Schema documentation for data.xsd

august 18, 2012

Table of Contents

Resource hierarchy:

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

data.xsd

sms.xsd

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

Schema(s)

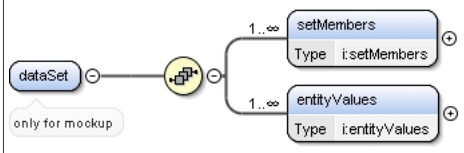
Main schema `data.xsd`

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

Element(s)

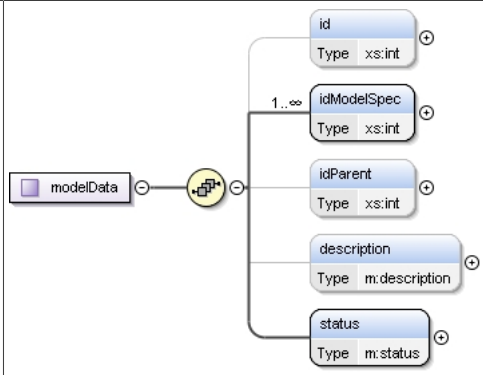
Element `i:dataSet`

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

Annotations	only for mockup
Diagram	
Properties	content: complex
Model	setMembers+ , entityValues+
Children	entityValues, setMembers
Instance	<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>

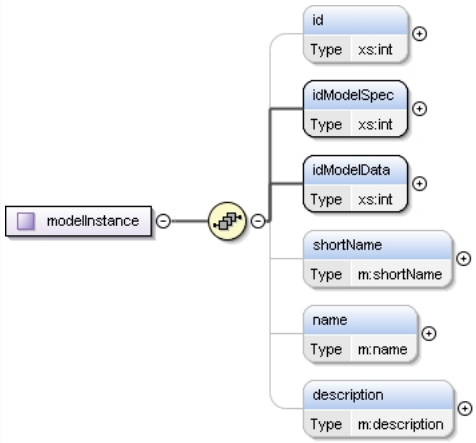
Complex Type(s)

Complex Type `i:modelData`

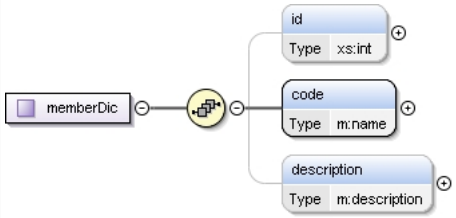
Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Model	id{0,1} , idModelSpec+ , idParent{0,1} , description{0,1} , status
Children	description, id, idModelSpec, idParent, status
Source	<xs:complexType name="modelData"> <xs:sequence> <xs:element name="id" type="xs:int" minOccurs="0" maxOccurs="1"/> <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="status" type="m:status" minOccurs="1" maxOccurs="1"/> </xs:sequence> </xs:complexType>

Complex Type `i:modelInstance`

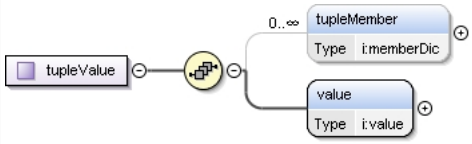
Namespace	http://www.ime.iiasa.ac.at/model/instance

Diagram	
Model	id{0,1} , idModelSpec , idModelData , shortName{0,1} , name{0,1} , description{0,1}
Children	description, id, idModelData, idModelSpec, name, shortName
Source	<xs:complexType name="modelInstance"> <xs:sequence> <xs:element name="id" type="xs:int" minOccurs="0" maxOccurs="1"/> <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:sequence> </xs:complexType>

Complex Type `i:memberDic`

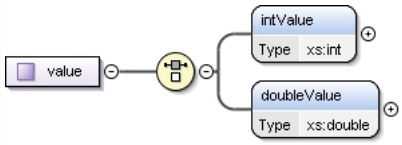
Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Used by	Elements i:batchValues/tupleMember, i:tupleValue/tupleMember
Model	id{0,1} , code , description{0,1}
Children	code, description, id
Source	<xs:complexType name="memberDic"> <xs:sequence> <xs:element name="id" type="xs:int" minOccurs="0"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

