Schema documentation for DR-GW.xsd

november 22, 2023

Table of Contents

Namespace: "DR-GW"

Schema(s)

Main schema DR-GW.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.select.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.call.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.ptt.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.keyExchange.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Call.unitInEmergency.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.check.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.login.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.supervise.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Session.logout.xsd

Namespace	DR-GW
Properties	attribute form default: qualified

	element form default: `qualified`

Included schema `DR-GW-Sds.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Sds.send.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Sds.types.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Sds.receive.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Status.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Status.send.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Status.types.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Status.receive.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-OrganisationBlock.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-OrganisationBlock.get.xsd`

Namespace	DR-GW

Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-OrganisationBlock.types.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-OrganisationBlock.getList.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-OrganisationBlock.event.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-Group.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-Group.get.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-Group.types.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-Group.getList.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-Group.getRadioMembers.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-Group.getAppMembers.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-Group.getCombinations.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.track.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.addRadioMember.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.removeRadioMember.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.addCombination.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.removeCombination.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.subscribeData.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Group.event.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Radio.xsd

Namespace	DR-GW
Properties	attribute form default: qualified element form default: qualified

Included schema DR-GW-Radio.types.xsd

Namespace	DR-GW
Properties	attribute form default: qualified

	element form default: `qualified`

Included schema `DR-GW-Radio.get.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Radio.getList.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Radio.getGroups.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Radio.track.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Radio.changeOpta.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Radio.enableDisable.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Radio.event.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Application.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Application.get.xsd`

Namespace	DR-GW
Properties	attribute form default: `qualified` element form default: `qualified`

Included schema `DR-GW-Application.types.xsd`

Namespace	DR-GW

Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-Application.getList.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-System.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-System.tetraStates.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-System.types.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-System.log.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-System.event.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-Management.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-Management.get.xsd

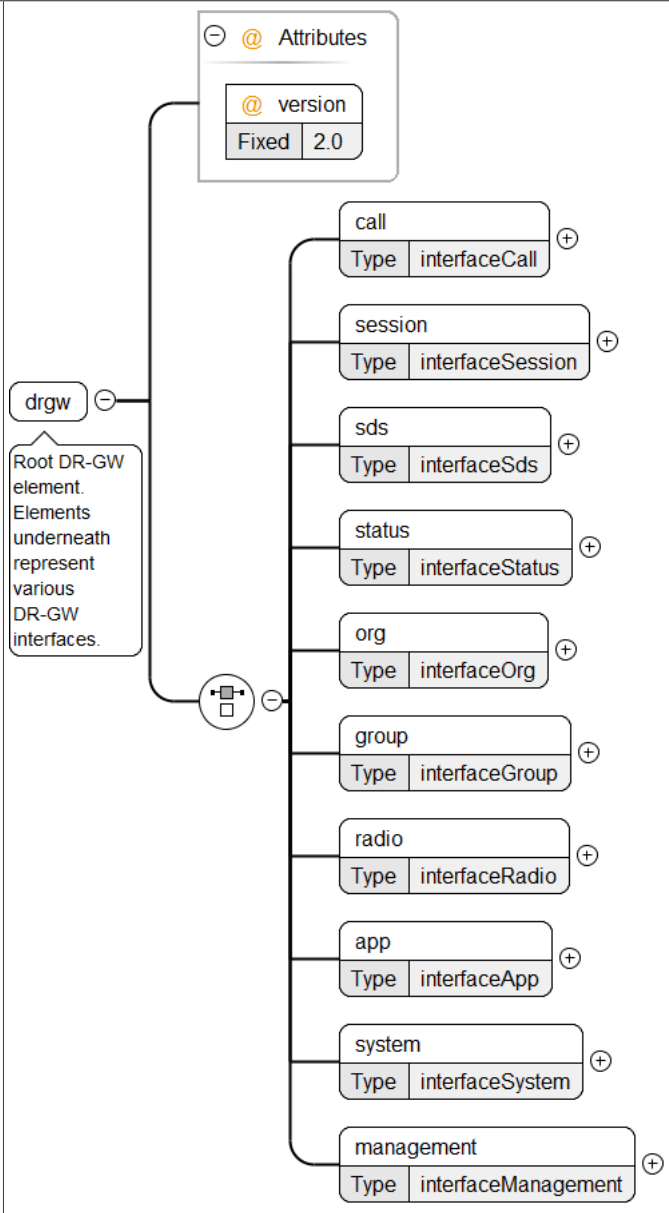
Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Included schema DR-GW-Management.types.xsd

Namespace	DR-GW	
Properties	attribute form default:	qualified
	element form default:	qualified

Element(s)

Element `drgw`

Namespace	DR-GW
Annotations	`Root DR-GW element. Elements underneath represent various DR-GW interfaces.`
Diagram	
Properties	content: `complex`
Model	call | session | sds | status | org | group | radio | app | system | management
Children	app, call, group, management, org, radio, sds, session, status, system

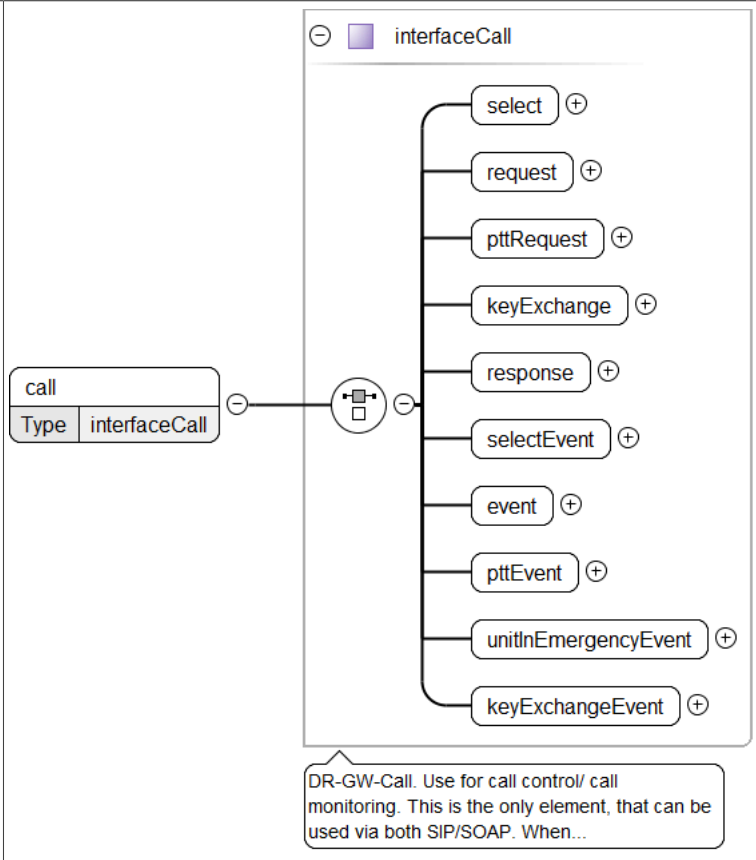
Instance	`<drgw version="2.0" xmlns="DR-GW">`
	`  <call>{1,1}</call>`
	`  <session>{1,1}</session>`
	`  <sds>{1,1}</sds>`
	`  <status>{1,1}</status>`
	`  <org>{1,1}</org>`
	`  <group>{1,1}</group>`
	`  <radio>{1,1}</radio>`
	`  <app>{1,1}</app>`
	`  <system>{1,1}</system>`
	`  <management>{1,1}</management>`
	`</drgw>`

Attributes	QName	Type	Fixed	Use	
	version		2.0	required	

Source

```
<xs:element name="drgw">
  <xs:annotation>
    <xs:documentation>Root DR-GW element. Elements underneath represent various DR-GW interfaces.</
xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:choice>
      <xs:element name="call" type="interfaceCall"/>
      <xs:element name="session" type="interfaceSession"/>
      <xs:element name="sds" type="interfaceSds"/>
      <xs:element name="status" type="interfaceStatus"/>
      <xs:element name="org" type="interfaceOrg"/>
      <xs:element name="group" type="interfaceGroup"/>
      <xs:element name="radio" type="interfaceRadio"/>
      <xs:element name="app" type="interfaceApp"/>
      <xs:element name="system" type="interfaceSystem"/>
      <xs:element name="management" type="interfaceManagement"/>
    </xs:choice>
    <xs:attribute name="version" fixed="2.0" use="required"/>
  </xs:complexType>
</xs:element>
```

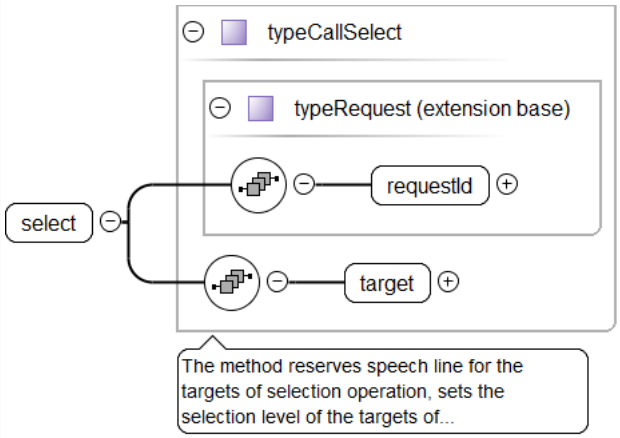
Element `drgw / call`

Namespace	DR-GW
Diagram	
Type	interfaceCall
Properties	content: complex
Model	select | request | pttRequest | keyExchange | response | selectEvent | event | pttEvent | unitInEmergencyEvent | keyExchangeEvent
Children	event, keyExchange, keyExchangeEvent, pttEvent, pttRequest, request, response, select, selectEvent, unitInEmergencyEvent

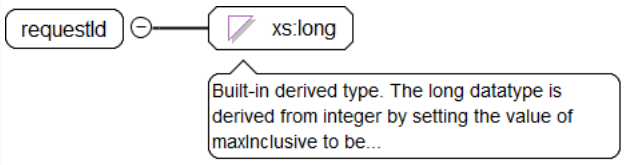
Instance	`<call xmlns="DR-GW">`
	`  <select>{1,1}</select>`
	`  <request>{1,1}</request>`
	`  <pttRequest>{1,1}</pttRequest>`
	`  <keyExchange>{1,1}</keyExchange>`
	`  <response>{1,1}</response>`

	<selectEvent>{1,1}</selectEvent> <event>{1,1}</event> <pttEvent>{1,1}</pttEvent> <unitInEmergencyEvent>{1,1}</unitInEmergencyEvent> <keyExchangeEvent>{1,1}</keyExchangeEvent> </call>
Source	<xs:element name="call" type="interfaceCall"/>

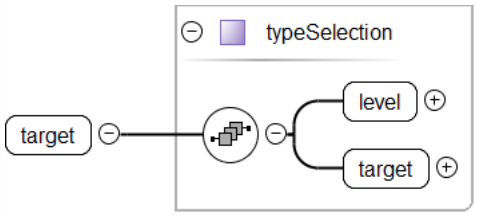
Element interfaceCall / select

Namespace	DR-GW
Diagram	
Type	typeCallSelect
Type hierarchy	• typeRequest • typeCallSelect
Properties	content: complex
Model	requestId , target
Children	requestId, target
Instance	<select xmlns="DR-GW"> <requestId>{1,1}</requestId> <target>{1,1}</target> </select>
Source	<xs:element name="select" type="typeCallSelect"/>

Element typeRequest / requestId

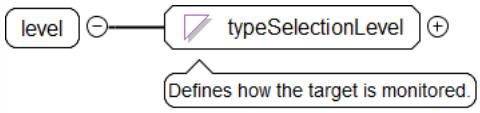
Namespace	DR-GW
Diagram	
Type	xs:long
Properties	content: simple
Source	<xs:element name="requestId" type="xs:long"/>

Element typeCallSelect / target

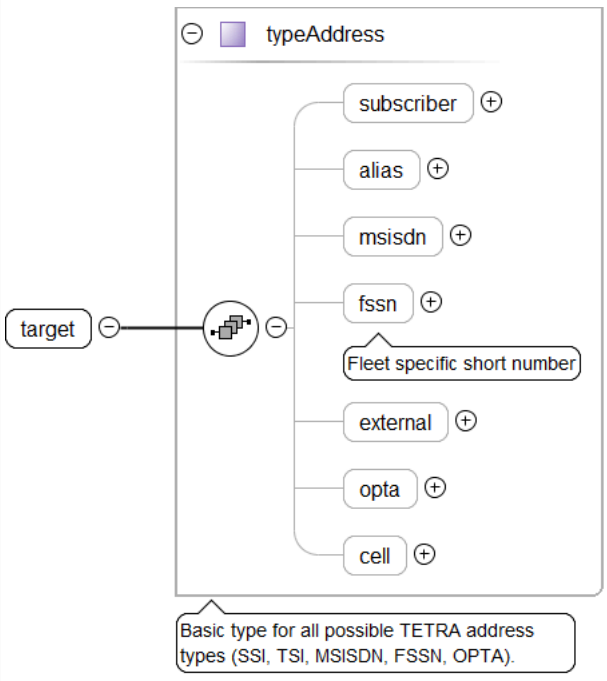
Namespace	DR-GW
Diagram	

Type	typeSelection
Properties	content: complex
Model	level , target
Children	level, target
Instance	<target xmlns="DR-GW"> <level>{1,1}</level> <target>{1,1}</target> </target>
Source	<xs:element name="target" type="typeSelection"/>

Element typeSelection / level

Namespace	DR-GW
Diagram	
Type	typeSelectionLevel
Properties	content: simple
Facets	enumeration no — No selection. Used to remove selection. enumeration event — Event monitoring. enumeration audio — Audio monitoring. enumeration use — Selection level use. enumeration a_use — Selection level active use.
Source	<xs:element name="level" type="typeSelectionLevel"/>

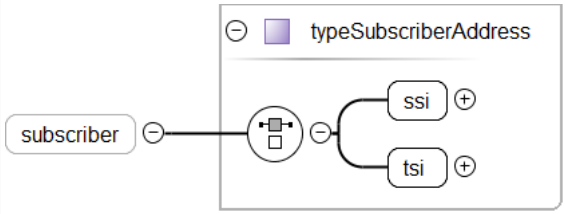
Element typeSelection / target

Namespace	DR-GW
Diagram	
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<target xmlns="DR-GW">

```
  <subscriber>{0,1}</subscriber>
  <alias>{0,1}</alias>
  <msisdn>{0,1}</msisdn>
  <fssn>{0,1}</fssn>
  <external>{0,1}</external>
  <opta>{0,1}</opta>
  <cell>{0,1}</cell>
</target>
```

Source	`<xs:element name="target" type="typeAddress"/>`

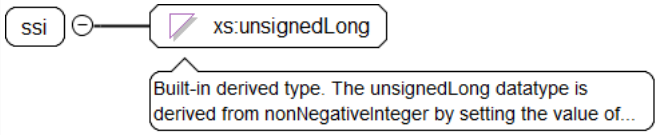
Element `typeAddress / subscriber`

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: `complex` minOccurs: `0`
Model	ssi | tsi
Children	ssi, tsi

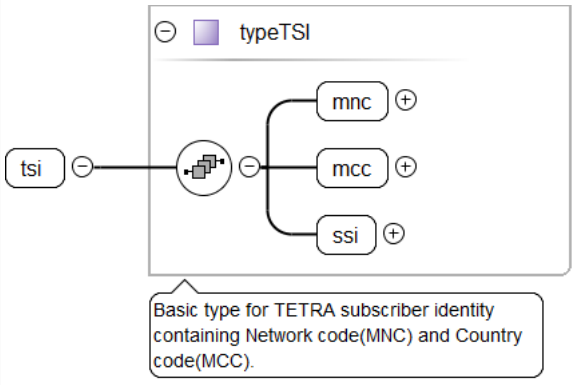
Instance	`<subscriber xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</subscriber>`

Source	`<xs:element name="subscriber" type="typeSubscriberAddress" minOccurs="0"/>`

Element `typeSubscriberAddress / ssi`

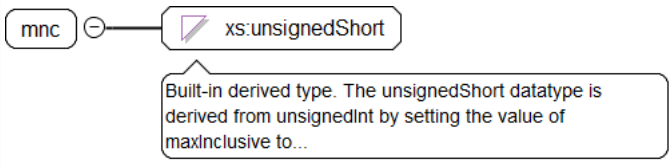
Namespace	DR-GW
Diagram	
Type	xs:unsignedLong
Properties	content: `simple`
Source	`<xs:element name="ssi" type="xs:unsignedLong"/>`

Element `typeSubscriberAddress / tsi`

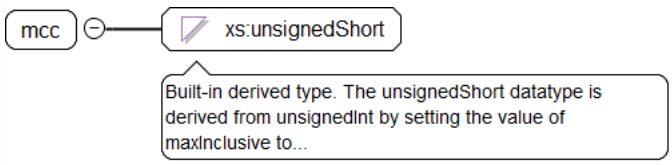
Namespace	DR-GW
Diagram	
Type	typeTSI

Properties	content: complex
Model	mnc , mcc , ssi
Children	mcc, mnc, ssi
Instance	<tsi xmlns="DR-GW"> <mnc>{1,1}</mnc> <mcc>{1,1}</mcc> <ssi>{1,1}</ssi> </tsi>
Source	<xs:element name="tsi" type="typeTSI"/>

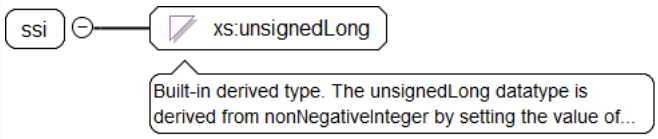
Element typeTSI / mnc

Namespace	DR-GW
Diagram	
Type	xs:unsignedShort
Properties	content: simple
Source	<xs:element name="mnc" type="xs:unsignedShort"/>

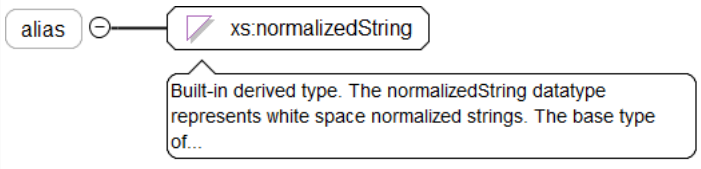
Element typeTSI / mcc

Namespace	DR-GW
Diagram	
Type	xs:unsignedShort
Properties	content: simple
Source	<xs:element name="mcc" type="xs:unsignedShort"/>

Element typeTSI / ssi

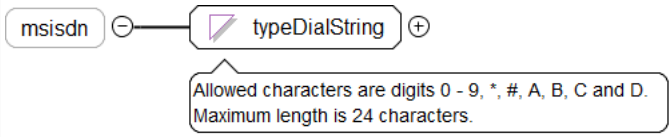
Namespace	DR-GW
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="ssi" type="xs:unsignedLong"/>

Element typeAddress / alias

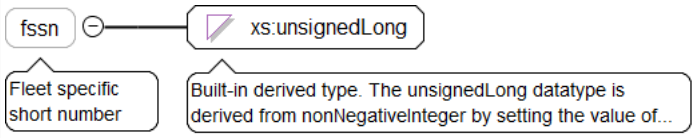
Namespace	DR-GW
Diagram	

Type	xs:normalizedString
Properties	content: simple minOccurs: 0
Source	`<xs:element name="alias" type="xs:normalizedString" minOccurs="0"/>`

Element typeAddress / msisdn

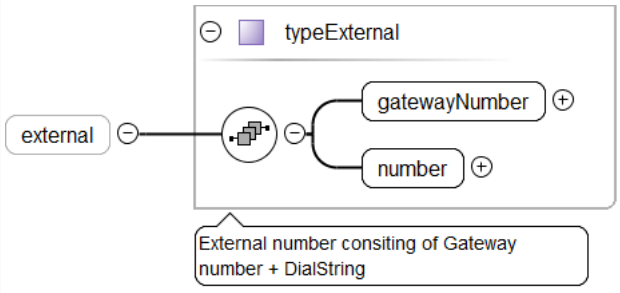
Namespace	DR-GW
Diagram	
Type	typeDialString
Properties	content: simple minOccurs: 0
Facets	maxLength 24
Source	`<xs:element name="msisdn" type="typeDialString" minOccurs="0"/>`

Element typeAddress / fssn

Namespace	DR-GW
Annotations	Fleet specific short number
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0

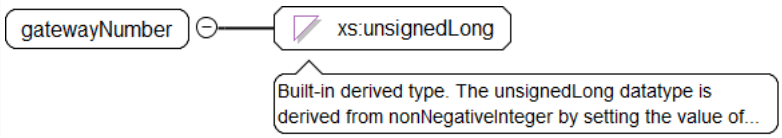
Source	`<xs:element name="fssn" type="xs:unsignedLong" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation>Fleet specific short number</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element typeAddress / external

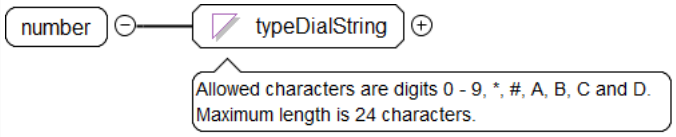
Namespace	DR-GW
Diagram	
Type	typeExternal
Properties	content: complex minOccurs: 0
Model	gatewayNumber , number
Children	gatewayNumber, number

Instance	<external xmlns="DR-GW"> <gatewayNumber>{1,1}</gatewayNumber> <number>{1,1}</number> </external>
Source	<xs:element name="external" type="typeExternal" minOccurs="0"/>

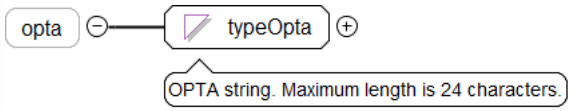
Element typeExternal / gatewayNumber

Namespace	DR-GW
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="gatewayNumber" type="xs:unsignedLong"/>

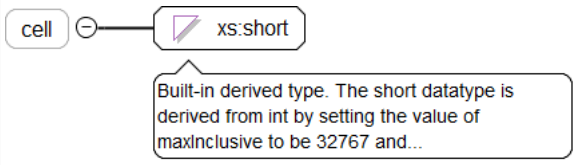
Element typeExternal / number

Namespace	DR-GW
Diagram	
Type	typeDialString
Properties	content: simple
Facets	maxLength 24
Source	<xs:element name="number" type="typeDialString"/>

Element typeAddress / opta

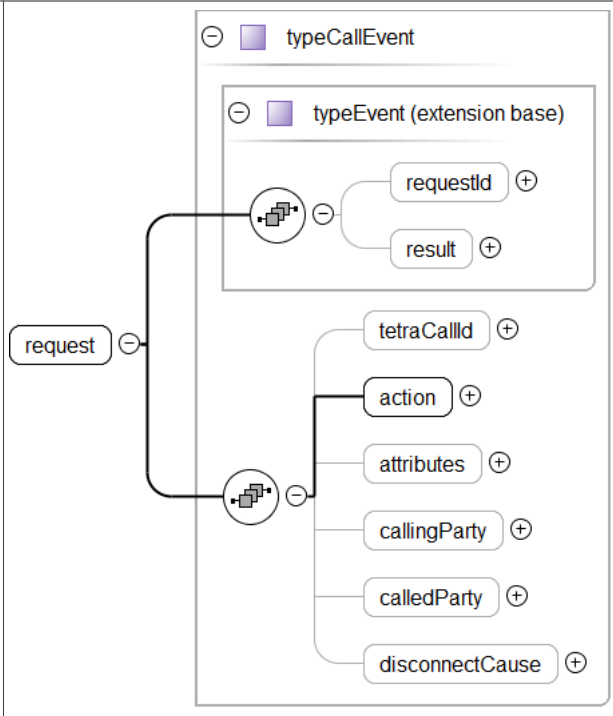
Namespace	DR-GW
Diagram	
Type	typeOpta
Properties	content: simple minOccurs: 0
Facets	maxLength 24
Source	<xs:element name="opta" type="typeOpta" minOccurs="0"/>

Element typeAddress / cell

Namespace	DR-GW
Diagram	
Type	xs:short
Properties	content: simple

	minOccurs: 0
Source	`<xs:element name="cell" type="xs:short" minOccurs="0"/>`

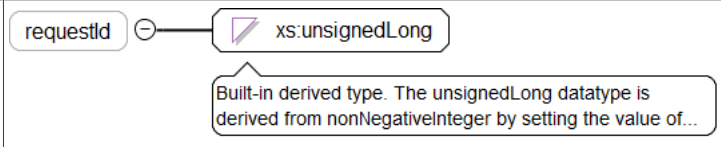
Element `interfaceCall / request`

Namespace	DR-GW
Diagram	
Type	typeCallEvent
Type hierarchy	• typeEvent • typeCallEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , tetraCallId{0,1} , action , attributes{0,1} , callingParty{0,1} , calledParty{0,1} , disconnectCause{0,1}
Children	action, attributes, calledParty, callingParty, disconnectCause, requestId, result, tetraCallId

Instance	`<request xmlns="DR-GW">`
	`  <requestId>{0,1}</requestId>`
	`  <result>{0,1}</result>`
	`  <tetraCallId>{0,1}</tetraCallId>`
	`  <action>{1,1}</action>`
	`  <attributes>{0,1}</attributes>`
	`  <callingParty>{0,1}</callingParty>`
	`  <calledParty>{0,1}</calledParty>`
	`  <disconnectCause>{0,1}</disconnectCause>`
	`</request>`

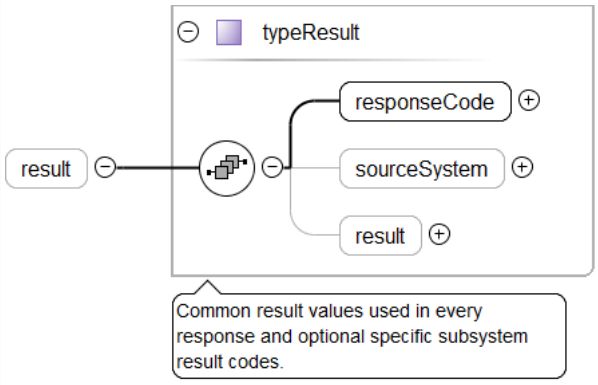
Source	`<xs:element name="request" type="typeCallEvent"/>`

Element `typeEvent / requestId`

Namespace	DR-GW
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0

Source	`<xs:element name="requestId" type="xs:unsignedLong" minOccurs="0"/>`

Element typeEvent / result

Namespace	DR-GW
Diagram	
Type	typeResult
Properties	content: complex minOccurs: 0
Model	responseCode , sourceSystem{0,1} , result{0,1}
Children	responseCode, result, sourceSystem

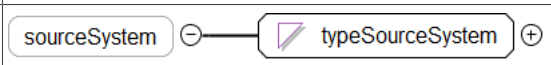
Instance	`<result xmlns="DR-GW">`
	`<responseCode>{1,1}</responseCode>`
	`<sourceSystem>{0,1}</sourceSystem>`
	`<result>{0,1}</result>`
	`</result>`

Source	`<xs:element name="result" type="typeResult" minOccurs="0"/>`

Element typeResult / responseCode

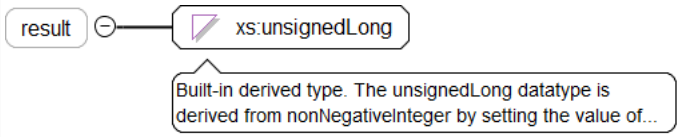
Namespace	DR-GW
Diagram	
Type	typeResponseCode
Properties	content: simple
Facets	enumeration: success enumeration: final_response_pending enumeration: error enumeration: not_authorized_error enumeration: temporary_failure enumeration: subscription_failed
Source	`<xs:element name="responseCode" type="typeResponseCode"/>`

Element typeResult / sourceSystem

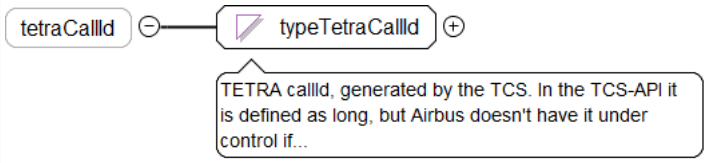
Namespace	DR-GW
Diagram	
Type	typeSourceSystem
Properties	content: simple minOccurs: 0
Facets	enumeration: DR-GW

	enumeration	TCS-API
	enumeration	TETRA
	enumeration	NEM-API
Source	<xs:element name="sourceSystem" type="typeSourceSystem" minOccurs="0"/>	

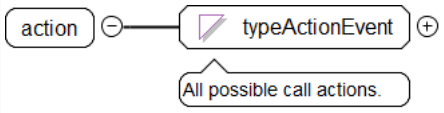
Element typeResult / result

Namespace	DR-GW
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	<xs:element name="result" type="xs:unsignedLong" minOccurs="0"/>

Element typeCallEvent / tetraCallId

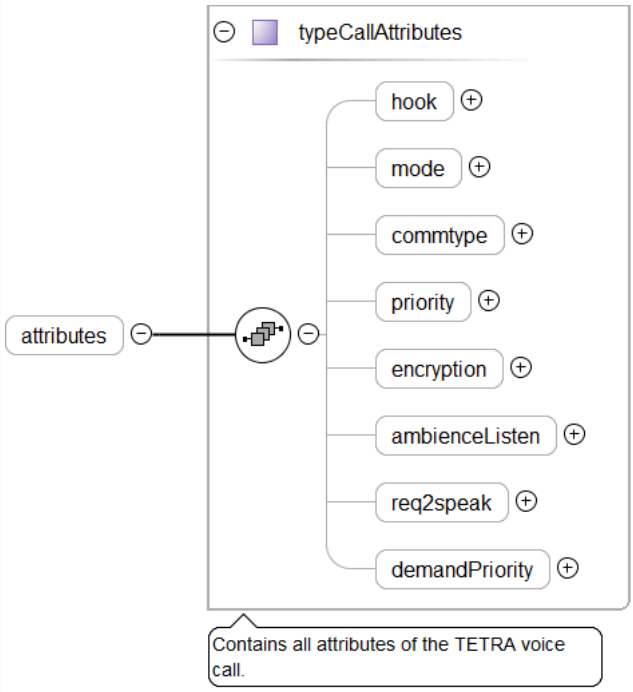
Namespace	DR-GW
Diagram	
Type	typeTetraCallId
Properties	content: simple minOccurs: 0
Source	<xs:element name="tetraCallId" type="typeTetraCallId" minOccurs="0"/>

Element typeCallEvent / action

Namespace	DR-GW	
Diagram		
Type	typeActionEvent	
Properties	content: simple	
Facets	enumeration: incoming	This event fired when there is an incoming call. This is the first indication of a new incoming call.
	enumeration: connected	This event is used to inform that call has been connected and call setup is finished.
	enumeration: held	This event is used to inform TCS Client that individual call was put to hold.
	enumeration: resumed	This event is used to inform that individual call has been taken from hold.
	enumeration: disconnected	This event is used to inform that the call was disconnected.
	enumeration: transferred	This event is a response to transfer method call and indicates the

	result of the request.
Source	`<xs:element name="action" type="typeActionEvent"/>`

