



Praktikum zu XML: Übung 07

Das „MOM“-Beispiel mittels
XML Schema, Teil 2



Organisatorisches



- Arbeitsverzeichnis:
`~/kurse/xml/07/`
- Dateinamen:
`07-bestell.xml` `XMLSchema.dtd`
`07-bestell.dtd` `datatypes.dtd`
`07-bestell.xsd`
- Abzugeben:
`07-bestell.xsd`
- Werkzeuge:
`emacs` `# NICHT X-Emacs`
`SAXPrint` `# bzw. "schemavalidator"`
`mozilla` `# zur Nachkontrolle`



- Zur Aufgabe:
 - Ziel ist der Umbau der DTD zur Bestellung in ein XML Schema.
 - Zweistufiges Vorgehen:
 - Aufgabe 06:
Zunächst reiner Umbau DTD → Schema
Dabei Kennenlernen der neuen Werkzeuge
 - Aufgabe 07:
Präzisere Datentypen u.a. Schema-Features!
- Abgabezeitpunkt:
 - Ausnahmsweise zusammen mit Teil 1 (Aufgabe 06).



- Dateien:
 - 07-bestell.xml, 07-bestell.dtd
 - Aus Ihren Daten der Aufgabe 06 kopieren
 - Dann geeignet modifizieren!
 - XMLSchema.dtd, datatypes.dtd
 - Aus Aufgabe 06 übernehmen (optional)
 - 07-bestell.xsd
 - Aus Ihren Daten der Aufgabe 06 kopieren
 - Dann selbständig weiterentwickeln!



- A: Definition neuer Datentypen (Elemente)
 - Verbesserungsfähig:
 - Belegnummer
 - Datum
 - IdentNr (Handelspartner)
 - ArtNr
 - Beschreibung
 - Menge
 - Summenterm
 - Definieren Sie mittels `<simpleType>` präzisere Datentypen für diese Elemente und verwenden Sie diese in den Element-Definitionen!



- B: Definition neuer Datentypen (Attribute)
 - Welche Attribute lassen sich mit XML Schema noch (deutlich) präziser fassen?
 - Definieren Sie auch hier mittels `<simpleType>` ggf. präzisere Datentypen und verwenden Sie diese in den Attribut-Definitionen!
- Bemerkungen zu A und B
 - Auch die Verwendung präziserer *eingebauter* Datentypen ist eine Verbesserung.
 - Auswahllisten sind i.d.R. schon sehr präzise. Hier besteht selten noch Handlungsbedarf



- C: „Nacharbeiten“
 - Besteht Ihre XML-Datei den Validierungstest?
 - Prüfen und ggf. korrigieren Sie die unteren und oberen Grenzen von Wiederholungen einiger Elemente:
 - minOccurs, maxOccurs.
 - Haben Sie alle Default-Deklarationen nach XML Schema übernommen?
 - default, fixed; use (aber: siehe auch **Bemerkungen** unten!)
 - Welche Anforderungen aus Aufgabe 04 sind auch jetzt noch nicht durch Schema-Validierung zu garantieren?
 - Schreiben Sie diese noch nicht erfüllten Anforderungen als Kommentare in Ihre Schema-Datei direkt unterhalb des tags <schema>.
 - Besteht die XML-Datei immer noch den Validierungstest?



- C: Bemerkungen:
 - Nach aktuellem Stand (12.5.03) erzeugt der Validierer Fehlermeldungen der Art „Message: NOTATION 'GLN' must be a valid QName“, wenn man einen Notations-Namen wie ‚GLN‘ als Wert des Attributs „default“ angibt. Das ist (für mich) nicht nachvollziehbar, denn Notationen sollten von Namensräumen nicht betroffen sein.
 - Lassen Sie ggf. diese Default-Deklarationen aus.
 - Sollte das Problem bei Ihnen nicht auftreten, bzw. sollten Sie sogar eine Erklärung finden, bitte ich um Rücksprache.



- D: Provokationstests
 - **Testen Sie abschließend Ihre neuen Datentypen auf sicheres Erkennen der Wertebereichsgrenzen.**
 - Ändern Sie dazu *vorübergehend* einen Element- bzw. Attributwert in der XML-Datei so, dass ein Fehler gemeldet werden müsste.
 - Prüfen Sie dies per Validierung!
 - Tragen Sie wieder den alten – korrekten – Wert ein und variieren Sie analog den Wert eines anderen Feldes.
 - Gelingt es Ihnen, einen unzulässigen Wert am Validierer „vorbeizuschmuggeln“? Wenn ja – lässt sich der Datentyp noch strenger definieren (mit vertretbarem Aufwand...)?
 - Nicht „perfekte“ Datentypen bitte in der Schema-Datei kommentieren (noch nicht erfüllte Anforderungen)!