Complex Type `i:tupleValue`

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
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="i:value"/>

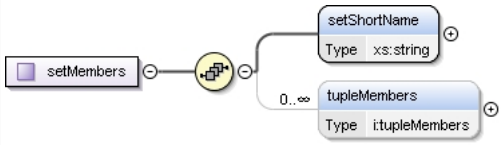
```
  </xs:sequence>
</xs:complexType>
```

Complex Type `i:value`

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Used by	Elements i:batchValues/value, i:tupleValue/value, i:tupleValues/value
Model	intValue | doubleValue
Children	doubleValue, intValue

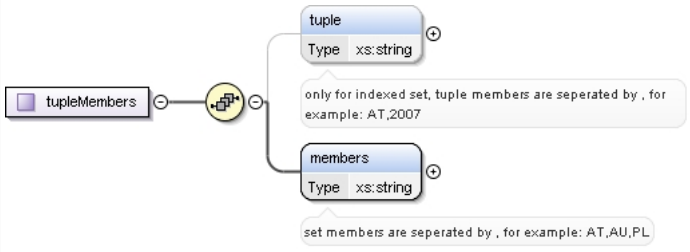
Source	`<xs:complexType name="value">`
	`  <xs:choice>`
	`  <xs:element name="intValue" type="xs:int"/>`
	`  <xs:element name="doubleValue" type="xs:double"/>`
	`  </xs:choice>`
	`</xs:complexType>`

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	`<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>`

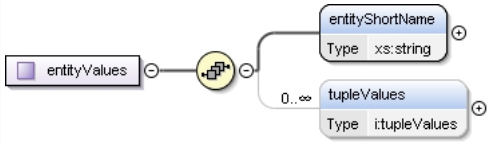
Complex Type `i:tupleMembers`

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	`<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 seperated by , for example: AT,2007</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element name="members" type="xs:string" minOccurs="1" maxOccurs="1">`