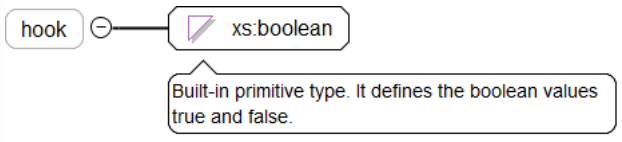
Element typeCallEvent / attributes

Namespace	DR-GW
Diagram	
Type	typeCallAttributes
Properties	content: complex minOccurs: 0
Model	hook{0,1} , mode{0,1} , commtype{0,1} , priority{0,1} , encryption{0,1} , ambienceListen{0,1} , req2speak{0,1} , demandPriority{0,1}
Children	ambienceListen, commtype, demandPriority, encryption, hook, mode, priority, req2speak

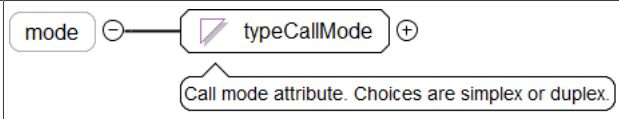
Instance	`<attributes xmlns="DR-GW">`
	`<hook>{0,1}</hook>`
	`<mode>{0,1}</mode>`
	`<commtype>{0,1}</commtype>`
	`<priority>{0,1}</priority>`
	`<encryption>{0,1}</encryption>`
	`<ambienceListen>{0,1}</ambienceListen>`
	`<req2speak>{0,1}</req2speak>`
	`<demandPriority>{0,1}</demandPriority>`
	`</attributes>`

Source	`<xs:element name="attributes" type="typeCallAttributes" minOccurs="0"/>`

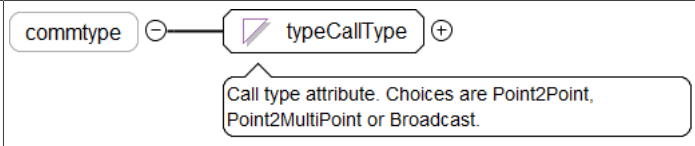
Element typeCallAttributes / hook

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0
Source	`<xs:element name="hook" type="xs:boolean" minOccurs="0"/>`

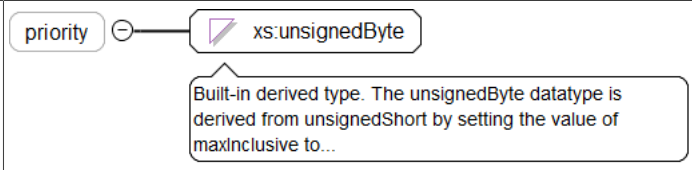
Element typeCallAttributes / mode

Namespace	DR-GW
Diagram	
Type	typeCallMode
Properties	content: simple minOccurs: 0
Facets	enumeration simplex enumeration duplex
Source	`<xs:element name="mode" type="typeCallMode" minOccurs="0"/>`

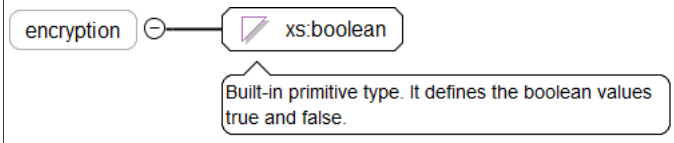
Element typeCallAttributes / commtype

Namespace	DR-GW
Diagram	
Type	typeCallType
Properties	content: simple minOccurs: 0
Facets	enumeration p2p enumeration p2mp enumeration bcast
Source	`<xs:element name="commtype" type="typeCallType" minOccurs="0"/>`

Element typeCallAttributes / priority

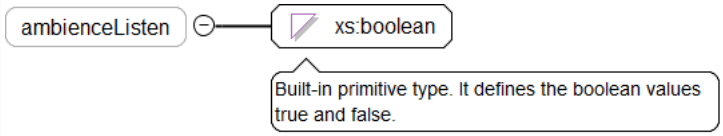
Namespace	DR-GW
Diagram	
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0 default: 1
Source	`<xs:element name="priority" type="xs:unsignedByte" default="1" minOccurs="0"/>`

Element typeCallAttributes / encryption

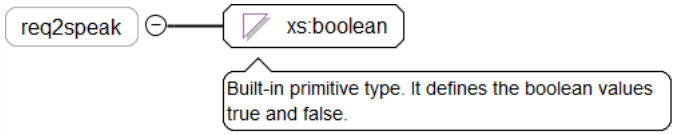
Namespace	DR-GW
Diagram	
Type	xs:boolean

Properties	content:	simple
	minOccurs:	0
	default:	true
Source	<xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/>	

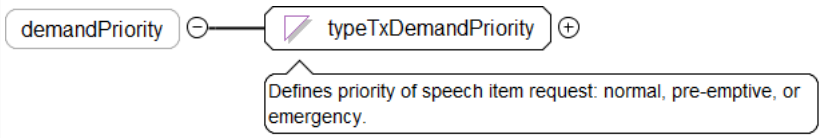
Element typeCallAttributes / ambienceListen

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: 0
Source	<xs:element name="ambienceListen" type="xs:boolean" default="0" minOccurs="0"/>

Element typeCallAttributes / req2speak

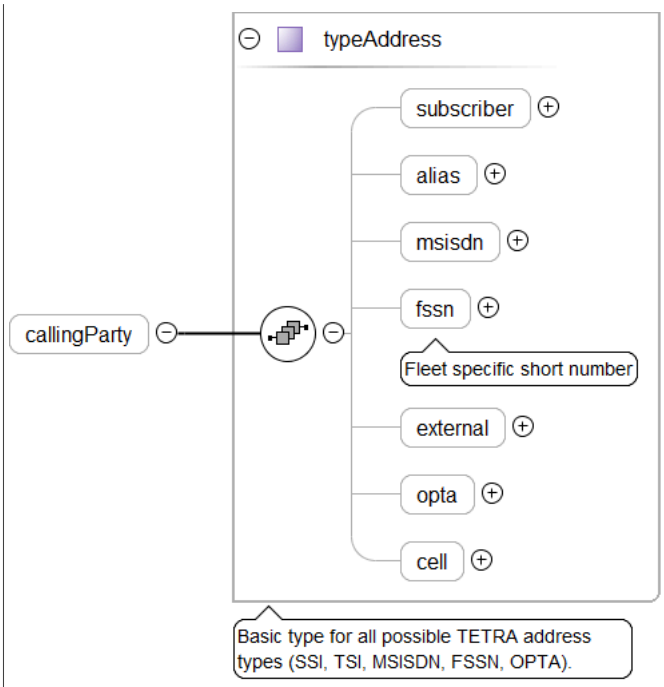
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: 1
Source	<xs:element name="req2speak" type="xs:boolean" default="1" minOccurs="0"/>

Element typeCallAttributes / demandPriority

Namespace	DR-GW
Diagram	
Type	typeTxDemandPriority
Properties	content: simple minOccurs: 0 default: normal
Facets	enumeration: normal enumeration: preemptive enumeration: emergency
Source	<xs:element name="demandPriority" type="typeTxDemandPriority" default="normal" minOccurs="0"/>

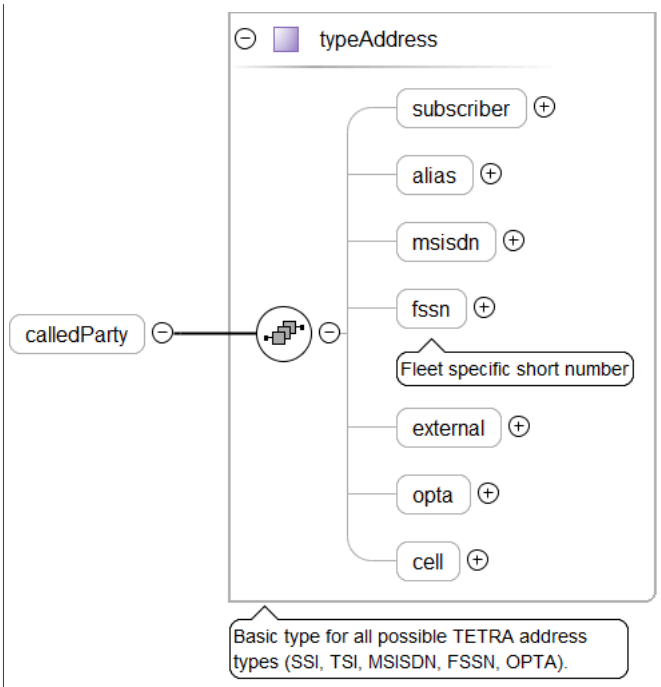
Element typeCallEvent / callingParty

Namespace	DR-GW

Diagram	
Type	typeAddress
Properties	content: complex minOccurs: 0
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<callingParty xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </callingParty>
Source	<xs:element name="callingParty" type="typeAddress" minOccurs="0"/>

Element typeCallEvent / calledParty

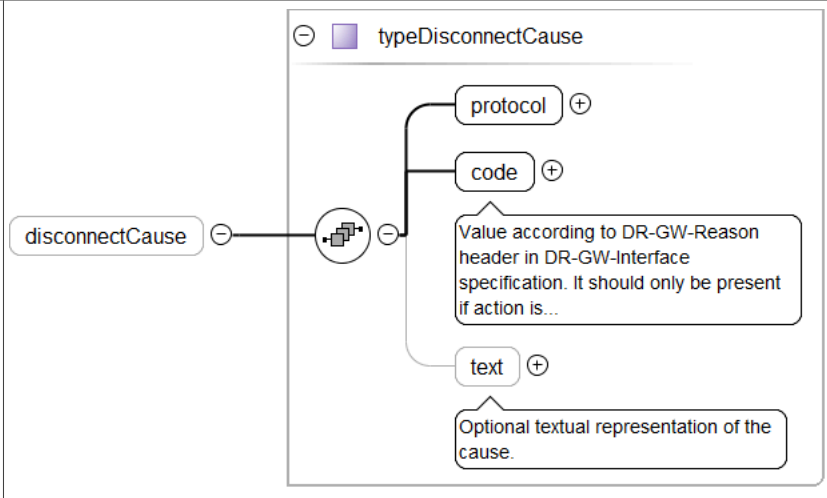
Namespace	DR-GW

Diagram	
Type	typeAddress
Properties	content: complex minOccurs: 0
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber

Instance	<pre><calledParty xmlns="DR-GW">
	  <subscriber>{0,1}</subscriber>
	  <alias>{0,1}</alias>
	  <msisdn>{0,1}</msisdn>
	  <fssn>{0,1}</fssn>
	  <external>{0,1}</external>
	  <opta>{0,1}</opta>
	  <cell>{0,1}</cell>
	</calledParty></pre>

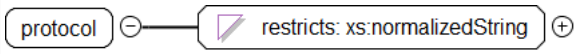
Source	<pre><xs:element name="calledParty" type="typeAddress" minOccurs="0"/></pre>

Element typeCallEvent / disconnectCause

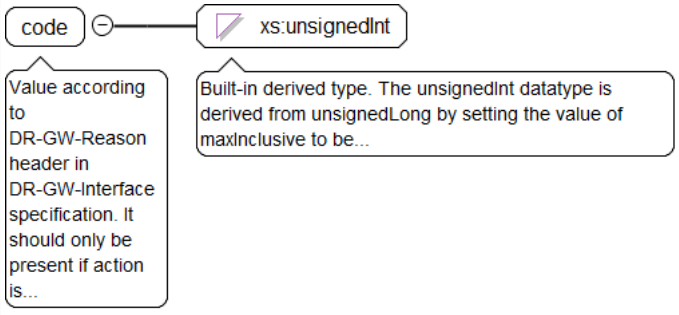
Namespace	DR-GW
Diagram	
Type	typeDisconnectCause
Properties	content: complex

	minOccurs: 0
Model	protocol , code , text{0,1}
Children	code, protocol, text
Instance	<disconnectCause xmlns="DR-GW"> <protocol>{1,1}</protocol> <code>{1,1}</code> <text>{0,1}</text> </disconnectCause>
Source	<xs:element name="disconnectCause" type="typeDisconnectCause" minOccurs="0"/>

Element typeDisconnectCause / protocol

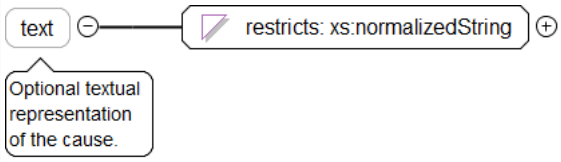
Namespace	DR-GW
Diagram	
Type	restriction of xs:normalizedString
Properties	content: simple
Facets	enumeration DR-GW enumeration TCS-API
Source	<xs:element name="protocol"> <xs:simpleType> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="DR-GW"/> <xs:enumeration value="TCS-API"/> </xs:restriction> </xs:simpleType> </xs:element>

Element typeDisconnectCause / code

Namespace	DR-GW
Annotations	Value according to DR-GW-Reason header in DR-GW-Interface specification. It should only be present if action is "disconnected" and holds the reason for call disconnection.
Diagram	
Type	xs:unsignedInt
Properties	content: simple
Source	<xs:element name="code" type="xs:unsignedInt"> <xs:annotation> <xs:documentation>Value according to DR-GW-Reason header in DR-GW-Interface specification. It should only be present if action is "disconnected" and holds the reason for call disconnection.</xs:documentation> </xs:annotation> </xs:element>

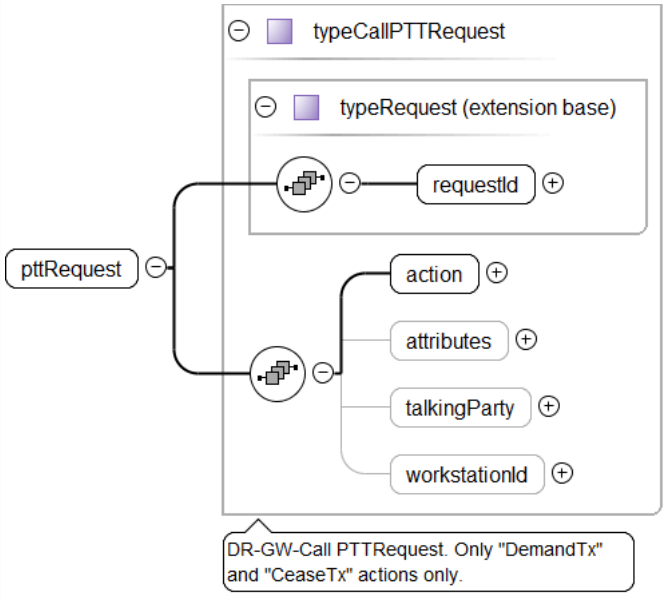
Element typeDisconnectCause / text

Namespace	DR-GW
Annotations	Optional textual representation of the cause.

Diagram	
Type	restriction of xs:normalizedString
Properties	content: simple minOccurs: 0
Facets	maxLength 80

Source	`<xs:element name="text" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation>Optional textual representation of the cause.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:simpleType>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:maxLength value="80"/>`
	`  </xs:restriction>`
	`  </xs:simpleType>`
	`</xs:element>`

Element `interfaceCall / pttRequest`

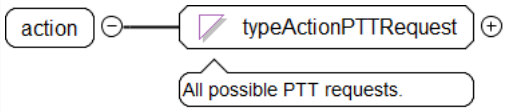
Namespace	DR-GW
Diagram	
Type	typeCallPTTRequest
Type hierarchy	• typeRequest • typeCallPTTRequest
Properties	content: complex
Model	requestId , action , attributes{0,1} , talkingParty{0,1} , workstationId{0,1}
Children	action, attributes, requestId, talkingParty, workstationId

Instance	`<pttRequest xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <action>{1,1}</action>`
	`  <attributes>{0,1}</attributes>`
	`  <talkingParty>{0,1}</talkingParty>`
	`  <workstationId>{0,1}</workstationId>`
	`</pttRequest>`

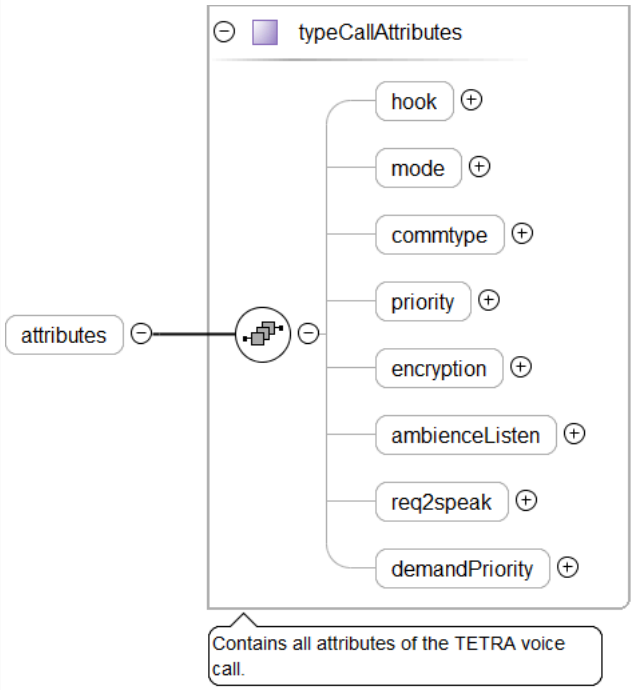
Source	`<xs:element name="pttRequest" type="typeCallPTTRequest"/>`

Element `typeCallPTTRequest / action`

Namespace	DR-GW

Diagram	
Type	typeActionPTTRequest
Properties	content: simple
Facets	enumeration: demandtx — This method can be used to request a speech item for a connected call. enumeration: ceasetx — This method is used to inform the system that the speech item is not needed any more.
Source	`<xs:element name="action" type="typeActionPTTRequest"/>`

Element typeCallPTTRequest / attributes

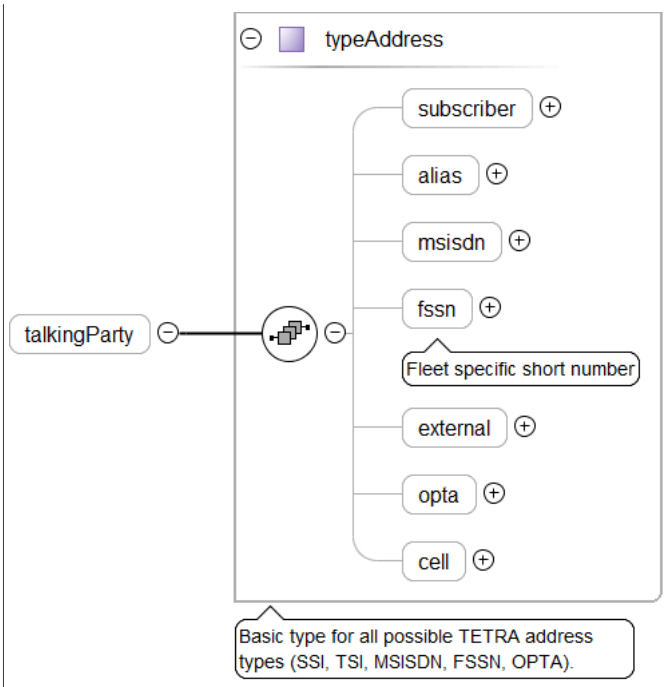
Namespace	DR-GW
Diagram	
Type	typeCallAttributes
Properties	content: complex minOccurs: 0
Model	hook{0,1} , mode{0,1} , commtype{0,1} , priority{0,1} , encryption{0,1} , ambienceListen{0,1} , req2speak{0,1} , demandPriority{0,1}
Children	ambienceListen, commtype, demandPriority, encryption, hook, mode, priority, req2speak

Instance	`<attributes xmlns="DR-GW">`
	`<hook>{0,1}</hook>`
	`<mode>{0,1}</mode>`
	`<commtype>{0,1}</commtype>`
	`<priority>{0,1}</priority>`
	`<encryption>{0,1}</encryption>`
	`<ambienceListen>{0,1}</ambienceListen>`
	`<req2speak>{0,1}</req2speak>`
	`<demandPriority>{0,1}</demandPriority>`
	`</attributes>`

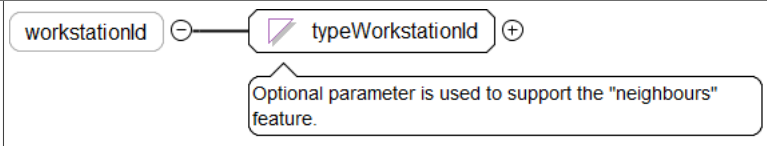
Source	`<xs:element name="attributes" type="typeCallAttributes" minOccurs="0"/>`

Element typeCallPTTRequest / talkingParty

Namespace	DR-GW

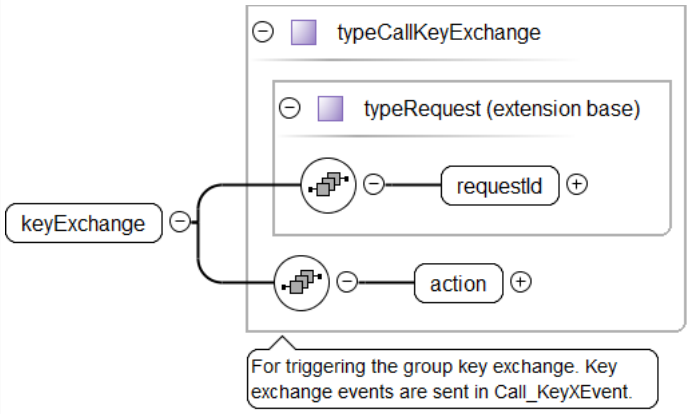
Diagram	
Type	typeAddress
Properties	content: complex minOccurs: 0
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<talkingParty xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </talkingParty>
Source	<xs:element name="talkingParty" type="typeAddress" minOccurs="0"/>

Element typeCallPTTRequest / workstationId

Namespace	DR-GW
Diagram	
Type	typeWorkstationId
Properties	content: simple minOccurs: 0
Source	<xs:element name="workstationId" type="typeWorkstationId" minOccurs="0"/>

Element interfaceCall / keyExchange

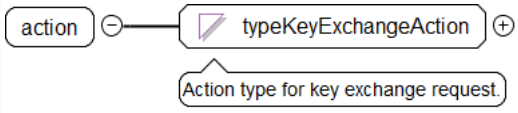
Namespace	DR-GW

Diagram	
Type	typeCallKeyExchange
Type hierarchy	• typeRequest • typeCallKeyExchange
Properties	content: complex
Model	requestId , action
Children	action, requestId

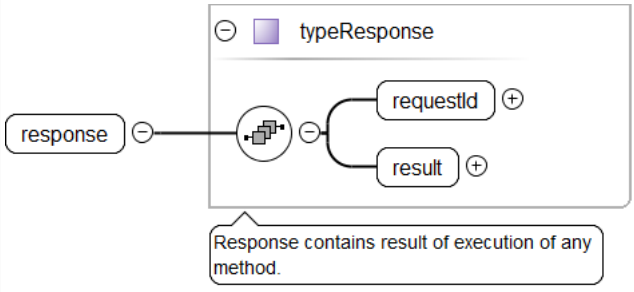
Instance	`<keyExchange xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <action>{1,1}</action>`
	`</keyExchange>`

Source	`<xs:element name="keyExchange" type="typeCallKeyExchange"/>`

Element `typeCallKeyExchange / action`

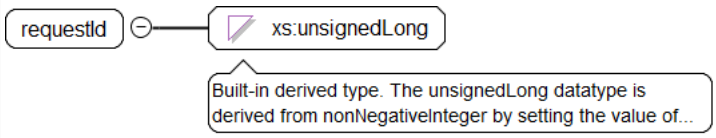
Namespace	DR-GW
Diagram	
Type	typeKeyExchangeAction
Properties	content: simple
Facets	enumeration start enumeration stop
Source	`<xs:element name="action" type="typeKeyExchangeAction"/>`

Element `interfaceCall / response`

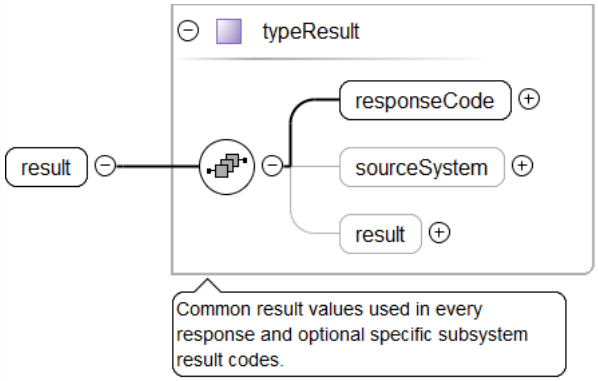
Namespace	DR-GW
Diagram	
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result

Instance	<response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response>
Source	<xs:element name="response" type="typeResponse"/>

Element typeResponse / requestId

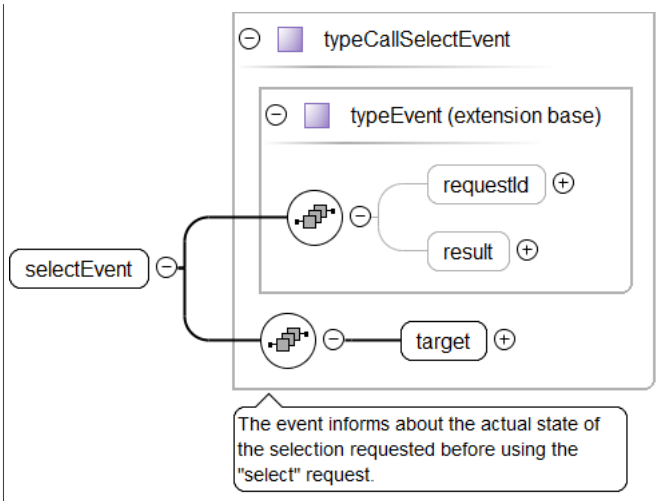
Namespace	DR-GW
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="requestId" type="xs:unsignedLong"/>

Element typeResponse / result

Namespace	DR-GW
Diagram	
Type	typeResult
Properties	content: complex
Model	responseCode , sourceSystem{0,1} , result{0,1}
Children	responseCode, result, sourceSystem
Instance	<result xmlns="DR-GW"> <responseCode>{1,1}</responseCode> <sourceSystem>{0,1}</sourceSystem> <result>{0,1}</result> </result>
Source	<xs:element name="result" type="typeResult"/>

Element interfaceCall / selectEvent

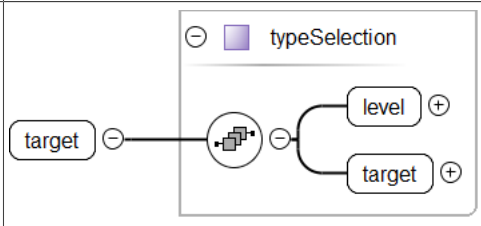
Namespace	DR-GW

Diagram	
Type	typeCallSelectEvent
Type hierarchy	• typeEvent • typeCallSelectEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , target
Children	requestId, result, target

Instance	`<selectEvent xmlns="DR-GW">`
	`<requestId>{0,1}</requestId>`
	`<result>{0,1}</result>`
	`<target>{1,1}</target>`
	`</selectEvent>`

Source	`<xs:element name="selectEvent" type="typeCallSelectEvent"/>`

Element `typeCallSelectEvent / target`

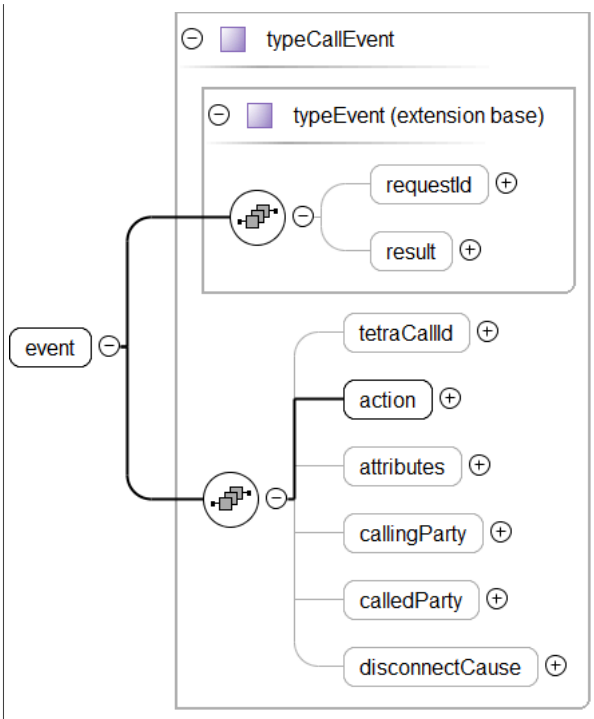
Namespace	DR-GW
Diagram	
Type	typeSelection
Properties	content: complex
Model	level , target
Children	level, target

Instance	`<target xmlns="DR-GW">`
	`<level>{1,1}</level>`
	`<target>{1,1}</target>`
	`</target>`

Source	`<xs:element name="target" type="typeSelection"/>`

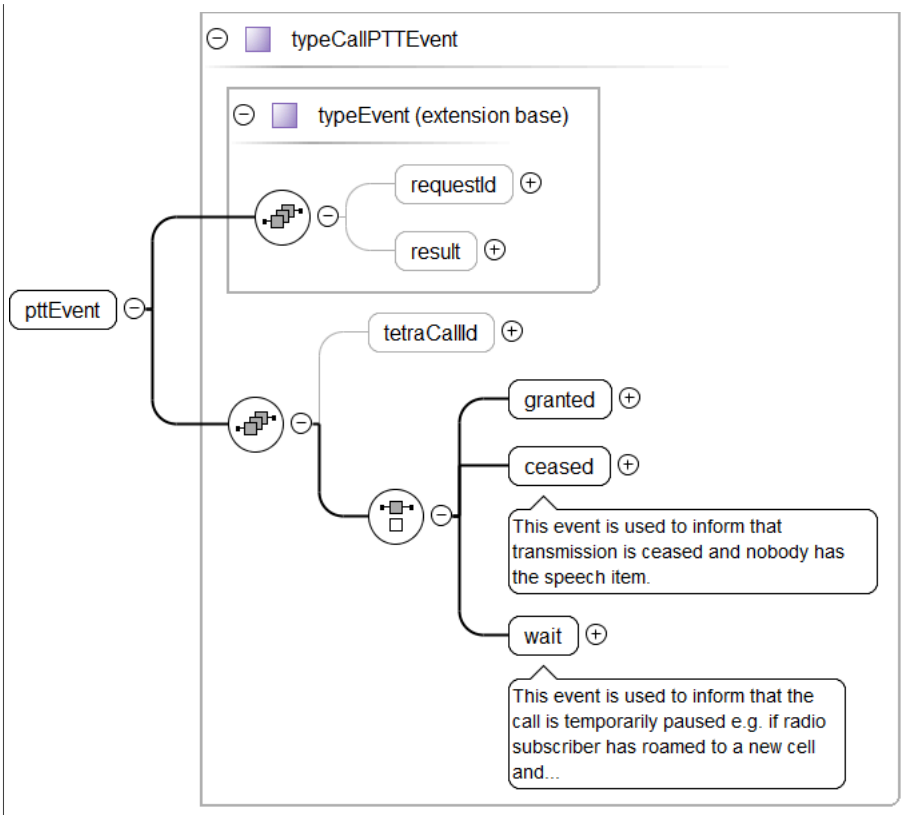
Element `interfaceCall / event`

Namespace	DR-GW

Diagram	
Type	typeCallEvent
Type hierarchy	• typeEvent • typeCallEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , tetraCallId{0,1} , action , attributes{0,1} , callingParty{0,1} , calledParty{0,1} , disconnectCause{0,1}
Children	action, attributes, calledParty, callingParty, disconnectCause, requestId, result, tetraCallId
Instance	<event xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <tetraCallId>{0,1}</tetraCallId> <action>{1,1}</action> <attributes>{0,1}</attributes> <callingParty>{0,1}</callingParty> <calledParty>{0,1}</calledParty> <disconnectCause>{0,1}</disconnectCause> </event>
Source	<xs:element name="event" type="typeCallEvent"/>

