



XML Schema und XQuery - Generierung

Christoph Böhm
HumMer-Workshop
Berlin, 16.7.2004

Zu meiner Person



- Informatikstudent im 8. Semester
- Spezialisierung Datenbanken und Informationssysteme
- Nebenfach Geographie
- seit Dezember 2003 bei der Arbeitsgruppe
- Aufgabe: XQuery-Generator



Agenda

- Was bisher geschah
- XML Schema Beispiel und Parser
- XQuery Generierung
- Folgende Schritte
- Visionen

3

Christoph Böhm, Arbeitsgruppe Informationsintegration

Berlin, 16.7.2004



Was bisher geschah

- XQuery Einarbeitung
- XML Schema Einarbeitung
- XML Schema Parsen
- XML Schema graphisch darstellen
- XQuery im XML Schema darstellen
- → Definition der Semantik (33 Fälle)
- [XQuery Objektmodell]

4

Christoph Böhm, Arbeitsgruppe Informationsintegration

Berlin, 16.7.2004





XML Schema Beispiel

```

<xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema" elementFormDefault="qualified">
  <xsd:element name="statisticsDB">
    [ . . . ]
  <xsd:complexType>
    <xsd:choice>
      <xsd:element name="cityStatistics" minOccurs="0" maxOccurs="unbounded">
        <xsd:complexType>
          <xsd:sequence>
            <xsd:element name="organization" minOccurs="0" maxOccurs="unbounded">
              <xsd:complexType>
                <xsd:sequence>
                  <xsd:element name="cid" type="xsd:string"/>
                  <xsd:element name="cname" type="xsd:string" nillable="true" maxOccurs="unbounded"/>
                  <xsd:element name="funding" minOccurs="0" maxOccurs="unbounded">
                    <xsd:complexType mixed="true">
                      [ . . . ]
                    </xsd:complexType>
                  </xsd:element>
                </xsd:sequence>
              </xsd:complexType>
            <xsd:attribute name="myattribute" type="xsd:string"/>
          </xsd:complexType>
        </xsd:element>
      </xsd:choice>
    </xsd:complexType>
  </xsd:element>
  [ . . . ]
</xsd:schema>

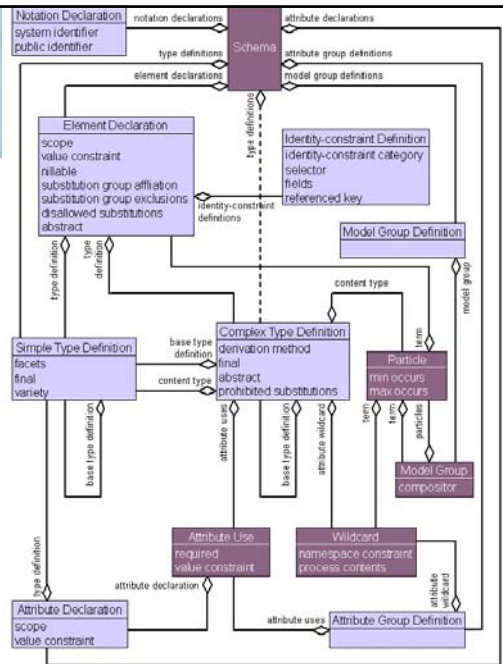
```



XML Schema Parser

XML Schema Object Model (XSOM)

By Kohsuke Kawaguchi (sun)



is a named component and has two additional properties - name and target namespace
 is an unnamed component

XML Schema Parser

Visitor → vereinfachter Baum zzgl. Informationen für XQueryGen
(isSelected(), getFather(), getChildren(), getPredicate())

```
+--de
| +-hu
| | +-mac
| | | +-hammer
| | | | +-xquerygen
| | | | | +-logic
| | | | | | +-xsparsing
| | | | | | | XAbstractTreeNode
| | | | | | | XSAttribute
| | | | | | | XSComplexElement
| | | | | | | XSElement
| | | | | | | XSModelGroup
| | | | | | | XSParserWrapper
| | | | | | | XSSimpleElement
| | | | | | | XSText
| | | | | | | XSTreeNode
```

7

Berlin, 16.7.2004



XQuery Beispiel - einfach

```
<result>
{
for $i in document("statisticsDB.xml")/
  statisticsDB/cityStatistics/organization/funding
let [ . . . ]
where [ . . . ]
order by [ . . . ]
return
(
  $i/proj,
  $i/recv
)
}
</result>
```

8



XQuery Beispiel - kompliziert

```
<result>
{
  for $i in doc("statisticsDB.xml")/statisticsDB/cityStatistics/*
  where local-name($i) = 'organization' or local-name($i) = 'financial'
  return
  (
    if ( local-name($i) = 'organization' ) then
      element {node-name($i)}
      {
        for $j in $i/*
        where local-name($j) = 'funding'
        return
          element {node-name($j)}
          {
            for $k in $j/*
            where local-name($k) = 'gid'
            return
              $k
          }
      }
    else
      (),
    if ( local-name($i) = 'financial' ) then
      element {node-name($i)}
      {
        for $l in $i/*
        where local-name($l) = 'aid'
        return
          $l
      }
    else
      ()
  )
}
</result>
```

