Schema documentation for wss.xsd

september 11, 2012

Table of Contents

Resource hierarchy:

Legend: 🔲 *Import,* 🔲 *Include,* 🔲 *Redefine,* 💠 *Cycle detected*

wss.xsd

   ↳ 🔲 data.xsd

      ↳ 🔲 sms.xsd

   ↳ 🔲 sms.xsd

   ↳ 🔲 ana.xsd

      ↳ 🔲 data.xsd

         ↳ 🔲 sms.xsd

      ↳ 🔲 sms.xsd

Namespace: "http://www.ime.iiasa.ac.at/enrima"

Schema(s)

Main schema `wss.xsd`

Namespace	http://www.ime.iiasa.ac.at/enrima	
Properties	attribute form default:	`unqualified`
	element form default:	`unqualified`

Element(s)

Element `e:getSMSRequest`
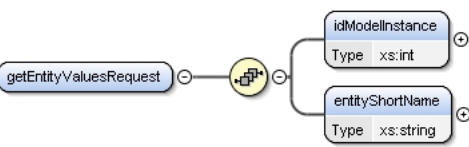
Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: `complex`
Model	ALL(idModelSpec)
Children	idModelSpec

Instance	`<e:getSMSRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">`
	  `<idModelSpec>{1,1}</idModelSpec>`
	`</e:getSMSRequest>`

Source	`<xs:element name="getSMSRequest">`
	  `<xs:complexType>`
	    `<xs:all>`
	      `<xs:element name="idModelSpec" type="xs:int" minOccurs="1" maxOccurs="1"/>`
	    `</xs:all>`
	  `</xs:complexType>`
	`</xs:element>`

Element `e:getSMSResponse`
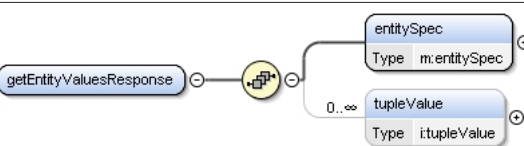
Namespace	http://www.ime.iiasa.ac.at/enrima

Diagram	
Properties	content: complex
Model	modelSpec
Children	modelSpec

Instance	`<e:getSMSResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">`
	`  <modelSpec>{1,1}</modelSpec>`
	`</e:getSMSResponse>`

Source	`<xs:element name="getSMSResponse">`
	`  <xs:complexType>`
	`    <xs:sequence>`
	`      <xs:element name="modelSpec" type="m:modelSpec"/>`
	`    </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

Element `e:getEntityValuesRequest`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	idModelInstance , entityShortName
Children	entityShortName, idModelInstance

Instance	`<e:getEntityValuesRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">`
	`  <idModelInstance>{1,1}</idModelInstance>`
	`  <entityShortName>{1,1}</entityShortName>`
	`</e:getEntityValuesRequest>`

Source	`<xs:element name="getEntityValuesRequest">`
	`  <xs:complexType>`
	`    <xs:sequence>`
	`      <xs:element name="idModelInstance" type="xs:int" minOccurs="1" maxOccurs="1"/>`
	`      <xs:element name="entityShortName" type="xs:string" minOccurs="1" maxOccurs="1"/>`
	`    </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

Element `e:getEntityValuesResponse`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	entitySpec , tupleValue*
Children	entitySpec, tupleValue

Instance	`<e:getEntityValuesResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">`
	`  <entitySpec>{1,1}</entitySpec>`
	`  <tupleValue>{0,unbounded}</tupleValue>`
	`</e:getEntityValuesResponse>`

Source	`<xs:element name="getEntityValuesResponse">`
	`  <xs:complexType>`
	`    <xs:sequence>`
	`      <xs:element name="entitySpec" type="m:entitySpec" minOccurs="1" maxOccurs="1"/>`
	`      <xs:element name="tupleValue" type="i:tupleValue" minOccurs="0" maxOccurs="unbounded"/>`
	`    </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

Element `e:getSetMembersRequest`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	idModelInstance , setShortName , tupleMember*
Children	idModelInstance, setShortName, tupleMember

Instance	```<e:getSetMembersRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">```
	 ``` <idModelInstance>{1,1}</idModelInstance>```
	 ``` <setShortName>{1,1}</setShortName>```
	 ``` <tupleMember>{0,unbounded}</tupleMember>```
	 ```</e:getSetMembersRequest>```

Source	```<xs:element name="getSetMembersRequest">```
	 ``` <xs:complexType>```
	 ``` <xs:sequence>```
	 ``` <xs:element name="idModelInstance" type="xs:int" minOccurs="1" maxOccurs="1"/>```
	 ``` <xs:element name="setShortName" type="xs:string" minOccurs="1" maxOccurs="1"/>```
	 ``` <xs:element name="tupleMember" type="i:memberDic" minOccurs="0" maxOccurs="unbounded"/>```
	 ``` </xs:sequence>```
	 ``` </xs:complexType>```
	 ```</xs:element>```

Element `e:getSetMembersResponse`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	member+
Children	member

Instance	```<e:getSetMembersResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">```
	 ``` <member>{1,unbounded}</member>```
	 ```</e:getSetMembersResponse>```

Source	```<xs:element name="getSetMembersResponse">```
	 ``` <xs:complexType>```
	 ``` <xs:sequence>```
	 ``` <xs:element name="member" type="i:memberDic" maxOccurs="unbounded"/>```
	 ``` </xs:sequence>```
	 ``` </xs:complexType>```
	 ```</xs:element>```

