



Praktikum zur Veranstaltung XML-Technologie: **Übung 06**

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



Organisatorisches



- Arbeitsverzeichnis:

`~/lv/xmltech/06/`

- Dateinamen:

`06-bestell.xml`

`06-bestell.dtd`

`06-bestell.xsd`

`XMLSchema.dtd`

`datatypes.dtd`

`06-myns.ent`

- Abzugeben:

`06-bestell.xsd, 06-bestell.xml [, 06-myns.ent]`

- Werkzeuge:

`emacs`

oder X-Emacs

`sval`

Zur Schemavalidierung

`firefox`

zur Nachkontrolle



- Zur Aufgabe:
 - Ziel ist der Umbau der DTD zur Bestellung in ein XML Schema.
 - Zweistufiges Vorgehen, über drei Wochen verteilt:
 - Aufgabe 05, Woche 1 & 2:
 - Zunächst reiner Umbau DTD → Schema
 - Dabei Kennenlernen der neuen Werkzeuge
 - Aufgabe 06, Woche 2 & 3:
 - Präzisere Datentypen u.a. Schema-Features!
- Abgabezeitpunkt:
 - Aufgabe 05 (spätestens) am Ende von Woche zwei,
 - Aufgabe 06 am Ende der dritten Woche



- Dateien:
 - 06-bestell.xml, 06-bestell.dtd
Aus Ihren Daten der Aufgabe 05 kopieren
Dann geeignet modifizieren!
 - XMLSchema.dtd, datatypes.dtd, 06-myns.ent
Aus Aufgabe 05 übernehmen (optional)
 - 06-bestell.xsd
Aus Ihren Daten der Aufgabe 05 kopieren
Dann selbständig weiterentwickeln!



- A: Definition neuer **Datentypen für Elemente**
 - Verbesserungsfähig (Beispiele):
 - 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 für 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 nur noch selten 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 03 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?



- Bemerkungen:
 - Nach aktuellem Stand (28.04.2009) erzeugt der Validierer Fehlermeldungen der Art
Message: Attribute 'public' must appear in global notation declarations
wenn man `<notation name="..." system="..." />` verwendet.
Das ist (für mich) nicht nachvollziehbar, denn Attribut „public“ sollte genauso optional sein wie „system“.
 - Verwenden Sie ggf. „public“ statt „system“.
- Hinweis:
 - In der zweiten Auflage von XML Schema wurde deutlicher zum Ausdruck gebracht, dass Attribut „public“ in der Tat optional ist!



- 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 Schemavalidierung!
 - Tragen Sie wieder den alten – korrekten – Wert ein und variieren Sie analog den Wert eines anderen Feldes.
 - Protokollieren Sie Ihre Tests in einem Kommentarblock der Schemadatei!
 - 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 (unter "noch nicht erfüllte Anforderungen")!