Element `interfaceCall / pttEvent`

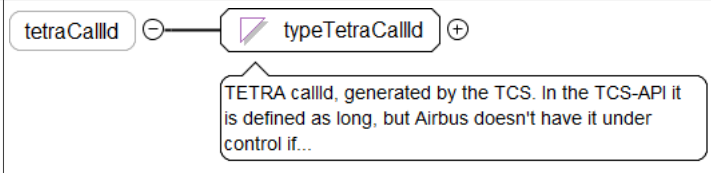
Namespace	DR-GW

Diagram	
Type	typeCallPTTEvent
Type hierarchy	• typeEvent • typeCallPTTEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , tetraCallId{0,1} , (granted | ceased | wait)
Children	ceased, granted, requestId, result, tetraCallId, wait

Instance	`<pttEvent xmlns="DR-GW">`
	`  <requestId>{0,1}</requestId>`
	`  <result>{0,1}</result>`
	`  <tetraCallId>{0,1}</tetraCallId>`
	`  <granted>{1,1}</granted>`
	`  <ceased>{1,1}</ceased>`
	`  <wait>{1,1}</wait>`
	`</pttEvent>`

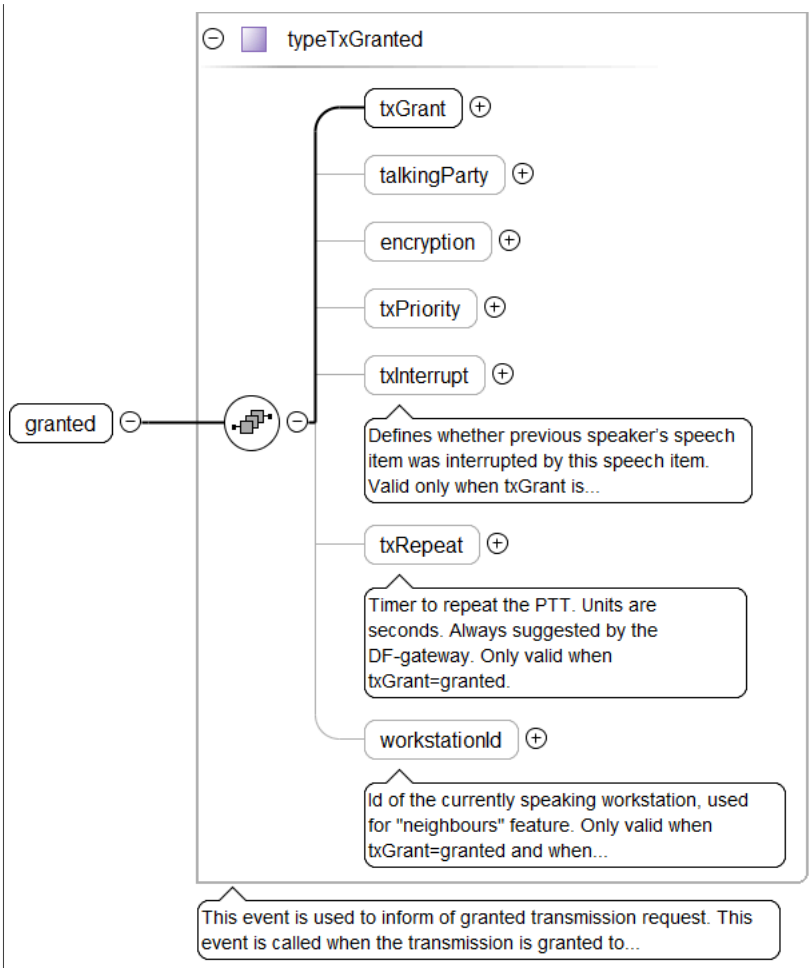
Source	`<xs:element name="pttEvent" type="typeCallPTTEvent"/>`

Element typeCallPTTEvent / tetraCallId

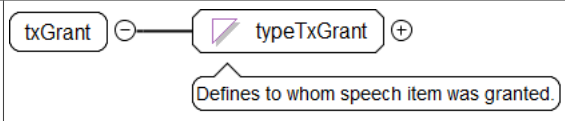
Namespace	DR-GW
Diagram	
Type	typeTetraCallId
Properties	content: simple minOccurs: 0
Source	`<xs:element name="tetraCallId" type="typeTetraCallId" minOccurs="0"/>`

Element typeCallPTTEvent / granted

Namespace	DR-GW

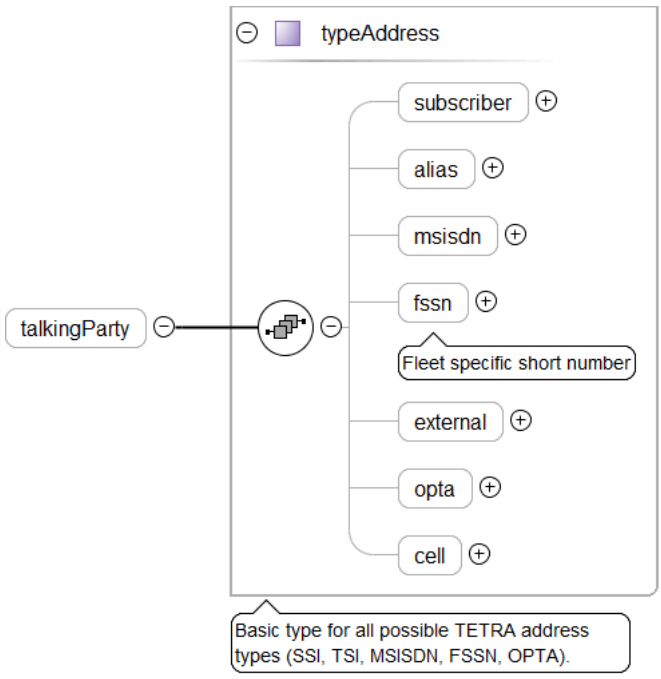
Diagram	
Type	typeTxGranted
Properties	content: complex
Model	txGrant , talkingParty{0,1} , encryption{0,1} , txPriority{0,1} , txInterrupt{0,1} , txRepeat{0,1} , workstationId{0,1}
Children	encryption, talkingParty, txGrant, txInterrupt, txPriority, txRepeat, workstationId
Instance	<granted xmlns="DR-GW"> <txGrant>{1,1}</txGrant> <talkingParty>{0,1}</talkingParty> <encryption>{0,1}</encryption> <txPriority>{0,1}</txPriority> <txInterrupt>{0,1}</txInterrupt> <txRepeat>{0,1}</txRepeat> <workstationId>{0,1}</workstationId> </granted>
Source	<xs:element name="granted" type="typeTxGranted"/>

Element typeTxGranted / txGrant

Namespace	DR-GW
Diagram	
Type	typeTxGrant
Properties	content: simple
Facets	enumeration granted enumeration notGranted enumeration queued enumeration granted2another

Source	`<xs:element name="txGrant" type="typeTxGrant"/>`

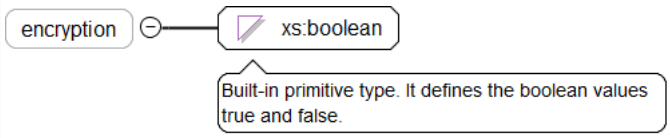
Element typeTxGranted / talkingParty

Namespace	DR-GW
Diagram	
Type	typeAddress
Properties	content: complex minOccurs: 0
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber

Instance	`<talkingParty xmlns="DR-GW">`
	`  <subscriber>{0,1}</subscriber>`
	`  <alias>{0,1}</alias>`
	`  <msisdn>{0,1}</msisdn>`
	`  <fssn>{0,1}</fssn>`
	`  <external>{0,1}</external>`
	`  <opta>{0,1}</opta>`
	`  <cell>{0,1}</cell>`
	`</talkingParty>`

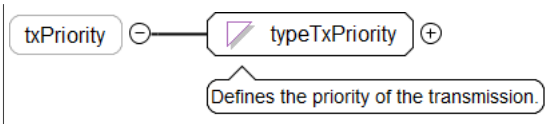
Source	`<xs:element name="talkingParty" type="typeAddress" minOccurs="0"/>`

Element typeTxGranted / encryption

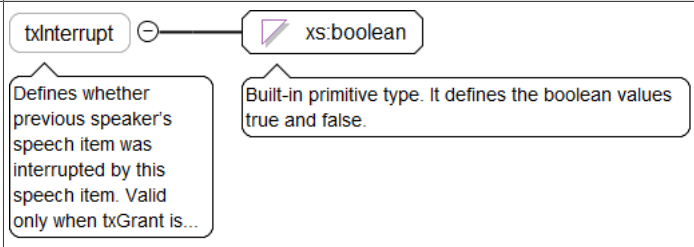
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: true
Source	`<xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/>`

Element typeTxGranted / txPriority

Namespace	DR-GW

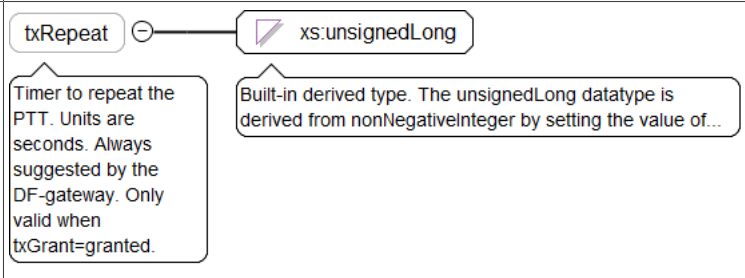
Diagram	
Type	typeTxPriority
Properties	content: simple minOccurs: 0 default: normal
Facets	enumeration normal enumeration emergency
Source	`<xs:element name="txPriority" type="typeTxPriority" minOccurs="0" default="normal"/>`

Element typeTxGranted / txInterrupt

Namespace	DR-GW
Annotations	Defines whether previous speaker's speech item was interrupted by this speech item. Valid only when txGrant is granted2another.
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: false

Source	`<xs:element name="txInterrupt" type="xs:boolean" default="false" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation>Defines whether previous speaker's speech item was interrupted by this speech item. Valid only when txGrant is granted2another.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

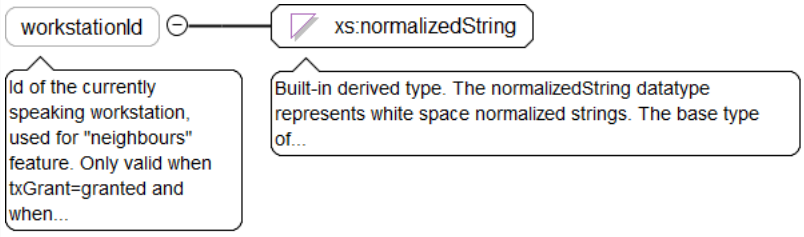
Element typeTxGranted / txRepeat

Namespace	DR-GW
Annotations	Timer to repeat the PTT. Units are seconds. Always suggested by the DF-gateway. Only valid when txGrant=granted.
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0 default: 0

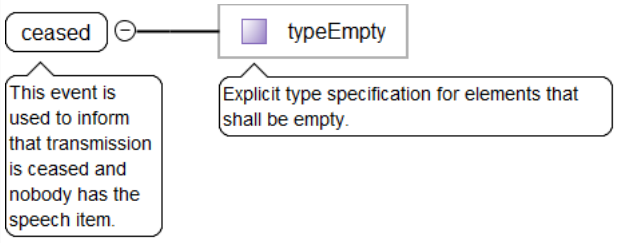
Source	`<xs:element name="txRepeat" type="xs:unsignedLong" minOccurs="0" default="0">`
	`  <xs:annotation>`
	`  <xs:documentation>Timer to repeat the PTT. Units are seconds. Always suggested by the DF-gateway. Only valid when txGrant=granted.</xs:documentation>`

```
  </xs:annotation>
</xs:element>
```

Element typeTxGranted / workstationId

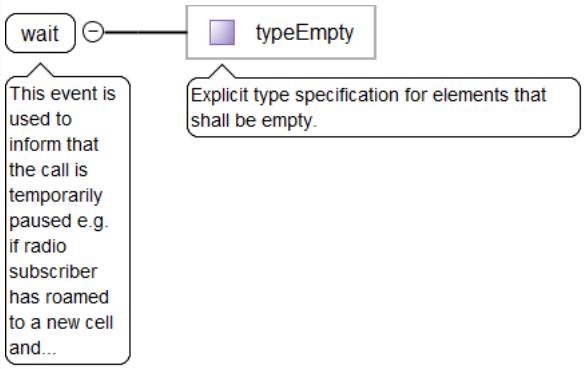
Namespace	DR-GW
Annotations	Id of the currently speaking workstation, used for "neighbours" feature. Only valid when txGrant=granted and when supplied by the DF-client in PTT request.
Diagram	
Type	xs:normalizedString
Properties	content: simple minOccurs: 0
Source	<xs:element name="workstationId" type="xs:normalizedString" minOccurs="0"> <xs:annotation> <xs:documentation>Id of the currently speaking workstation, used for "neighbours" feature. Only valid when txGrant=granted and when supplied by the DF-client in PTT request.</xs:documentation> </xs:annotation> </xs:element>

Element typeCallPTTEvent / ceased

Namespace	DR-GW
Annotations	This event is used to inform that transmission is ceased and nobody has the speech item.
Diagram	
Type	typeEmpty
Properties	content: complex
Source	<xs:element name="ceased" type="typeEmpty"> <xs:annotation> <xs:documentation>This event is used to inform that transmission is ceased and nobody has the speech item.</xs:documentation> </xs:annotation> </xs:element>

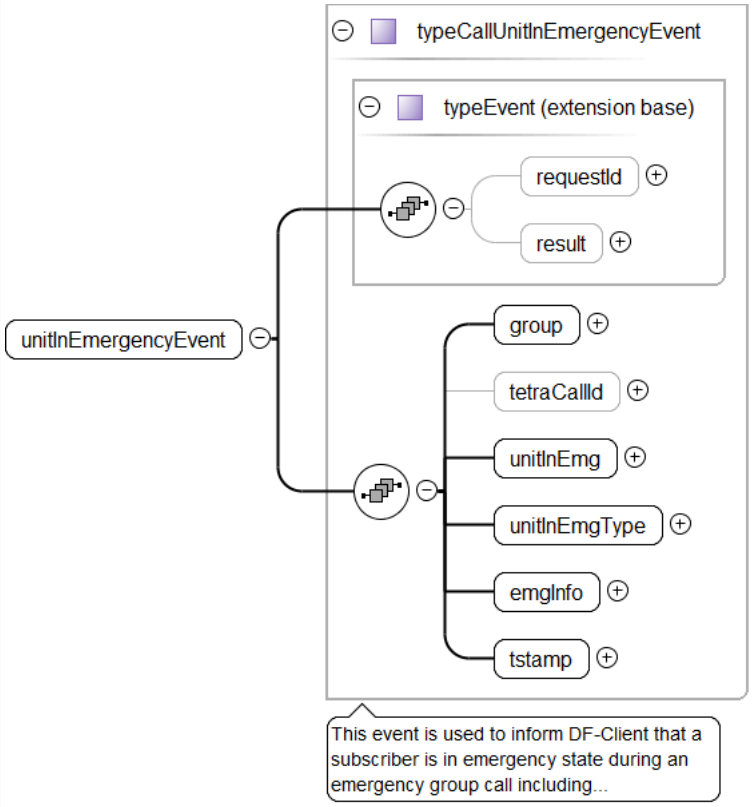
Element typeCallPTTEvent / wait

Namespace	DR-GW
Annotations	This event is used to inform that the call is temporarily paused e.g. if radio subscriber has roamed to a new cell and there are no free resources available.

Diagram	
Type	typeEmpty
Properties	content: complex

Source	`<xs:element name="wait" type="typeEmpty">`
	`  <xs:annotation>`
	`  <xs:documentation>This event is used to inform that the call is temporarily paused e.g. if radio subscriber has roamed to a new cell and there are no free resources available.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

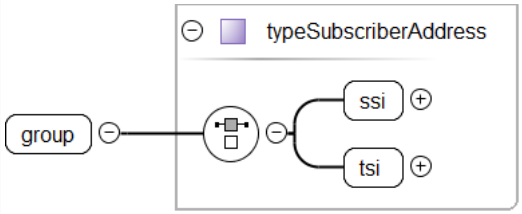
Element `interfaceCall / unitInEmergencyEvent`

Namespace	DR-GW
Diagram	
Type	typeCallUnitInEmergencyEvent
Type hierarchy	• typeEvent • typeCallUnitInEmergencyEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , tetraCallId{0,1} , unitInEmg , unitInEmgType , emgInfo , tstamp
Children	emgInfo, group, requestId, result, tetraCallId, tstamp, unitInEmg, unitInEmgType
Instance	`<unitInEmergencyEvent xmlns="DR-GW">` `  <requestId>{0,1}</requestId>`

```
  <result>{0,1}</result>
  <group>{1,1}</group>
  <tetraCallId>{0,1}</tetraCallId>
  <unitInEmg>{1,1}</unitInEmg>
  <unitInEmgType>{1,1}</unitInEmgType>
  <emgInfo>{1,1}</emgInfo>
  <tstamp>{1,1}</tstamp>
</unitInEmergencyEvent>
```

Source	`<xs:element name="unitInEmergencyEvent" type="typeCallUnitInEmergencyEvent"/>`

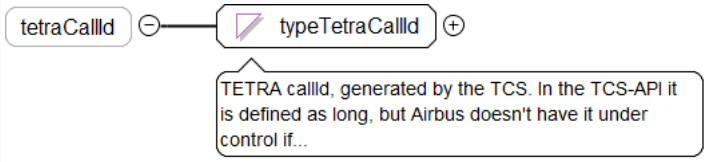
Element typeCallUnitInEmergencyEvent / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<group xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</group>`

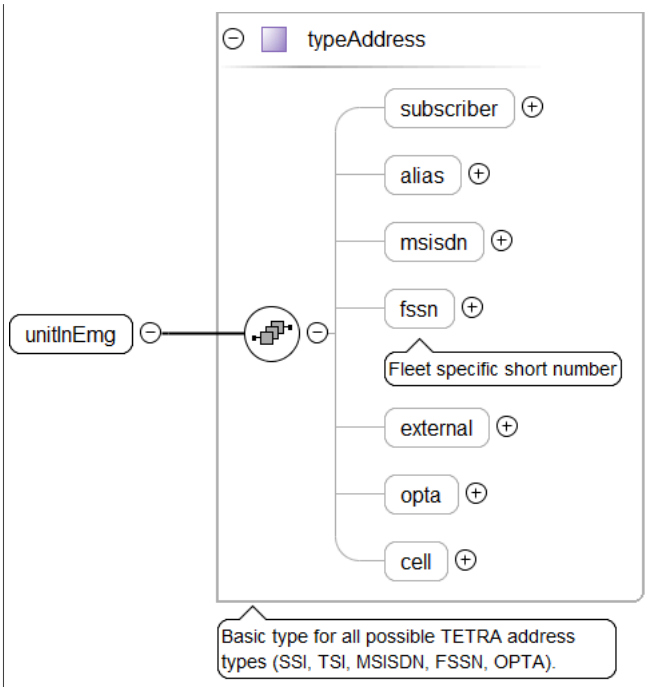
Source	`<xs:element name="group" type="typeSubscriberAddress"/>`

Element typeCallUnitInEmergencyEvent / tetraCallId

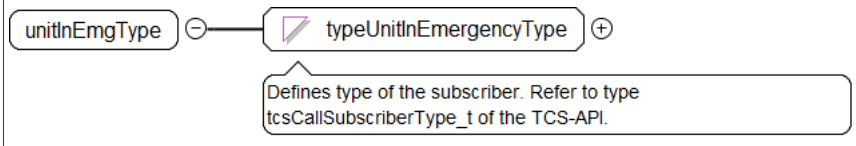
Namespace	DR-GW
Diagram	
Type	typeTetraCallId
Properties	content: simple minOccurs: 0
Source	`<xs:element name="tetraCallId" type="typeTetraCallId" minOccurs="0"/>`

Element typeCallUnitInEmergencyEvent / unitInEmg

Namespace	DR-GW

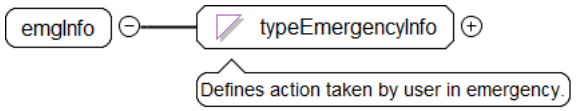
Diagram	
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<unitInEmg xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </unitInEmg>
Source	<xs:element name="unitInEmg" type="typeAddress"/>

Element typeCallUnitInEmergencyEvent / unitInEmgType

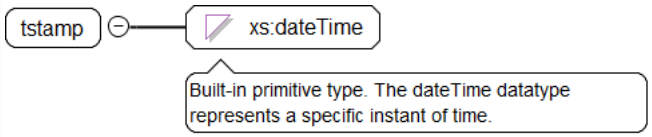
Namespace	DR-GW
Diagram	
Type	typeUnitInEmergencyType
Properties	content: simple
Facets	enumeration dummy enumeration ms enumeration g4wif enumeration external enumeration ws
Source	<xs:element name="unitInEmgType" type="typeUnitInEmergencyType"/>

Element typeCallUnitInEmergencyEvent / emgInfo

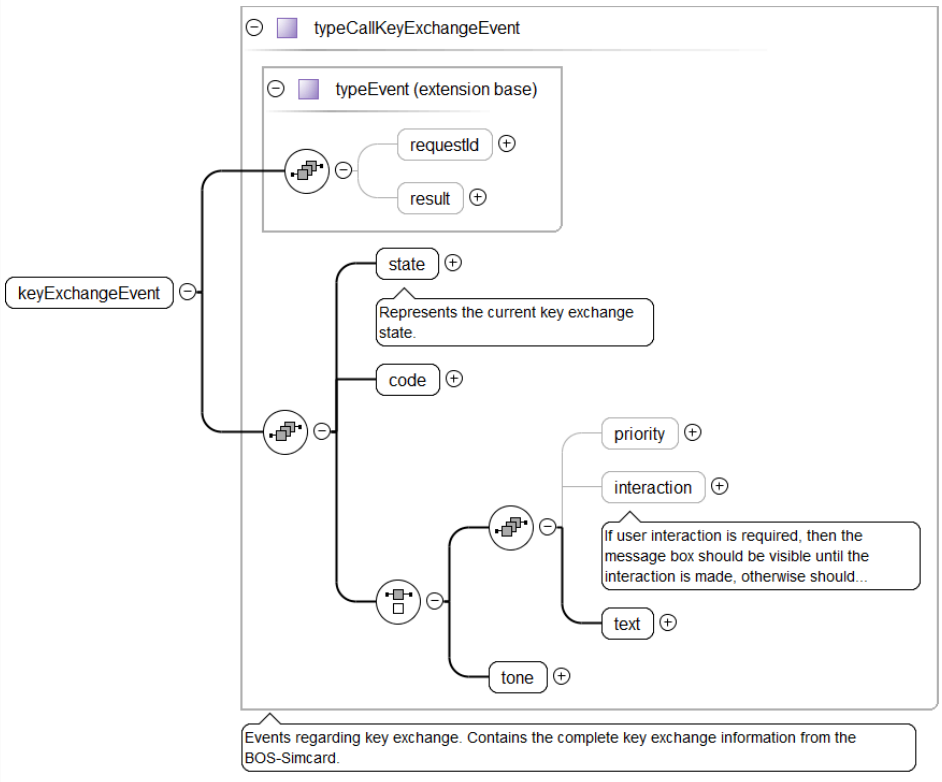
Namespace	DR-GW

Diagram	
Type	typeEmergencyInfo
Properties	content: simple
Facets	enumeration addTx enumeration add enumeration ceased enumeration demandTx enumeration removed enumeration emergencyCallDisconnected
Source	`<xs:element name="emgInfo" type="typeEmergencyInfo"/>`

Element `typeCallUnitInEmergencyEvent / tstamp`

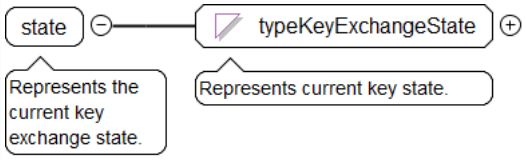
Namespace	DR-GW
Diagram	
Type	xs:dateTime
Properties	content: simple
Source	`<xs:element name="tstamp" type="xs:dateTime"/>`

Element `interfaceCall / keyExchangeEvent`

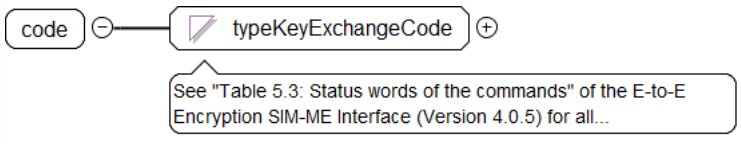
Namespace	DR-GW
Diagram	
Type	typeCallKeyExchangeEvent
Type hierarchy	• typeEvent

	• typeCallKeyExchangeEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , state , code , ((priority{0,1} , interaction{0,1} , text) | tone)
Children	code, interaction, priority, requestId, result, state, text, tone
Instance	<keyExchangeEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <state>{1,1}</state> <code>{1,1}</code> <priority>{0,1}</priority> <interaction>{0,1}</interaction> <text>{1,1}</text> <tone>{1,1}</tone> </keyExchangeEvent>
Source	<xs:element name="keyExchangeEvent" type="typeCallKeyExchangeEvent"/>

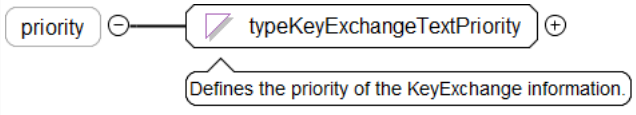
Element typeCallKeyExchangeEvent / state

Namespace	DR-GW		
Annotations	Represents the current key exchange state.		
Diagram			
Type	typeKeyExchangeState		
Properties	content:	simple	
Facets	enumeration	keyValid	current key is valid, no user action required.
	enumeration	keyInvalid	Key invalid, user must request key exchange.
	enumeration	keyExchangeInProgress	Key exchange in progress, user may abort exchange or wait until it gets finished.
Source	<xs:element name="state" type="typeKeyExchangeState"> <xs:annotation> <xs:documentation>Represents the current key exchange state.</xs:documentation> </xs:annotation> </xs:element>		

Element typeCallKeyExchangeEvent / code

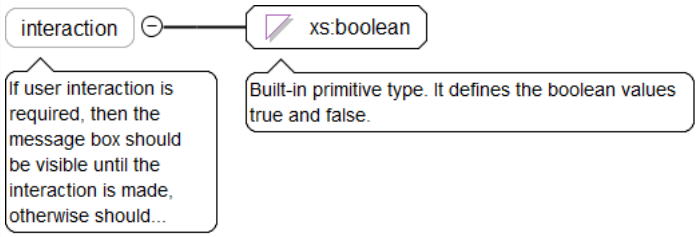
Namespace	DR-GW
Diagram	
Type	typeKeyExchangeCode
Properties	content: simple
Facets	length 2
Source	<xs:element name="code" type="typeKeyExchangeCode"/>

Element typeCallKeyExchangeEvent / priority

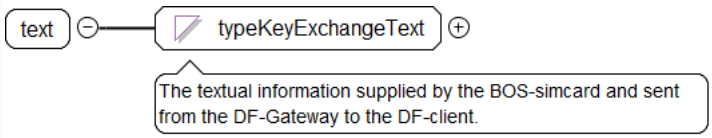
Namespace	DR-GW
Diagram	
Type	typeKeyExchangeTextPriority

Properties	content:	simple
	minOccurs:	0
	default:	normal
Facets	enumeration	normal
	enumeration	high
Source	<xs:element name="priority" type="typeKeyExchangeTextPriority" minOccurs="0" default="normal"/>	

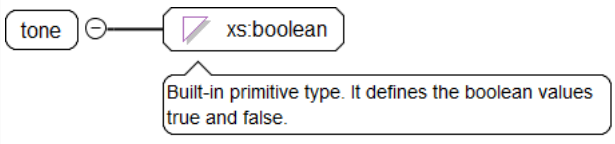
Element typeCallKeyExchangeEvent / interaction

Namespace	DR-GW
Annotations	If user interaction is required, then the message box should be visible until the interaction is made, otherwise should be hidden after delay.
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: false
Source	<xs:element name="interaction" type="xs:boolean" minOccurs="0" default="false"> <xs:annotation> <xs:documentation>If user interaction is required, then the message box should be visible until the interaction is made, otherwise should be hidden after delay.</xs:documentation> </xs:annotation> </xs:element>

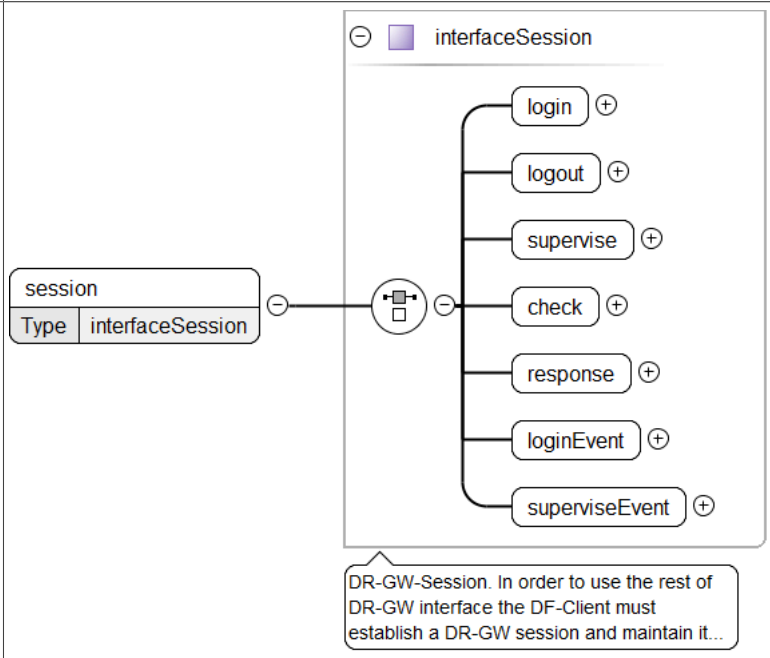
Element typeCallKeyExchangeEvent / text

Namespace	DR-GW
Diagram	
Type	typeKeyExchangeText
Properties	content: simple
Facets	maxLength 100
Source	<xs:element name="text" type="typeKeyExchangeText"/>

Element typeCallKeyExchangeEvent / tone

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple fixed: true
Source	<xs:element name="tone" type="xs:boolean" fixed="true"/>

