

# Import of the XSD Namespace

The target namespace of the each imported schema is attached to each UML class corresponding to types and elements of this schema as it is shown in the following table:

Figure: XSD target namespace and its equivalent in the UML model

<pre>&lt;xsd:schema   xmlns:xsd="http://www.w3.org/2001/XMLSchema"   xmlns:typens= "urn:my.schema.type.namespace"   targetNamespace="urn:my.schema.type.namespace"&gt; &lt;/xsd:schema&gt;</pre>	<div><div>PurchaseOrderType</div><div>{xmlns:urn:my.schema.type.namespace}</div><div><div>+orderDate : DateTime</div><div>+purchaseDate : DateTime</div></div></div>
--	--

Another possibility is to map schema target namespaces to UML packages. This is actually done because the PurchaseOrderType class is located in the package urn:my.schema.type.namespace. This package in turn is located in the Imported Types package, which illustrates one problem quite nicely: if the actual XML namespace is bound to UML packages, the serialization of these classes as XML strings depends on their location. This is not sensible all the time. Thus, the E2E Model Compiler supports both, if there is a tagged value xmlns it will use this namespace for serializing a UML class into a XML string. If this tagged value does not exist, it will take the full qualified package name as namespace. However, the Importer follows the mapping rule below.

**Mapping Rule:** The schema target namespace is stored in the xmlns tagged value for each UML class corresponding to a XSD type or element of this namespace. Furthermore, the Importer generates for each schema namespace a package of the same name to avoid name conflicts in the UML model.