Element `e:storeSetMembersRequest`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	idModeldata , idSetSpec , setMembers+
Children	idModeldata, idSetSpec, setMembers
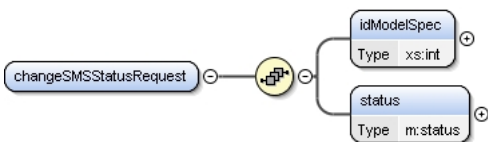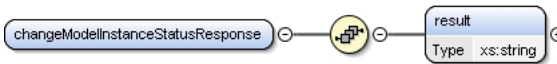
| Instance | ```
<e:storeSetMembersRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">
  <idModeldata>{1,1}</idModeldata>
  <idSetSpec>{1,1}</idSetSpec>
  <setMembers>{1,unbounded}</setMembers>
</e:storeSetMembersRequest>
``` |
|---|---|
| Source | ```
<xs:element name="storeSetMembersRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="idModeldata" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="idSetSpec" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="setMembers" type="i:setMembers" minOccurs="1" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
``` |
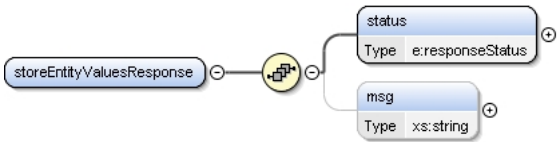
Element `e:storeSetMembersResponse`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	status , msg{0,1}
Children	msg, status
| Instance | ```
<e:storeSetMembersResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">
  <status>{1,1}</status>
  <msg>{0,1}</msg>
</e:storeSetMembersResponse>
``` |
| Source | ```
<xs:element name="storeSetMembersResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="status" type="e:responseStatus" minOccurs="1" maxOccurs="1"/>
      <xs:element name="msg" type="xs:string" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
``` |

Element `e:storeEntityValuesRequest`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	idModeldata , tupleValue*
Children	idModeldata, tupleValue
| Instance | ```
<e:storeEntityValuesRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">
  <idModeldata>{1,1}</idModeldata>
  <tupleValue>{0,unbounded}</tupleValue>
</e:storeEntityValuesRequest>
``` |
| Source | ```
<xs:element name="storeEntityValuesRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="idModeldata" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="tupleValue" type="i:tupleValue" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
``` |

Element `e:storeEntityValuesResponse`

Namespace	http://www.ime.iiasa.ac.at/enrima

Diagram	
Properties	content: complex
Model	status , msg{0,1}
Children	msg, status
| Instance | ```xml
<e:storeEntityValuesResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">
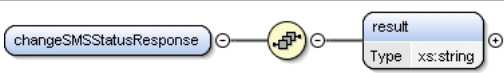  <status>{1,1}</status>
  <msg>{0,1}</msg>
</e:storeEntityValuesResponse>
``` |
| Source | ```xml
<xs:element name="storeEntityValuesResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="status" type="e:responseStatus" minOccurs="1" maxOccurs="1"/>
      <xs:element name="msg" type="xs:string" minOccurs="0" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
``` |

Element e:changeSMSStatusRequest

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	idModelSpec , status
Children	idModelSpec, status
| Instance | ```xml
<e:changeSMSStatusRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">
  <idModelSpec>{1,1}</idModelSpec>
  <status>{1,1}</status>
</e:changeSMSStatusRequest>
``` |
| Source | ```xml
<xs:element name="changeSMSStatusRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="idModelSpec" type="xs:int" minOccurs="1" maxOccurs="1"/>
      <xs:element name="status" type="m:status" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
``` |
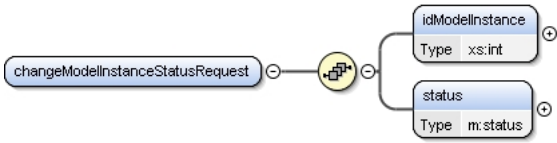
Element e:changeSMSStatusResponse

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	result
Children	result
| Instance | ```xml
<e:changeSMSStatusResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">
  <result>{1,1}</result>
</e:changeSMSStatusResponse>
``` |
| Source | ```xml
<xs:element name="changeSMSStatusResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="result" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
``` |

Element e:changeModelInstanceStatusRequest

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	idModelInstance , status
Children	idModelInstance, status

Instance	```xml
	<e:changeModelInstanceStatusRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">
	  <idModelInstance>{1,1}</idModelInstance>
	  <status>{1,1}</status>
	</e:changeModelInstanceStatusRequest>
	``

Source	```xml
	<xs:element name="changeModelInstanceStatusRequest">
	  <xs:complexType>
	    <xs:sequence>
	      <xs:element name="idModelInstance" type="xs:int" minOccurs="1" maxOccurs="1"/>
	      <xs:element name="status" type="m:status" minOccurs="1" maxOccurs="1"/>
	    </xs:sequence>
	  </xs:complexType>
	</xs:element>
	``

Element e:changeModelInstanceStatusResponse

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	result
Children	result

Instance	```xml
	<e:changeModelInstanceStatusResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">
	  <result>{1,1}</result>
	</e:changeModelInstanceStatusResponse>
	``