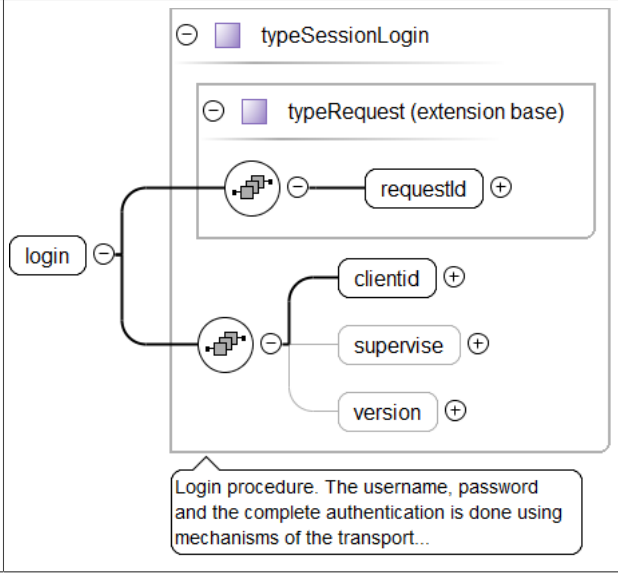
Element `drgw / session`

Namespace	DR-GW
Diagram	
Type	interfaceSession
Properties	content: `complex`
Model	login | logout | supervise | check | response | loginEvent | superviseEvent
Children	check, login, loginEvent, logout, response, supervise, superviseEvent

Instance	`<session xmlns="DR-GW">`
	`  <login>{1,1}</login>`
	`  <logout>{1,1}</logout>`
	`  <supervise>{1,1}</supervise>`
	`  <check>{1,1}</check>`
	`  <response>{1,1}</response>`
	`  <loginEvent>{1,1}</loginEvent>`
	`  <superviseEvent>{1,1}</superviseEvent>`
	`</session>`

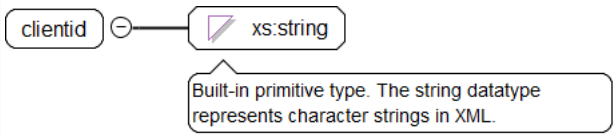
Source	`<xs:element name="session" type="interfaceSession"/>`

Element `interfaceSession / login`

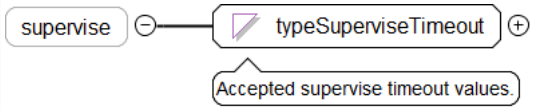
Namespace	DR-GW
Diagram	

Type	typeSessionLogin
Type hierarchy	• typeRequest • typeSessionLogin
Properties	content: complex
Model	requestId , clientid , supervise{0,1} , version{0,1}
Children	clientid, requestId, supervise, version
Instance	<login xmlns="DR-GW"> <requestId>{1,1}</requestId> <clientid>{1,1}</clientid> <supervise>{0,1}</supervise> <version>{0,1}</version> </login>
Source	<xs:element name="login" type="typeSessionLogin"/>

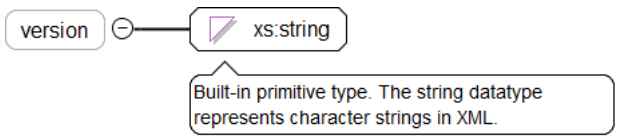
Element typeSessionLogin / clientid

Namespace	DR-GW
Diagram	
Type	xs:string
Properties	content: simple
Source	<xs:element name="clientid" type="xs:string"/>

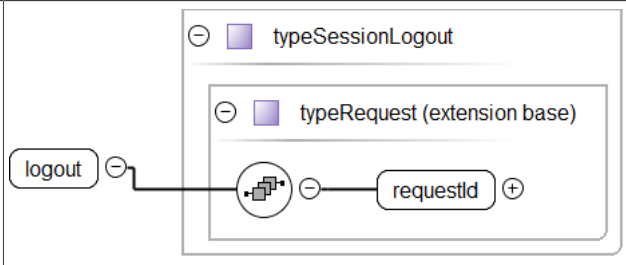
Element typeSessionLogin / supervise

Namespace	DR-GW
Diagram	
Type	typeSuperviseTimeout
Properties	content: simple minOccurs: 0 default: 60
Facets	enumeration 20 enumeration 30 enumeration 60
Source	<xs:element name="supervise" type="typeSuperviseTimeout" default="60" minOccurs="0"/>

Element typeSessionLogin / version

Namespace	DR-GW
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0
Source	<xs:element name="version" type="xs:string" minOccurs="0"/>

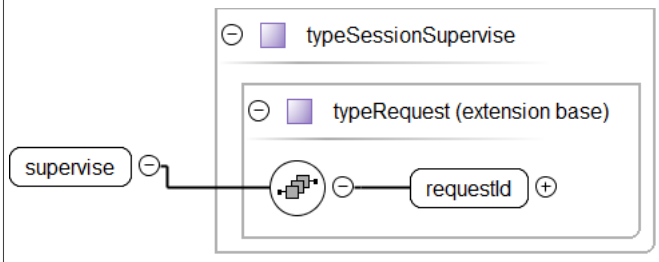
Element `interfaceSession / logout`

Namespace	DR-GW
Diagram	
Type	typeSessionLogout
Type hierarchy	• typeRequest • typeSessionLogout
Properties	content: complex
Model	requestId
Children	requestId

Instance	`<logout xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`</logout>`

Source	`<xs:element name="logout" type="typeSessionLogout"/>`

Element `interfaceSession / supervise`

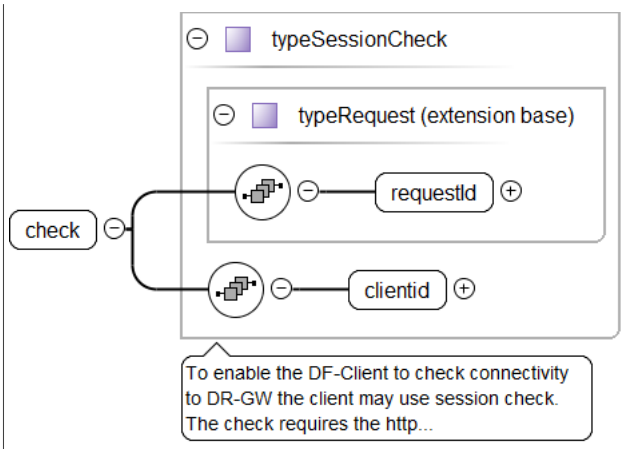
Namespace	DR-GW
Diagram	
Type	typeSessionSupervise
Type hierarchy	• typeRequest • typeSessionSupervise
Properties	content: complex
Model	requestId
Children	requestId

Instance	`<supervise xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`</supervise>`

Source	`<xs:element name="supervise" type="typeSessionSupervise"/>`

Element `interfaceSession / check`

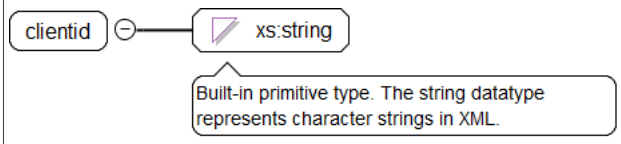
Namespace	DR-GW

Diagram	
Type	typeSessionCheck
Type hierarchy	• typeRequest • typeSessionCheck
Properties	content: complex
Model	requestId , clientid
Children	clientid, requestId

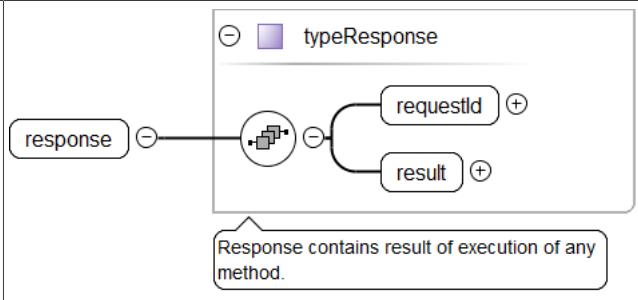
Instance	`<check xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <clientid>{1,1}</clientid>`
	`</check>`

Source	`<xs:element name="check" type="typeSessionCheck"/>`

Element `typeSessionCheck / clientid`

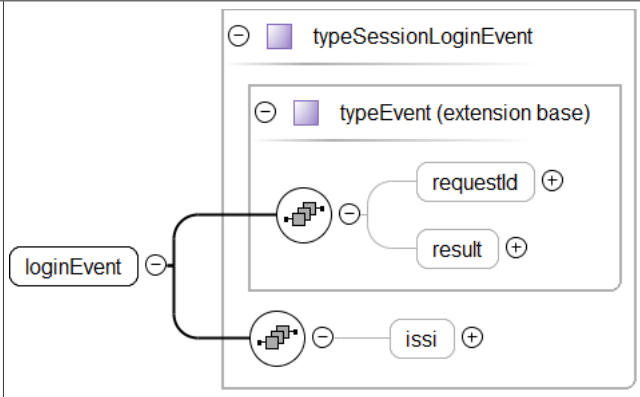
Namespace	DR-GW
Diagram	
Type	xs:string
Properties	content: simple
Source	`<xs:element name="clientid" type="xs:string"/>`

Element `interfaceSession / response`

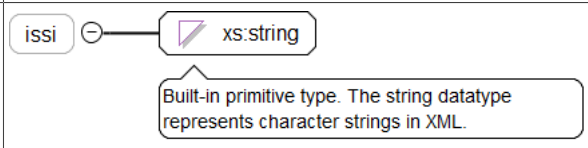
Namespace	DR-GW
Diagram	
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	`<response xmlns="DR-GW">` `  <requestId>{1,1}</requestId>`

	<result>{1,1}</result> </response>
Source	<xs:element name="response" type="typeResponse"/>

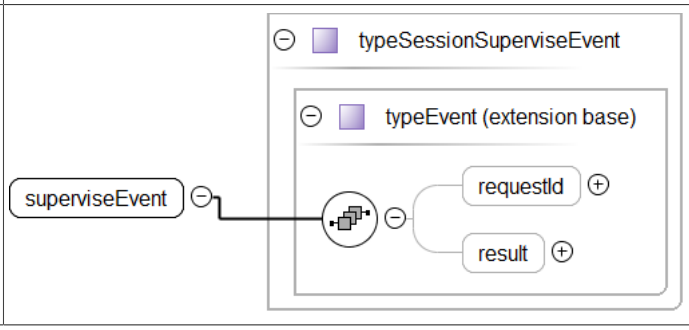
Element interfaceSession / loginEvent

Namespace	DR-GW
Diagram	
Type	typeSessionLoginEvent
Type hierarchy	• typeEvent • typeSessionLoginEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , issi{0,1}
Children	issi, requestId, result
Instance	<loginEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <issi>{0,1}</issi> </loginEvent>
Source	<xs:element name="loginEvent" type="typeSessionLoginEvent"/>

Element typeSessionLoginEvent / issi

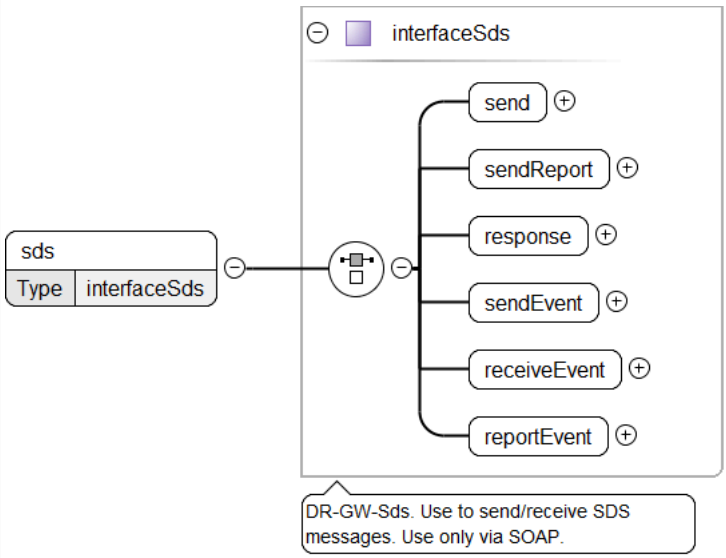
Namespace	DR-GW
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0
Source	<xs:element name="issi" type="xs:string" minOccurs="0"/>

Element interfaceSession / superviseEvent

Namespace	DR-GW
Diagram	

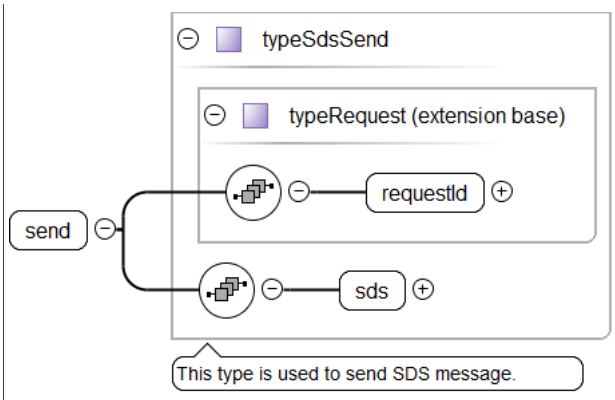
Type	typeSessionSuperviseEvent
Type hierarchy	• typeEvent • typeSessionSuperviseEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1}
Children	requestId, result
Instance	<superviseEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> </superviseEvent>
Source	<xs:element name="superviseEvent" type="typeSessionSuperviseEvent"/>

Element drgw / sds

Namespace	DR-GW
Diagram	
Type	interfaceSds
Properties	content: complex
Model	send | sendReport | response | sendEvent | receiveEvent | reportEvent
Children	receiveEvent, reportEvent, response, send, sendEvent, sendReport
Instance	<sds xmlns="DR-GW"> <send>{1,1}</send> <sendReport>{1,1}</sendReport> <response>{1,1}</response> <sendEvent>{1,1}</sendEvent> <receiveEvent>{1,1}</receiveEvent> <reportEvent>{1,1}</reportEvent> </sds>
Source	<xs:element name="sds" type="interfaceSds"/>

Element interfaceSds / send

Namespace	DR-GW

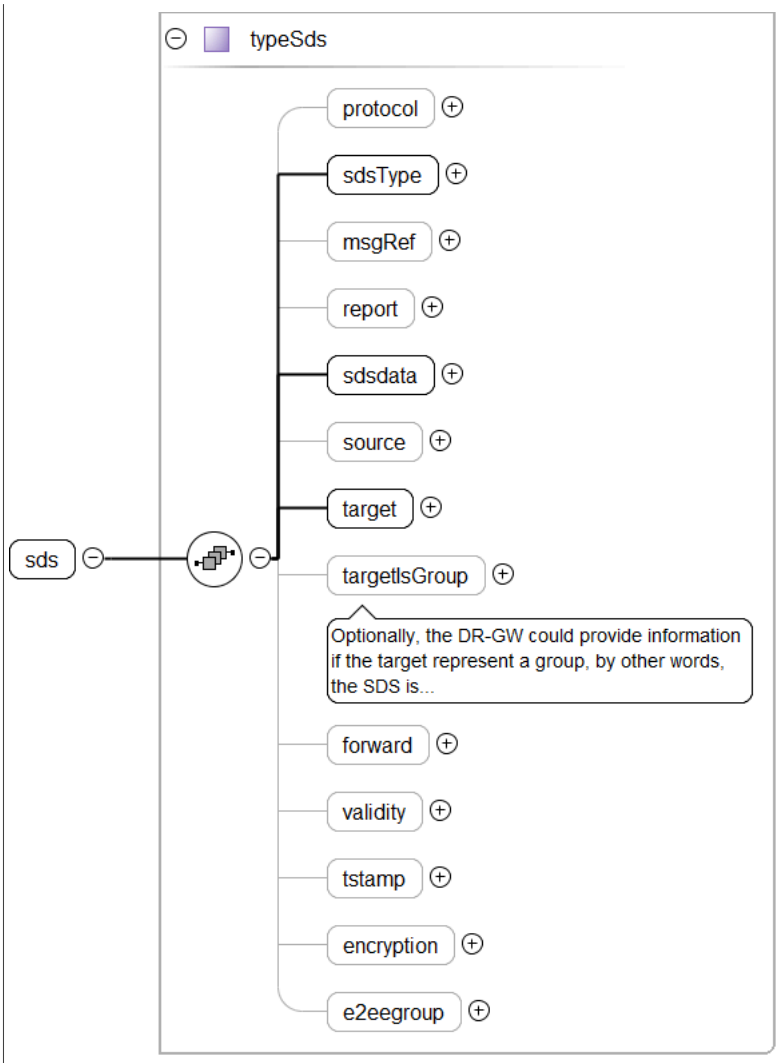
Diagram	
Type	typeSdsSend
Type hierarchy	• typeRequest • typeSdsSend
Properties	content: complex
Model	requestId , sds
Children	requestId, sds

Instance	`<send xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <sds>{1,1}</sds>`
	`</send>`

Source	`<xs:element name="send" type="typeSdsSend"/>`

Element typeSdsSend / sds

Namespace	DR-GW

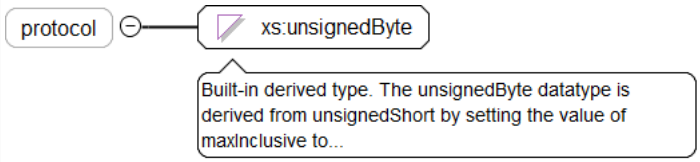
Diagram	
Type	typeSds
Properties	content: complex
Model	protocol{0,1} , sdsType , msgRef{0,1} , report{0,1} , sdsdata , source{0,1} , target , targetIsGroup{0,1} , forward{0,1} , validity{0,1} , tstamp{0,1} , encryption{0,1} , e2eegroup{0,1}
Children	e2eegroup, encryption, forward, msgRef, protocol, report, sdsType, sdsdata, source, target, targetIsGroup, tstamp, validity

Instance	<pre><sds xmlns="DR-GW">
	  <protocol>{0,1}</protocol>
	  <sdsType>{1,1}</sdsType>
	  <msgRef>{0,1}</msgRef>
	  <report>{0,1}</report>
	  <sdsdata>{1,1}</sdsdata>
	  <source>{0,1}</source>
	  <target>{1,1}</target>
	  <targetIsGroup>{0,1}</targetIsGroup>
	  <forward>{0,1}</forward>
	  <validity>{0,1}</validity>
	  <tstamp>{0,1}</tstamp>
	  <encryption>{0,1}</encryption>
	  <e2eegroup>{0,1}</e2eegroup>
	</sds></pre>

Source	<pre><xs:element name="sds" type="typeSds"/></pre>

Element typeSds / protocol

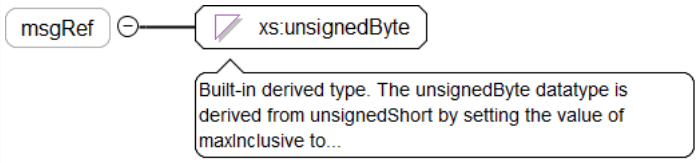
Namespace	DR-GW

Diagram	
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0
Source	`<xs:element name="protocol" type="xs:unsignedByte" minOccurs="0"/>`

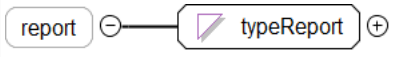
Element typeSds / sdsType

Namespace	DR-GW
Diagram	
Type	typeSdsType
Properties	content: simple
Facets	enumeration 0 SDS1. enumeration 1 SDS2. enumeration 2 SDS3. enumeration 3 SDS4. enumeration 4 SDS-TL. enumeration 5 Status.
Source	`<xs:element name="sdsType" type="typeSdsType"/>`

Element typeSds / msgRef

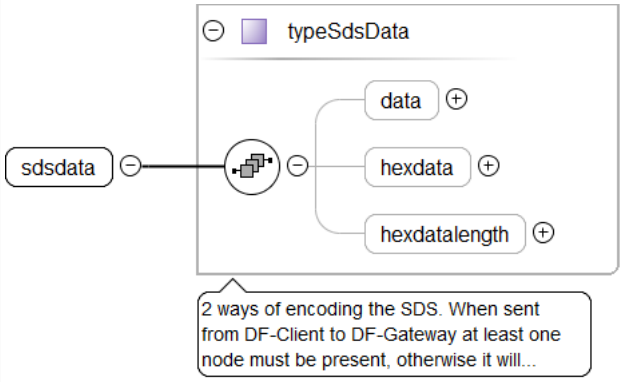
Namespace	DR-GW
Diagram	
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0
Source	`<xs:element name="msgRef" type="xs:unsignedByte" minOccurs="0"/>`

Element typeSds / report

Namespace	DR-GW
Diagram	
Type	typeReport
Properties	content: simple minOccurs: 0 default: none
Facets	enumeration none enumeration delivery enumeration consume enumeration both

Source	`<xs:element name="report" type="typeReport" default="none" minOccurs="0"/>`

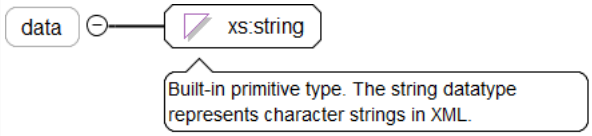
Element typeSds / sdsdata

Namespace	DR-GW
Diagram	
Type	typeSdsData
Properties	content: complex
Model	data{0,1} , hexdata{0,1} , hexdatalength{0,1}
Children	data, hexdata, hexdatalength

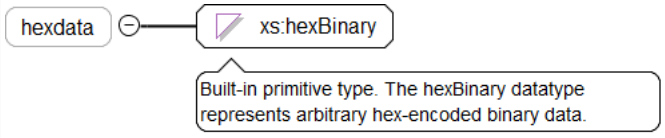
Instance	`<sdsdata xmlns="DR-GW">`
	`  <data>{0,1}</data>`
	`  <hexdata>{0,1}</hexdata>`
	`  <hexdatalength>{0,1}</hexdatalength>`
	`</sdsdata>`

Source	`<xs:element name="sdsdata" type="typeSdsData"/>`

Element typeSdsData / data

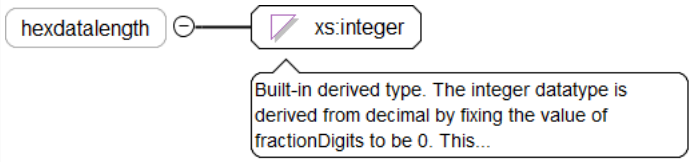
Namespace	DR-GW
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0
Source	`<xs:element name="data" type="xs:string" minOccurs="0"/>`

Element typeSdsData / hexdata

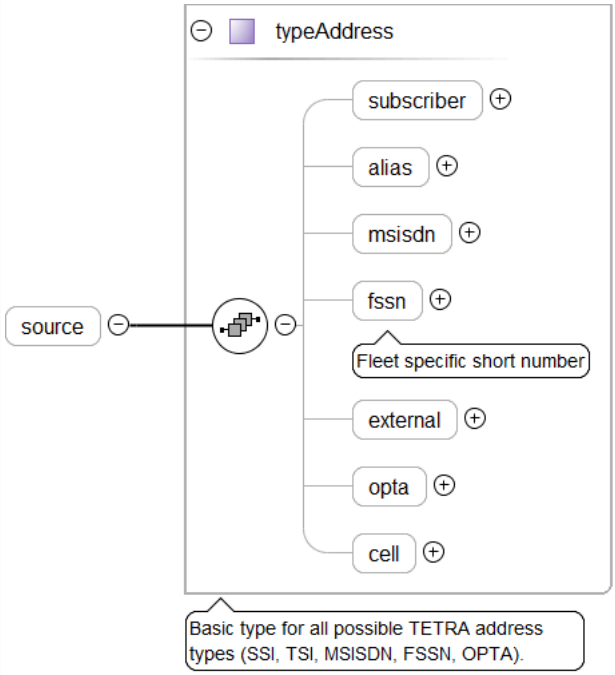
Namespace	DR-GW
Diagram	
Type	xs:hexBinary
Properties	content: simple minOccurs: 0
Source	`<xs:element name="hexdata" type="xs:hexBinary" minOccurs="0"/>`

Element typeSdsData / hexdatalength

Namespace	DR-GW

Diagram	
Type	xs:integer
Properties	content: simple minOccurs: 0
Source	`<xs:element name="hexdatalength" type="xs:integer" minOccurs="0"/>`

Element typeSds / source

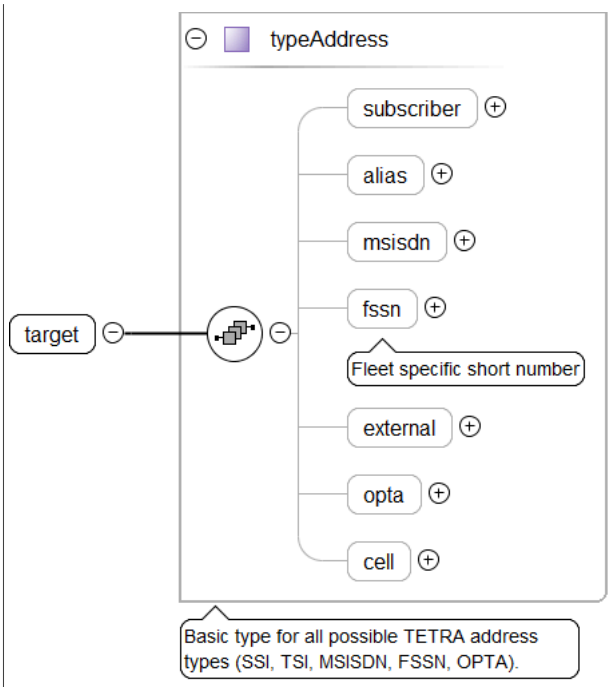
Namespace	DR-GW
Diagram	
Type	typeAddress
Properties	content: complex minOccurs: 0
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber

Instance	`<source xmlns="DR-GW">`
	`  <subscriber>{0,1}</subscriber>`
	`  <alias>{0,1}</alias>`
	`  <msisdn>{0,1}</msisdn>`
	`  <fssn>{0,1}</fssn>`
	`  <external>{0,1}</external>`
	`  <opta>{0,1}</opta>`
	`  <cell>{0,1}</cell>`
	`</source>`

Source	`<xs:element name="source" type="typeAddress" minOccurs="0"/>`

Element typeSds / target

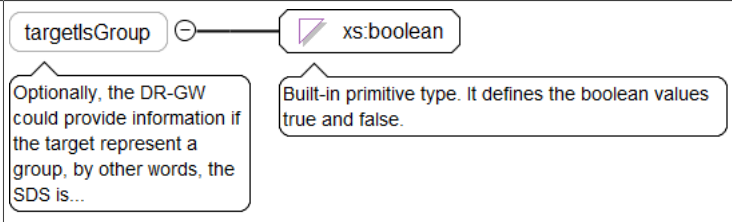
Namespace	DR-GW

Diagram	
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber

Instance	`<target xmlns="DR-GW">`
	`<subscriber>{0,1}</subscriber>`
	`<alias>{0,1}</alias>`
	`<msisdn>{0,1}</msisdn>`
	`<fssn>{0,1}</fssn>`
	`<external>{0,1}</external>`
	`<opta>{0,1}</opta>`
	`<cell>{0,1}</cell>`
	`</target>`

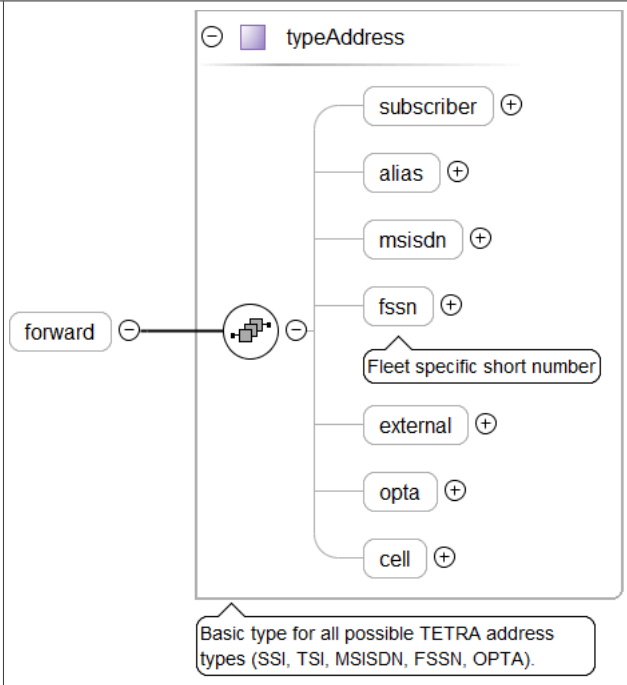
Source	`<xs:element name="target" type="typeAddress"/>`

Element typeSds / targetIsGroup

Namespace	DR-GW
Annotations	Optionally, the DR-GW could provide information if the target represent a group, by other words, the SDS is group-addressed. Also the Df-Client can provide this information to help DR-GW decide which resource to use for actual sending.
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0

Source	`<xs:element name="targetIsGroup" type="xs:boolean" minOccurs="0">`
	`<xs:annotation>`
	`<xs:documentation>Optionally, the DR-GW could provide information if the target represent a group, by other words, the SDS is group-addressed. Also the Df-Client can provide this information to help DR-GW decide which resource to use for actual sending.</xs:documentation>`
	`</xs:annotation>`
	`</xs:element>`

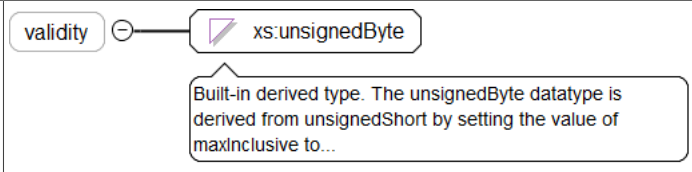
Element typeSds / forward

Namespace	DR-GW
Diagram	
Type	typeAddress
Properties	content: complex minOccurs: 0
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber

Instance	<pre><forward xmlns="DR-GW">
	  <subscriber>{0,1}</subscriber>
	  <alias>{0,1}</alias>
	  <msisdn>{0,1}</msisdn>
	  <fssn>{0,1}</fssn>
	  <external>{0,1}</external>
	  <opta>{0,1}</opta>
	  <cell>{0,1}</cell>
	</forward></pre>

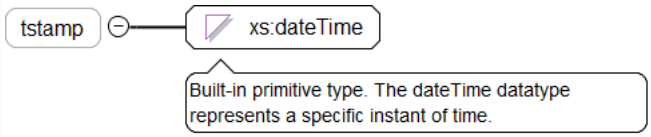
Source	<pre><xs:element name="forward" type="typeAddress" minOccurs="0"/></pre>

Element typeSds / validity

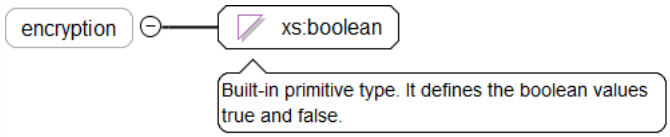
Namespace	DR-GW
Diagram	
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0
Source	<pre><xs:element name="validity" type="xs:unsignedByte" minOccurs="0"/></pre>

Element typeSds / tstamp

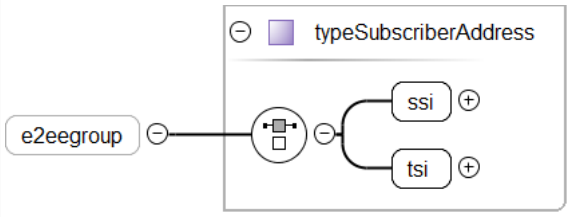
Namespace	DR-GW

Diagram	
Type	xs:dateTime
Properties	content: simple minOccurs: 0
Source	`<xs:element name="tstamp" type="xs:dateTime" minOccurs="0"/>`

Element typeSds / encryption

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: true
Source	`<xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/>`

