
MOVE $source [BEFORE | AFTER $child] |

FOR $l IN xpath-subexpr1, ...
LET $m := xpath-subexpr2, ...
WHERE predicate, ...
UPDATE $l { updateOp {, updateOp}* }
```

Konzeption Syntaxdefinition (Beispiel)

```
FOR $a in document("staff.xml")/staff/assistant[given='Gunnar']
LET $s := $a/salary
UPDATE $a {
    INSERT <degree>Dr.</degree> BEFORE family,
    INSERT createRefs(worksWith,"no007"),
    DELETE $s
}
```



Auswertungsstrategie

- Integration in bestehendes System zur XQuery-Verarbeitung
- Erweiterung der Grammatik um SUNFLWR-Ausdrücke
- Berechnung der *content*-Ausdrücke (ähnlich *RETURN*)
- Umsetzung der generierten Basis-Updates in SQL-Statements

Sprachabbildung (Umsetzungsbeispiel)

- *SUN*-Basisoperation zum Einfügen von Attributen

```
UPDATE /books/book[0] { INSERT attribute lang {EN} }
```

- Umsetzung in *SQL*-Statements

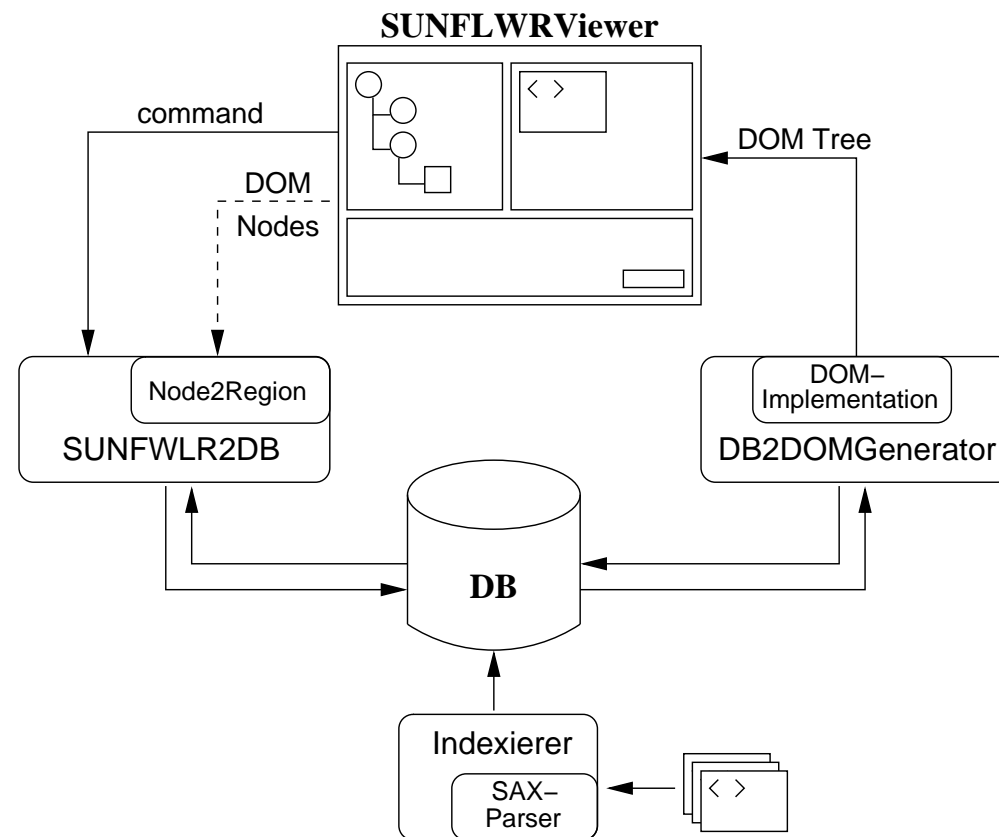
```
INSERT INTO path  
VALUES ('/books/book/@lang', 13);
```

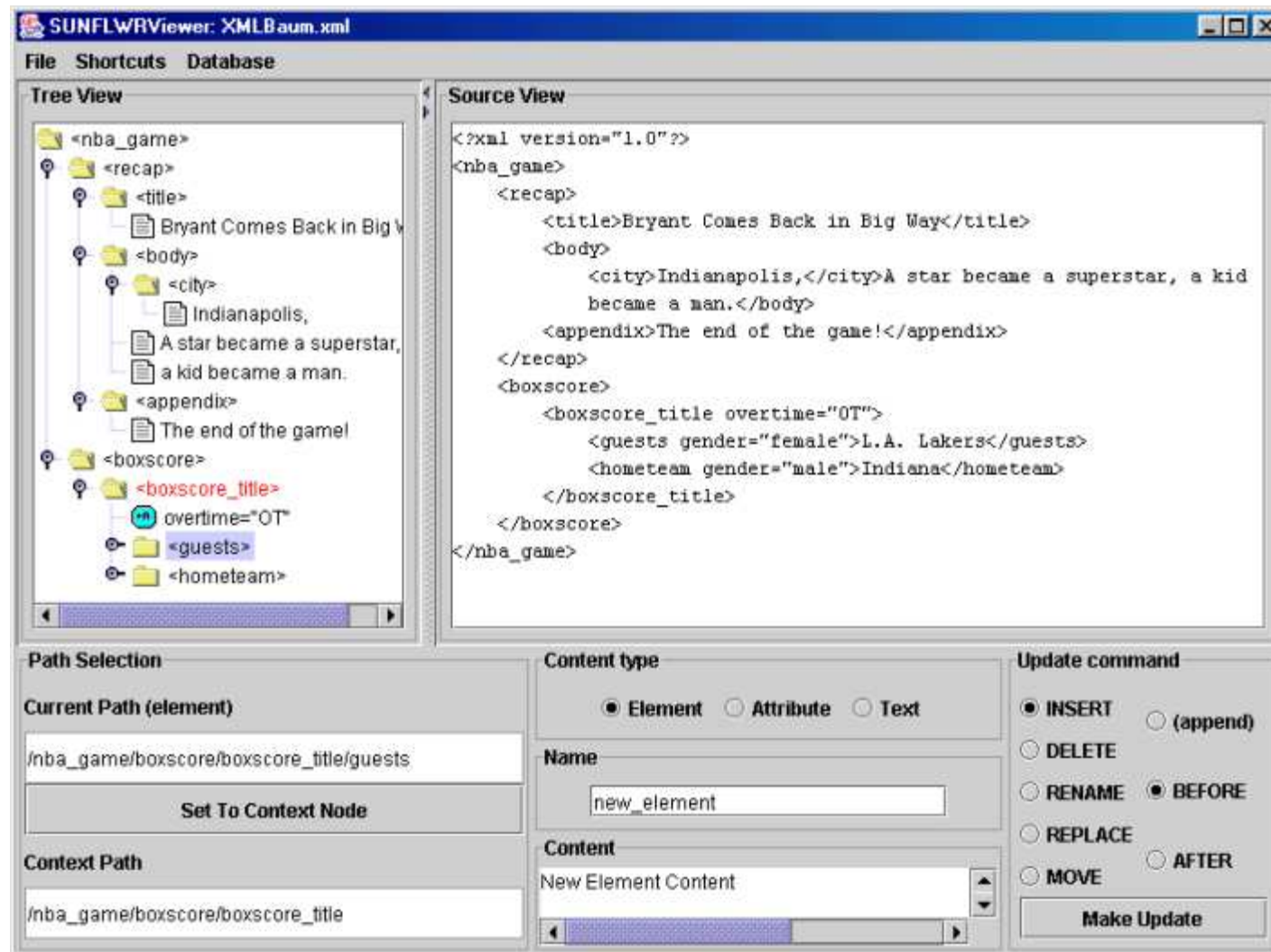
```
INSERT INTO attribut (docID, pathID, value, wi, ei, lwi, lei)  
VALUES (1, 13, 'EN', 0, 2, 0, 2)
```

Sprachabbildung (Umsetzung in SQL)

- Problem SQL: Single-Relation-Update
- Vielzahl an SQL-Updates für eine XML-Änderung erforderlich