Source	```xml
	<xs:element name="changeModelInstanceStatusResponse">
	  <xs:complexType>
	    <xs:sequence>
	      <xs:element name="result" type="xs:string" minOccurs="1" maxOccurs="1"/>
	    </xs:sequence>
	  </xs:complexType>
	</xs:element>
	``

Element e:defineInstanceRequest

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	modelInstance
Children	modelInstance

Instance	```xml
	<e:defineInstanceRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">
	  <modelInstance>{1,1}</modelInstance>
	</e:defineInstanceRequest>
	``

Source	```xml
	<xs:element name="defineInstanceRequest">
	  <xs:complexType>
	    <xs:sequence>
	      <xs:element name="modelInstance" type="i:modelInstance" minOccurs="1" maxOccurs="1"/>
	    </xs:sequence>
	  </xs:complexType>
	</xs:element>
	``

Element `e:defineInstanceResponse`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: `complex`
Model	result
Children	result

Instance	```xml
	<e:defineInstanceResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">
	  <result>{1,1}</result>
	</e:defineInstanceResponse>
	``

Source	```xml
	<xs:element name="defineInstanceResponse">
	  <xs:complexType>
	    <xs:sequence>
	      <xs:element name="result" type="xs:string" minOccurs="1" maxOccurs="1"/>
	    </xs:sequence>
	  </xs:complexType>
	</xs:element>
	``

Element `e:getDataSummaryRequest`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: `complex`
Model	idModelData
Children	idModelData

Instance	```xml
	<e:getDataSummaryRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">
	  <idModelData>{1,1}</idModelData>
	</e:getDataSummaryRequest>
	``

Source	```xml
	<xs:element name="getDataSummaryRequest">
	  <xs:complexType>
	    <xs:sequence>
	      <xs:element name="idModelData" type="xs:int" minOccurs="1" maxOccurs="1"/>
	    </xs:sequence>
	  </xs:complexType>
	</xs:element>
	``

Element `e:getDataSummaryResponse`

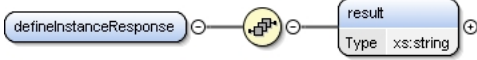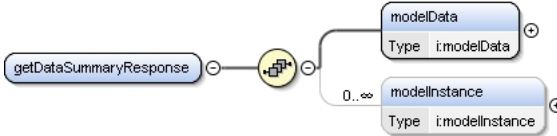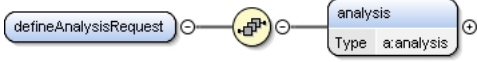|	
Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: `complex`
Model	modelData , modelInstance*
Children	modelData, modelInstance

Instance	```xml
	<e:getDataSummaryResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">
	  <modelData>{1,1}</modelData>
	  <modelInstance>{0,unbounded}</modelInstance>
	</e:getDataSummaryResponse>
	``

Source	```xml
	<xs:element name="getDataSummaryResponse">
	  <xs:complexType>
	    <xs:sequence>
	      <xs:element name="modelData" type="i:modelData" minOccurs="1" maxOccurs="1"/>
	      <xs:element name="modelInstance" type="i:modelInstance" minOccurs="0" maxOccurs="unbounded"/>
	    </xs:sequence>
	  </xs:complexType>
	</xs:element>
	``

Element `e:defineAnalysisRequest`

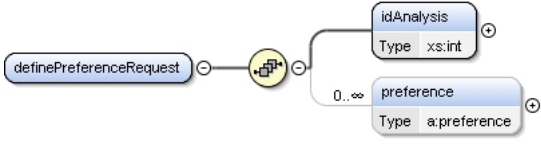| Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	analysis
Children	analysis

Instance	`<e:defineAnalysisRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">`
	`  <analysis>{1,1}</analysis>`
	`</e:defineAnalysisRequest>`

Source	`<xs:element name="defineAnalysisRequest">`
	`  <xs:complexType>`
	`    <xs:sequence>`
	`      <xs:element name="analysis" type="a:analysis" minOccurs="1" maxOccurs="1"/>`
	`    </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

Element `e:defineAnalysisResponse`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	result
Children	result

Instance	`<e:defineAnalysisResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">`
	`  <result>{1,1}</result>`
	`</e:defineAnalysisResponse>`

Source	`<xs:element name="defineAnalysisResponse">`
	`  <xs:complexType>`
	`    <xs:sequence>`
	`      <xs:element name="result" type="xs:string" minOccurs="1" maxOccurs="1"/>`
	`    </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

Element `e:definePreferenceRequest`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	idAnalysis , preference*
Children	idAnalysis, preference

Instance	`<e:definePreferenceRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">`
	`  <idAnalysis>{1,1}</idAnalysis>`
	`  <preference>{0,unbounded}</preference>`
	`</e:definePreferenceRequest>`

Source	`<xs:element name="definePreferenceRequest">`
	`  <xs:complexType>`
	`    <xs:sequence>`
	`      <xs:element name="idAnalysis" type="xs:int" minOccurs="1" maxOccurs="1"/>`
	`      <xs:element name="preference" type="a:preference" minOccurs="0" maxOccurs="unbounded"/>`
	`    </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

Element `e:definePreferenceResponse`

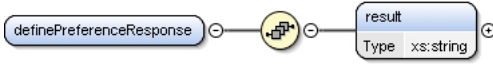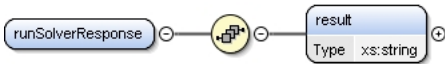|	
Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	result
Children	result