Element typeSds / e2eegroup

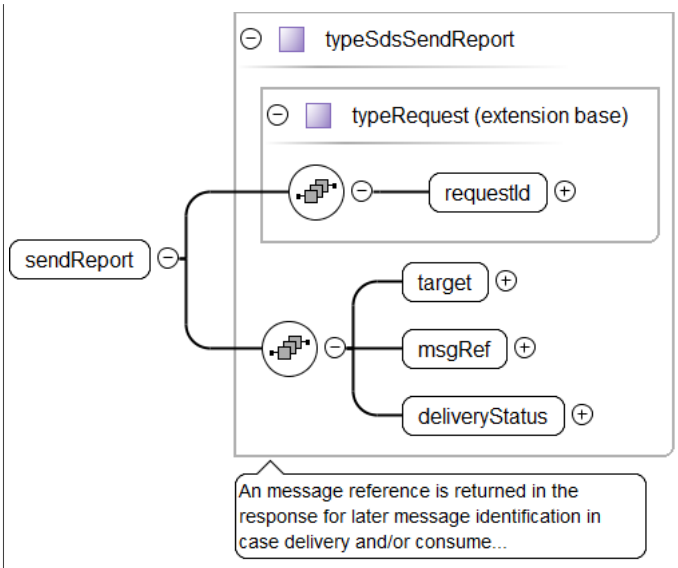
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex minOccurs: 0
Model	ssi | tsi
Children	ssi, tsi

Instance	`<e2eegroup xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</e2eegroup>`

Source	`<xs:element name="e2eegroup" type="typeSubscriberAddress" minOccurs="0"/>`

Element interfaceSds / sendReport

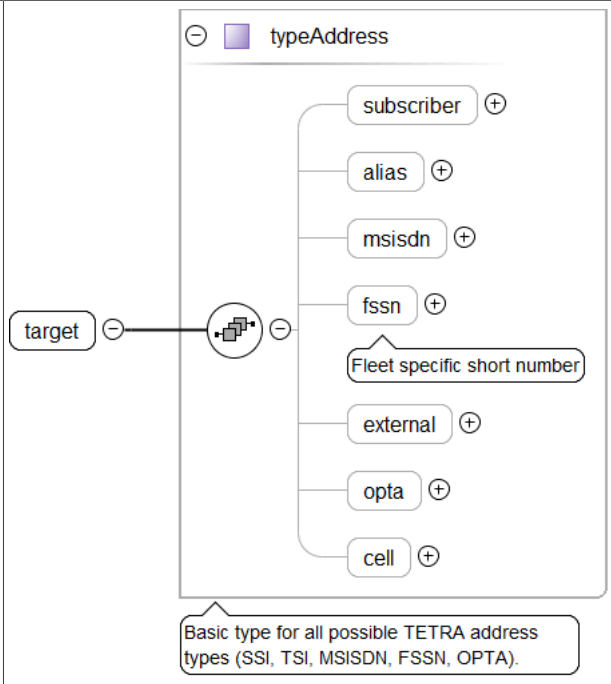
Namespace	DR-GW

Diagram	
Type	typeSdsSendReport
Type hierarchy	• typeRequest • typeSdsSendReport
Properties	content: complex
Model	requestId , target , msgRef , deliveryStatus
Children	deliveryStatus, msgRef, requestId, target

Instance	`<sendReport xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <target>{1,1}</target>`
	`  <msgRef>{1,1}</msgRef>`
	`  <deliveryStatus>{1,1}</deliveryStatus>`
	`</sendReport>`

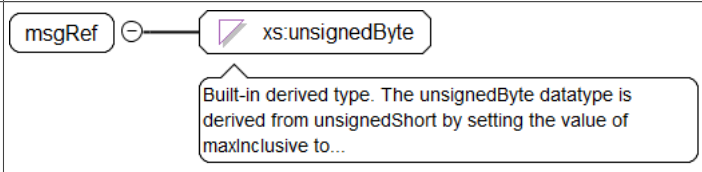
Source	`<xs:element name="sendReport" type="typeSdsSendReport"/>`

Element typeSdsSendReport / target

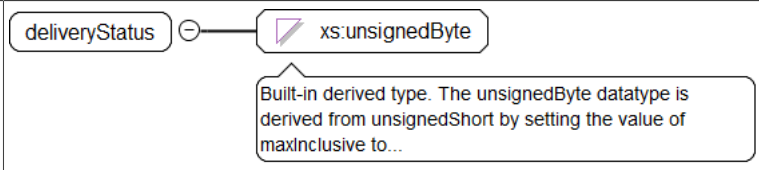
Namespace	DR-GW
Diagram	
Type	typeAddress

Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<target xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </target>
Source	<xs:element name="target" type="typeAddress"/>

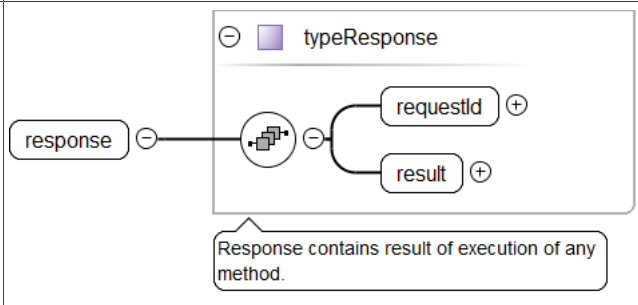
Element typeSdsSendReport / msgRef

Namespace	DR-GW
Diagram	
Type	xs:unsignedByte
Properties	content: simple
Source	<xs:element name="msgRef" type="xs:unsignedByte"/>

Element typeSdsSendReport / deliveryStatus

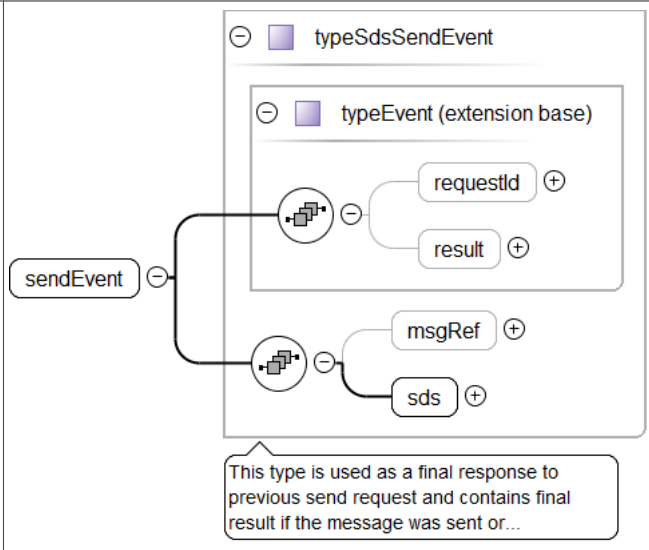
Namespace	DR-GW
Diagram	
Type	xs:unsignedByte
Properties	content: simple
Source	<xs:element name="deliveryStatus" type="xs:unsignedByte"/>

Element interfaceSds / response

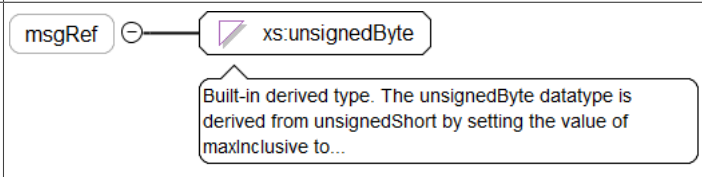
Namespace	DR-GW
Diagram	
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<response xmlns="DR-GW">

	<requestId>{1,1}</requestId> <result>{1,1}</result> </response>
Source	<xs:element name="response" type="typeResponse"/>

Element interfaceSds / sendEvent

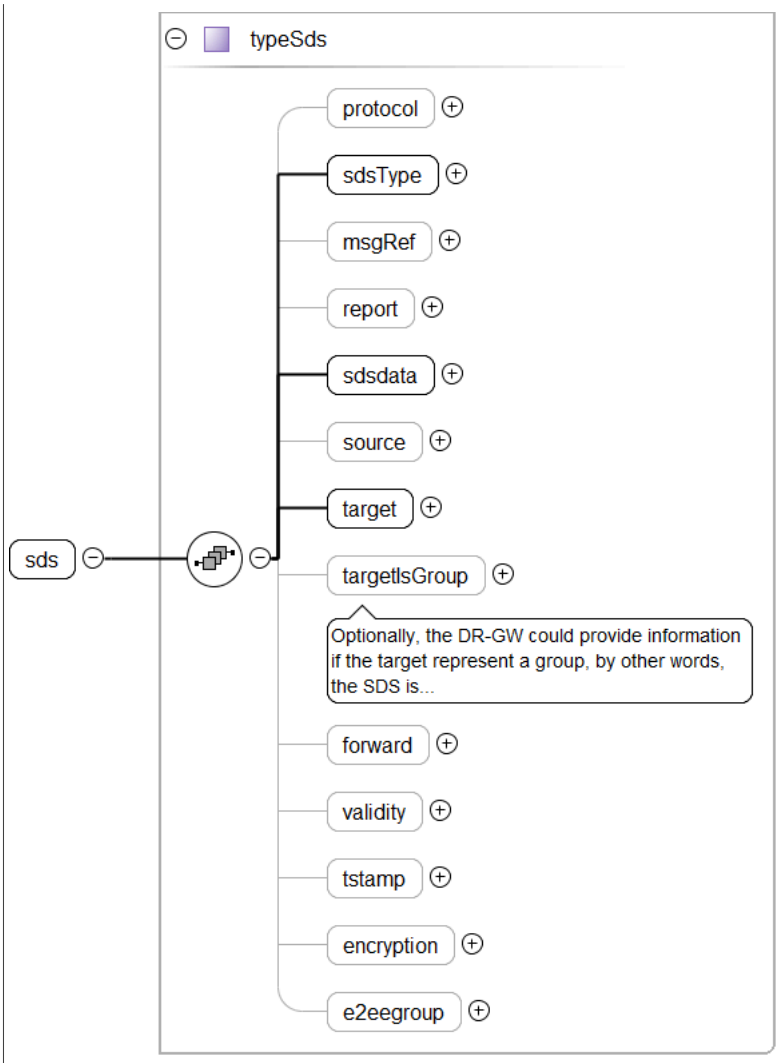
Namespace	DR-GW
Diagram	
Type	typeSdsSendEvent
Type hierarchy	• typeEvent • typeSdsSendEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , msgRef{0,1} , sds
Children	msgRef, requestId, result, sds
Instance	<sendEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <msgRef>{0,1}</msgRef> <sds>{1,1}</sds> </sendEvent>
Source	<xs:element name="sendEvent" type="typeSdsSendEvent"/>

Element typeSdsSendEvent / msgRef

Namespace	DR-GW
Diagram	
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0 default: 0
Source	<xs:element name="msgRef" type="xs:unsignedByte" minOccurs="0" default="0"/>

Element typeSdsSendEvent / sds

Namespace	DR-GW

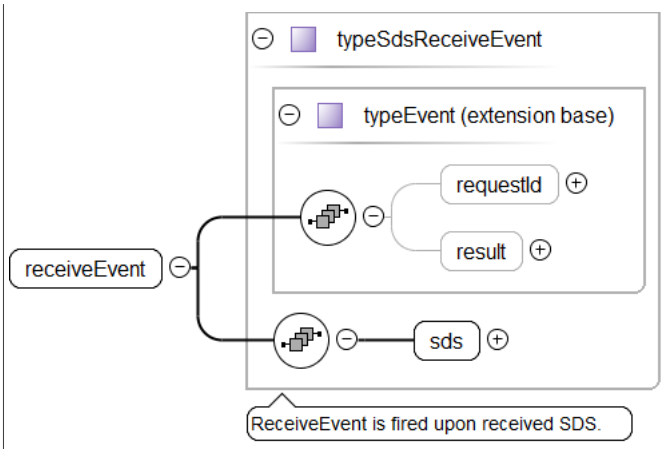
Diagram	
Type	typeSds
Properties	content: complex
Model	protocol{0,1} , sdsType , msgRef{0,1} , report{0,1} , sdsdata , source{0,1} , target , targetIsGroup{0,1} , forward{0,1} , validity{0,1} , tstamp{0,1} , encryption{0,1} , e2eegroup{0,1}
Children	e2eegroup, encryption, forward, msgRef, protocol, report, sdsType, sdsdata, source, target, targetIsGroup, tstamp, validity

Instance	<pre><sds xmlns="DR-GW">
	  <protocol>{0,1}</protocol>
	  <sdsType>{1,1}</sdsType>
	  <msgRef>{0,1}</msgRef>
	  <report>{0,1}</report>
	  <sdsdata>{1,1}</sdsdata>
	  <source>{0,1}</source>
	  <target>{1,1}</target>
	  <targetIsGroup>{0,1}</targetIsGroup>
	  <forward>{0,1}</forward>
	  <validity>{0,1}</validity>
	  <tstamp>{0,1}</tstamp>
	  <encryption>{0,1}</encryption>
	  <e2eegroup>{0,1}</e2eegroup>
	</sds></pre>

Source	<pre><xs:element name="sds" type="typeSds"/></pre>

Element `interfaceSds / receiveEvent`

Namespace	DR-GW

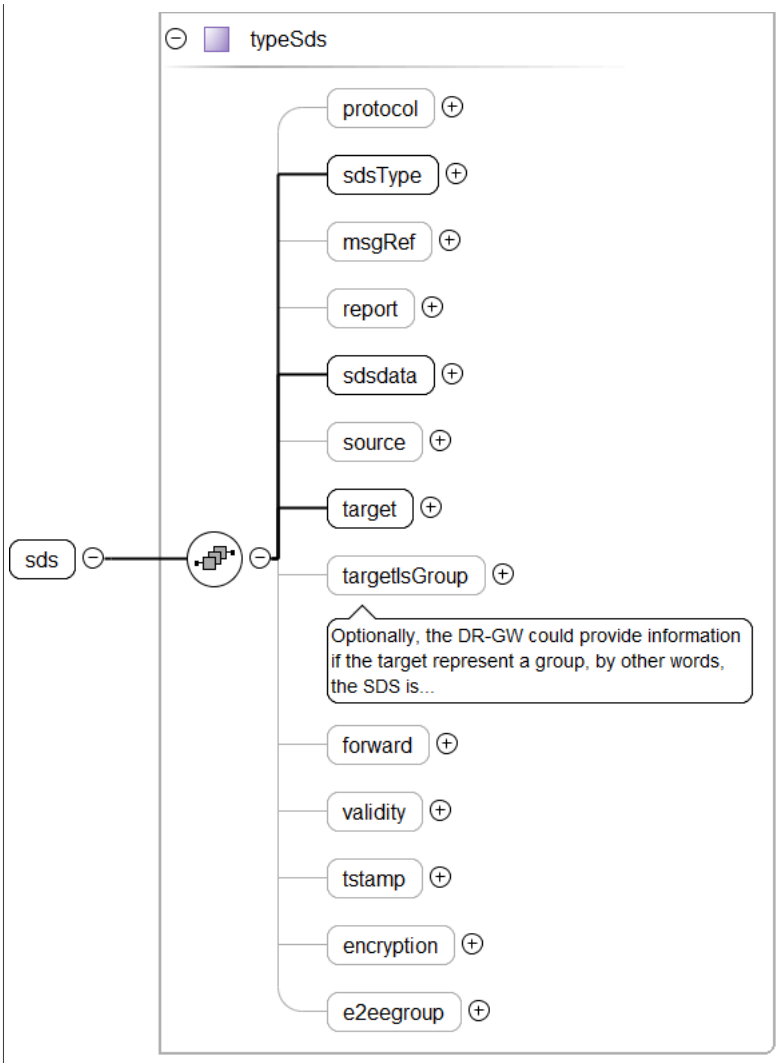
Diagram	
Type	typeSdsReceiveEvent
Type hierarchy	• typeEvent • typeSdsReceiveEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , sds
Children	requestId, result, sds

Instance	`<receiveEvent xmlns="DR-GW">`
	`<requestId>{0,1}</requestId>`
	`<result>{0,1}</result>`
	`<sds>{1,1}</sds>`
	`</receiveEvent>`

Source	`<xs:element name="receiveEvent" type="typeSdsReceiveEvent"/>`

Element typeSdsReceiveEvent / sds

Namespace	DR-GW

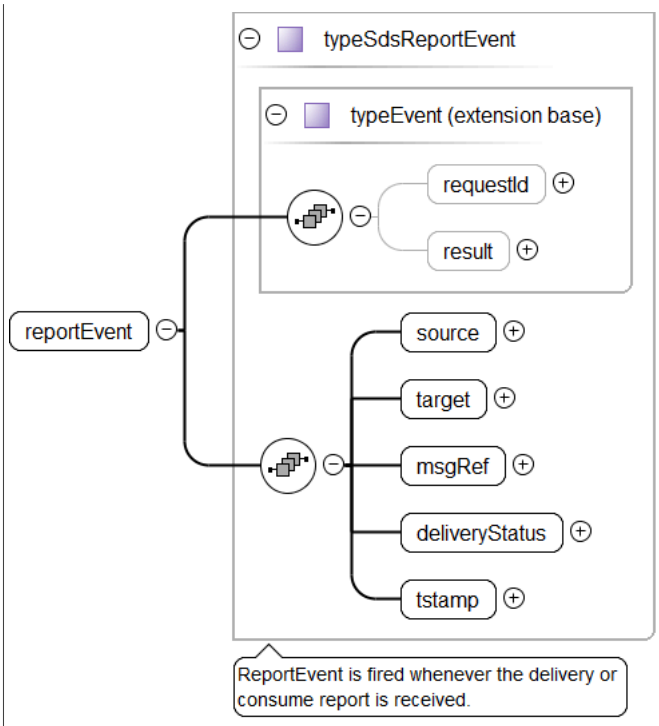
Diagram	
Type	typeSds
Properties	content: complex
Model	protocol{0,1} , sdsType , msgRef{0,1} , report{0,1} , sdsdata , source{0,1} , target , targetIsGroup{0,1} , forward{0,1} , validity{0,1} , tstamp{0,1} , encryption{0,1} , e2eegroup{0,1}
Children	e2eegroup, encryption, forward, msgRef, protocol, report, sdsType, sdsdata, source, target, targetIsGroup, tstamp, validity

Instance	<pre><sds xmlns="DR-GW">
	  <protocol>{0,1}</protocol>
	  <sdsType>{1,1}</sdsType>
	  <msgRef>{0,1}</msgRef>
	  <report>{0,1}</report>
	  <sdsdata>{1,1}</sdsdata>
	  <source>{0,1}</source>
	  <target>{1,1}</target>
	  <targetIsGroup>{0,1}</targetIsGroup>
	  <forward>{0,1}</forward>
	  <validity>{0,1}</validity>
	  <tstamp>{0,1}</tstamp>
	  <encryption>{0,1}</encryption>
	  <e2eegroup>{0,1}</e2eegroup>
	</sds></pre>

Source	<pre><xs:element name="sds" type="typeSds"/></pre>

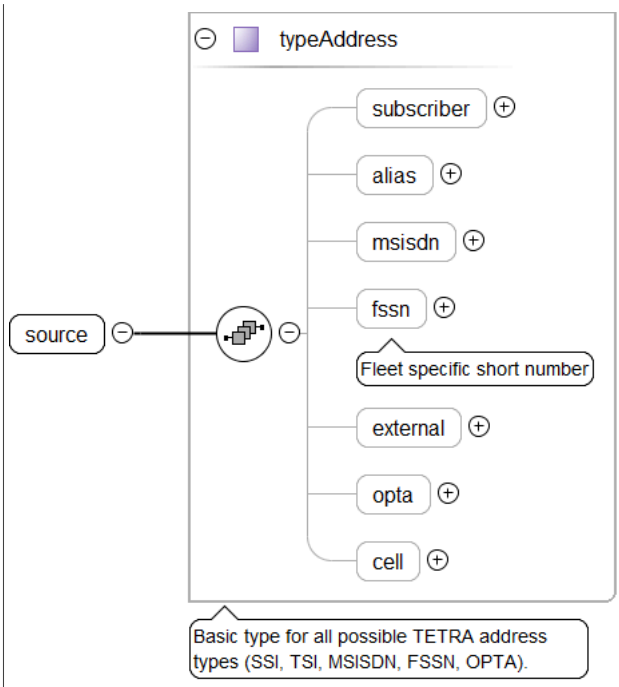
Element `interfaceSds / reportEvent`

Namespace	DR-GW

Diagram	
Type	typeSdsReportEvent
Type hierarchy	• typeEvent • typeSdsReportEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , source , target , msgRef , deliveryStatus , tstamp
Children	deliveryStatus, msgRef, requestId, result, source, target, tstamp
Instance	<reportEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <source>{1,1}</source> <target>{1,1}</target> <msgRef>{1,1}</msgRef> <deliveryStatus>{1,1}</deliveryStatus> <tstamp>{1,1}</tstamp> </reportEvent>
Source	<xs:element name="reportEvent" type="typeSdsReportEvent"/>

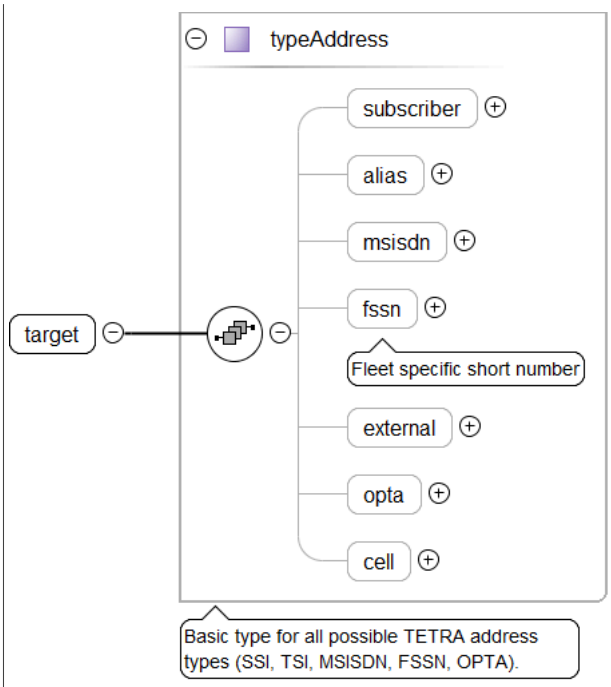
Element typeSdsReportEvent / source

Namespace	DR-GW

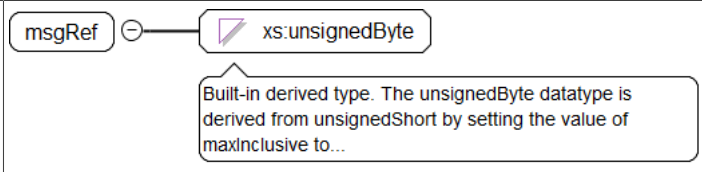
Diagram	
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<source xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </source>
Source	<xs:element name="source" type="typeAddress"/>

Element typeSdsReportEvent / target

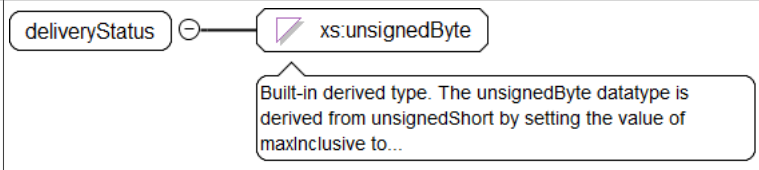
Namespace	DR-GW

Diagram	
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<target xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </target>
Source	<xs:element name="target" type="typeAddress"/>

Element typeSdsReportEvent / msgRef

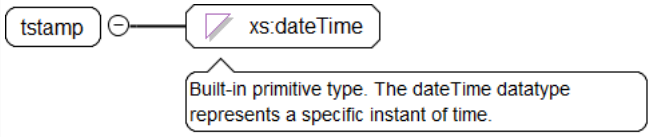
Namespace	DR-GW
Diagram	
Type	xs:unsignedByte
Properties	content: simple
Source	<xs:element name="msgRef" type="xs:unsignedByte"/>

Element typeSdsReportEvent / deliveryStatus

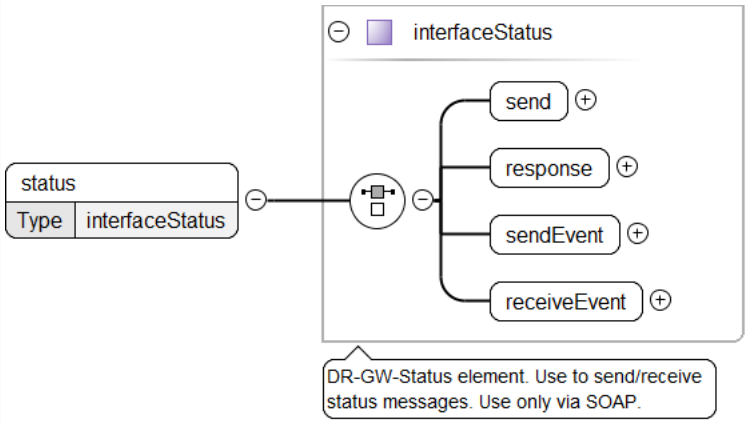
Namespace	DR-GW
Diagram	
Type	xs:unsignedByte

Properties	content: simple
Source	`<xs:element name="deliveryStatus" type="xs:unsignedByte"/>`

Element typeSdsReportEvent / tstamp

Namespace	DR-GW
Diagram	
Type	xs:dateTime
Properties	content: simple
Source	`<xs:element name="tstamp" type="xs:dateTime"/>`

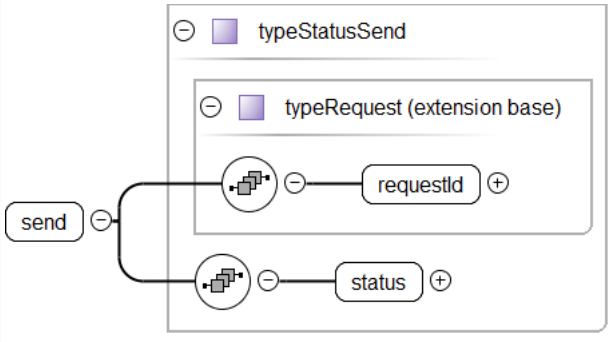
Element drgw / status

Namespace	DR-GW
Diagram	
Type	interfaceStatus
Properties	content: complex
Model	send | response | sendEvent | receiveEvent
Children	receiveEvent, response, send, sendEvent

Instance	`<status xmlns="DR-GW">`
	`  <send>{1,1}</send>`
	`  <response>{1,1}</response>`
	`  <sendEvent>{1,1}</sendEvent>`
	`  <receiveEvent>{1,1}</receiveEvent>`
	`</status>`

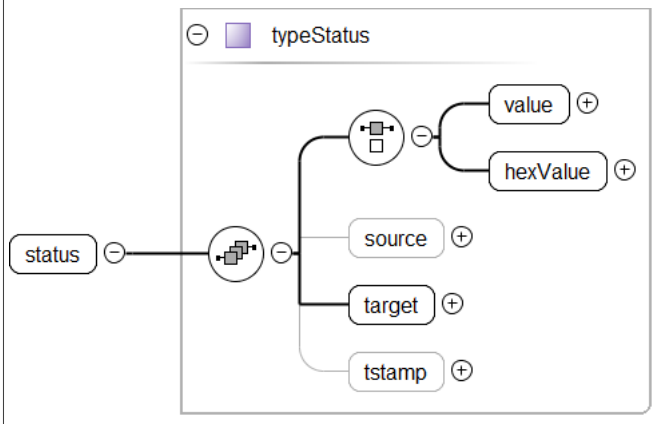
Source	`<xs:element name="status" type="interfaceStatus"/>`

Element interfaceStatus / send

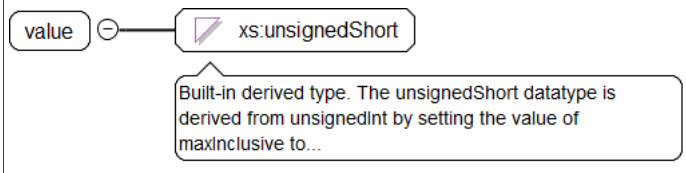
Namespace	DR-GW
Diagram	

Type	typeStatusSend
Type hierarchy	• typeRequest • typeStatusSend
Properties	content: complex
Model	requestId , status
Children	requestId, status
Instance	<send xmlns="DR-GW"> <requestId>{1,1}</requestId> <status>{1,1}</status> </send>
Source	<xs:element name="send" type="typeStatusSend"/>

Element typeStatusSend / status

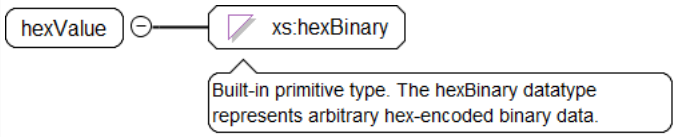
Namespace	DR-GW
Diagram	
Type	typeStatus
Properties	content: complex
Model	(value | hexValue) , source{0,1} , target , tstamp{0,1}
Children	hexValue, source, target, tstamp, value
Instance	<status xmlns="DR-GW"> <value>{1,1}</value> <hexValue>{1,1}</hexValue> <source>{0,1}</source> <target>{1,1}</target> <tstamp>{0,1}</tstamp> </status>
Source	<xs:element name="status" type="typeStatus"/>

Element typeStatus / value

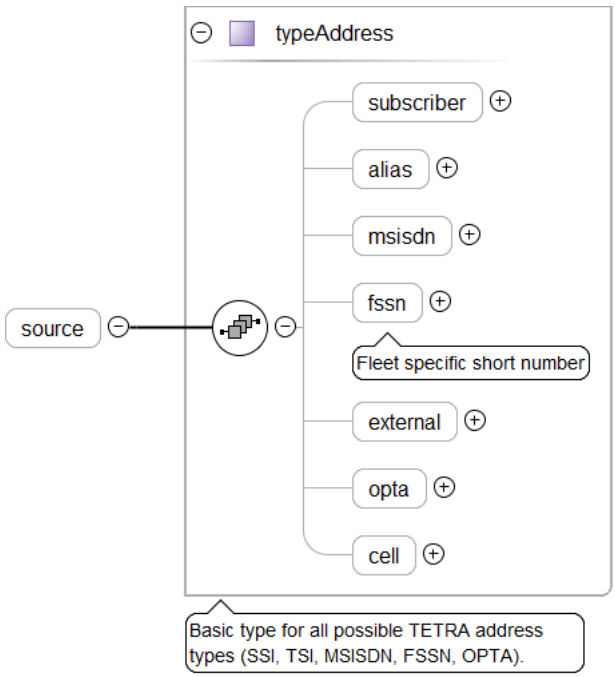
Namespace	DR-GW
Diagram	
Type	xs:unsignedShort
Properties	content: simple
Source	<xs:element name="value" type="xs:unsignedShort"/>

Element typeStatus / hexValue

Namespace	DR-GW

Diagram	
Type	xs:hexBinary
Properties	content: simple
Source	`<xs:element name="hexValue" type="xs:hexBinary"/>`

