6. XML Schema (3)

In the exercises of Sheets 4 and 5, you have created XML schemas for XML documents about juicers. This sheet contains further exercises that build on the preceding ones and in which you explore more features of XML schema.[1] The schema `juicers0.xsd` and the file `juicers0.xml` represent a possible outcome of those exercises.

Hint: In all the exercises, use the Eclipse "design" view as far as possible to create your schema.

1. Sets instead of Sequences

In the last schema you defined the content of a `juicer` element as a sequence of other elements. There is no inherent reason, however, that the children of the juicer element should occur in a specified order. Change the schema so that the juicer element contains the same set of child elements but allow them to occur in any order.

Hint 1: (1) Whenever a type extends a base type, the extension elements are always appended to the base type's elements:

$$b_1, b_2, e_1, e_2$$

where b_1 and b_2 are the base type's elements and e_1, e_2 are the elements from the type that is extending the base type.
Thus, even if one makes the base type unordered and the type extending the base type unordered, the resulting set of elements will still be partially ordered as

$$\{\, b_1, b_2 \,\},\ \{\, e_1, e_2 \,\}$$

where the curly braces indicates an unordered set. We have an unordered first set followed (sequentially) by an unordered second set.

[1]The exercises are based on the labs in the XML Schema tutorial by Roger L. Costello, `http://www.xfront.com/xml-schema.html`

So, for this exercise you are to delete the `appliance` type and place its element declarations in with the `juiceAppliance` type.

Hint 2: One of the restrictions on using the <`all`> element is that the elements declared within <`all`> must have a `maxOccurs="1"`. So, remove the cost that specifies the cost in Canadian Dollars. (We will just deal in US currency.)

The file `juicers8.xml` contains `juicer` elements in arbitrary order. Validate (1) your new schema and (2) validate the `juicers8.xml` against the new schema.

2. Empty Elements

Reuse again the schema `juicers0.xsd`, which has the appliance complexType. Currently the retailer element has as its content a URI. Modify it so that its content is empty and it instead has an attribute—`href`—whose type is `anyURI`.

The file `juicers9.xml` has empty retailer elements. Validate the instance document against your schema.

3. Creating an Appliances Repository Schema

Modify the XML file and the schema that you created in the last exercise.
In all of the exercises thus far we have stored all of our schema components in a single schema. Typically, however, one will put elements and types that may be used by many schemas into a separate schema (a repository schema).
In our juicer example the appliance type is an example of something that could be used in many schemas. Hence, pull it out of `juicers.xsd` and put it into a generic schema, `appliances.xsd.` Then modify `juicers.xsd` to use the definition of appliances in `appliances.xsd`.
For this lab you are to use the same namespace for the two schemas.

4. Demonstating the Chameleon Effect

Modify the `appliances.xsd` schema that you created in Exercise 3 to have no targetNamespace.

5. Using Multiple Namespaces

Modify again the juicers file and the schemas that you created in Exercise 3. Put the `appliances.xsd` schema in its own namespace (`http://www.appliances.org`). Modify `juicers.xsd` to use the `appliances` complexType, which is now in a different namespace. You will need to modify the instance document to reflect the fact that it is using multiple namespaces.

6. Creating Lists

Modify the juicers file and the schemas that you created in Exercise 5. Assume there are multiple retailers for each juicer. Modify `juicers.xsd` to allow for a list of values for the `href` in the retailers element.

7. Declaring Key and Uniqueness Constraints

Take again `juicers0.xsd` and `juicers0.xml` as a starting point.
Add the constraints that for juicers, the `name` attribute is a key and that values of `image` are unique. Introduce errors into the XML file to check whether they are detected by the validator.