

Schema documentation for data.xsd

september 7, 2012

Table of Contents

Resource hierarchy:	1
Namespace: "http://www.ime.iiasa.ac.at/model/instance"	1
Schema(s)	1
Main schema data.xsd	1
Element(s)	1
Element i:dataSet	1
Complex Type(s)	2
Complex Type i:modelData	2
Complex Type i:modelInstance	2
Complex Type i:memberDic	3
Complex Type i:tupleValue	3
Complex Type i:setMembers	4
Complex Type i:tupleMembers	4
Complex Type i:entityValues	4
Complex Type i:tupleValues	5
Complex Type i:value	5
Complex Type i:groupEntityValues	6
Namespace: "http://www.ime.iiasa.ac.at/model/spec"	6
Schema(s)	6
Imported schema sms.xsd	6
Simple Type(s)	6
Simple Type m:description	6
Simple Type m:shortName	6
Simple Type m:name	7
Simple Type m:status	7
Simple Type m:setSpecType	7
Simple Type m:entityRole	8
Simple Type m:mathType	8
Complex Type(s)	9
Complex Type m:modelSpec	9
Complex Type m:setSpec	9
Complex Type m:iteratorContainer	10
Complex Type m:entitySpec	10

Resource hierarchy:

Legend:  Import,  Include,  Redefine,  Cycle detected

data.xsd

  sms.xsd

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

Schema(s)

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

Complex Type(s)

Complex Type i:modelData

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Model	id, idModelSpec+, idParent{0,1}, description{0,1}, locked
Children	description, id, idModelSpec, idParent, locked
Source	<pre><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></pre>

Complex Type i:modelInstance

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

Diagram	
Model	id , idModelSpec , idModelData , shortName{0,1} , name{0,1} , description{0,1} , status
Children	description, id, idModelData, idModelSpec, name, shortName, status
Source	<pre> <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> </pre>

Complex Type i:memberDic

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Used by	Elements i:groupEntityValues/tupleMember, i:tupleValue/tupleMember
Model	id , code , description{0,1}
Children	code, description, id
Source	<pre> <xs:complexType name="memberDic"> <xs:sequence> <xs:element name="id" type="xs:int"/> <xs:element name="code" type="m:name"/> <xs:element name="description" type="m:description" minOccurs="0" maxOccurs="1"/> </xs:sequence> </xs:complexType> </pre>

Complex Type i:tupleValue

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Model	tupleMember* , value
Children	tupleMember, value

Source	<pre> <xs:complexType name="tupleValue"> <xs:sequence> <xs:element name="tupleMember" type="i:memberDic" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="value"> <xs:complexType> <xs:choice> <xs:element name="intValue" type="xs:int"/> <xs:element name="doubleValue" type="xs:double"/> </xs:choice> </xs:complexType> </xs:element> </xs:sequence> </xs:complexType> </pre>
--------	--

Complex Type i:setMembers

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Used by	Element i:dataSet/setMembers
Model	setShortName , tupleMembers*
Children	setShortName, tupleMembers
Source	<pre> <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> </pre>

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	<pre> <xs:complexType name="tupleMembers"> <xs:sequence> <xs:element name="tuple" type="xs:string" minOccurs="0" maxOccurs="1"> <xs:annotation> <xs:documentation>only for indexed set, tuple members are separated 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 separated by , for example: AT,AU,PL</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType> </pre>

Complex Type i:entityValues

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

Annotations	only for mockup
Diagram	
Used by	Element <code>i:dataSet/entityValues</code>
Model	<code>entityShortName</code> , <code>tupleValues</code> *
Children	<code>entityShortName</code> , <code>tupleValues</code>
Source	<pre> <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> </pre>

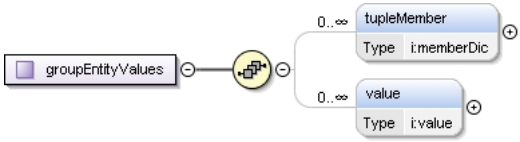
Complex Type `i:tupleValues`

Namespace	<code>http://www.ime.iiasa.ac.at/model/instance</code>
Annotations	only for mockup
Diagram	
Used by	Element <code>i:entityValues/tupleValues</code>
Model	<code>tuple{0,1}</code> , <code>value</code>
Children	<code>tuple</code> , <code>value</code>
Source	<pre> <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 separated by , for example: AT,2007</xs:documentation> </xs:annotation> </xs:element> <xs:element name="value" type="i:value"/> </xs:sequence> </xs:complexType> </pre>

Complex Type `i:value`

Namespace	<code>http://www.ime.iiasa.ac.at/model/instance</code>
Diagram	
Used by	Elements <code>i:groupEntityValues/value</code> , <code>i:tupleValues/value</code>
Model	<code>intValue</code> <code>doubleValue</code>
Children	<code>doubleValue</code> , <code>intValue</code>
Source	<pre> <xs:complexType name="value"> <xs:choice> <xs:element name="intValue" type="xs:int"/> <xs:element name="doubleValue" type="xs:double"/> </xs:choice> </xs:complexType> </pre>

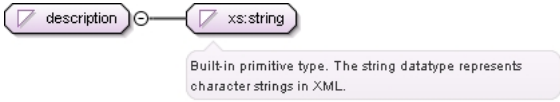
Complex Type `i:groupEntityValues`

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Model	<code>tupleMember*</code> , <code>value*</code>
Children	<code>tupleMember</code> , <code>value</code>
Source	<pre> <xs:complexType name="groupEntityValues"> <xs:sequence> <xs:element name="tupleMember" type="i:memberDic" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="value" type="i:value" minOccurs="0" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </pre>