Element typeStatus / source

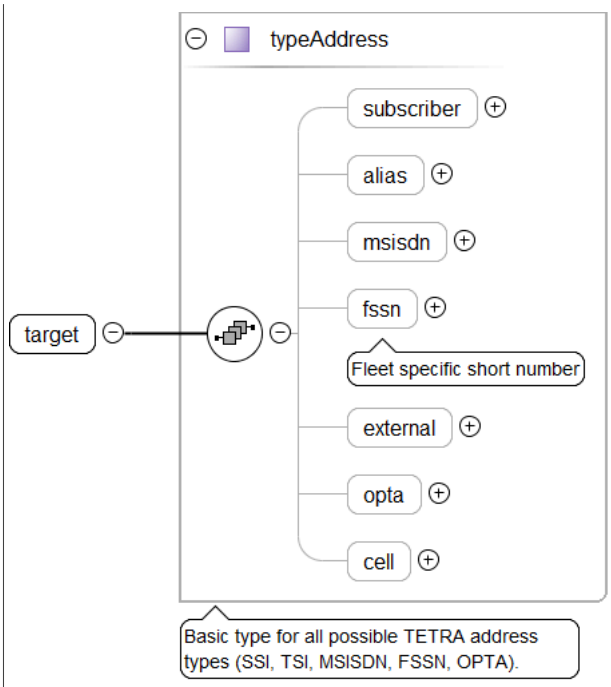
Namespace	DR-GW
Diagram	
Type	typeAddress
Properties	content: complex minOccurs: 0
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber

Instance	`<source xmlns="DR-GW">`
	`  <subscriber>{0,1}</subscriber>`
	`  <alias>{0,1}</alias>`
	`  <msisdn>{0,1}</msisdn>`
	`  <fssn>{0,1}</fssn>`
	`  <external>{0,1}</external>`
	`  <opta>{0,1}</opta>`
	`  <cell>{0,1}</cell>`
	`</source>`

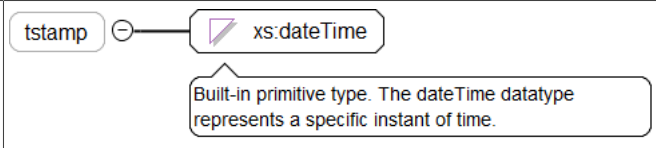
Source	`<xs:element name="source" type="typeAddress" minOccurs="0"/>`

Element typeStatus / target

Namespace	DR-GW

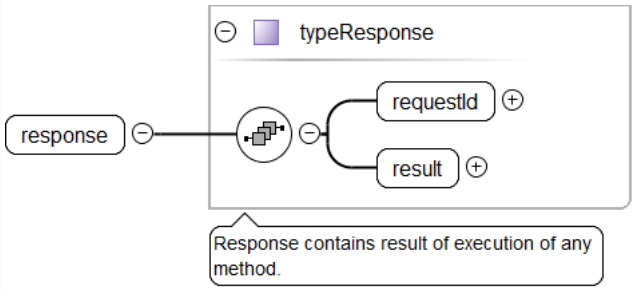
Diagram	
Type	typeAddress
Properties	content: complex
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<target xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </target>
Source	<xs:element name="target" type="typeAddress"/>

Element typeStatus / tstamp

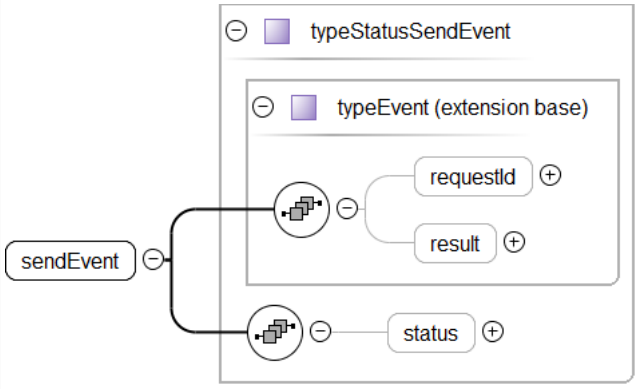
Namespace	DR-GW
Diagram	
Type	xs:dateTime
Properties	content: simple minOccurs: 0
Source	<xs:element name="tstamp" type="xs:dateTime" minOccurs="0"/>

Element interfaceStatus / response

Namespace	DR-GW

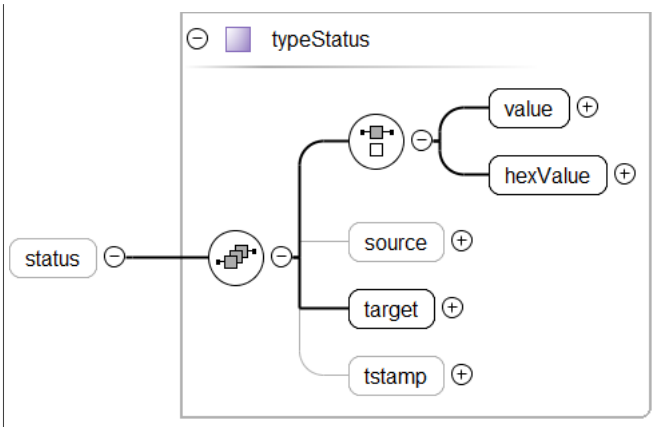
Diagram	
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response>
Source	<xs:element name="response" type="typeResponse"/>

Element interfaceStatus / sendEvent

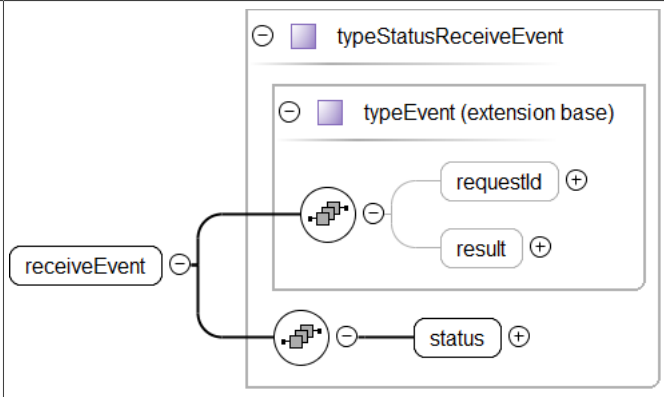
Namespace	DR-GW
Diagram	
Type	typeStatusSendEvent
Type hierarchy	• typeEvent • typeStatusSendEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , status{0,1}
Children	requestId, result, status
Instance	<sendEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <status>{0,1}</status> </sendEvent>
Source	<xs:element name="sendEvent" type="typeStatusSendEvent"/>

Element typeStatusSendEvent / status

Namespace	DR-GW

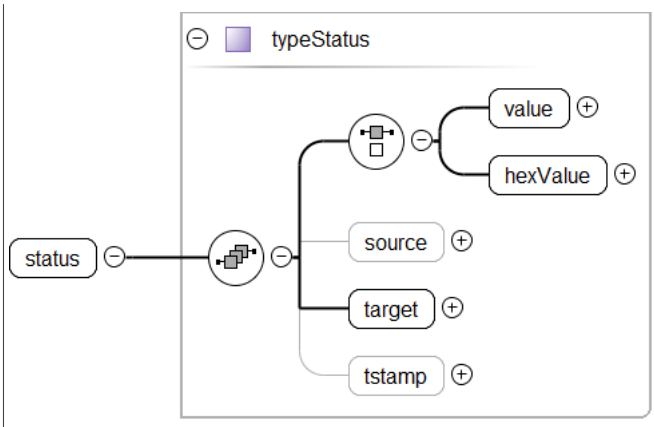
Diagram	
Type	typeStatus
Properties	content: complex minOccurs: 0
Model	(value | hexValue) , source{0,1} , target , tstamp{0,1}
Children	hexValue, source, target, tstamp, value
Instance	<status xmlns="DR-GW"> <value>{1,1}</value> <hexValue>{1,1}</hexValue> <source>{0,1}</source> <target>{1,1}</target> <tstamp>{0,1}</tstamp> </status>
Source	<xs:element name="status" type="typeStatus" minOccurs="0"/>

Element interfaceStatus / receiveEvent

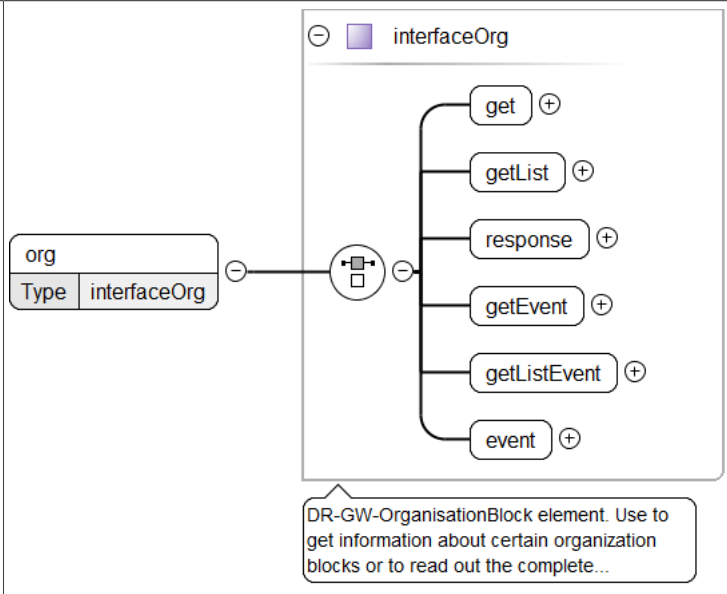
Namespace	DR-GW
Diagram	
Type	typeStatusReceiveEvent
Type hierarchy	• typeEvent • typeStatusReceiveEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , status
Children	requestId, result, status
Instance	<receiveEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <status>{1,1}</status> </receiveEvent>
Source	<xs:element name="receiveEvent" type="typeStatusReceiveEvent"/>

Element typeStatusReceiveEvent / status

Namespace	DR-GW

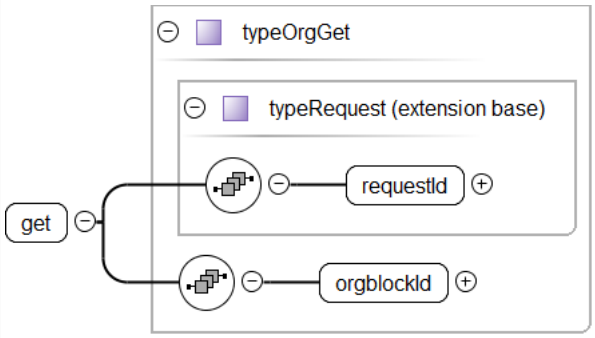
Diagram	
Type	typeStatus
Properties	content: complex
Model	(value | hexValue) , source{0,1} , target , tstamp{0,1}
Children	hexValue, source, target, tstamp, value
Instance	<status xmlns="DR-GW"> <value>{1,1}</value> <hexValue>{1,1}</hexValue> <source>{0,1}</source> <target>{1,1}</target> <tstamp>{0,1}</tstamp> </status>
Source	<xs:element name="status" type="typeStatus"/>

Element drgw / org

Namespace	DR-GW
Diagram	
Type	interfaceOrg
Properties	content: complex
Model	get | getList | response | getEvent | getListEvent | event
Children	event, get, getEvent, getList, getListEvent, response
Instance	<org xmlns="DR-GW"> <get>{1,1}</get> <getList>{1,1}</getList> <response>{1,1}</response> <getEvent>{1,1}</getEvent> <getListEvent>{1,1}</getListEvent> <event>{1,1}</event> </org>

Source	`<xs:element name="org" type="interfaceOrg"/>`

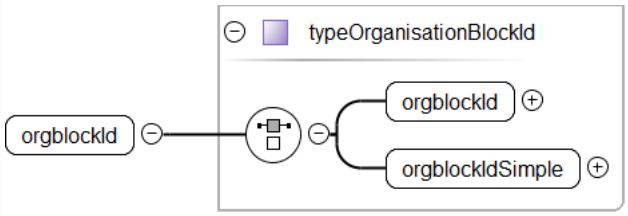
Element `interfaceOrg / get`

Namespace	DR-GW
Diagram	
Type	typeOrgGet
Type hierarchy	• typeRequest • typeOrgGet
Properties	content: complex
Model	requestId , orgblockId
Children	orgblockId, requestId

Instance	`<get xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <orgblockId>{1,1}</orgblockId>`
	`</get>`

Source	`<xs:element name="get" type="typeOrgGet"/>`

Element `typeOrgGet / orgblockId`

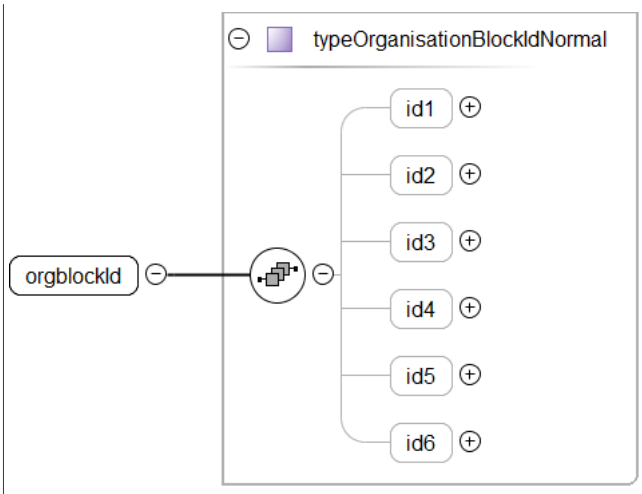
Namespace	DR-GW
Diagram	
Type	typeOrganisationBlockId
Properties	content: complex
Model	orgblockId | orgblockIdSimple
Children	orgblockId, orgblockIdSimple

Instance	`<orgblockId xmlns="DR-GW">`
	`  <orgblockId>{1,1}</orgblockId>`
	`  <orgblockIdSimple>{1,1}</orgblockIdSimple>`
	`</orgblockId>`

Source	`<xs:element name="orgblockId" type="typeOrganisationBlockId"/>`

Element `typeOrganisationBlockId / orgblockId`

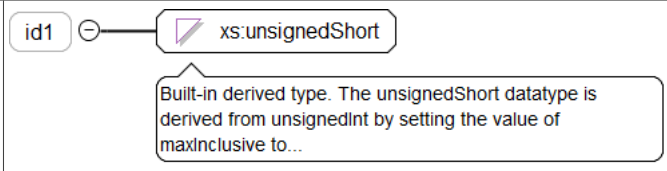
Namespace	DR-GW

Diagram	
Type	typeOrganisationBlockIdNormal
Properties	content: complex
Model	id1{0,1} , id2{0,1} , id3{0,1} , id4{0,1} , id5{0,1} , id6{0,1}
Children	id1, id2, id3, id4, id5, id6

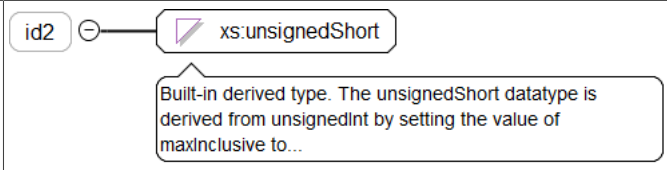
Instance	<pre><orgblockId xmlns="DR-GW">
	  <id1>{0,1}</id1>
	  <id2>{0,1}</id2>
	  <id3>{0,1}</id3>
	  <id4>{0,1}</id4>
	  <id5>{0,1}</id5>
	  <id6>{0,1}</id6>
	</orgblockId></pre>

Source	<pre><xs:element name="orgblockId" type="typeOrganisationBlockIdNormal"/></pre>

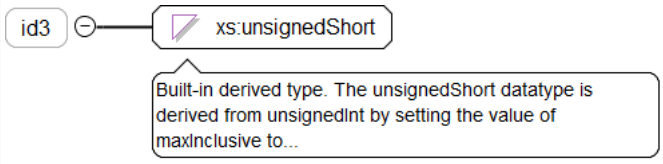
Element typeOrganisationBlockIdNormal / id1

Namespace	DR-GW
Diagram	
Type	xs:unsignedShort
Properties	content: simple minOccurs: 0
Source	<pre><xs:element name="id1" type="xs:unsignedShort" minOccurs="0"/></pre>

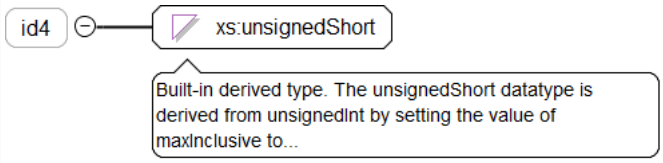
Element typeOrganisationBlockIdNormal / id2

Namespace	DR-GW
Diagram	
Type	xs:unsignedShort
Properties	content: simple minOccurs: 0
Source	<pre><xs:element name="id2" type="xs:unsignedShort" minOccurs="0"/></pre>

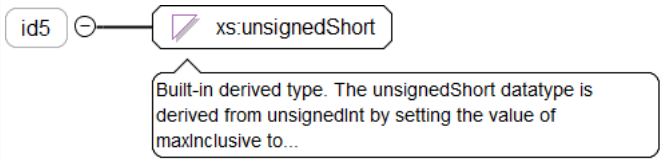
Element typeOrganisationBlockIdNormal / id3

Namespace	DR-GW
Diagram	
Type	xs:unsignedShort
Properties	content: simple minOccurs: 0
Source	`<xs:element name="id3" type="xs:unsignedShort" minOccurs="0"/>`

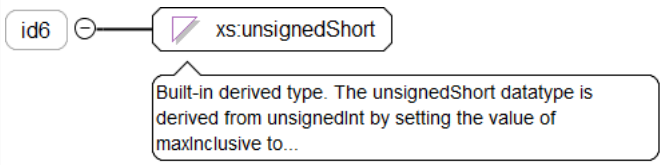
Element typeOrganisationBlockIdNormal / id4

Namespace	DR-GW
Diagram	
Type	xs:unsignedShort
Properties	content: simple minOccurs: 0
Source	`<xs:element name="id4" type="xs:unsignedShort" minOccurs="0"/>`

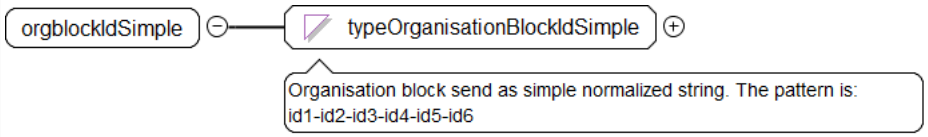
Element typeOrganisationBlockIdNormal / id5

Namespace	DR-GW
Diagram	
Type	xs:unsignedShort
Properties	content: simple minOccurs: 0
Source	`<xs:element name="id5" type="xs:unsignedShort" minOccurs="0"/>`

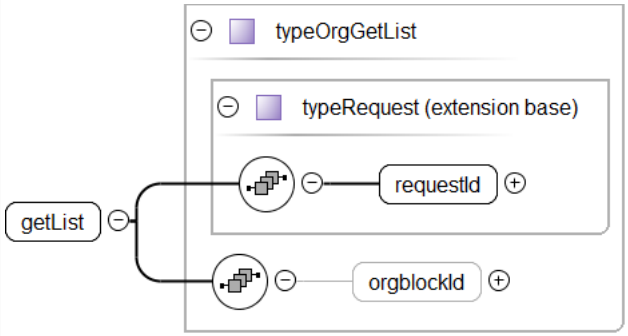
Element typeOrganisationBlockIdNormal / id6

Namespace	DR-GW
Diagram	
Type	xs:unsignedShort
Properties	content: simple minOccurs: 0
Source	`<xs:element name="id6" type="xs:unsignedShort" minOccurs="0"/>`

Element `typeOrganisationBlockId / orgblockIdSimple`

Namespace	DR-GW
Diagram	
Type	typeOrganisationBlockIdSimple
Properties	content: `simple`
Facets	pattern `(([0-9]|[1-9]\d{0,3}|[1-5]\d{4}|6[0-4]\d{3}|65[0-4]\d{2}|655[0-2]\d|6553[0-5])-){0,5}([0-9]|[1-9]\d{0,3}|[1-5]\d{4}|6[0-4]\d{3}|65[0-4]\d{2}|655[0-2]\d|6553[0-5])`
Source	`<xs:element name="orgblockIdSimple" type="typeOrganisationBlockIdSimple"/>`

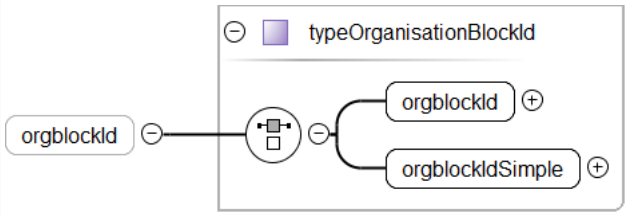
Element `interfaceOrg / getList`

Namespace	DR-GW
Diagram	
Type	typeOrgGetList
Type hierarchy	• typeRequest • typeOrgGetList
Properties	content: `complex`
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId

Instance	`<getList xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <orgblockId>{0,1}</orgblockId>`
	`</getList>`

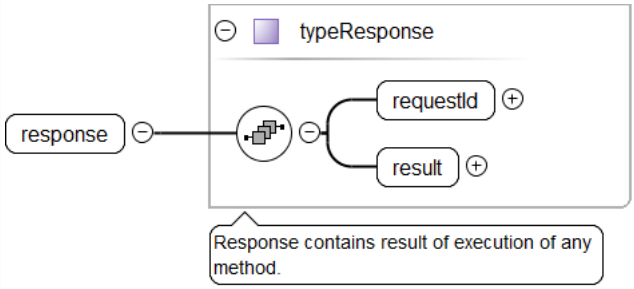
Source	`<xs:element name="getList" type="typeOrgGetList"/>`

Element `typeOrgGetList / orgblockId`

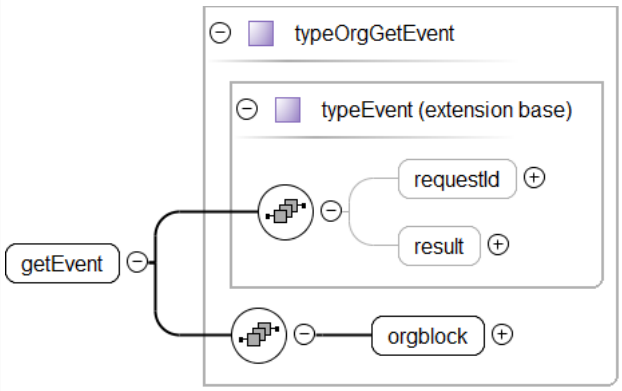
Namespace	DR-GW
Diagram	
Type	typeOrganisationBlockId

Properties	content: complex minOccurs: 0
Model	orgblockId | orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId>
Source	<xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/>

Element interfaceOrg / response

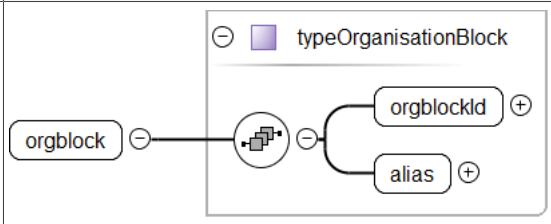
Namespace	DR-GW
Diagram	
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response>
Source	<xs:element name="response" type="typeResponse"/>

Element interfaceOrg / getEvent

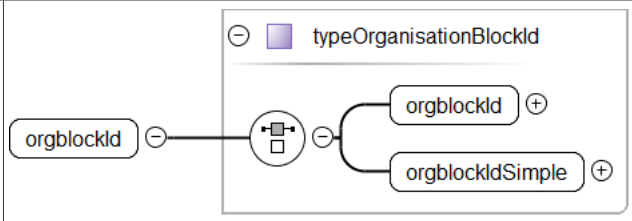
Namespace	DR-GW
Diagram	
Type	typeOrgGetEvent
Type hierarchy	• typeEvent • typeOrgGetEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , orgblock
Children	orgblock, requestId, result
Instance	<getEvent xmlns="DR-GW"> <requestId>{0,1}</requestId>

	<result>{0,1}</result> <orgblock>{1,1}</orgblock> </getEvent>
Source	<xs:element name="getEvent" type="typeOrgGetEvent"/>

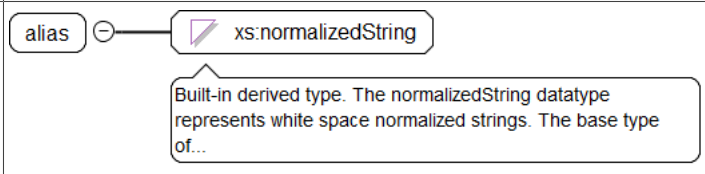
Element typeOrgGetEvent / orgblock

Namespace	DR-GW
Diagram	
Type	typeOrganisationBlock
Properties	content: complex
Model	orgblockId , alias
Children	alias, orgblockId
Instance	<orgblock xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <alias>{1,1}</alias> </orgblock>
Source	<xs:element name="orgblock" type="typeOrganisationBlock"/>

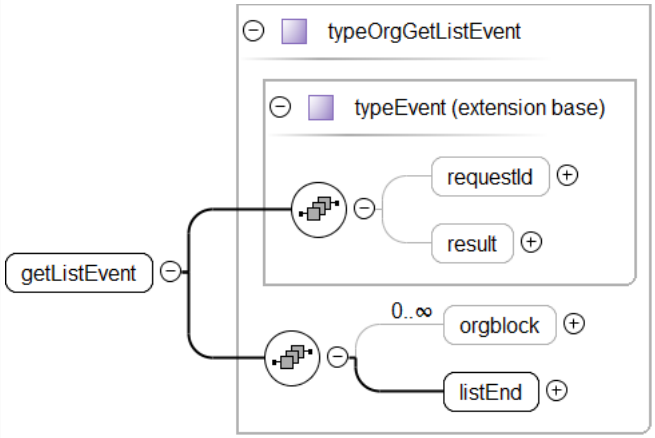
Element typeOrganisationBlock / orgblockId

Namespace	DR-GW
Diagram	
Type	typeOrganisationBlockId
Properties	content: complex
Model	orgblockId | orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId>
Source	<xs:element name="orgblockId" type="typeOrganisationBlockId"/>

Element typeOrganisationBlock / alias

Namespace	DR-GW
Diagram	
Type	xs:normalizedString
Properties	content: simple
Source	<xs:element name="alias" type="xs:normalizedString"/>

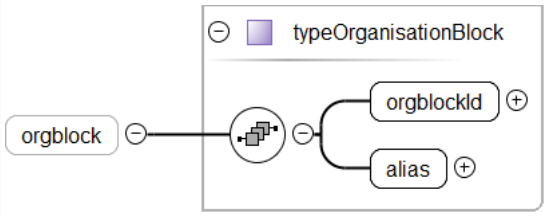
Element `interfaceOrg / getListEvent`

Namespace	DR-GW
Diagram	
Type	typeOrgGetListEvent
Type hierarchy	• typeEvent • typeOrgGetListEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , orgblock* , listEnd
Children	listEnd, orgblock, requestId, result

Instance	`<getListEvent xmlns="DR-GW">`
	`  <requestId>{0,1}</requestId>`
	`  <result>{0,1}</result>`
	`  <orgblock>{0,unbounded}</orgblock>`
	`  <listEnd>{1,1}</listEnd>`
	`</getListEvent>`

Source	`<xs:element name="getListEvent" type="typeOrgGetListEvent"/>`

Element `typeOrgGetListEvent / orgblock`

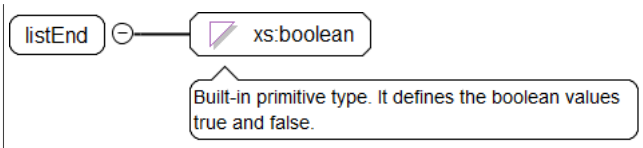
Namespace	DR-GW
Diagram	
Type	typeOrganisationBlock
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	orgblockId , alias
Children	alias, orgblockId

Instance	`<orgblock xmlns="DR-GW">`
	`  <orgblockId>{1,1}</orgblockId>`
	`  <alias>{1,1}</alias>`
	`</orgblock>`

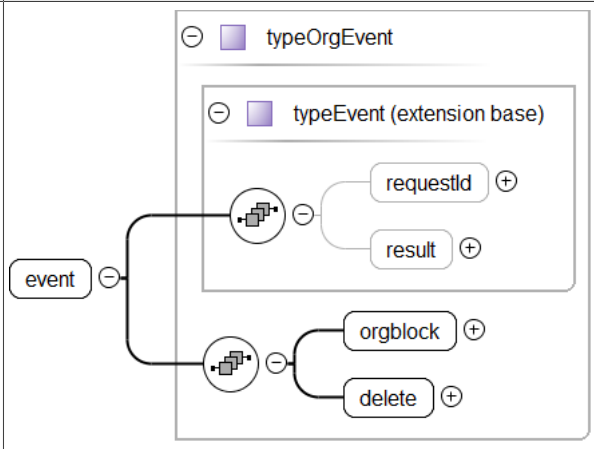
Source	`<xs:element name="orgblock" type="typeOrganisationBlock" minOccurs="0" maxOccurs="unbounded"/>`

Element `typeOrgGetListEvent / listEnd`

Namespace	DR-GW

Diagram	
Type	xs:boolean
Properties	content: simple
Source	`<xs:element name="listEnd" type="xs:boolean"/>`

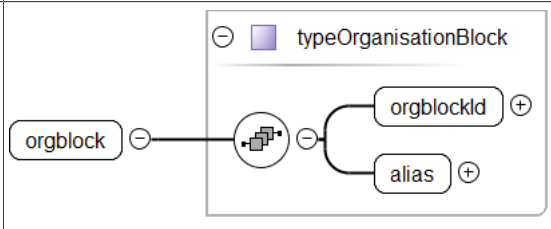
Element `interfaceOrg / event`

Namespace	DR-GW
Diagram	
Type	typeOrgEvent
Type hierarchy	• typeEvent • typeOrgEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , orgblock , delete
Children	delete, orgblock, requestId, result

Instance	`<event xmlns="DR-GW">`
	`  <requestId>{0,1}</requestId>`
	`  <result>{0,1}</result>`
	`  <orgblock>{1,1}</orgblock>`
	`  <delete>{1,1}</delete>`
	`</event>`

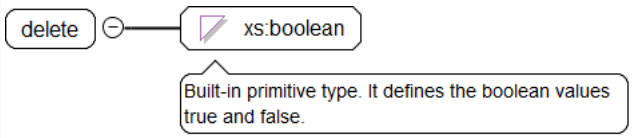
Source	`<xs:element name="event" type="typeOrgEvent"/>`

Element `typeOrgEvent / orgblock`

Namespace	DR-GW
Diagram	
Type	typeOrganisationBlock
Properties	content: complex
Model	orgblockId , alias
Children	alias, orgblockId
Instance	`<orgblock xmlns="DR-GW">` `  <orgblockId>{1,1}</orgblockId>`

	<alias>{1,1}</alias> </orgblock>
Source	<xs:element name="orgblock" type="typeOrganisationBlock"/>

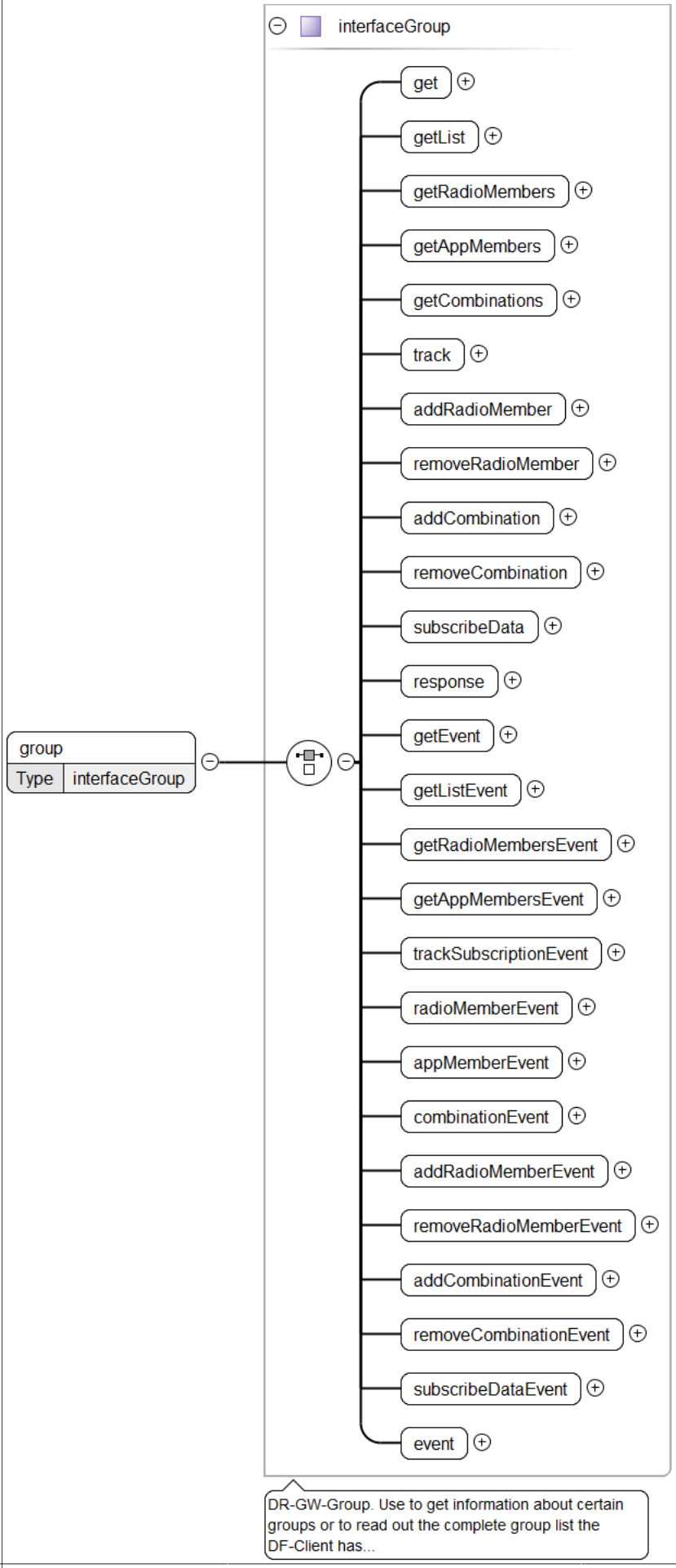
Element typeOrgEvent / delete

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="delete" type="xs:boolean"/>

Element drgw / group

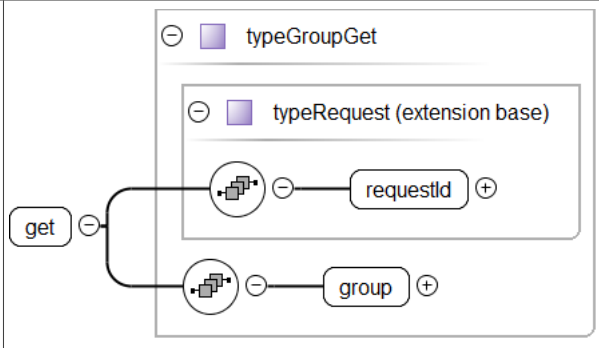
Namespace	DR-GW

Diagram

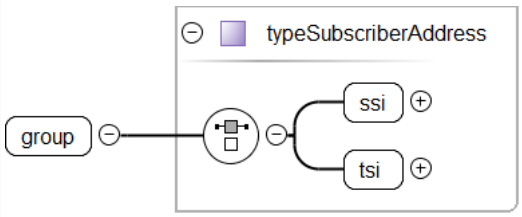

interfaceGroup
get
getList
getRadioMembers
getAppMembers
getCombinations
track
addRadioMember
removeRadioMember
addCombination
removeCombination
subscribeData
response
getEvent
getListEvent
getRadioMembersEvent
getAppMembersEvent
trackSubscriptionEvent
radioMemberEvent
appMemberEvent
combinationEvent
addRadioMemberEvent
removeRadioMemberEvent
addCombinationEvent
removeCombinationEvent
subscribeDataEvent
event
group
Type
interfaceGroup
DR-GW-Group. Use to get information about certain groups or to read out the complete group list the DF-Client has...