Instance	`<e:definePreferenceResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">`
	  `<result>{1,1}</result>`
	`</e:definePreferenceResponse>`

Source	`<xs:element name="definePreferenceResponse">`
	  `<xs:complexType>`
	    `<xs:sequence>`
	      `<xs:element name="result" type="xs:string" minOccurs="1" maxOccurs="1"/>`
	    `</xs:sequence>`
	  `</xs:complexType>`
	`</xs:element>`

Element `e:runSolverRequest`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	idAnalysis
Children	idAnalysis

Instance	`<e:runSolverRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">`
	  `<idAnalysis>{1,1}</idAnalysis>`
	`</e:runSolverRequest>`

Source	`<xs:element name="runSolverRequest">`
	  `<xs:complexType>`
	    `<xs:sequence>`
	      `<xs:element name="idAnalysis" type="xs:int" minOccurs="1" maxOccurs="1"/>`
	    `</xs:sequence>`
	  `</xs:complexType>`
	`</xs:element>`

Element `e:runSolverResponse`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	result
Children	result

Instance	`<e:runSolverResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">`
	  `<result>{1,1}</result>`
	`</e:runSolverResponse>`

Source	`<xs:element name="runSolverResponse">`
	  `<xs:complexType>`
	    `<xs:sequence>`
	      `<xs:element name="result" type="xs:string" minOccurs="1" maxOccurs="1"/>`
	    `</xs:sequence>`
	  `</xs:complexType>`
	`</xs:element>`

Element `e:runScenarioRequest`

Namespace	http://www.ime.iiasa.ac.at/enrima

Diagram	
Properties	content: complex
Model	idAnalysis
Children	idAnalysis
| Instance | ```xml
<e:runScenarioRequest xmlns:e="http://www.ime.iiasa.ac.at/enrima">
  <idAnalysis>{1,1}</idAnalysis>
</e:runScenarioRequest>
``` |
| Source | ```xml
<xs:element name="runScenarioRequest">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="idAnalysis" type="xs:int" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
``` |

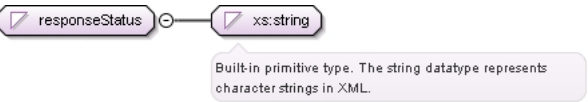## Element `e:runScenarioResponse`

Namespace	http://www.ime.iiasa.ac.at/enrima
Diagram	
Properties	content: complex
Model	result
Children	result
| Instance | ```xml
<e:runScenarioResponse xmlns:e="http://www.ime.iiasa.ac.at/enrima">
  <result>{1,1}</result>
</e:runScenarioResponse>
``` |
| Source | ```xml
<xs:element name="runScenarioResponse">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="result" type="xs:string" minOccurs="1" maxOccurs="1"/>
    </xs:sequence>
  </xs:complexType>
</xs:element>
``` |

Simple Type(s)

Simple Type `e:responseStatus`

Namespace	http://www.ime.iiasa.ac.at/enrima	
Diagram	 Built-in primitive type. The string datatype represents character strings in XML.	
Type	restriction of xs:string	
Facets	enumeration	OK
	enumeration	INVALID_ITERATOR
	enumeration	INVALID_SET_MEMBER
	enumeration	INVALID_VALUE
	enumeration	BOUND_VIOLATION
Used by	Elements	e:storeEntityValuesResponse/status, e:storeSetMembersResponse/status
| Source | ```xml
<xs:simpleType name="responseStatus">
  <xs:restriction base="xs:string">
    <xs:enumeration value="OK"/>
    <xs:enumeration value="INVALID_ITERATOR"/>
    <xs:enumeration value="INVALID_SET_MEMBER"/>
    <xs:enumeration value="INVALID_VALUE"/>
    <xs:enumeration value="BOUND_VIOLATION"/>
  </xs:restriction>
</xs:simpleType>
``` | |

Namespace: "http://www.ime.iiasa.ac.at/model/instance"

Schema(s)

Imported schema `data.xsd`

Namespace	http://www.ime.iiasa.ac.at/model/instance	
Properties	attribute form default:	`unqualified`
	element form default:	`unqualified`

Element(s)

Element `i:dataSet`

Namespace	http://www.ime.iiasa.ac.at/model/instance
Annotations	`only for mockup`
Diagram	
Properties	content: `complex`
Model	setMembers+ , entityValues+
Children	entityValues, setMembers

Instance	`<i:dataSet xmlns:i="http://www.ime.iiasa.ac.at/model/instance">`
	`  <setMembers>{1,unbounded}</setMembers>`
	`  <entityValues>{1,unbounded}</entityValues>`
	`</i:dataSet>`

Source	`<xs:element name="dataSet">`
	`  <xs:annotation>`
	`    <xs:documentation>only for mockup</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`    <xs:sequence>`
	`      <xs:element name="setMembers" type="i:setMembers" minOccurs="1" maxOccurs="unbounded"/>`
	`      <xs:element name="entityValues" type="i:entityValues" minOccurs="1" maxOccurs="unbounded"/>`
	`    </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

Complex Type(s)

Complex Type `i:tupleValue`

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Used by	Elements e:getEntityValuesResponse/tupleValue, e:storeEntityValuesRequest/tupleValue
Model	tupleMember* , value
Children	tupleMember, value

Source	`<xs:complexType name="tupleValue">`
	`  <xs:sequence>`
	`    <xs:element name="tupleMember" type="i:memberDic" minOccurs="0" maxOccurs="unbounded"/>`
	`    <xs:element name="value" type="xs:double"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `i:memberDic`

Namespace	http://www.ime.iiasa.ac.at/model/instance

Diagram	
Used by	Elements e:getSetMembersRequest/tupleMember, e:getSetMembersResponse/member, i:tupleValue/tupleMember
Model	id , code , description{0,1}
Children	code, description, id
| Source | ```xml
<xs:complexType name="memberDic">
  <xs:sequence>
    <xs:element name="id" type="xs:int"/>
    <xs:element name="code" type="xs:string"/>
    <xs:element name="description" type="m:description" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
``` |