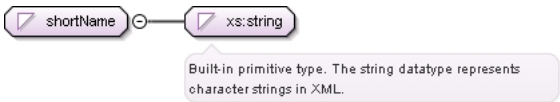
Namespace: "http://www.ime.iiasa.ac.at/model/spec"**Schema(s)****Imported schema `sms.xsd`**

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

Simple Type(s)**Simple Type `m:description`**

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

Simple Type `m:shortName`

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

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

Simple Type **m:name**

Namespace	http://www.ime.iiasa.ac.at/model/spec				
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>				
Type	restriction of xs:string				
Facets	<table> <tr> <td>minLength</td><td>1</td></tr> <tr> <td>maxLength</td><td>32</td></tr> </table>	minLength	1	maxLength	32
minLength	1				
maxLength	32				
Used by	Elements: i:memberDic/code, i:modelInstance/name, m:entitySpec/name, m:modelSpec/name, m:setSpec/name				
Source	<pre> <xs:simpleType name="name"> <xs:restriction base="xs:string"> <xs:minLength value="1" /> <xs:maxLength value="32" /> </xs:restriction> </xs:simpleType> </pre>				

Simple Type **m:status**

Namespace	http://www.ime.iiasa.ac.at/model/spec						
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>						
Type	restriction of xs:string						
Facets	<table> <tr> <td>enumeration</td><td>EDIT</td></tr> <tr> <td>enumeration</td><td>TEST</td></tr> <tr> <td>enumeration</td><td>COMMITTED</td></tr> </table>	enumeration	EDIT	enumeration	TEST	enumeration	COMMITTED
enumeration	EDIT						
enumeration	TEST						
enumeration	COMMITTED						
Used by	Elements: i:modelInstance/status, m:modelSpec/status						
Source	<pre> <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> </pre>						

Simple Type **m:setSpecType**

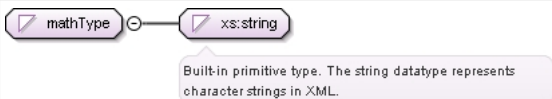
Namespace	http://www.ime.iiasa.ac.at/model/spec				
Diagram	<p>Built-in primitive type. The string datatype represents character strings in XML.</p>				
Type	restriction of xs:string				
Facets	<table> <tr> <td>enumeration</td><td>COLLECTION</td></tr> <tr> <td>enumeration</td><td>SEQUENCE</td></tr> </table>	enumeration	COLLECTION	enumeration	SEQUENCE
enumeration	COLLECTION				
enumeration	SEQUENCE				
Used by	Element: m:setSpec/type				
Source	<pre> <xs:simpleType name="setSpecType"> <xs:restriction base="xs:string"> <xs:enumeration value="COLLECTION" /> <xs:enumeration value="SEQUENCE" /> </xs:restriction> </pre>				

```
</xs:simpleType>
```

Simple Type **m:entityRole**

Namespace	http://www.ime.iiasa.ac.at/model/spec		
Diagram	<div><div>entityRole</div><div>xs:string</div><div>Built-in primitive type. The string datatype represents character strings in XML.</div></div>		
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	<pre><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></pre>		

Simple Type **m:mathType**

Namespace	http://www.ime.iiasa.ac.at/model/spec		
Diagram			
Type	restriction of xs:string		
Facets	enumeration	INTEGER	
	enumeration	REAL	
Used by	Element	m:entitySpec/mathType	
Source	<pre><xs:simpleType name="mathType"> <xs:restriction base="xs:string"> <xs:enumeration value="INTEGER"/> <!-- <xs:enumeration value="BINARY"/> --> <xs:enumeration value="REAL"/> </xs:restriction></pre>		


```
</xs:simpleType>
```

Complex Type(s)

Complex Type `m:modelSpec`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Model	id , shortName , name{0,1} , description{0,1} , status , setSpec+ , entitySpec+
Children	description, entitySpec, id, name, setSpec, shortName, status
Source	<pre><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></pre>

Complex Type `m:setSpec`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Used by	Element <code>m:modelSpec/setSpec</code>
Model	id , idParent{0,1} , shortName , name{0,1} , description{0,1} , type , idx{0,1} , iteratorContainer{0,1}
Children	description, id, idParent, idx, iteratorContainer, name, shortName, type
Source	<pre><xs:complexType name="setSpec"> <xs:sequence> <xs:element name="id" type="xs:int"/> <xs:element name="idParent" type="xs:int" minOccurs="0" maxOccurs="1"> <xs:annotation> <xs:documentation>only for sub set</xs:documentation> </xs:annotation> </xs:element> <xs:element name="shortName" type="m:shortName"/> <xs:element name="name" type="m:name" minOccurs="0" maxOccurs="1"/></pre>

```

<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	<pre> <xs:complexType name="iteratorContainer"> <xs:sequence> <xs:element name="idSetSpec" type="xs:int" minOccurs="1" maxOccurs="unbounded"/> </xs:sequence> </xs:complexType> </pre>

Complex Type m:entitySpec

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Used by	Element 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*, modelSpec{0,1}

Children	constantValue, description, formula, group, id, idLowerBound, idUpperBound, iteratorContainer, mathType, modelSpec, name, role, shortName, source, unit
Source	<pre> <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:element name="modelSpec" type="xs:string" minOccurs="0" maxOccurs="1"/> </xs:sequence> </xs:complexType> </pre>