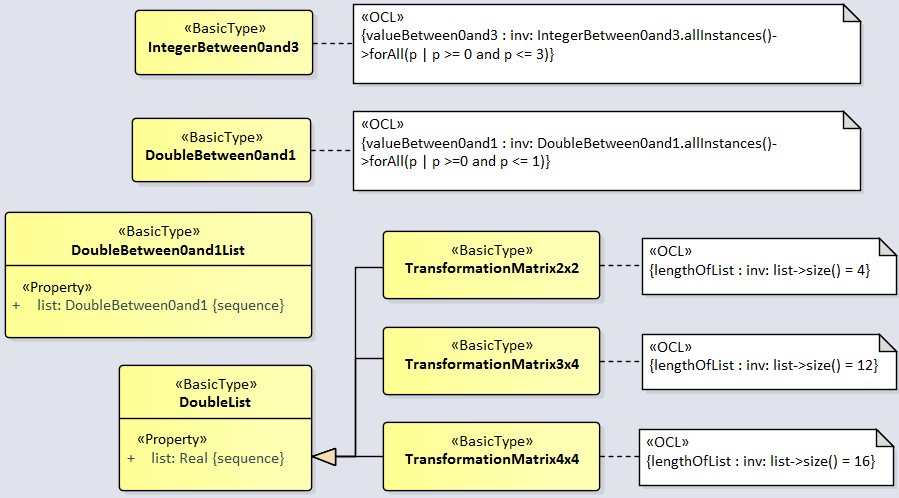
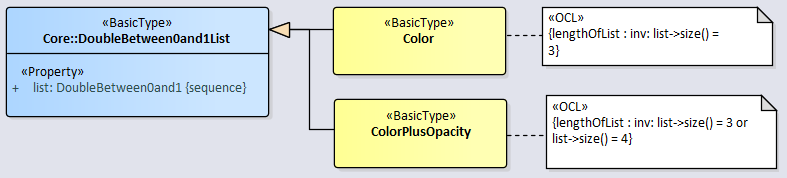
**List Representation in the CityGML 3.0 CM**

The OGC TB17 Model Driven Standards task is developing and demonstrating an automated process flow from UML-based model specifications to standards documents and platform-specific model encodings; the CityGML 3.0 CM document is one use-case being prototyped and the CityGML 3.0 XML Schema-based PSM is one of several target encodings included in that work.

In the CityGML 3.0 CM, lists of basic types are principally modeled in UML Package «ApplicationSchema» Core, as follows:



They are also modeled in UML Package «Application Schema» Appearance, as follows:



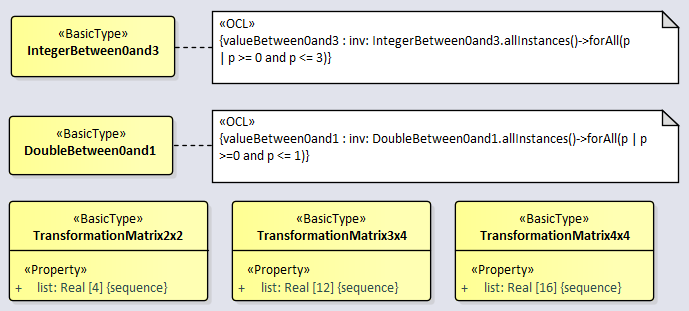
After reviewing the 3.0 CM and the ShapeChange-derived XSDs developed by Tatjana Kutzner, including the source code modifications made by Katjana to a non-current ShapeChange baseline, we arrived at the following recommendations which we have made the basis for our ongoing TB17 activities.

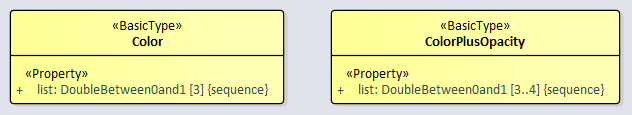
As part of prototyping and testing these recommendations the current ShapeChange software baseline (v2.10) has been extended with one new XML Schema-related rule: [rule-xsd-cls-basictype-list](https://shapechange.net/targets/xsd/extensions/#rule-xsd-cls-basictype-list) This rule is independent of the CityGML 3.0 CM, addressing the more general topic of the modeling of lists of basic types in Application Schemas for an XML Schema PSM. This new rule will be included in the v2.11.0 release of ShapeChange and is available for immediate use in the developmental branch v2.10.1-SNAPSHOT.

