XML Schema a jeho implementace v prostředí relační databáze

Autor: Irena Mlýnková

Vedoucí: Prof. RNDr. Jaroslav Pokorný, CSc.

Katedra Softwarového Inženýrství

2003

Cíl práce

- Využít pro ukládání XML dat do relačních databází jazyk XML Schema
- Návrh tří algoritmů:
  1. Transformace schématu XML dat popsaném v jazyce XML Schema do relačního DB schématu
  2. Uložení dat z odpovídajících XML dokumentů do takto vytvořené relační DB
  3. Mapování podmnožiny dotazovacího jazyka nad XML dokumenty na posloupnost SELECT příkazů nad takto vytvořenou relační DB
     - Výsledek: zrekonstruovaný fragment XML dokumentu
- Ukázková implementace

Jazyk XML Schema

- **Definuje přípustnou strukturu XML dokumentu (elementy, atributy, pořadí a počet podelementů, datové typy…)**
- **Zaměření práce na:**
  - **Objektově orientované rysy (uživatelsky definované typy, dědičnost, substituovatelnost…)**
  - **Vlastnost: Schéma XML dat popsané v jazyce XML Schema (dále jen XML schéma) je opět XML dokumentem**

Algoritmus 1.

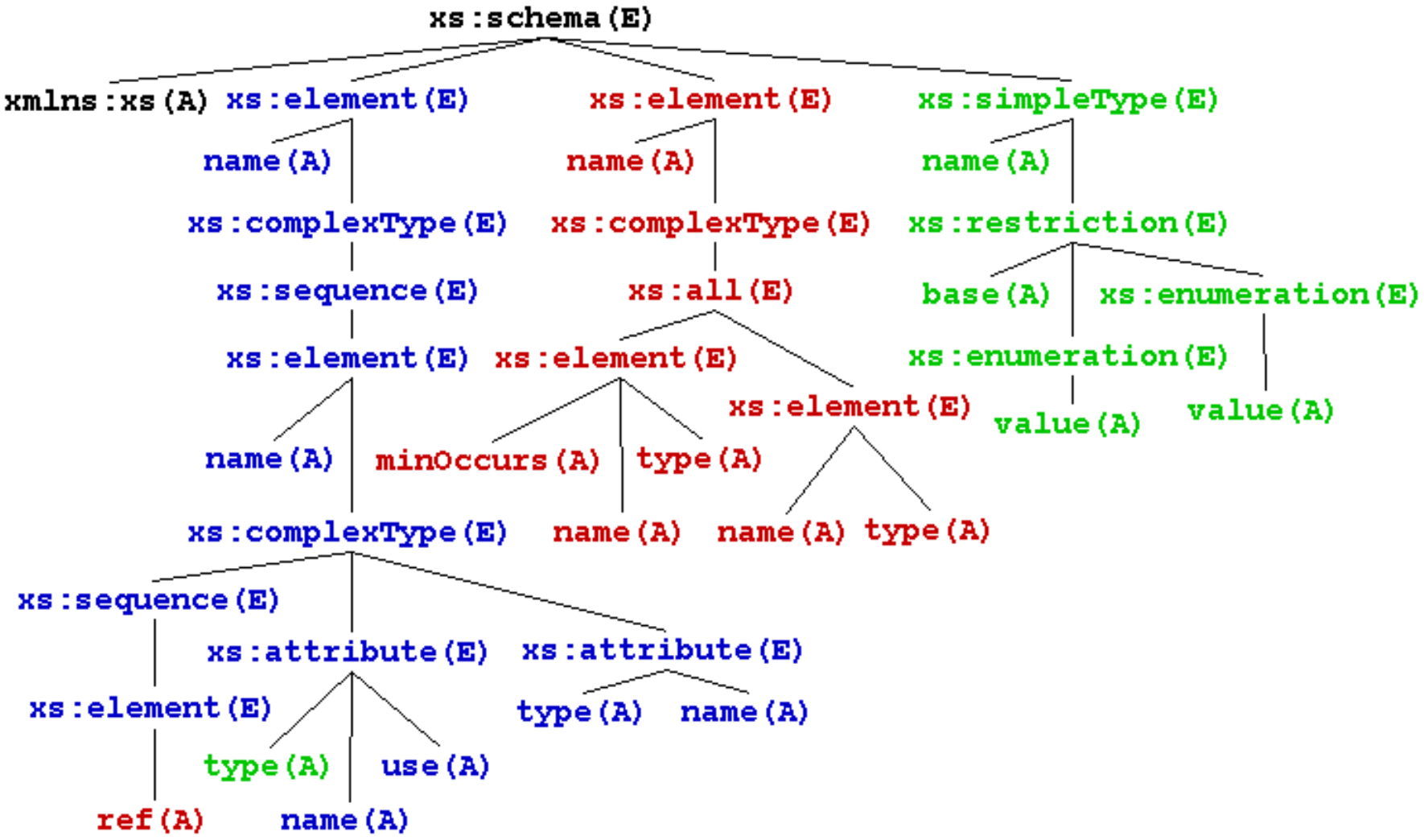
- **Cílové schéma:**
  - **Objektově relační (OR): SQL:1999**
  - **Ukázková implementace: Oracle9i Release 2**
- **XML schéma = XML dokument -> lze zpracovat pomocí rozhraní DOM:**
  - **Modifikace DOM stromu = DOM graf**
    - **Orientovaný graf**
    - **Vzájemné použití prvků schématu, tj. pořadí při vytváření prvků OR schématu**
    - **Problém: cykly**

