

Compositors

Complex type maps to UML Class with the stereotype `XSDcomplexType`. In order to have a group in a complex type, the same UML Class must also have the `XSDall`, `XSDchoice` or `XSDsequence` stereotype.

A UML model can have a `ModelClass` with only a single stereotype `XSDall`, `XSDchoice`, or `XSDsequence`. In this case, the class maps to the inner part of another group.

Element order in a sequence group is very important. These elements are ordered according to the values of `TaggedValue sequenceOrder`.

compositors XML representation summary

```
<all
    id = ID
    maxOccurs = 1 : 1
    minOccurs = (0 | 1) : 1
    {any attributes with non-schema namespace...}>
Content: (annotation?, element*)
</all>

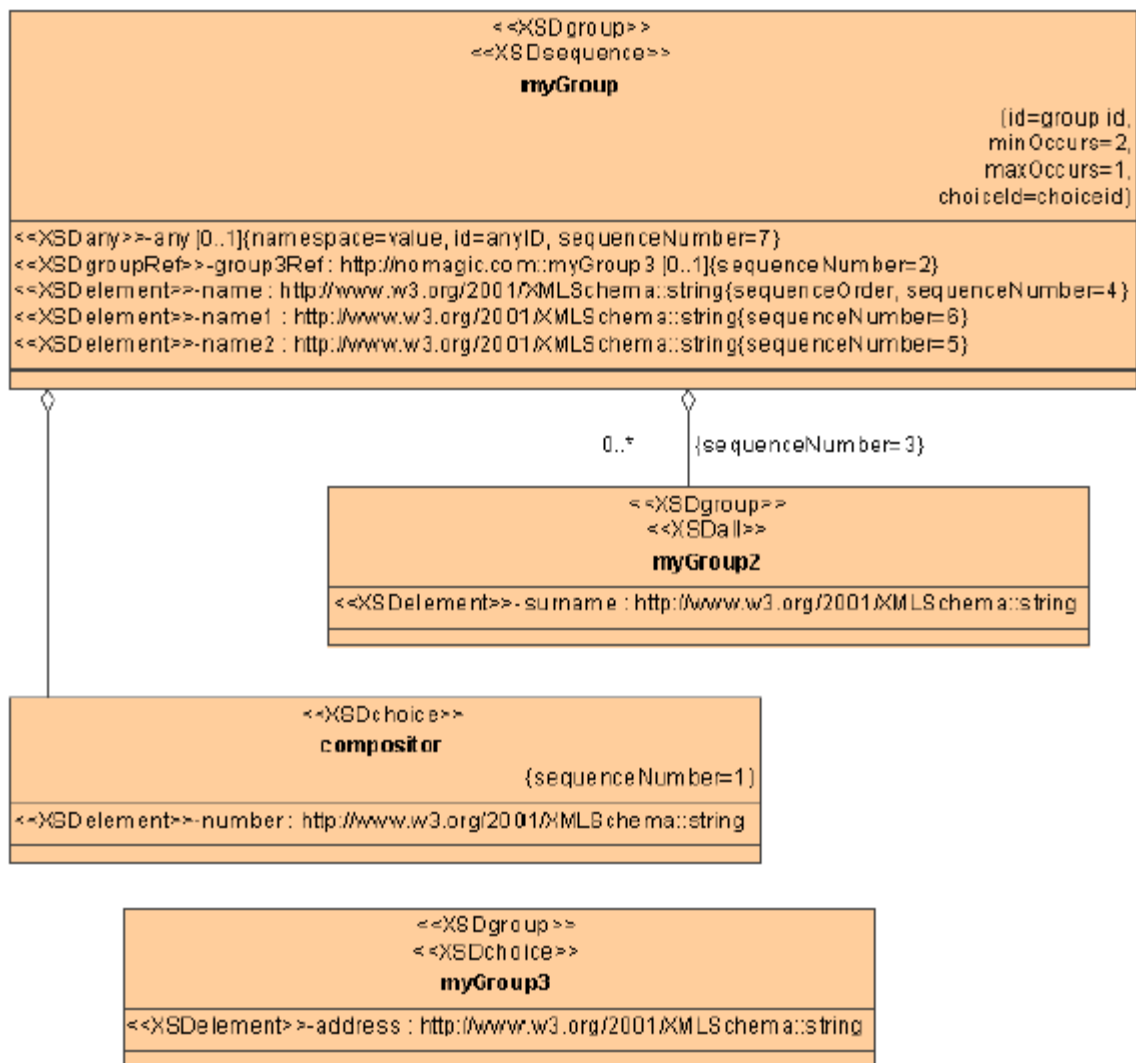
<choice
    id = ID
    maxOccurs = (nonNegativeInteger | unbounded):1
    minOccurs = nonNegativeInteger : 1
    {any attributes with non-schema namespace...}>
Content: (annotation?, (element | group | choice | sequence | any)*)
</choice>

<sequence
    id = ID
    maxOccurs = (nonNegativeInteger | unbounded) : 1
    minOccurs = nonNegativeInteger : 1
    {any attributes with non-schema namespace...}>
Content: (annotation?, (element | group | choice | sequence | any)*)
</sequence>
```

XML representations for the three kinds of model group

```
<xs:all>
    <xs:element ref="cats"/>
    <xs:element ref="dogs"/>
</xs:all>

<xs:sequence>
    <xs:choice>
        <xs:element ref="left"/>
        <xs:element ref="right"/>
    </xs:choice>
    <xs:element ref="landmark"/>
</xs:sequence>
```



compositors UML model example.

compositors XML code sample

```
<?xml version = '1.0' encoding = 'Cp1252'?>

<xs:schema xmlns:nm = "http://nomagic.com" xmlns:xs = "http://www.w3.org/2001/XMLSchema" targetNamespace = "http://nomagic.com">

    <xs:group name = "myGroup" >
        <xs:annotation>
            <xs:documentation>my group documentation</xs:documentation>
        </xs:annotation>
        <xs:sequence minOccurs = "2" maxOccurs = "1">
            <xs:choice>
                <xs:element name = "number" type = "xs:string" />
            </xs:choice>
            <xs:group ref = "nm:myGroup3" minOccurs = "0" maxOccurs = "1" >
                <xs:annotation >
                    <xs:documentation>ref documentation</xs:documentation>
                </xs:annotation>
            </xs:group>
            <xs:group ref = "nm:myGroup2" minOccurs = "0" maxOccurs = "unbounded" >
                <xs:annotation >
                    <xs:documentation >another ref documentation</xs:documentation>
                </xs:annotation>
            </xs:group>
            <xs:element name = "name" type = "xs:string" />
            <xs:element name = "name2" type = "xs:string" />
            <xs:element name = "name1" type = "xs:string" />
            <xs:any id = "anyID" namespace = "value" minOccurs = "0" maxOccurs = "1" />
        </xs:sequence>
    </xs:group>

    <xs:group name = "myGroup3" >
        <xs:choice >
            <xs:element name = "address" type = "xs:string" />
        </xs:choice>
    </xs:group>

    <xs:group name = "myGroup2" >
        <xs:all>
            <xs:element name = "surname" type = "xs:string" />
        </xs:all>
    </xs:group>
</xs:schema>
```