Complex Type `i:setMembers`

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Used by	Elements e:storeSetMembersRequest/setMembers, i:dataSet/setMembers
Model	setShortName , tupleMembers*
Children	setShortName, tupleMembers
| Source | ```xml
<xs:complexType name="setMembers">
  <xs:sequence>
    <xs:element name="setShortName" type="xs:string"/>
    <xs:element name="tupleMembers" type="i:tupleMembers" minOccurs="0" maxOccurs="unbounded"/>
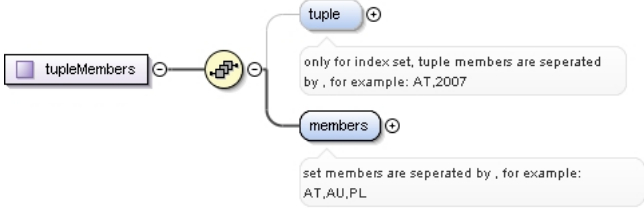  </xs:sequence>
</xs:complexType>
``` |

Complex Type `i:tupleMembers`

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Used by	Element i:setMembers/tupleMembers
Model	tuple{0,1} , members
Children	members, tuple
| Source | ```xml
<xs:complexType name="tupleMembers">
  <xs:sequence>
    <xs:element name="tuple" type="xs:string" minOccurs="0" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>only for index set, tuple members are seperated by , for example: AT,2007</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="members" type="xs:string" minOccurs="1" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>set members are seperated by , for example: AT,AU,PL</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
``` |

Complex Type `i:modelInstance`

Namespace	http://www.ime.iiasa.ac.at/model/instance

Diagram	
Used by	**Elements** e:defineInstanceRequest/modelInstance, e:getDataSummaryResponse/modelInstance
Model	id , idModelSpec , idModelData , shortName{0,1} , name{0,1} , description{0,1} , status
Children	description, id, idModelData, idModelSpec, name, shortName, status
| Source | ```xml
<xs:complexType name="modelInstance">
  <xs:sequence>
    <xs:element name="id" type="xs:int"/>
    <xs:element name="idModelSpec" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="idModelData" type="xs:int" minOccurs="1" maxOccurs="1"/>
    <xs:element name="shortName" type="m:shortName" minOccurs="0" maxOccurs="1"/>
    <xs:element name="name" type="m:name" minOccurs="0" maxOccurs="1"/>
    <xs:element name="description" type="m:description" minOccurs="0" maxOccurs="1"/>
    <xs:element name="status" type="m:status" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
``` |

Complex Type `i:modelData`

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Used by	**Element** e:getDataSummaryResponse/modelData
Model	id , idModelSpec+ , idParent{0,1} , description{0,1} , locked
Children	description, id, idModelSpec, idParent, locked
| Source | ```xml
<xs:complexType name="modelData">
  <xs:sequence>
    <xs:element name="id" type="xs:int"/>
    <xs:element name="idModelSpec" type="xs:int" minOccurs="1" maxOccurs="unbounded"/>
    <xs:element name="idParent" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="description" type="m:description" minOccurs="0" maxOccurs="1"/>
    <xs:element name="locked" type="xs:boolean" minOccurs="1" maxOccurs="1"/>
  </xs:sequence>
</xs:complexType>
``` |

Complex Type `i:entityValues`

Namespace	http://www.ime.iiasa.ac.at/model/instance
Annotations	only for mockup
Diagram	
Used by	**Element** i:dataSet/entityValues
Model	entityShortName , tupleValues*
Children	entityShortName, tupleValues
| Source | ```xml
<xs:complexType name="entityValues">
``` |