Příklad 1. XML schéma

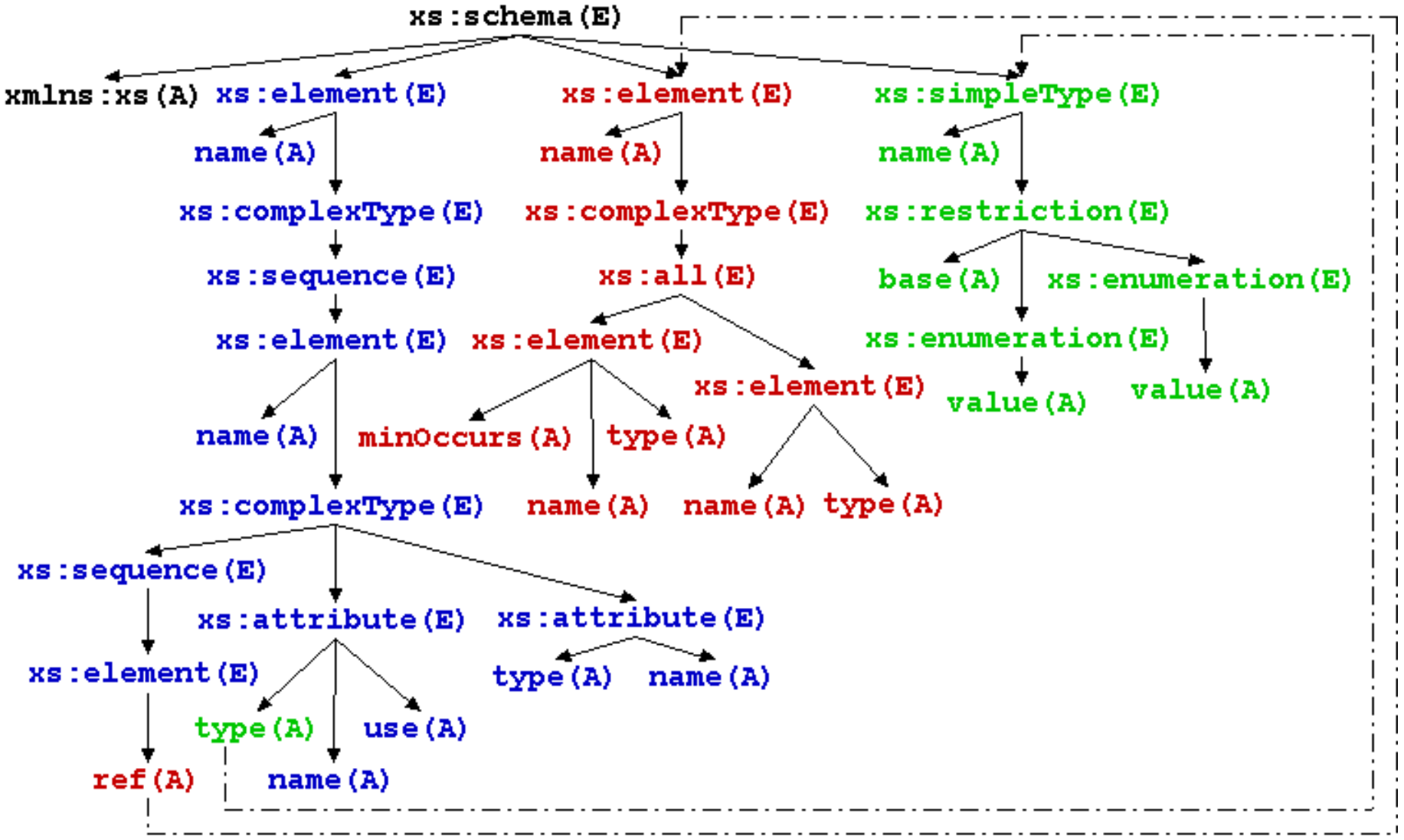
```
<xs:schema xmlns:xs="...">
 <xs:element name="zamestnanci">
  <xs:complexType>
   <xs:sequence>
    <xs:element name="osoba">
     <xs:complexType>
      <xs:sequence>
       <xs:element ref="jmeno"/>
      </xs:sequence>
      <xs:attribute name="volno"
                    type="anoNe"
                    use="required"/>
      <xs:attribute name="poznamka"
                    type="xs:string"/>
     </xs:complexType>
    </xs:element>
   </xs:sequence>
  </xs:complexType>
 </xs:element>

 <xs:element name="jmeno">
  <xs:complexType>
   <xs:all>
    <xs:element name="krestni"
                type="xs:string"
                minOccurs="0"/>
    <xs:element name="prijmeni"
                type="xs:string"/>
   </xs:all>
  </xs:complexType>
 </xs:element>
 <xs:simpleType name="anoNe">
  <xs:restriction base="xs:string">
   <xs:enumeration value="ano"/>
   <xs:enumeration value="ne"/>
  </xs:restriction>
 </xs:simpleType>
</xs:schema>
```

