XSD 1.1 labs

**Lab 1. Approving/Rejecting Auto Loan Applications**

A certain car insurance company, called New England, has a

rule regarding at what age an applicant will be

considered for receiving a loan for a car. This table

shows the minimum age requirements:

Minimum Age Requirements for the New England states:

State Age

------------------------

CT 19

MA 18

ME 19

NH 17

RI 18

VT 16

In the labs\lab01 folder you will find a file, auto-loan-application.xml

Open it and examine it. You see that the applicant

lives in Vermont (VT) and is 17 years old. According

to the table above that qualifies him for consideration.

This means that the XML document is valid.

There is a schema, auto-loan-application.xsd

It defines the the structure of

auto loan applications but it doesn't enforce the

business rule (the age rule). You are to add an

<assert> element that enforces the age rule.

After you've validated the XML document against the

schema, change the data and revalidate, e.g. change Johnny Brown's

age to 15.

# Lab 2. Division of Labor

In the previous lab your schema contained two kinds of things:

1. Structure rules about the form (structure) of instance documents.

2. A business rule that expressed a policy the insurance

company has regarding the age of the applicant.

The structure rules should be implemented by a technical person

, while the business rule should be implemented by

a business person.

Let's implement this division of labor.

auto-loan-application.xsd just contains structure rules.

Open this document. The design has been changed from Russian

Doll to Salami Slice.

Create a second schema, auto-loan-min-age.xsd that

expresses the business rule.

You will need to use the xs:override element. It's pretty

straightforward:

<xs:override schemaLocation="auto-loan-application.xsd">

-- In here you redefine a globally

-- declared/defined component in

-- auto-loan-application.xsd

</xs:override>

After you've created this new schema validate auto-loan-application.xml

against it.

# Lab 3. Check Election Results

In the labs\lab03 folder is election-results.xml

Open it and examine its contents. It contains

the results of an election. It shows the

percentage that each candidate received.

In this folder you will also find election-results.xsd

Open it and examine its contents. It expresses

the structure of the instance document.

For the instance document to be valid it

must have the correct structure and it

must conform to this extra rule:

The sum of all the percentages

must be 100.

Add an assertion to the schema to implement

this rule.

# Lab 4. Prohibit Reserved Words and enforce classification

In the lab folder you will find an xml document that represents a document with paragraphs. The document has a security classification. The paragraphs also have a classification.

No para element may have a classification higher than the

Document's classification. That means if the document has a classification of Secret, the paragraphs may not have Top Secret.

Add an assertion for this rule.

You must also check each Para-element for use of the

reserved words 'script' and 'function'.

Open classification.xsd and examine it. Notice

the simpleType called paraType. Add an assertion

facet that checks for 'script' and 'function'.

Finally, use schema wide attributes for the classification attribute, but make sure only the document and para elements inherit it.

# Lab 5. Conditional Appliance

Open homeAppliances.xml and examine it. Notice

that each <appliance> element has a kind attribute,

which specifies whether the appliance is a stereo

appliance or a juicer appliance. If there is an unknown type specified throw an error!

Update homeAppliances.xsd so that the <appliance> element

has a content (type) that depends on (is conditional to)

the value of the kind attribute.

# Lab 06. Open Content

Open Cellphone.xml and examine it. It contains

features of the Nokia 252 cellphone (mobile).

Update Cellphone.xsd such that:

1. The features can be listed in any order

2. New (extension) elements can be interleaved

among the features. In the sample instance

document there are two extension elements:

<new:GPS/>

<new:Camera/>