



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

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



## Organisatorisches



- Arbeitsverzeichnis:

`~/lv/xmltech/06/`

- Dateinamen:

<code>06-bestell.xml</code>	<code>XMLSchema.dtd</code>
<code>06-bestell.dtd</code>	<code>datatypes.dtd</code>
<code>06-bestell.xsd</code>	<code>06-myns.ent</code>

- Abzugeben:

`06-bestell.xsd`

- Werkzeuge:

<code>emacs</code>	# oder X-Emacs
<code>sval</code>	# Zur Schemavalidierung
<code>firefox</code>	# zur Nachkontrolle



- Zur Aufgabe:
  - Ziel ist der Umbau der DTD zur Bestellung in ein XML Schema.
  - Zweistufiges Vorgehen:
    - Aufgabe 05:
      - Zunächst reiner Umbau DTD → Schema
      - Dabei Kennenlernen der neuen Werkzeuge
    - Aufgabe 06:
      - Präzisere Datentypen u.a. Schema-Features!
- Abgabezeitpunkt:
  - Am Ende der 3-Wochen-Frist für Aufgaben 05 & 06.



- 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!



## Aufgabe



- 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!



## Aufgabe



- 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 selten noch Handlungsbedarf



## Aufgabe

10

- 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?



## Aufgabe

10

- C: Bemerkungen:
  - Nach aktuellem Stand (29.11.2005) 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“.
  - 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 Schemavalidierung!
  - 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 (unter "noch nicht erfüllte Anforderungen")!