FHNW: Vorlesung XML Sommersemester 2006

Test XML Schema

MUSTERLÖSUNG (Lösungen in grün & underline)

12.6.2006, 8^{30} – 9^{30}

Bitte schreiben Sie *sorgfältig* und *gut lesbar*! Danke!
Für diesen Test dürfen Sie *keinerlei Unterlagen* verwenden, insbesondere auch keine elektronischen Hilfsmittel wie Laptop oder Organizer!

1. XML Schema kennt die Konzepte der *Simple* und *Complex Types*, und die Konzepte des *Simple* und *Complex Content*.
   a. Erklären Sie diese Konzepte jeweils im Vergleich zueinander (die beiden *Types* und die beiden *Contents*).
      Simple und Complex Types sind die *beiden Typarten*, die XML Schema unterscheidet. Simple Types sind einfache Werte, die *nicht durch XML Markup strukturiert* werden (Attribute haben immer Simple Types). Complex Types sind strukturierte Werte, also *Element-Inhalt mit Attributen*. Complex Types können Simple oder Complex Content haben. Simple Content heisst, dass der *Inhalt des Elements ein Simple Type* ist. Unter Complex Content sind alle anderen Arten von Inhalt zusammengefasst (*Element only*, *Mixed*, *Empty*).
   b. Erklären Sie diese Konzepte im Vergleich zwischen den beiden Gruppen (*Types* im Vergleich zu *Content*).
      Types sind die *grundlegenden Bausteine*, aus denen eine Schema zusammengesetzt wird (alle Attribute und Elemente haben einen Type), während die Contents nur eine *Klassifizierung von Complex Types* sind, abhängig davon, welchen Inhalt ein Complex Type im Element erlaubt.
   c. Erklären Sie, wie diese Konzepte in die *Typableitungsmechanismen* von XML Schema einzugliedern sind.
      Types erlauben die *Ableitung abhängig vom Typ*, Simple Types können per *Restriction, Union oder List* abgeleitet werden, während Complex Types per *Restriction oder Extension* abgeleitet werden können. Contents bestimmen für Complex Types, *in welcher Weise* Restriction oder Extension Typableitungen angewendet werden kann.
2. XML Schema Identity Constraints dienen dazu, Bedingungen zu definieren, die über die reine Grammatik hinausgehen und zusätzlich zu ihr weitere Einschränkungen dafür sind, die ein Dokument erfüllen muss.
   a. Es gibt drei Aspekte, die für Identity Constraints wichtig sind. Beschreiben Sie diese drei Aspekte und die Art, wie sich diese Aspekte auf die Struktur der Instanzen beziehen.

Der *1. Aspekt* ist der Ort der Identity Constraints, der bestimmt, von wo aus der Constraint interpretiert wird. Dieser Ort bestimmt sich durch die Element-Definition, in der der Constraint vorkommt. Der *2. Aspekt* ist die Auswahl der Knoten, auf die sich der Constraint bezieht. Dies wird als XPath im Constraint selber angegeben und relativ zu dem Element interpretiert, in dem der Constraint definiert wurde (1. Aspekt). Der *3. Aspekt* ist die Identifikation der Werte, auf denen der Constraint definiert ist. Dies wird als XPath im Constraint selber angegeben und relativ zu den Knoten interpretiert, auf die sich der Constraint bezieht (2. Aspekt).

   b. Identity Constraints werden in Element-Definitionen angegeben, sie stehen also "ausserhalb" der Typenwelt von XML Schema. Was sind die Konsequenzen dieser Tatsache und wie kann man damit umgehen?
   Weil die Identity Constraints nicht in Typen definiert werden, sind sie *nicht Bestandteil der Typableitungen*. Es gibt aus diesem keine Möglichkeit, Identity Constraints an Typen zu binden, sie sind immer erst in der Element-Definition vorhanden. Sollen die Constraints auch auf Elementen abgeleiteter Typen existieren, so müssen sie dort *von Hand* eingefügt werden. In einem Typ könnte es bestenfalls einen Kommentar geben, der dokumentiert, dass es weitergehende Konsistenzbedingungen für diesen Typ gibt, die bei dessen Verwendung in Identity Constraints definiert werden sollten.

3. Was die Grammatik und Constraints betrifft, lässt sich in einem XML Schema alles definieren, was in einer DTD definiert werden kann. Aus welchem Grund kann es trotzdem notwendig sein, weiterhin eine (u.U. minimale) DTD zu benutzen für die Verarbeitung von XML Dokumenten?
   Ein möglicher Grund ist, dass ein Partner *keine XML Schemas verarbeiten kann* und man sich deshalb darauf geeinigt hat, DTDs zu unterstützen. Ein weiterer und wichtigerer Grund kann sein, dass man *Entities* benutzen möchte. Diese werden von XML Schema nicht unterstützt, es muss also zumindest eine Minimal-DTD geben, in der die Entities definiert werden, die man in seinen XML-Dokumenten verwenden möchte.
4. Bei der *Ableitung komplexer Typen* in XML Schema durch Erweiterung und Einschränkung gibt es hinsichtlich des Umgangs mit Typen mit *Mixed Content* gewisse Grundsätze.
   a. Erklären Sie, warum diese Grundsätze existieren.
   Ob ein Complex Type Mixed ist oder nicht, bezieht sich prinzipiell auf den *gesamten Inhalt* eines Typen (es ist ein binäres Flag für den Typen). Es kann also in XML Schema keine Complex Types geben, die sozusagen nur "zum Teil" oder auf andersartig spezielle Art Mixed Content haben.
   b. Erklären Sie, was diese Grundsätze für *Typ-Erweiterung* sind, und weshalb das so ist.
   Erweiterung fügt Inhalte an einen bestehenden Typ an, ohne dass der Inhalt des Basistypes verändert werden darf. Aus diesem Grund *darf sich das "Mixed Flag" nicht verändern* im erweiterten Typ, weil sich dies ja sonst auch auf den "vorderen Teil des Typs" (den Inhalt des Basistyps) auswirken würde.
   c. Erklären Sie, was diese Grundsätze für *Typ-Einschränkung* sind, und weshalb das so ist.

Eine Einschränkung *darf einen Mixed Content Type zu Nicht-Mixed ändern*, weil dies einfach eine Einschränkung der erlaubten Werte darstellt. Nicht-Mixed Typen als Basistypen erlauben ohnehin keinen Mixed Content und *müssen daher Nicht-Mixed bleiben*.

5. Sie verwenden ein fremdes Schema per `redefine`, indem Sie dieses benutzen (mit `redefine` darauf verweisen) und eine einzige Attributgruppe umdefinieren, in der Sie eigene Attribute hinzufügen (dies war der einzige Aspekt, der Sie am ursprünglichen Schema gestört hat). Funktioniert dieses Vorgehen in jedem Fall? Was könnte schief gehen?
   Es kann Konflikte mit bestehenden Elementen geben, die die Attributgruppe verwenden, und ein Attribut des von Ihnen hinzugefügten Namens bereits deklarieren.