9

Tool - Arbeitsablauf

1. Lade XML Schema (später auch Schema aus Daten extrahieren)
2. Anzeigen des Schemas
3. Auswahl der Zielstruktur
4. Zielschemamodifikation (vorerst nur durch Umbenennung der Knoten)
5. Prädikate an den Knoten antragen
6. XQuery generieren
7. XQuery ausführen

Stand der Dinge



HuMer: XQuery-Generator

E:\christoph\xml-data\ciol\statisticsDB.xsd 3 statisticsDB parse XML Schema generate Query execute Query

Element	Incl.Tag	Type Information	Name	Predicate
statisticsDB	n.a.	CT, isMixed = false, (-11--11)	statisticsDB	-
choice	n.a.	choice		
cityStatistics	n.a.	CT, isMixed = false, (0--1)	cityStatistics	-
sequence	n.a.	sequence		
organization	n.a.	CT, isMixed = false, (0--1)	organization	-
myattribute : string	n.a.	Attribute	myattribute	-
sequence	n.a.	sequence		
cid	<input checked="" type="checkbox"/>	ST, (1-1)	cid	-
cname	<input checked="" type="checkbox"/>	ST, (1--1)	cname	-
funding	n.a.	CT, isMixed = true, (0--1)	funding	-
text	n.a.	text		
all	n.a.	all		
gid	<input checked="" type="checkbox"/>	ST, (1-1)	gid	-
proj	<input checked="" type="checkbox"/>	ST, (1-1)	proj	Hummer
faid	<input checked="" type="checkbox"/>	ST, (1-1)	faid	-
recv	<input checked="" type="checkbox"/>	ST, (0-1)	recv	=1000
financial	n.a.	CT, isMixed = false, (0--1)	financial	-
project	n.a.	CT, isMixed = false, (0--1)	project	-
city	<input checked="" type="checkbox"/>	ST, (1--1)	city	-

no query generatet

no query executed

11 Christoph Böhm, Arbeitsgruppe Informationsintegration Berlin, 16.7.2004

Stand der Dinge



HuMer: XQuery-Generator

E:\christoph\xml-data\ciol\statisticsDB.xsd 3 statisticsDB parse XML Schema generate Query execute Query

Element	Incl.Tag	Type Information	Name	Predicate
statisticsDB	n.a.	CT, isMixed = false, (-11--11)	statisticsDB	-
choice	n.a.	choice		
cityStatistics	n.a.	CT, isMixed = false, (0--1)	cityStatistics	-
sequence	n.a.	sequence		
organization	n.a.	CT, isMixed = false, (0--1)	organization	-
myattribute : string	n.a.	Attribute	myattribute	-
sequence	n.a.	sequence		
cid	<input checked="" type="checkbox"/>	ST, (1-1)	cid	-
cname	<input checked="" type="checkbox"/>	ST, (1--1)	cname	-
funding	n.a.	CT, isMixed = true, (0--1)	funding	-
text	n.a.	text		
all	n.a.	all		
gid	<input checked="" type="checkbox"/>	ST, (1-1)	gid	-
proj	<input checked="" type="checkbox"/>	ST, (1-1)	proj	-
faid	<input checked="" type="checkbox"/>	ST, (1-1)	faid	-
recv	<input checked="" type="checkbox"/>	ST, (0-1)	recv	-
financial	n.a.	CT, isMixed = false, (0--1)	financial	-
myattribute : string	n.a.	Attribute	myattribute	-
all	n.a.	all		
aid	<input checked="" type="checkbox"/>	ST, (1-1)	aid	-
amount	<input checked="" type="checkbox"/>	ST, (1-1)	amount	-
date	<input checked="" type="checkbox"/>	ST, (0-1)	date	-
project	n.a.	CT, isMixed = false, (0--1)	project	-
city	<input checked="" type="checkbox"/>	ST, (1--1)	city	-

no query generatet

no query executed

Sinn, Zweck, Ideen und Visionen 1



Funktionen:

- Schema aus Daten berechnen
- Vorschlag für Joins anhand Key-KeyRefs
- Unsinnige Anfragen mittels Schemainformationen vermeiden
- „bessere“ Anfragen mittels Schemainformationen generieren
- Schnittstellen für XML Schema und XQuery zur Einbettung in Drittsysteme
- Veränderung des Zielschemas (Umbenennung und Knotenhierarchie)
- Simplify Query – Button
- Query – Example – Result – Function



Sinn, Zweck, Ideen und Visionen 2



- zur Erschließung / zum Verstehen von komplexen computergenerierten XML Dokumenten
- → Verstehen und Debuggen von Internettransfer (OGC Webservices)
- Als Anfragehilfe für XML – Datenbanken (mittels XML modellierte Bilderdatenbank in der die Bilder in hierarchischen Kategorien gespeichert sind)
- Flexibles Tool zur Einbettung in Systeme



Noch Fragen ?