Type	interfaceGroup
Properties	content: complex
Model	get | getList | getRadioMembers | getAppMembers | getCombinations | track | addRadioMember | removeRadioMember | addCombination | removeCombination | subscribeData | response | getEvent | getListEvent | getRadioMembersEvent | getAppMembersEvent | trackSubscriptionEvent | radioMemberEvent | appMemberEvent | combinationEvent | addRadioMemberEvent | removeRadioMemberEvent | addCombinationEvent | removeCombinationEvent | subscribeDataEvent | event
Children	addCombination, addCombinationEvent, addRadioMember, addRadioMemberEvent, appMemberEvent, combinationEvent, event, get, getAppMembers, getAppMembersEvent, getCombinations, getEvent, getList, getListEvent, getRadioMembers, getRadioMembersEvent, radioMemberEvent, removeCombination, removeCombinationEvent, removeRadioMember, removeRadioMemberEvent, response, subscribeData, subscribeDataEvent, track, trackSubscriptionEvent
Instance	<group xmlns="DR-GW"> <get>{1,1}</get> <getList>{1,1}</getList> <getRadioMembers>{1,1}</getRadioMembers> <getAppMembers>{1,1}</getAppMembers> <getCombinations>{1,1}</getCombinations> <track>{1,1}</track> <addRadioMember>{1,1}</addRadioMember> <removeRadioMember>{1,1}</removeRadioMember> <addCombination>{1,1}</addCombination> <removeCombination>{1,1}</removeCombination> <subscribeData>{1,1}</subscribeData> <response>{1,1}</response> <getEvent>{1,1}</getEvent> <getListEvent>{1,1}</getListEvent> <getRadioMembersEvent>{1,1}</getRadioMembersEvent> <getAppMembersEvent>{1,1}</getAppMembersEvent> <trackSubscriptionEvent>{1,1}</trackSubscriptionEvent> <radioMemberEvent>{1,1}</radioMemberEvent> <appMemberEvent>{1,1}</appMemberEvent> <combinationEvent>{1,1}</combinationEvent> <addRadioMemberEvent>{1,1}</addRadioMemberEvent> <removeRadioMemberEvent>{1,1}</removeRadioMemberEvent> <addCombinationEvent>{1,1}</addCombinationEvent> <removeCombinationEvent>{1,1}</removeCombinationEvent> <subscribeDataEvent>{1,1}</subscribeDataEvent> <event>{1,1}</event> </group>
Source	<xs:element name="group" type="interfaceGroup"/>

Element interfaceGroup / get

Namespace	DR-GW
Diagram	
Type	typeGroupGet
Type hierarchy	• typeRequest • typeGroupGet
Properties	content: complex
Model	requestId , group
Children	group, requestId
Instance	<get xmlns="DR-GW"> <requestId>{1,1}</requestId> <group>{1,1}</group> </get>
Source	<xs:element name="get" type="typeGroupGet"/>

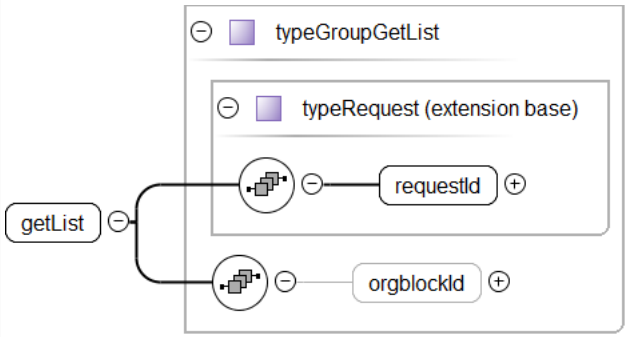
Element `typeGroupGet / group`

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<group xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</group>`

Source	`<xs:element name="group" type="typeSubscriberAddress"/>`

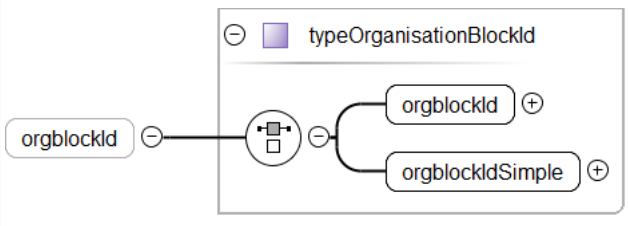
Element `interfaceGroup / getList`

Namespace	DR-GW
Diagram	
Type	typeGroupGetList
Type hierarchy	• typeRequest • typeGroupGetList
Properties	content: complex
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId

Instance	`<getList xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <orgblockId>{0,1}</orgblockId>`
	`</getList>`

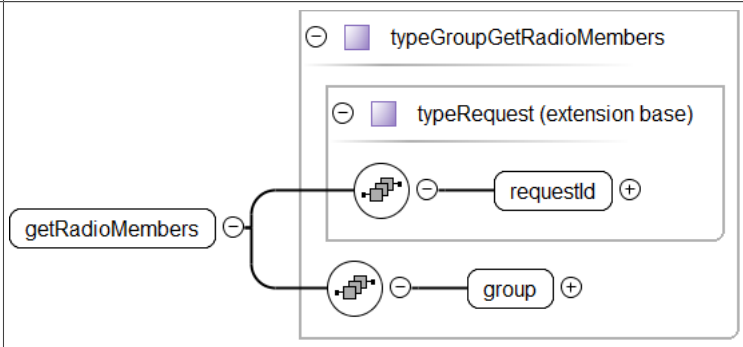
Source	`<xs:element name="getList" type="typeGroupGetList"/>`

Element `typeGroupGetList / orgblockId`

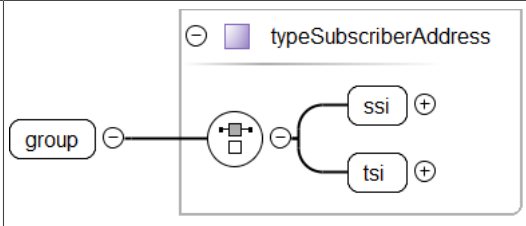
Namespace	DR-GW
Diagram	

Type	typeOrganisationBlockId
Properties	content: complex minOccurs: 0
Model	orgblockId | orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId>
Source	<xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/>

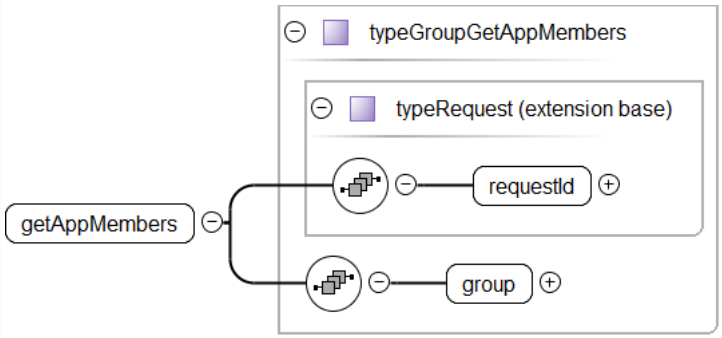
Element interfaceGroup / getRadioMembers

Namespace	DR-GW
Diagram	
Type	typeGroupGetRadioMembers
Type hierarchy	• typeRequest • typeGroupGetRadioMembers
Properties	content: complex
Model	requestId , group
Children	group, requestId
Instance	<getRadioMembers xmlns="DR-GW"> <requestId>{1,1}</requestId> <group>{1,1}</group> </getRadioMembers>
Source	<xs:element name="getRadioMembers" type="typeGroupGetRadioMembers"/>

Element typeGroupGetRadioMembers / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

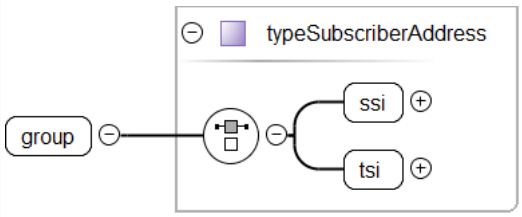
Element `interfaceGroup / getAppMembers`

Namespace	DR-GW
Diagram	
Type	typeGroupGetAppMembers
Type hierar-chy	• typeRequest • typeGroupGetAppMembers
Properties	content: complex
Model	requestId , group
Children	group, requestId

Instance	`<getAppMembers xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <group>{1,1}</group>`
	`</getAppMembers>`

Source	`<xs:element name="getAppMembers" type="typeGroupGetAppMembers"/>`

Element `typeGroupGetAppMembers / group`

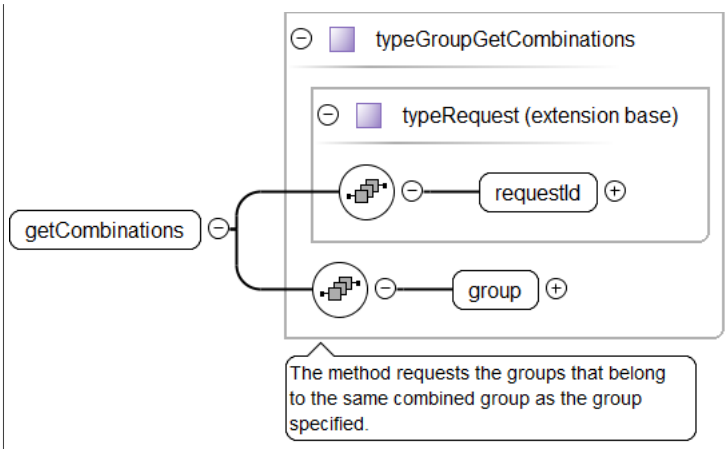
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<group xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</group>`

Source	`<xs:element name="group" type="typeSubscriberAddress"/>`

Element `interfaceGroup / getCombinations`

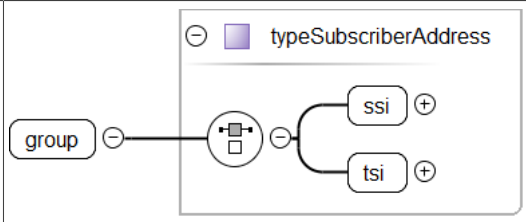
Namespace	DR-GW

Diagram	
Type	typeGroupGetCombinations
Type hierarchy	• typeRequest • typeGroupGetCombinations
Properties	content: complex
Model	requestId , group
Children	group, requestId

Instance	`<getCombinations xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <group>{1,1}</group>`
	`</getCombinations>`

Source	`<xs:element name="getCombinations" type="typeGroupGetCombinations"/>`

Element `typeGroupGetCombinations / group`

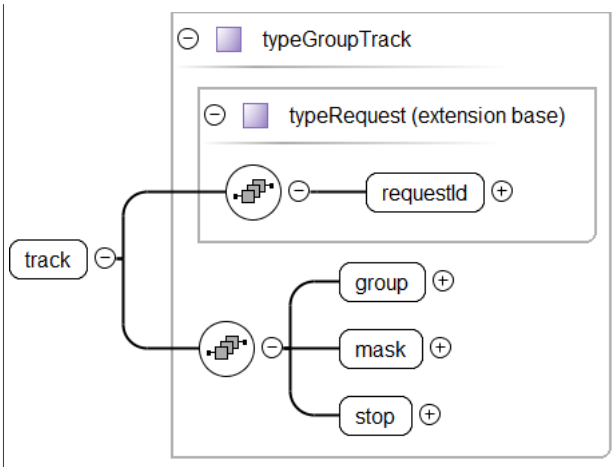
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<group xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</group>`

Source	`<xs:element name="group" type="typeSubscriberAddress"/>`

Element `interfaceGroup / track`

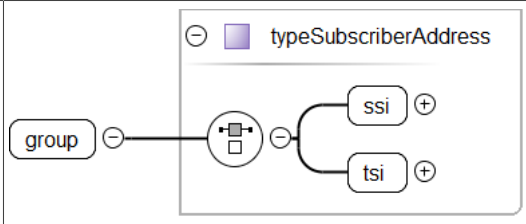
Namespace	DR-GW

Diagram	
Type	typeGroupTrack
Type hierarchy	• typeRequest • typeGroupTrack
Properties	content: complex
Model	requestId , group , mask , stop
Children	group, mask, requestId, stop

Instance	`<track xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <group>{1,1}</group>`
	`  <mask>{1,1}</mask>`
	`  <stop>{1,1}</stop>`
	`</track>`

Source	`<xs:element name="track" type="typeGroupTrack"/>`

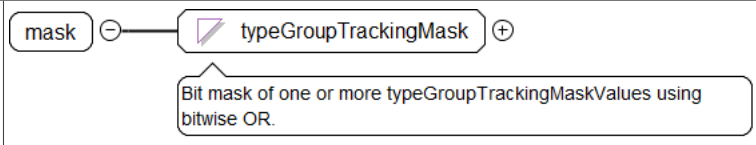
Element typeGroupTrack / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<group xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</group>`

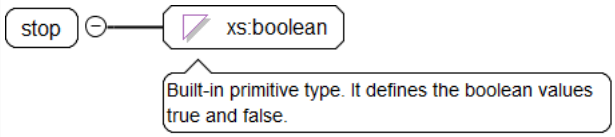
Source	`<xs:element name="group" type="typeSubscriberAddress"/>`

Element typeGroupTrack / mask

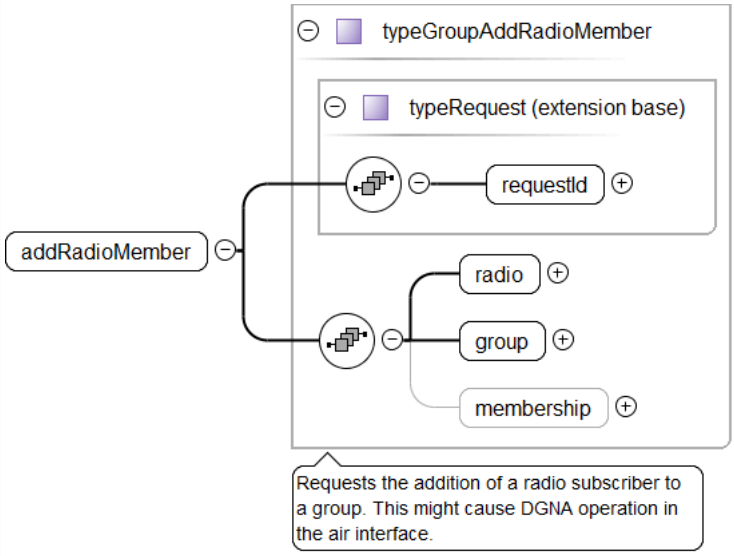
Namespace	DR-GW
Diagram	
Type	typeGroupTrackingMask

Properties	content: simple
Source	<xs:element name="mask" type="typeGroupTrackingMask"/>

Element typeGroupTrack / stop

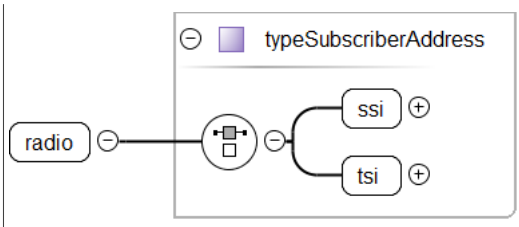
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="stop" type="xs:boolean"/>

Element interfaceGroup / addRadioMember

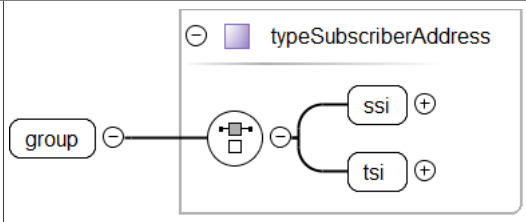
Namespace	DR-GW
Diagram	
Type	typeGroupAddRadioMember
Type hierarchy	• typeRequest • typeGroupAddRadioMember
Properties	content: complex
Model	requestId , radio , group , membership{0,1}
Children	group, membership, radio, requestId
Instance	<addRadioMember xmlns="DR-GW"> <requestId>{1,1}</requestId> <radio>{1,1}</radio> <group>{1,1}</group> <membership>{0,1}</membership> </addRadioMember>
Source	<xs:element name="addRadioMember" type="typeGroupAddRadioMember"/>

Element typeGroupAddRadioMember / radio

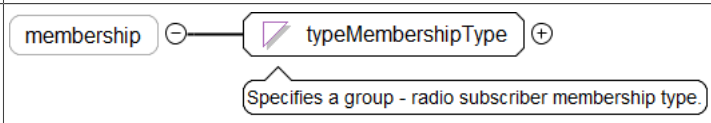
Namespace	DR-GW

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

Element typeGroupAddRadioMember / group

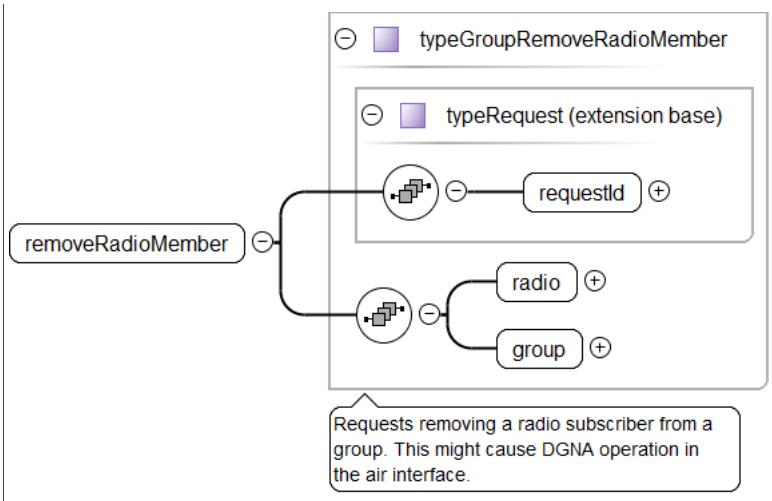
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

Element typeGroupAddRadioMember / membership

Namespace	DR-GW
Diagram	
Type	typeMembershipType
Properties	content: simple minOccurs: 0
Facets	enumeration unknown enumeration permanent enumeration visiting
Source	<xs:element name="membership" type="typeMembershipType" minOccurs="0"/>

Element interfaceGroup / removeRadioMember

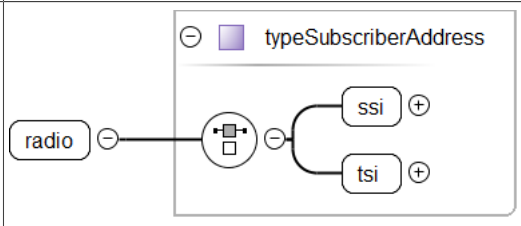
Namespace	DR-GW

Diagram	
Type	typeGroupRemoveRadioMember
Type hierarchy	• typeRequest • typeGroupRemoveRadioMember
Properties	content: complex
Model	requestId , radio , group
Children	group, radio, requestId

Instance	`<removeRadioMember xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <radio>{1,1}</radio>`
	`  <group>{1,1}</group>`
	`</removeRadioMember>`

Source	`<xs:element name="removeRadioMember" type="typeGroupRemoveRadioMember"/>`

Element typeGroupRemoveRadioMember / radio

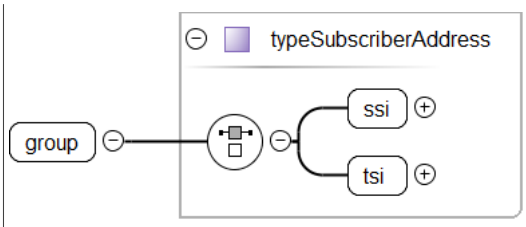
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<radio xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</radio>`

Source	`<xs:element name="radio" type="typeSubscriberAddress"/>`

Element typeGroupRemoveRadioMember / group

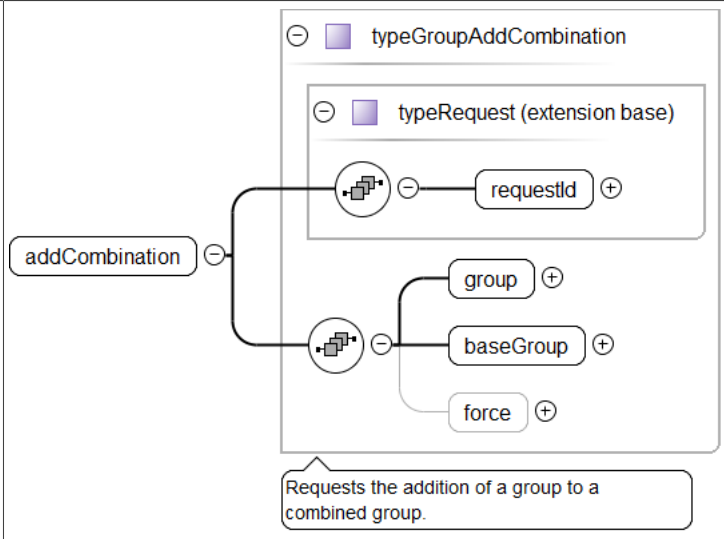
Namespace	DR-GW

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<group xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</group>`

Source	`<xs:element name="group" type="typeSubscriberAddress"/>`

Element `interfaceGroup / addCombination`

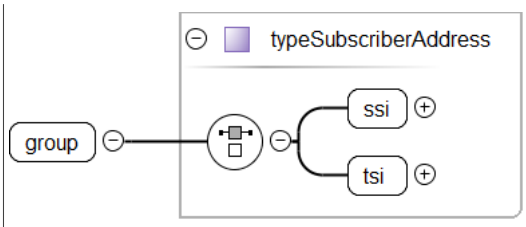
Namespace	DR-GW
Diagram	
Type	typeGroupAddCombination
Type hierarchy	• typeRequest • typeGroupAddCombination
Properties	content: complex
Model	requestId , group , baseGroup , force{0,1}
Children	baseGroup, force, group, requestId

Instance	`<addCombination xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <group>{1,1}</group>`
	`  <baseGroup>{1,1}</baseGroup>`
	`  <force>{0,1}</force>`
	`</addCombination>`

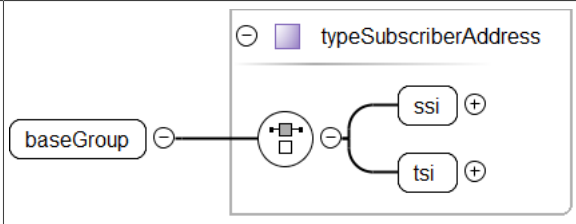
Source	`<xs:element name="addCombination" type="typeGroupAddCombination"/>`

Element `typeGroupAddCombination / group`

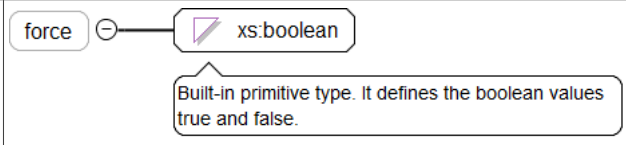
Namespace	DR-GW

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

Element typeGroupAddCombination / baseGroup

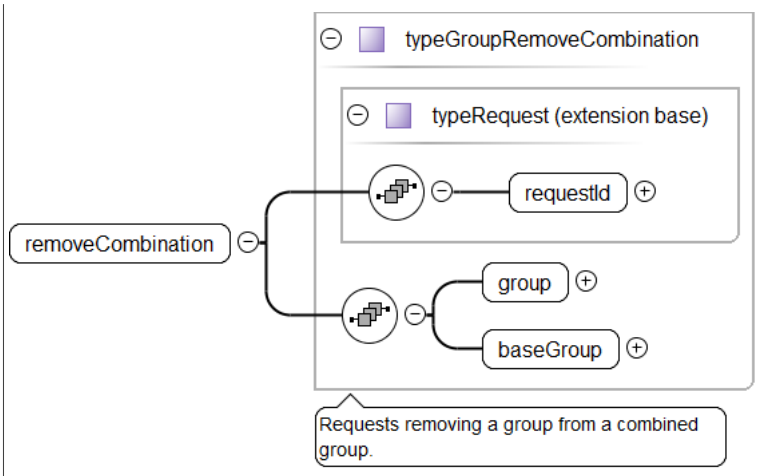
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup>
Source	<xs:element name="baseGroup" type="typeSubscriberAddress"/>

Element typeGroupAddCombination / force

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0 default: true
Source	<xs:element name="force" type="xs:boolean" minOccurs="0" default="true"/>

Element interfaceGroup / removeCombination

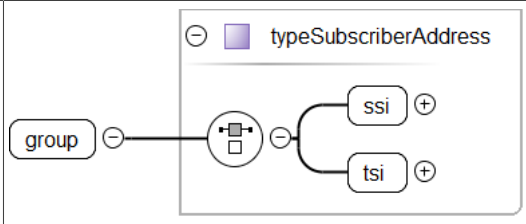
Namespace	DR-GW

Diagram	
Type	typeGroupRemoveCombination
Type hierarchy	• typeRequest • typeGroupRemoveCombination
Properties	content: complex
Model	requestId , group , baseGroup
Children	baseGroup, group, requestId

Instance	`<removeCombination xmlns="DR-GW">`
	`<requestId>{1,1}</requestId>`
	`<group>{1,1}</group>`
	`<baseGroup>{1,1}</baseGroup>`
	`</removeCombination>`

Source	`<xs:element name="removeCombination" type="typeGroupRemoveCombination"/>`

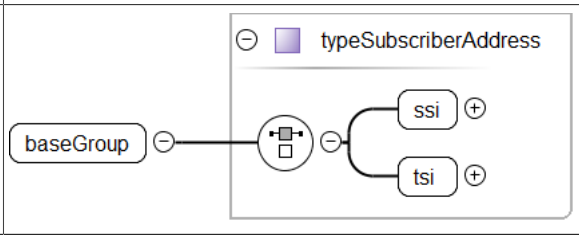
Element typeGroupRemoveCombination / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<group xmlns="DR-GW">`
	`<ssi>{1,1}</ssi>`
	`<tsi>{1,1}</tsi>`
	`</group>`

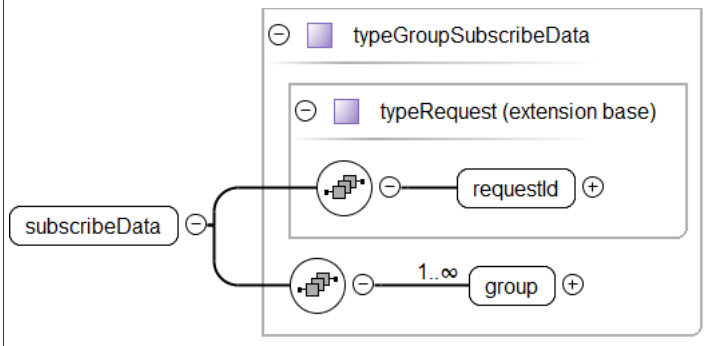
Source	`<xs:element name="group" type="typeSubscriberAddress"/>`