```
    <xs:annotation>
      <xs:documentation>only for mockup</xs:documentation>
    </xs:annotation>
    <xs:sequence>
      <xs:element name="entityShortName" type="xs:string"/>
      <xs:element name="tupleValues" type="i:tupleValues" minOccurs="0" maxOccurs="unbounded"/>
    </xs:sequence>
</xs:complexType>
```

Complex Type `i:tupleValues`

Namespace	http://www.ime.iiasa.ac.at/model/instance
Annotations	only for mockup
Diagram	
Used by	Element i:entityValues/tupleValues
Model	tuple{0,1} , value
Children	tuple, value

Source	`<xs:complexType name="tupleValues">`
	`  <xs:annotation>`
	`    <xs:documentation>only for mockup</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`    <xs:element name="tuple" type="xs:string" minOccurs="0" maxOccurs="1">`
	`      <xs:annotation>`
	`        <xs:documentation>tuple members are seperated by , for example: AT,2007</xs:documentation>`
	`      </xs:annotation>`
	`    </xs:element>`
	`    <xs:element name="value" type="xs:double"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Namespace: "http://www.ime.iiasa.ac.at/model/spec"

Schema(s)

Imported schema `sms.xsd`

Namespace	http://www.ime.iiasa.ac.at/model/spec	
Properties	attribute form default:	unqualified
	element form default:	unqualified

Complex Type(s)

Complex Type `m:modelSpec`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Used by	Element e:getSMSResponse/modelSpec

Model	id , shortName , name{0,1} , description{0,1} , status , setSpec+ , entitySpec+
Children	description, entitySpec, id, name, setSpec, shortName, status

Source	```xml
	<xs:complexType name="modelSpec">
	  <xs:sequence>
	    <xs:element name="id" type="xs:int"/>
	    <xs:element name="shortName" type="m:shortName"/>
	    <xs:element name="name" type="m:name" minOccurs="0" maxOccurs="1"/>
	    <xs:element name="description" type="m:description" minOccurs="0" maxOccurs="1"/>
	    <xs:element name="status" type="m:status"/>
	    <xs:element name="setSpec" type="m:setSpec" maxOccurs="unbounded"/>
	    <xs:element name="entitySpec" type="m:entitySpec" maxOccurs="unbounded"/>
	  </xs:sequence>
	</xs:complexType>
	``

Complex Type `m:setSpec`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Used by	Element m:modelSpec/setSpec
Model	id , idParent , shortName , name{0,1} , description{0,1} , type , idx{0,1} , iteratorContainer{0,1}
Children	description, id, idParent, idx, iteratorContainer, name, shortName, type

Source	```xml
	<xs:complexType name="setSpec">
	  <xs:sequence>
	    <xs:element name="id" type="xs:int"/>
	    <!--sub set-->
	    <xs:element name="idParent" type="xs:int"/>
	    <xs:element name="shortName" type="m:shortName"/>
	    <xs:element name="name" type="m:name" minOccurs="0" maxOccurs="1"/>
	    <xs:element name="description" type="m:description" minOccurs="0" maxOccurs="1"/>
	    <xs:element name="type" type="m:setSpecType"/>
	    <xs:element name="idx" type="xs:string" minOccurs="0" maxOccurs="1">
	      <xs:annotation>
	        <xs:documentation>default index name (for main set only)</xs:documentation>
	      </xs:annotation>
	    </xs:element>
	    <xs:element name="iteratorContainer" type="m:iteratorContainer" minOccurs="0" maxOccurs="1">
	      <xs:annotation>
	        <xs:documentation>for index set only</xs:documentation>
	      </xs:annotation>
	    </xs:element>
	  </xs:sequence>
	</xs:complexType>
	``

Complex Type `m:iteratorContainer`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Used by	Elements m:entitySpec/iteratorContainer, m:setSpec/iteratorContainer
Model	idSetSpec+
Children	idSetSpec

| Source | ```xml
<xs:complexType name="iteratorContainer">
  <xs:sequence>
    <xs:element name="idSetSpec" type="xs:int" minOccurs="1" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
``` |

Complex Type `m:entitySpec`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Used by	Elements e:getEntityValuesResponse/entitySpec, m:modelSpec/entitySpec
Model	id , shortName , name{0,1} , description{0,1} , idLowerBound{0,1} , idUpperBound{0,1} , constantValue{0,1} , iteratorContainer{0,1} , role{0,1} , mathType{0,1} , unit{0,1} , formula{0,1} , source{0,1} , group*
Children	constantValue, description, formula, group, id, idLowerBound, idUpperBound, iteratorContainer, mathType, name, role, shortName, source, unit
| Source | ```xml
<xs:complexType name="entitySpec">
  <xs:sequence>
    <xs:element name="id" type="xs:int"/>
    <xs:element name="shortName" type="m:shortName"/>
    <xs:element name="name" type="m:name" minOccurs="0" maxOccurs="1"/>
    <xs:element name="description" type="m:description" minOccurs="0" maxOccurs="1"/>
    <xs:element name="idLowerBound" type="xs:int" minOccurs="0" maxOccurs="1"/>
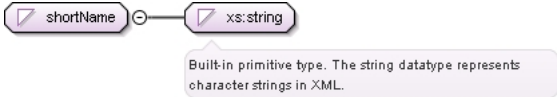    <xs:element name="idUpperBound" type="xs:int" minOccurs="0" maxOccurs="1"/>
    <xs:element name="constantValue" type="xs:double" minOccurs="0" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>optional, only for constant type of entity</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="iteratorContainer" type="m:iteratorContainer" minOccurs="0" maxOccurs="1"/>
    <xs:element name="role" type="m:entityRole" minOccurs="0" maxOccurs="1"/>
    <xs:element name="mathType" type="m:mathType" minOccurs="0" maxOccurs="1"/>
    <xs:element name="unit" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="formula" type="xs:string" minOccurs="0" maxOccurs="1"/>
    <xs:element name="source" type="xs:string" minOccurs="0" maxOccurs="1">
      <xs:annotation>
        <xs:documentation>specification of the data source (optional, but very useful for organizing data management processes</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="group" type="xs:string" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
``` |