The semantics of this new rule are that a basic type that has a single property with maximum multiplicity greater than 1 will be encoded as a list-based XML Schema simple type. The list item type is the XSD type of the UML property value type. If the minimum multiplicity of the UML property is 0 and the maximum multiplicity is unbounded (‘\*’), then the length of the resulting list is not restricted. Otherwise, length restrictions are defined according to the multiplicity of the property.

The benefits of this rule are that a more natural approach to modeling lists in UML can be adopted, one that leverages the standard mechanisms for specifying UML properties (*e.g.,* multiplicity) and thus avoids the uses of OCL constraints in the CM simply to add restrictions on the allowed list length as well as avoids the employment of mechanisms such as Schematron for validating XML Instance documents based on such OCL-specified constraints in CMs.

The revised CM no longer requires the inclusion of “accessory” «BasicType» **DoubleBetween0and1List** and **DoubleList** generalization classes; the revised five subclasses are easier to read and understand. The design pattern is simple and clear should developers of ADEs desire to create new list-of-basic-type classes of their own. The revised CityGML 3.0 CM appears as follows, noting that the special tagged-values added by Tatjana in coordination with her revised ShapeChange software are no longer desirable and have been removed from all five list (formerly sub)classes in the CM:





Note also that it seems like a better modeling solution to have adopted the existing **ISO 19136:2007 Geography Markup Language (GML)::basicTypes::doubleList** rather than to define a new **Core::DoubleList** – which is anyway removed in this recommendation.

When the new XML Schema-related rule-xsd-cls-basictype-list is applied to this revised CM the following changes result:

1. Removed from cityGMLBase.xsd:

<simpleType name="DoubleBetween0and1ListType">

<list itemType="core:DoubleBetween0and1Type"/>

</simpleType>

1. Revised in cityGMLBase.xsd:

<simpleType name="TransformationMatrix2x2Type">

<restriction base="gml:doubleList">

<length value="4"/>

</restriction>

</simpleType>

<simpleType name="TransformationMatrix3x4Type">

<restriction base="gml:doubleList">

<length value="12"/>

</restriction>

</simpleType>

<simpleType name="TransformationMatrix4x4Type">

<restriction base="gml:doubleList">

<length value="16"/>

</restriction>

</simpleType>

Now as, respectively:

<simpleType name="TransformationMatrix2x2Type">

<restriction>

<simpleType>

<list itemType="double"/>

</simpleType>

<length value="4"/>

</restriction>

</simpleType>

<simpleType name="TransformationMatrix3x4Type">

<restriction>

<simpleType>

<list itemType="double"/>

</simpleType>

<length value="12"/>

</restriction>

</simpleType>

<simpleType name="TransformationMatrix4x4Type">

<restriction>

<simpleType>

<list itemType="double"/>

</simpleType>

<length value="16"/>

</restriction>

</simpleType>

1. Revised in appearance.xsd:

<simpleType name="ColorType">

<restriction base="core:DoubleBetween0and1ListType">

<length value="3"/>

</restriction>

</simpleType>

<simpleType name="ColorPlusOpacityType">

<restriction base="core:DoubleBetween0and1ListType">

<minLength value="3"/>

<maxLength value="4"/>

</restriction>

</simpleType>

Now as, respectively:

<simpleType name="ColorType">

<restriction>

<simpleType>

<list itemType="core:DoubleBetween0and1Type"/>

</simpleType>

<length value="3"/>

</restriction>

</simpleType>

<simpleType name="ColorPlusOpacityType">

<restriction>

<simpleType>

<list itemType="core:DoubleBetween0and1Type"/>

</simpleType>

<minLength value="3"/>

<maxLength value="4"/>

</restriction>

</simpleType>

The generated XSDs using the added rule and the ShapeChange developmental branch v2.10.1-SNAPSHOT are otherwise identical to those currently prepared by Tatjana. In the future CityGML ADE developers can use the (now maintained) ShapeChange capability to generate their community-specific XSD encodings consistent with the CityGML 3.0 CM.

----------