

## Vorlesung XML im Wintersemester 2003/2004

### Test XML Schemasprachen (DTD und XML Schema)

16.12.2003, 13<sup>15</sup> - 14<sup>15</sup>

## MUSTERLÖSUNG

Name: \_\_\_\_\_

In diesem Test sollen Sie Ihr im bisherigen Verlauf der Vorlesung erstelltes Schema (bzw. Ihre Schemas, denn wir haben ja bereits eine DTD und ein XML Schema erstellt) beschreiben und bewerten. Dabei geht es nicht darum, die Schemas detailliert zu dokumentieren, sondern sie sollen Ihr *Modell* (das mit allen Randbedingungen nur in Ihrem Kopf existiert) mit den *Schemas* vergleichen, und an interessanten Stellen erläutern, warum und welche *Unterschiede* zwischen Ihrem Modell und den Schemas bestehen.

- 1) Identifizieren und Beschreiben Sie die Entities und Relationships in Ihrem Anwendungsbeispiel, auf die Sie in den folgenden Fragen eingehen wollen (d.h., höchstwahrscheinlich nicht alle Entities und Relationships).

Entities sind Künstler, Komponisten, Labels, Alben, und Musikstücke.  
Relationships sind Release (Album-Label), Interpret (Künstler-Musikstück),  
Enthalten (Musikstück-Album), Komponiert (Komponist-Musikstück).

- 2) Beschreiben Sie die Abbildung interessanter Entities und Relationships (je *mindestens ein Beispiel*) auf DTD Konstrukte. Beachten Sie dabei besonders, welche Aspekte Ihres Modells Sie formal darstellen konnten und welche nicht.

Abbildung auf Elemente und Attribute, Künstler und Komponisten sind sehr ähnlich, von daher sind ihre Eigenschaften (Elemente/Attribute) als Parameter Entities definiert und für beide Entities wiederbenutzt. Relationships werden grundsätzlich als ID/IDREF-Beziehungen dargestellt. Abhängigkeiten untereinander (z.B. Todesdatum > Geburtsdatum) sind nicht darstellbar.

- 3) Beschreiben Sie die Abbildung interessanter Entities und Relationships (je *mindestens ein Beispiel*) auf XML Schema Konstrukte. Beachten Sie dabei besonders, welche Aspekte Ihres Modells Sie formal darstellen konnten und welche nicht, unter besonderer Beachtung des Vergleichs zum DTD Modell.

Die Entities Künstler und Komponist müssen nun nicht mehr mittels Parameter Entities die gleichen Eigenschaften zugewiesen bekommen,

sondern sie können einen gemeinsamen Typ benutzen, der in beiden Elementdeklarationen referenziert wird. Die Relationships benutzen nun nicht mehr die ID/IDREF-Typen der DTDs, sondern es wird für alle Referenzen ein eigener Typ verwendet, der die lexikalischen Einschränkungen für IDs definiert. Die eigentlichen Konsistenzbedingungen werden durch XML Schema Identity Constraints definiert.

- 4) Beschreiben Sie für ausgewählte interessante Aspekte (mindestens zwei), welche der von Ihnen weder in DTD noch in XML Schema formalisierten Aspekte Sie deklarativ beschreiben könnten (*mindestens einer*) und welche Mechanismen Sie dazu bräuchten; und welche Aspekte ihnen andererseits so schwer deklarativ zu beschreiben erscheinen (*mindestens einer*), dass Sie sie eher auf Programmebene implementieren würden.

Das Todesdatum einer Person muss nach dem Geburtsdatum liegen, diese Abhängigkeit zweier Wert kann man weder in einer DTD noch in einem XML Schema ausdrücken. Nötig wäre dazu eine Sprache wie Schematron, die solche Co-Constraints unterstützt. Programmieren sollte man Aspekte wie die Einschränkung, dass ein Album nicht Musikstücke von mehr als 90min Dauer enthalten darf (bzw. eine Doppelalbum 3h), die arithmetische Bedingung kann man Programmgesteuert überprüfen.

- 5) Falls sie XML Schema Complex Type Derivation benutzt haben, beschreiben Sie, an welchen interessanten Stellen und welche Art der Complex Type Derivation Sie verwendet haben und warum. Falls Sie keine Complex Type Derivation in Ihrem XML Schema benutzt haben, begründen Sie weshalb, und wo dies allenfalls möglich wäre.

Um die Entities Künstler und Komponist spezifischer beschreiben zu können, könnte man einen Personentyp definieren, den man dann als Basistyp verwendet für Typableitungen, in denen man die Typen für die Entities um Informationen erweitert, die spezifisch sind für Künstler bzw. Komponisten.