```
            <xs:documentation>for grouping parameters (optional, but very useful for management of the
 data access rights).</xs:documentation>
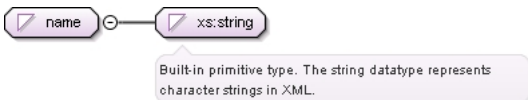        </xs:annotation>
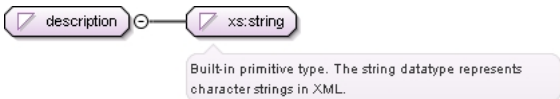      </xs:element>
    </xs:sequence>
</xs:complexType>
```

Simple Type(s)

Simple Type `m:shortName`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Type	restriction of xs:string
Facets	minLength 1 maxLength 8
Used by	Elements a:analysis/shortName, i:modelInstance/shortName, m:entitySpec/shortName, m:modelSpec/shortName, m:setSpec/shortName

Source	`<xs:simpleType name="shortName">`
	  `<xs:restriction base="xs:string">`
	    `<xs:minLength value="1"/>`
	    `<xs:maxLength value="8"/>`
	  `</xs:restriction>`
	`</xs:simpleType>`

Simple Type `m:name`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Type	restriction of xs:string
Facets	minLength 1 maxLength 32
Used by	Elements a:analysis/name, a:task/name, i:modelInstance/name, m:entitySpec/name, m:modelSpec/name, m:setSpec/name

Source	`<xs:simpleType name="name">`
	  `<xs:restriction base="xs:string">`
	    `<xs:minLength value="1"/>`
	    `<xs:maxLength value="32"/>`
	  `</xs:restriction>`
	`</xs:simpleType>`

Simple Type `m:description`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Type	restriction of xs:string
Facets	minLength 1 maxLength 512
Used by	Elements a:analysis/description, a:task/description, i:memberDic/description, i:modelData/description, i:modelInstance/description, m:entitySpec/description, m:modelSpec/description, m:setSpec/description
Source	`<xs:simpleType name="description">` `<xs:restriction base="xs:string">`

```
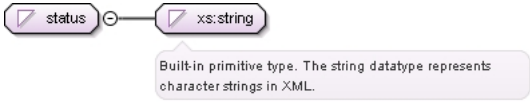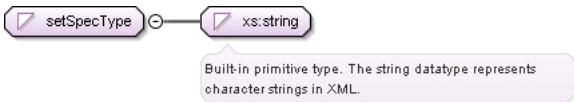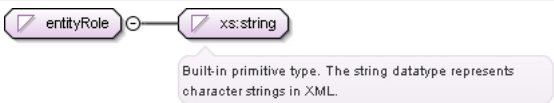        <xs:minLength value="1"/>
        <xs:maxLength value="512"/>
    </xs:restriction>
</xs:simpleType>
```

Simple Type `m:status`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Type	restriction of xs:string
Facets	enumeration EDIT
	enumeration TEST
	enumeration COMMITTED
Used by	Elements e:changeModelInstanceStatusRequest/status, e:changeSMSStatusRequest/status, i:modelInstance/status, m:modelSpec/status

Source	`<xs:simpleType name="status">`
	`  <xs:restriction base="xs:string">`
	`    <xs:enumeration value="EDIT"/>`
	`    <xs:enumeration value="TEST"/>`
	`    <xs:enumeration value="COMMITTED"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `m:setSpecType`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Type	restriction of xs:string
Facets	enumeration COLLECTION
	enumeration SEQUENCE
Used by	Element m:setSpec/type

Source	`<xs:simpleType name="setSpecType">`
	`  <xs:restriction base="xs:string">`
	`    <xs:enumeration value="COLLECTION"/>`
	`    <xs:enumeration value="SEQUENCE"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `m:entityRole`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Type	restriction of xs:string
Facets	enumeration CONSTANT
	enumeration PARAMETER
	enumeration DECISION_VAR
	enumeration EXTERNAL_DECISION_VAR
	enumeration OUTCOME_VAR
	enumeration AUXILIARY_VAR
	enumeration ASSIGNMENT

enumeration	CONSTRAINT		
enumeration	A_VARIABLE		any type of variable: DECISION_VAR, EXTERNAL_DECISION_VAR,OUTCOME_VAR,or AUXILIARY_VAR
enumeration	A_FORMULA		any type of formula: ASSIGNMENT or CONSTRAINT

Used by	Element	m:entitySpec/role

Source	
| | ```xml
<xs:simpleType name="entityRole">
  <xs:restriction base="xs:string">
    <xs:enumeration value="CONSTANT"/>
    <xs:enumeration value="PARAMETER"/>
    <xs:enumeration value="DECISION_VAR"/>
    <xs:enumeration value="EXTERNAL_DECISION_VAR"/>
    <xs:enumeration value="OUTCOME_VAR"/>
    <xs:enumeration value="AUXILIARY_VAR"/>
    <xs:enumeration value="ASSIGNMENT"/>
    <xs:enumeration value="CONSTRAINT"/>
    <xs:enumeration value="A_VARIABLE">
      <xs:annotation>
        <xs:documentation>any type of variable: DECISION_VAR, EXTERNAL_DECISION_VAR,OUTCOME_VAR,or
 AUXILIARY_VAR</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="A_FORMULA">
      <xs:annotation>
        <xs:documentation>any type of formula: ASSIGNMENT or CONSTRAINT</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
``` |
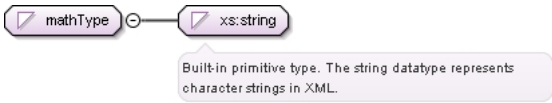
Simple Type `m:mathType`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	Built-in primitive type. The string datatype represents character strings in XML.
Type	restriction of xs:string