- Einsatz von Triggern zur Vereinfachung des Löschens
- Evtl. Nutzung systemspezifischer Mechanismen (ADT)
- Nummerierungsschema leicht verständlich und umsetzbar, jedoch nicht besonders leistungsfähig

Implementierung (prototypisch)





Zusammenfassung

- Prototypische Umsetzung von XQuery+SUNFLWR auf (objekt-)relationalen DBS (DB2)
- einfache Adaption an andere Datenbanksysteme möglich
- Einsatz von objektorientierten Mechanismen kann zur Verbesserung der Performance genutzt werden, erschwert jedoch die Austauschbarkeit der Backend-Systeme
- Umsetzung auf *realen Systemen* zeigt, dass OO-Features in DBS meistens unausgereift sind /

Ausblick

- laufende Diplomarbeit: "Effiziente XML-Speicherung und XQuery-Auswertung mit Oracle9i"
- Ziel: Gesamtsystem basierend auf bisherigen Arbeiten
 - Optimierung des eingesetzten Nummerierungsschemas
 - Bündelung der einzelnen SQL-Anfragen in Stored Procedures
 - Integration der SUNFLWR-Ausdrücke
- Zeithorizont: II. Quartal 2003

**Vielen Dank
für Ihre Aufmerksamkeit!**



Meike Klettke / Holger Meyer
XML & Datenbanken
Konzepte, Sprachen und Systeme

xml.bibliothek

dpunkt.verlag
Dezember 2002
444 Seiten, Broschur
42 Euro (D)
ISBN 3-89864-148-1