Příklad 2. DOM strom



Příklad 3. DOM graf



Mapování

<table>
<tr><th colspan="2">XML Schema:</th><th colspan="2">OR schéma:</th></tr>
<tr><td colspan="2">Vestavěný jednoduchý typ</td><td colspan="2">Jednoduchý typ (+ IO)</td></tr>
<tr><td colspan="2">Uživatelsky def. jednoduchý typ</td><td colspan="2">Jednoduchý typ + IO</td></tr>
<tr><td rowspan="2">Složený typ</td><td>Atributy</td><td rowspan="2">UDT</td><td>Atributy – jednoduchý typ</td></tr>
<tr><td>Obsah:
• Jednoduchý
• Složený (vztah element-podelement)</td><td>Obsah:
• Atribut – jednoduchý typ
• Atribut – typový sloupec / reference / pole referencí</td></tr>
<tr><td colspan="2">Odvozený složený typ</td><td colspan="2">Dědičnost</td></tr>
<tr><td colspan="2">Element</td><td colspan="2">Vlastní typová tabulka / typový sloupec nadelementu</td></tr>
</table>

OR schéma = množina typových tabulek „propojených“ referencemi

Algoritmus 2.

- **Průchod DOM stromu ukládaného XML dokumentu**
- **Vytváření konstruktorů UDT**

Algoritmus 3.

- Zvolený jazyk: XQL (podmnožina)
  - Navigace hierarchií elementů a podelementů
  - Filtrace výsledků
- Odpovídající SQL dotaz = navigace hierarchií UDT/tabulek
- Přechod na následující element: dle mapování vztahu element-podelement
  - Typový sloupec: přechod přes ‘.’
  - Reference: přechod přes ‘->’
  - Pole referencí: odhnízdění + přechod přes ‘->’

Závěr

- **Hlavní přínosy práce:**
  - **Vzhledem k existujícím řešením originální volba cílového schématu a jemu odpovídající návrh algoritmů**
  - **Návrh modifikace DOM stromu (DOM graf)**
    - **Algoritmus zpracování (včetně ošetření cyklů)**
- **Možná rozšíření:**
  - **Využití prvků, které nebyly do řešení začleněny**
  - **Nedořešené detaily OO rysů normy SQL:1999 -> využití nové normy jazyka SQL**

Konec...

Vyjádření k připomínkám

1. **Zmínit nevýhody mapování do několika tabulek**

   Tyto nevýhody jsou zmíněny až v pozdějších kapitolách popisujících algoritmy, kterých se především týkají.

2. **Mapování ID, IDREF, IDREFS**

   Jistě existují metody částečně postihující vlastnosti těchto typů. Text ovšem (naneštěstí zřejmě nejasně) říká, že není možné postihnout všechny (příliš obecné) vlastnosti.

Konec