Facets	enumeration	INTEGER
	enumeration	BINARY
	enumeration	REAL

Used by	Element	m:entitySpec/mathType

Source	
| | ```xml
<xs:simpleType name="mathType">
  <xs:restriction base="xs:string">
    <xs:enumeration value="INTEGER"/>
    <xs:enumeration value="BINARY"/>
    <xs:enumeration value="REAL"/>
  </xs:restriction>
</xs:simpleType>
``` |

Namespace: "http://www.ime.iiasa.ac.at/model/analysis"

Schema(s)

Imported schema `ana.xsd`

Namespace	http://www.ime.iiasa.ac.at/model/analysis
Properties	attribute form default: unqualified
	element form default: unqualified

Complex Type(s)

Complex Type `a:analysis`

Namespace	http://www.ime.iiasa.ac.at/model/analysis

Diagram	
Used by	Element e:defineAnalysisRequest/analysis
Model	id , idModelInstance , shortName , name{0,1} , description{0,1}
Children	description, id, idModelInstance, name, shortName

Source	<pre><xs:complexType name="analysis">
	  <xs:sequence>
	    <xs:element name="id" type="xs:int"/>
	    <xs:element name="idModelInstance" type="xs:int" minOccurs="1" maxOccurs="1"/>
	    <xs:element name="shortName" type="m:shortName"/>
	    <xs:element name="name" type="m:name" minOccurs="0" maxOccurs="1"/>
	    <xs:element name="description" type="m:description" minOccurs="0" maxOccurs="1"/>
	  </xs:sequence>
	</xs:complexType></pre>

Complex Type `a:preference`

Namespace	http://www.ime.iiasa.ac.at/model/analysis
Diagram	
Used by	Element e:definePreferenceRequest/preference
Model	idEntitySpec , preferenceType , prefValue
Children	idEntitySpec, prefValue, preferenceType

Source	<pre><xs:complexType name="preference">
	  <xs:sequence>
	    <xs:element name="idEntitySpec" type="xs:int" minOccurs="1" maxOccurs="1"/>
	    <xs:element name="preferenceType" type="a:preferenceType" minOccurs="1" maxOccurs="1"/>
	    <xs:element name="prefValue" type="xs:double" minOccurs="1" maxOccurs="1"/>
	  </xs:sequence>
	</xs:complexType></pre>

Complex Type `a:task`

Namespace	http://www.ime.iiasa.ac.at/model/analysis
Diagram	
Model	id , idAnalysis , idSolver , name{0,1} , description{0,1} , process
Children	description, id, idAnalysis, idSolver, name, process

Source	<pre><xs:complexType name="task">
	  <xs:sequence>
	    <xs:element name="id" type="xs:int"/>
	    <xs:element name="idAnalysis" type="xs:int" minOccurs="1" maxOccurs="1"/>
	    <xs:element name="idSolver" type="xs:int" minOccurs="1" maxOccurs="1"/>
	    <xs:element name="name" type="m:name" minOccurs="0" maxOccurs="1"/>
	    <xs:element name="description" type="m:description" minOccurs="0" maxOccurs="1"/>
	    <xs:element name="process" type="xs:string" minOccurs="1" maxOccurs="1"/>
	  </xs:sequence></pre>

```
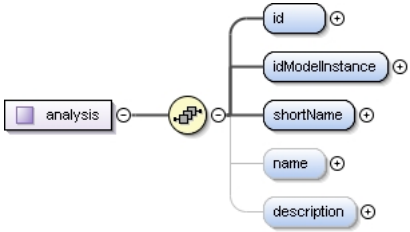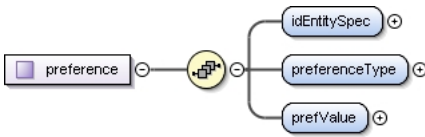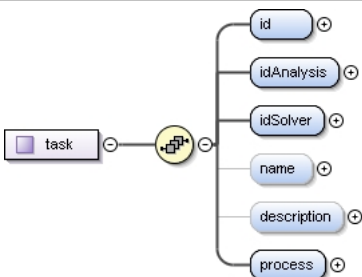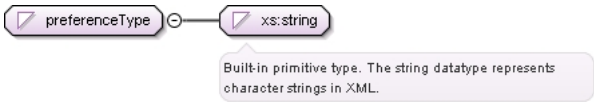</xs:complexType>
```

Simple Type(s)

Simple Type `a:preferenceType`

Namespace	http://www.ime.iiasa.ac.at/model/analysis
Diagram	
Type	restriction of xs:string
Facets	enumeration MIN
	enumeration MAX
	enumeration LOWER_BND
	enumeration UPPER_BND
Used by	Element a:preference/preferenceType

Source	`<xs:simpleType name="preferenceType">`
	  `<xs:restriction base="xs:string">`
	    `<xs:enumeration value="MIN"/>`
	    `<xs:enumeration value="MAX"/>`
	    `<xs:enumeration value="LOWER_BND"/>`
	    `<xs:enumeration value="UPPER_BND"/>`
	  `</xs:restriction>`
	`</xs:simpleType>`

```
</xs:complexType>
```