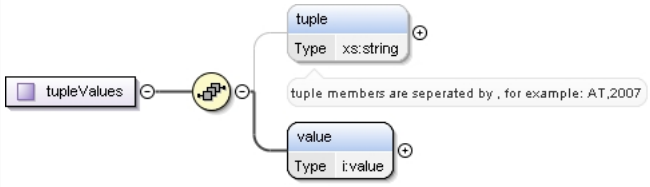
```
      <xs:annotation>
        <xs:documentation>set members are seperated by , for example: AT,AU,PL</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Complex Type `i:entityValues`

Namespace	http://www.ime.iiasa.ac.at/model/instance
Diagram	
Used by	Element i:dataSet/entityValues
Model	entityShortName , tupleValues*
Children	entityShortName, tupleValues

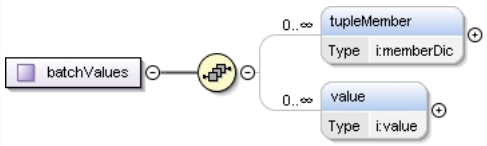
Source	`<xs:complexType name="entityValues">`
	`  <xs:sequence>`
	`  <xs:element name="entityShortName" type="xs:string"/>`
	`  <xs:element name="tupleValues" type="i:tupleValues" minOccurs="0" maxOccurs="unbounded"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `i:tupleValues`

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

Source	`<xs:complexType name="tupleValues">`
	`  <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="i:value"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `i:batchValues`

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

Source	`<xs:complexType name="batchValues">`
	`  <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>`

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

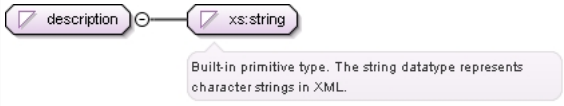
Schema(s)

Imported schema `sms.xsd`

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

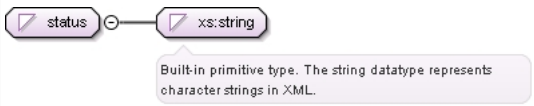
Simple Type(s)

Simple Type `m:description`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Type	restriction of xs:string
Facets	minLength: `1` maxLength: `1000`
Used by	Elements: i:memberDic/description, i:modelData/description, i:modelInstance/description, m:entitySpec/description, m:modelSpec/description, m:setSpec/description

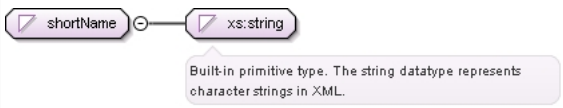
Source	`<xs:simpleType name="description">`
	`  <xs:restriction base="xs:string">`
	`  <xs:minLength value="1"/>`
	`  <xs:maxLength value="1000"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

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: `COMMITED`
Used by	Elements: i:modelData/status, m:modelSpec/status

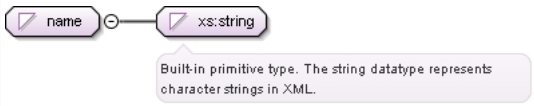
Source	`<xs:simpleType name="status">`
	`  <xs:restriction base="xs:string">`
	`  <xs:enumeration value="EDIT"/>`
	`  <xs:enumeration value="TEST"/>`
	`  <xs:enumeration value="COMMITED"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `m:shortName`

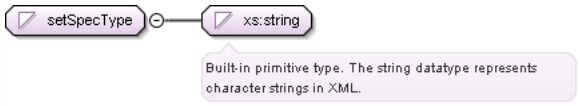
Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Type	restriction of xs:string
Facets	minLength: `1` maxLength: `16`

Used by	Elements	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="16"/> </xs:restriction> </xs:simpleType>	

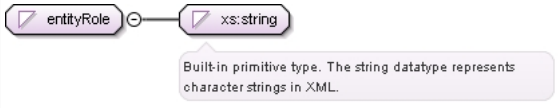
Simple Type `m:name`

Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Type	restriction of xs:string
Facets	minLength 1 maxLength 64
Used by	Elements i:memberDic/code, 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="64"/> </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 MAINSET enumeration SUBSET enumeration INDEXEDSUBSET
Used by	Element m:setSpec/type
Source	<xs:simpleType name="setSpecType"> <xs:restriction base="xs:string"> <xs:enumeration value="MAINSET"/> <xs:enumeration value="SUBSET"/> <xs:enumeration value="INDEXEDSUBSET"/> </xs:restriction> </xs:simpleType>

Simple Type `m:entityRole`

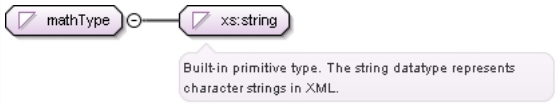
Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Type	restriction of xs:string
Facets	enumeration CONSTANT enumeration PARAMETER enumeration DECISION_VAR enumeration EXTERNAL_DECISION_VAR enumeration OUTCOME_VAR enumeration AUXILIARY_VAR

	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

```
<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`

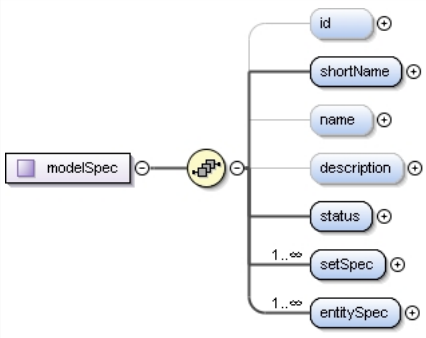
Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Type	restriction of xs:string
Facets	enumeration INTEGER enumeration REAL
Used by	Element m:entitySpec/mathType

Source

```
<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>
```

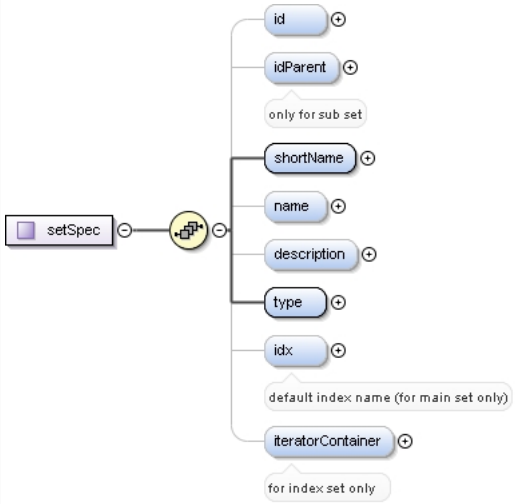
Complex Type(s)

