

Building Content Models

The definitions of complex types in the purchase order schema all declare sequences of elements that must appear in the instance document. The occurrence of individual elements declared in the so-called content models of these types may be optional, as indicated by a 0 value for the attribute `minOccurs` (e.g. in `comment`), or be otherwise constrained depending upon the values of `minOccurs` and `maxOccurs`. XML Schema also provides constraints that apply to groups of elements appearing in a content model. These constraints mirror those available in XML 1.0 plus some additional constraints. Note that the constraints do not apply to attributes.

XML Schema enables groups of elements to be defined and named, so that the elements can be used to build up the content models of complex types. Un-named groups of elements can also be defined, and along with elements in named groups, they can be constrained to appear in the same order (sequence) as they are declared.

To illustrate, we introduce two groups into the `PurchaseOrderType` definition from the purchase order schema so that purchase orders may contain either separate shipping and billing addresses, or a single address for those cases in which the shipper and biller are co-located:

Figure: Nested Choice and Sequence Groups

```
<xsd:complexType name="PurchaseOrderType">
  <xsd:sequence>
    <xsd:choice>
      <xsd:group ref="typens:shipAndBill"/>
      <xsd:element name="singleUSAddress" type="typens:USAddress"/>
    </xsd:choice>
    <xsd:element ref="comment" minOccurs="0"/>
    <xsd:element name="items" type="typens:Items"/>
  </xsd:sequence>
  <xsd:attribute name="orderDate" type="xsd:date"/>
</xsd:complexType>

<xsd:group name="shipAndBill">
  <xsd:sequence>
    <xsd:element name="shipTo" type="typens:USAddress"/>
    <xsd:element name="billTo" type="typens:USAddress"/>
  </xsd:sequence>
</xsd:group>
```

Such `xsd:group` references are resolved by the Importer, therefore they are not visible in the imported classes. [Figure 447](#) depicts the situation where the `PurchaseOrderType` class contains the `shipTo` and `billTo` attributes corresponding to the elements of the imported `shipAndBill` group which is not visible anymore.

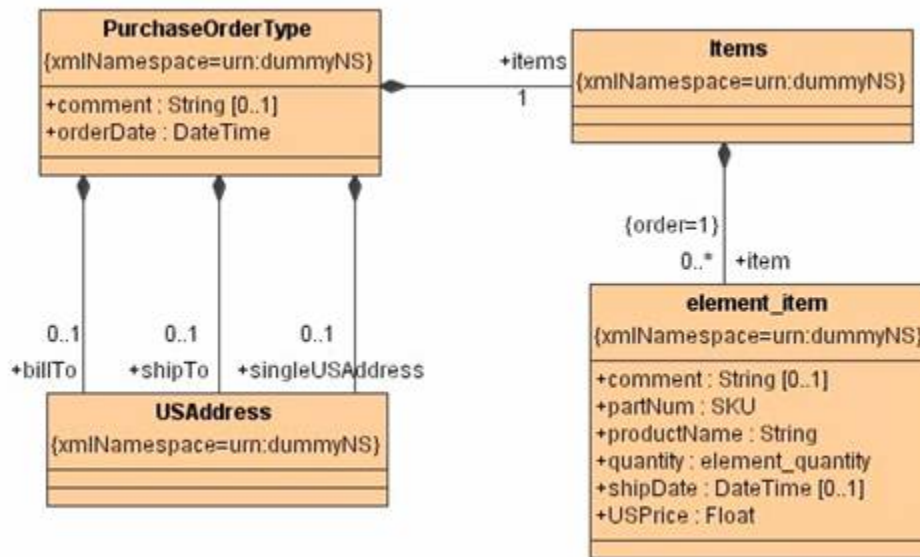
Mapping Rule: Group references like in `xsd:group` are resolved by the Importer and therefore not visible in the generated UML classes. Group members become attributes and associations of the generated UML class.

The choice group element allows only one of its children to appear in an instance.

One child is an inner group element that references the named group `shipAndBill` consisting of the element sequence `shipTo`, `billTo`, and the second child is a `singleUSAddress`. Hence, in an instance document, the `purchaseOrder` element must contain either a `shipTo` element followed by a `billTo` element or a `singleUSAddress` element. The choice group is followed by the `comment` and `items` element declarations, and both the choice group and the element declarations are children of a sequence group. The effect of these various groups is that the address element(s) must be followed by `comment` and `items` elements in that order.

The following diagram shows, how this construct gets imported into UML:

Figure: Nested Choice and Sequence Groups imported into UML



The UML class PurchaseOrderType is created in order to hold all possible instances of the XSD PurchaseOrderType. This is meant by the following

Mapping Rule: All attributes and elements found in the choice content model must be given as optional UML class attributes respectively associations.

For the above example this implies that billTo, shipTo and singleUSAddress elements must be given as optional – that is, lower multiplicity = 0 – associations. billTo and shipTo are defined using an elementGroup element which is explained in the next section. Note that there is no order tagged value, because the UML attributes and associations are not fully ordered anymore, if there is at least one choice content model present. Which leads to another rule:

Mapping Rule: All UML attributes and associations corresponding to XSD elements do have an order tagged value if and only if the type contains only [xsd:sequence](#) content model.

Note: The mapping of the content model in a class constraint is not implemented yet.

There exists a third option for constraining elements in a group: All the elements in the group may appear once or not at all, and they may appear in any order. The all group is limited to the top-level of any content model. Moreover, the group's children must all be individual elements (no groups), and no element in the content model may appear more than once, i.e. the permissible values of minOccurs and maxOccurs are 0 and 1. This content model corresponds exactly to the "content" model of UML classes meaning that:

Mapping Rule: [xsd:elements](#) of simple and complex types being member of a [xsd:all](#) content model are mapped to UML attributes respectively associations without any constraint.