Complex Type `m:modelSpec`

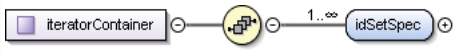
Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	

Model	id{0,1} , shortName , name{0,1} , description{0,1} , status , setSpec+ , entitySpec+
Children	description, entitySpec, id, name, setSpec, shortName, status
Source	<xs:complexType name="modelSpec"> <xs:sequence> <xs:element name="id" type="xs:int" minOccurs="0" 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: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`

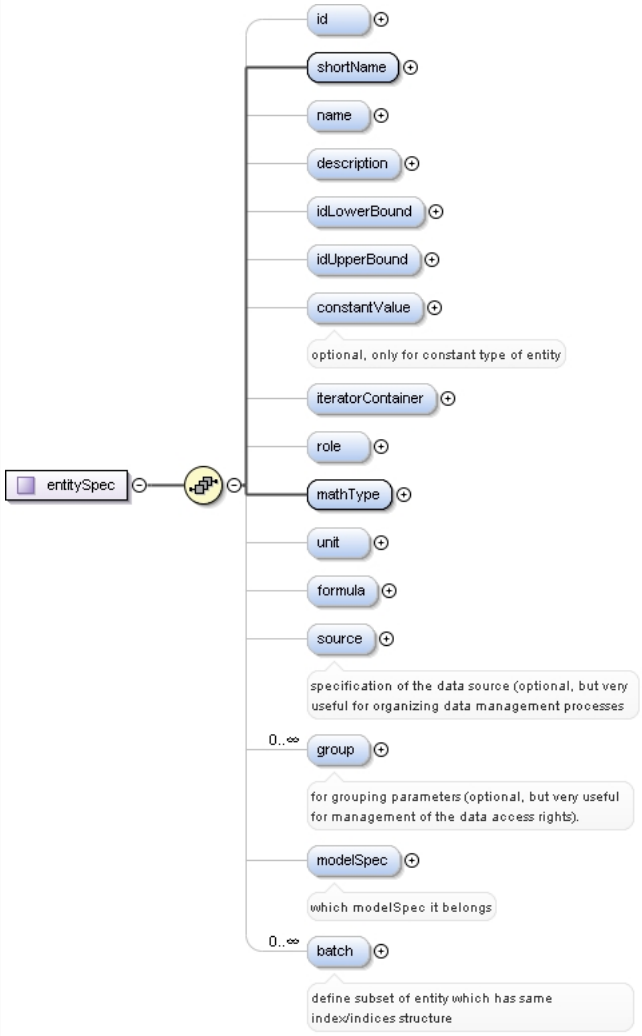
Namespace	http://www.ime.iiasa.ac.at/model/spec
Diagram	
Used by	Element m:modelSpec/setSpec
Model	id{0,1} , idParent{0,1} , shortName , name{0,1} , description{0,1} , type , idx{0,1} , iteratorContainer{0,1}
Children	description, id, idParent, idx, iteratorContainer, name, shortName, type
Source	<xs:complexType name="setSpec"> <xs:sequence> <xs:element name="id" type="xs:int" minOccurs="0" maxOccurs="1"/> <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"/> <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	<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	Element m:modelSpec/entitySpec
Model	id{0,1} , shortName , name{0,1} , description{0,1} , idLowerBound{0,1} , idUpperBound{0,1} , constantValue{0,1} , iteratorContainer{0,1} , role{0,1} , mathType , unit{0,1} , formula{0,1} , source{0,1} , group* , modelSpec{0,1} , batch*
Children	batch, constantValue, description, formula, group, id, idLowerBound, idUpperBound, iteratorContainer, mathType, modelSpec, name, role, shortName, source, unit
Source	<xs:complexType name="entitySpec"> <xs:sequence> <xs:element name="id" type="xs:int" minOccurs="0" 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: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="1" 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:annotation>
        <xs:documentation>which modelSpec it belongs</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="batch" type="xs:string" minOccurs="0" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation>define subset of entity which has same index/indices structure</
xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```