Element typeGroupRemoveCombination / baseGroup

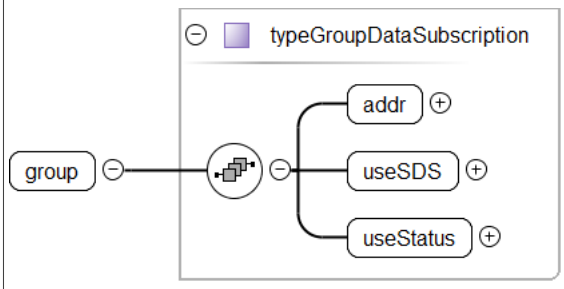
Namespace	DR-GW
Diagram	

Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup>
Source	<xs:element name="baseGroup" type="typeSubscriberAddress"/>

Element interfaceGroup / subscribeData

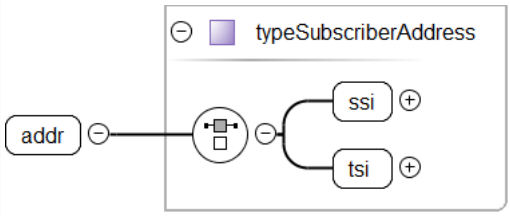
Namespace	DR-GW
Diagram	
Type	typeGroupSubscribeData
Type hierarchy	• typeRequest • typeGroupSubscribeData
Properties	content: complex
Model	requestId , group+
Children	group, requestId
Instance	<subscribeData xmlns="DR-GW"> <requestId>{1,1}</requestId> <group>{1,unbounded}</group> </subscribeData>
Source	<xs:element name="subscribeData" type="typeGroupSubscribeData"/>

Element typeGroupSubscribeData / group

Namespace	DR-GW
Diagram	
Type	typeGroupDataSubscription
Properties	content: complex maxOccurs: unbounded
Model	addr , useSDS , useStatus
Children	addr, useSDS, useStatus
Instance	<group xmlns="DR-GW"> <addr>{1,1}</addr> <useSDS>{1,1}</useSDS> <useStatus>{1,1}</useStatus> </group>

Source	`<xs:element name="group" type="typeGroupDataSubscription" maxOccurs="unbounded"/>`

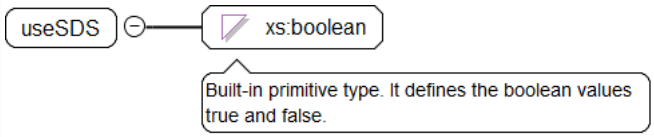
Element typeGroupDataSubscription / addr

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

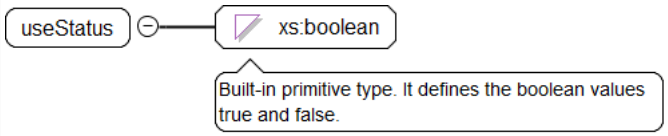
Instance	`<addr xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</addr>`

Source	`<xs:element name="addr" type="typeSubscriberAddress"/>`

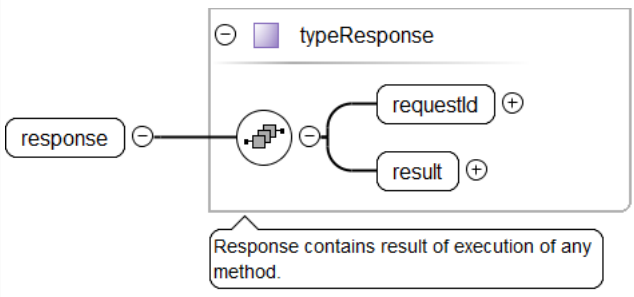
Element typeGroupDataSubscription / useSDS

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	`<xs:element name="useSDS" type="xs:boolean"/>`

Element typeGroupDataSubscription / useStatus

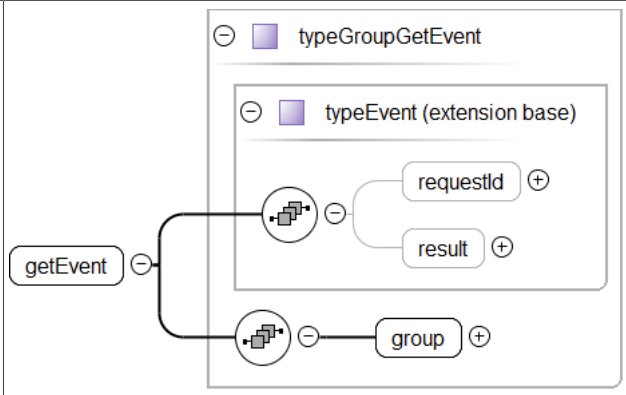
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	`<xs:element name="useStatus" type="xs:boolean"/>`

Element interfaceGroup / response

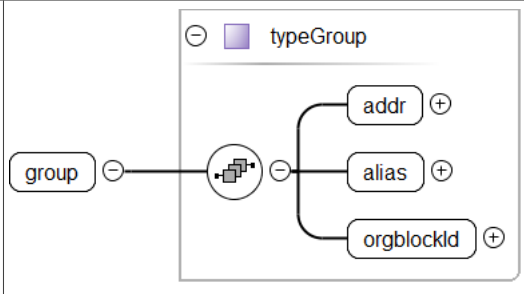
Namespace	DR-GW
Diagram	

Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response>
Source	<xs:element name="response" type="typeResponse"/>

Element interfaceGroup / getEvent

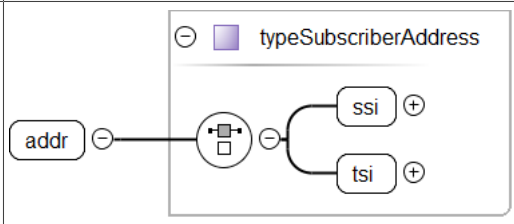
Namespace	DR-GW
Diagram	
Type	typeGroupGetEvent
Type hierarchy	• typeEvent • typeGroupGetEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group
Children	group, requestId, result
Instance	<getEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> </getEvent>
Source	<xs:element name="getEvent" type="typeGroupGetEvent"/>

Element typeGroupGetEvent / group

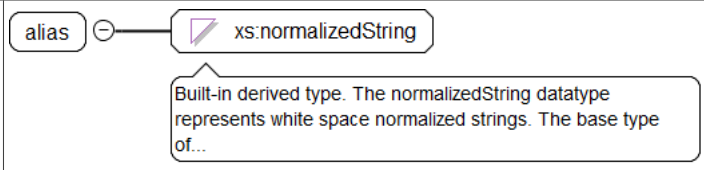
Namespace	DR-GW
Diagram	
Type	typeGroup
Properties	content: complex
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId
Instance	<group xmlns="DR-GW"> <addr>{1,1}</addr>

	<alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> </group>
Source	<xs:element name="group" type="typeGroup"/>

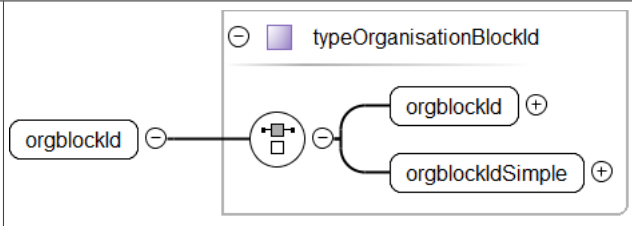
Element typeGroup / addr

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<addr xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </addr>
Source	<xs:element name="addr" type="typeSubscriberAddress"/>

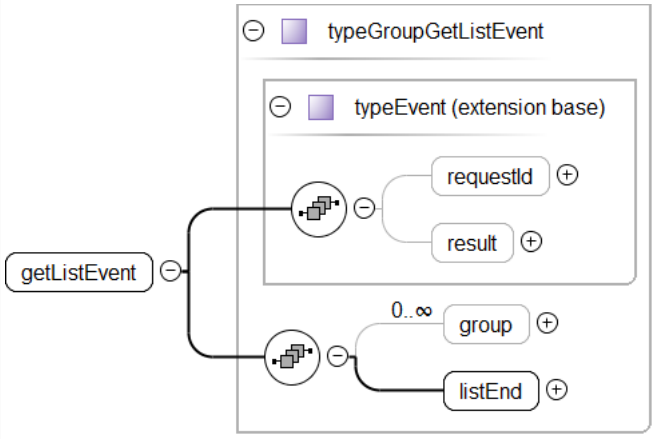
Element typeGroup / alias

Namespace	DR-GW
Diagram	
Type	xs:normalizedString
Properties	content: simple
Source	<xs:element name="alias" type="xs:normalizedString"/>

Element typeGroup / orgblockId

Namespace	DR-GW
Diagram	
Type	typeOrganisationBlockId
Properties	content: complex
Model	orgblockId | orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId>
Source	<xs:element name="orgblockId" type="typeOrganisationBlockId"/>

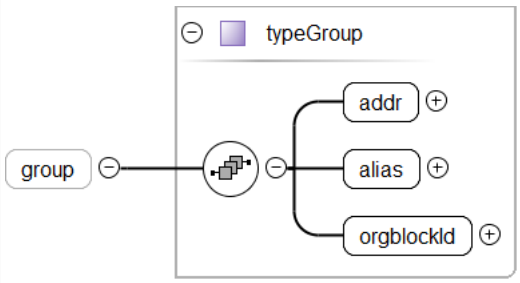
Element `interfaceGroup / getListEvent`

Namespace	DR-GW
Diagram	
Type	typeGroupGetListEvent
Type hierarchy	• typeEvent • typeGroupGetListEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group* , listEnd
Children	group, listEnd, requestId, result

Instance	`<getListEvent xmlns="DR-GW">`
	`  <requestId>{0,1}</requestId>`
	`  <result>{0,1}</result>`
	`  <group>{0,unbounded}</group>`
	`  <listEnd>{1,1}</listEnd>`
	`</getListEvent>`

Source	`<xs:element name="getListEvent" type="typeGroupGetListEvent"/>`

Element `typeGroupGetListEvent / group`

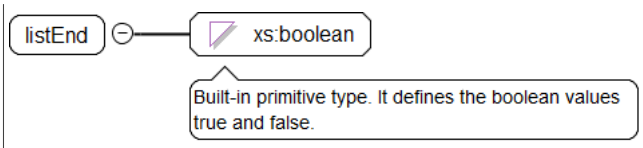
Namespace	DR-GW
Diagram	
Type	typeGroup
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId

Instance	`<group xmlns="DR-GW">`
	`  <addr>{1,1}</addr>`
	`  <alias>{1,1}</alias>`
	`  <orgblockId>{1,1}</orgblockId>`
	`</group>`

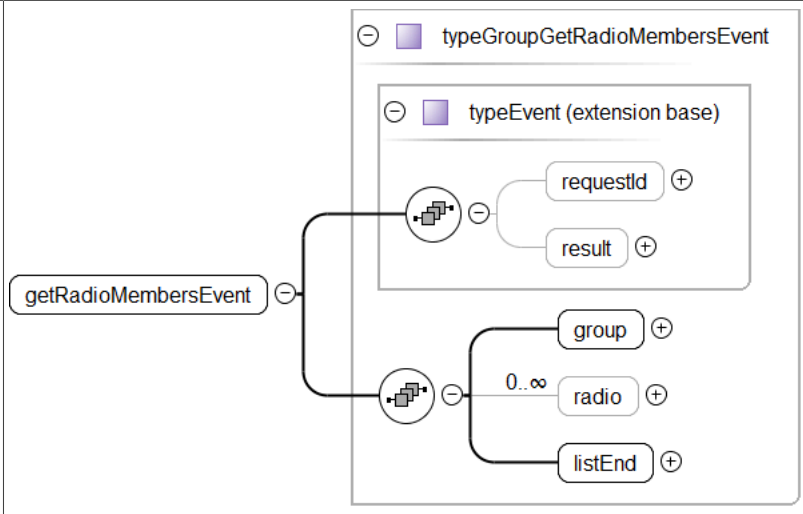
Source	`<xs:element name="group" type="typeGroup" minOccurs="0" maxOccurs="unbounded"/>`

Element `typeGroupGetListEvent / listEnd`

Namespace	DR-GW

Diagram	
Type	xs:boolean
Properties	content: simple
Source	`<xs:element name="listEnd" type="xs:boolean"/>`

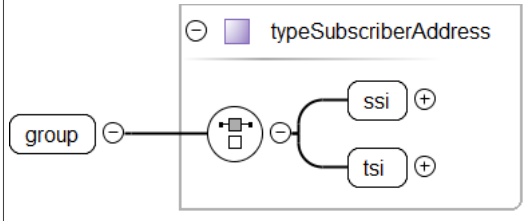
Element `interfaceGroup / getRadioMembersEvent`

Namespace	DR-GW
Diagram	
Type	typeGroupGetRadioMembersEvent
Type hierarchy	• typeEvent • typeGroupGetRadioMembersEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , radio* , listEnd
Children	group, listEnd, radio, requestId, result

Instance	`<getRadioMembersEvent xmlns="DR-GW">`
	`  <requestId>{0,1}</requestId>`
	`  <result>{0,1}</result>`
	`  <group>{1,1}</group>`
	`  <radio>{0,unbounded}</radio>`
	`  <listEnd>{1,1}</listEnd>`
	`</getRadioMembersEvent>`

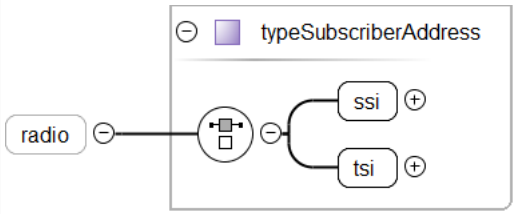
Source	`<xs:element name="getRadioMembersEvent" type="typeGroupGetRadioMembersEvent"/>`

Element `typeGroupGetRadioMembersEvent / group`

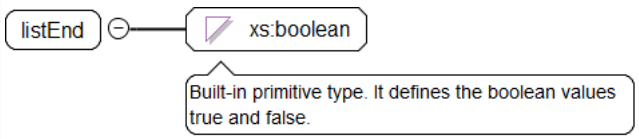
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

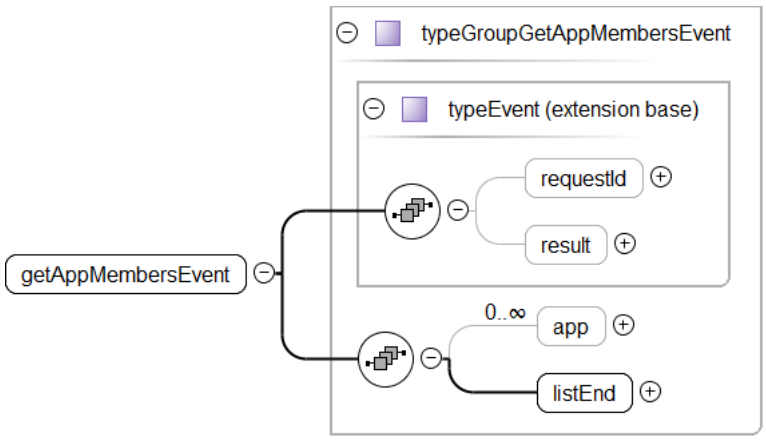
Element typeGroupGetRadioMembersEvent / radio

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ssi | tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress" minOccurs="0" maxOccurs="unbounded"/>

Element typeGroupGetRadioMembersEvent / listEnd

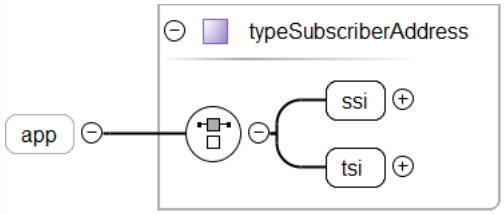
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="listEnd" type="xs:boolean"/>

Element interfaceGroup / getAppMembersEvent

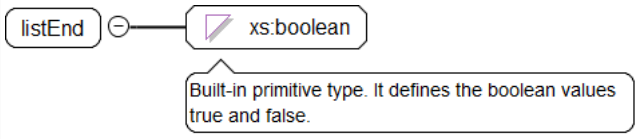
Namespace	DR-GW
Diagram	
Type	typeGroupGetAppMembersEvent

Type hierarchy	• typeEvent • typeGroupGetAppMembersEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , app* , listEnd
Children	app, listEnd, requestId, result
Instance	<getAppMembersEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <app>{0,unbounded}</app> <listEnd>{1,1}</listEnd> </getAppMembersEvent>
Source	<xs:element name="getAppMembersEvent" type="typeGroupGetAppMembersEvent"/>

Element typeGroupGetAppMembersEvent / app

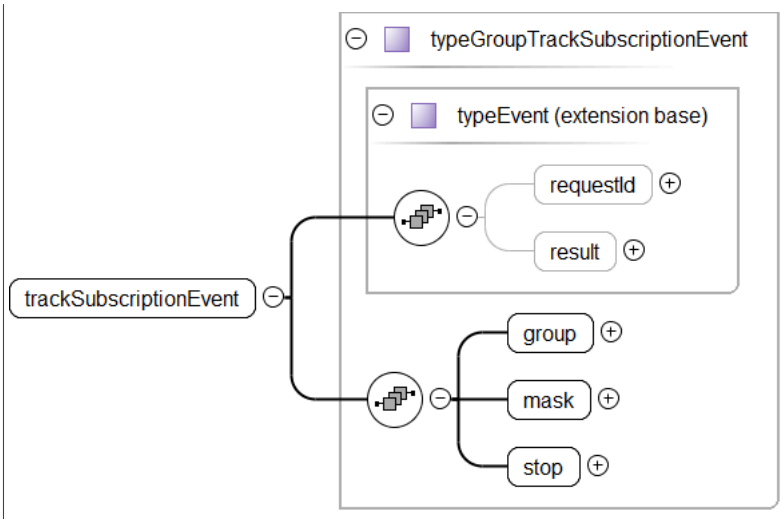
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	ssi | tsi
Children	ssi, tsi
Instance	<app xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </app>
Source	<xs:element name="app" type="typeSubscriberAddress" minOccurs="0" maxOccurs="unbounded"/>

Element typeGroupGetAppMembersEvent / listEnd

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="listEnd" type="xs:boolean"/>

Element interfaceGroup / trackSubscriptionEvent

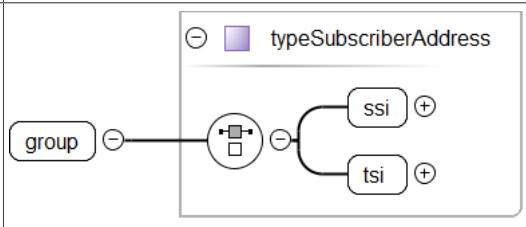
Namespace	DR-GW

Diagram	
Type	typeGroupTrackSubscriptionEvent
Type hierarchy	• typeEvent • typeGroupTrackSubscriptionEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , mask , stop
Children	group, mask, requestId, result, stop

Instance	`<trackSubscriptionEvent xmlns="DR-GW">`
	`<requestId>{0,1}</requestId>`
	`<result>{0,1}</result>`
	`<group>{1,1}</group>`
	`<mask>{1,1}</mask>`
	`<stop>{1,1}</stop>`
	`</trackSubscriptionEvent>`

Source	`<xs:element name="trackSubscriptionEvent" type="typeGroupTrackSubscriptionEvent"/>`

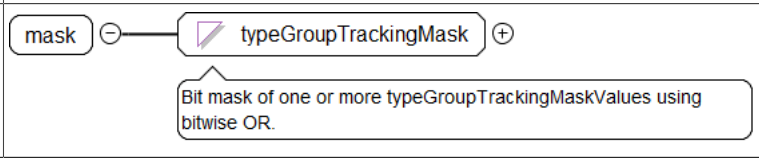
Element typeGroupTrackSubscriptionEvent / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<group xmlns="DR-GW">`
	`<ssi>{1,1}</ssi>`
	`<tsi>{1,1}</tsi>`
	`</group>`

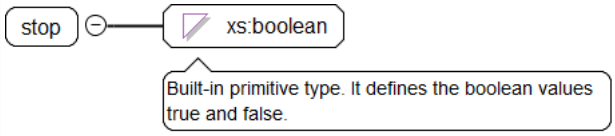
Source	`<xs:element name="group" type="typeSubscriberAddress"/>`

Element typeGroupTrackSubscriptionEvent / mask

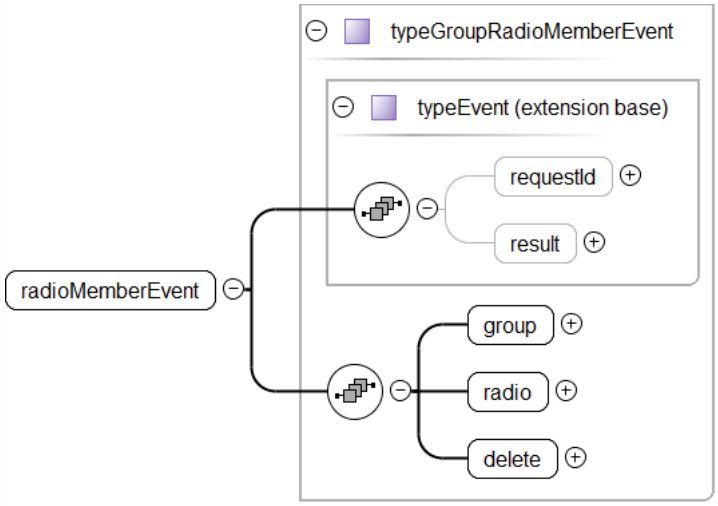
Namespace	DR-GW
Diagram	

Type	typeGroupTrackingMask
Properties	content: simple
Source	`<xs:element name="mask" type="typeGroupTrackingMask"/>`

Element typeGroupTrackSubscriptionEvent / stop

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	`<xs:element name="stop" type="xs:boolean"/>`

Element interfaceGroup / radioMemberEvent

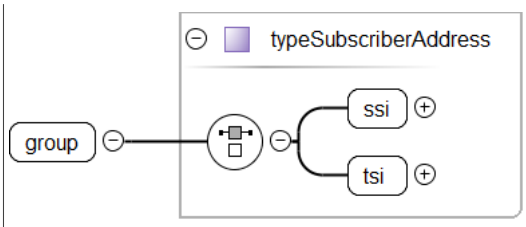
Namespace	DR-GW
Diagram	
Type	typeGroupRadioMemberEvent
Type hierarchy	• typeEvent • typeGroupRadioMemberEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , radio , delete
Children	delete, group, radio, requestId, result

Instance	`<radioMemberEvent xmlns="DR-GW">`
	`  <requestId>{0,1}</requestId>`
	`  <result>{0,1}</result>`
	`  <group>{1,1}</group>`
	`  <radio>{1,1}</radio>`
	`  <delete>{1,1}</delete>`
	`</radioMemberEvent>`

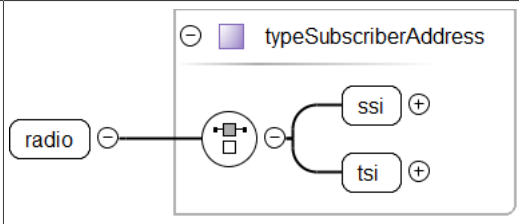
Source	`<xs:element name="radioMemberEvent" type="typeGroupRadioMemberEvent"/>`

Element typeGroupRadioMemberEvent / group

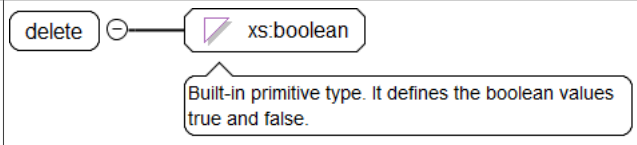
Namespace	DR-GW

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

Element typeGroupRadioMemberEvent / radio

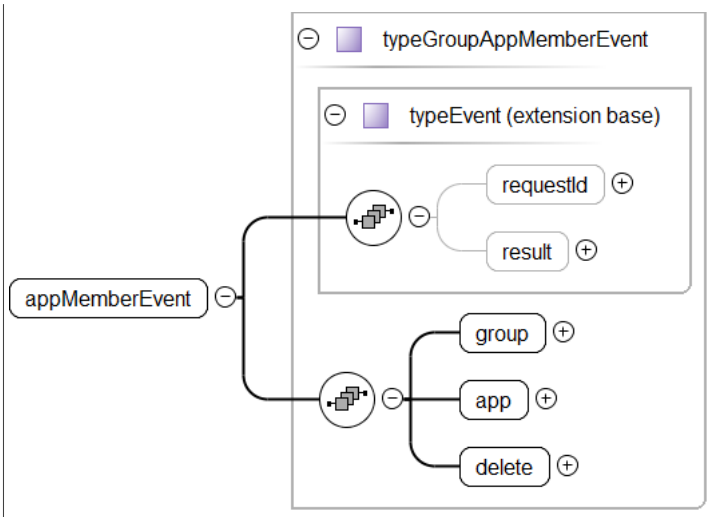
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

Element typeGroupRadioMemberEvent / delete

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="delete" type="xs:boolean"/>

Element interfaceGroup / appMemberEvent

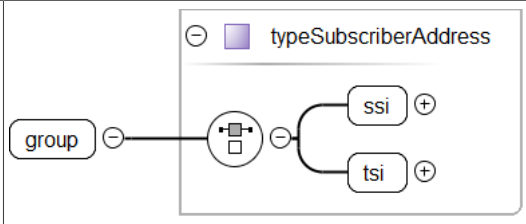
Namespace	DR-GW

Diagram	
Type	typeGroupAppMemberEvent
Type hierarchy	• typeEvent • typeGroupAppMemberEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , app , delete
Children	app, delete, group, requestId, result

Instance	`<appMemberEvent xmlns="DR-GW">`
	`  <requestId>{0,1}</requestId>`
	`  <result>{0,1}</result>`
	`  <group>{1,1}</group>`
	`  <app>{1,1}</app>`
	`  <delete>{1,1}</delete>`
	`</appMemberEvent>`

Source	`<xs:element name="appMemberEvent" type="typeGroupAppMemberEvent"/>`

Element typeGroupAppMemberEvent / group

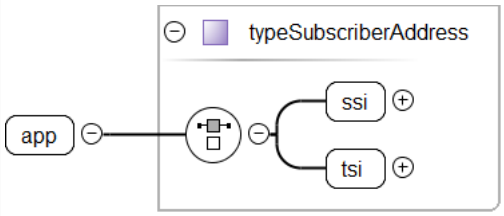
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<group xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</group>`

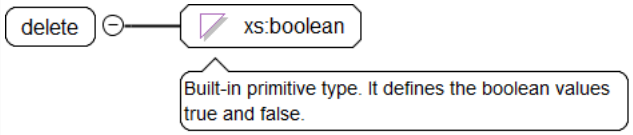
Source	`<xs:element name="group" type="typeSubscriberAddress"/>`

Element typeGroupAppMemberEvent / app

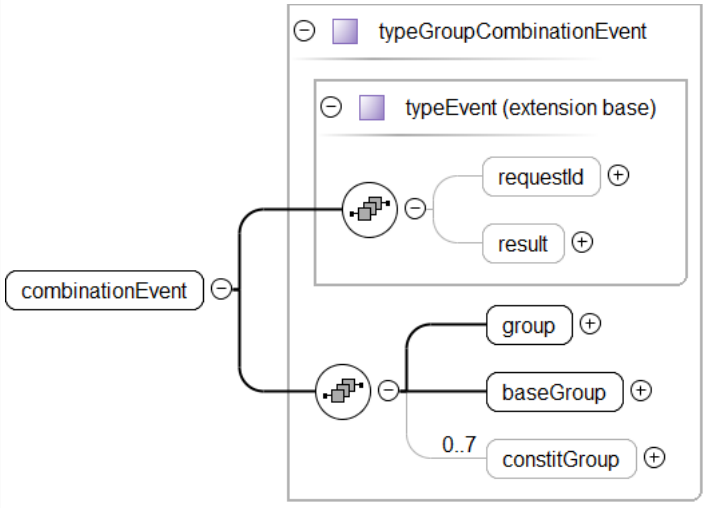
Namespace	DR-GW

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<app xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </app>
Source	<xs:element name="app" type="typeSubscriberAddress"/>

Element typeGroupAppMemberEvent / delete

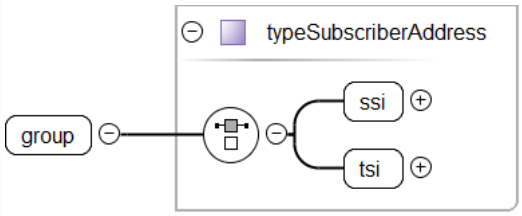
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="delete" type="xs:boolean"/>

Element interfaceGroup / combinationEvent

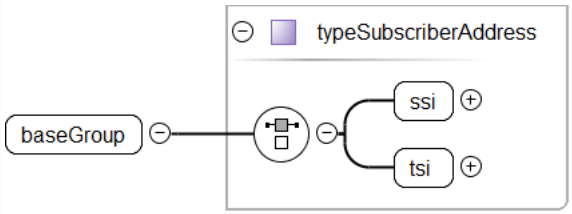
Namespace	DR-GW
Diagram	
Type	typeGroupCombinationEvent
Type hierarchy	• typeEvent • typeGroupCombinationEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , baseGroup , constitGroup{0,7}
Children	baseGroup, constitGroup, group, requestId, result
Instance	<combinationEvent xmlns="DR-GW">

	<requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <baseGroup>{1,1}</baseGroup> <constitGroup>{0,7}</constitGroup> </combinationEvent>
Source	<xs:element name="combinationEvent" type="typeGroupCombinationEvent"/>

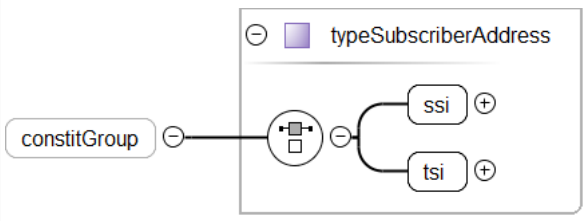
Element typeGroupCombinationEvent / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

Element typeGroupCombinationEvent / baseGroup

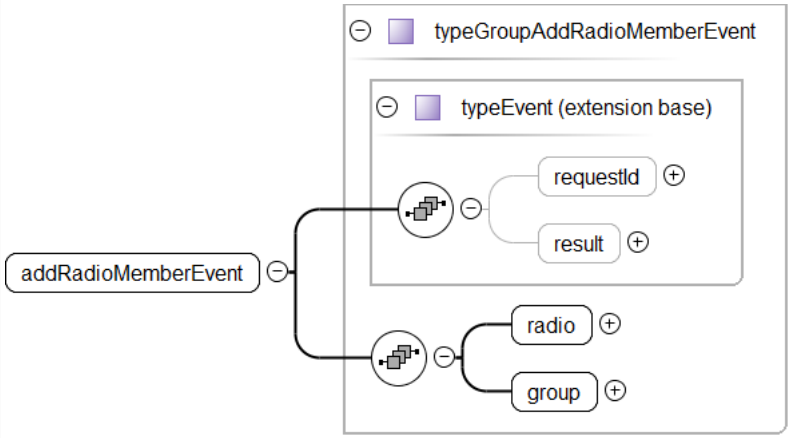
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup>
Source	<xs:element name="baseGroup" type="typeSubscriberAddress"/>

Element typeGroupCombinationEvent / constitGroup

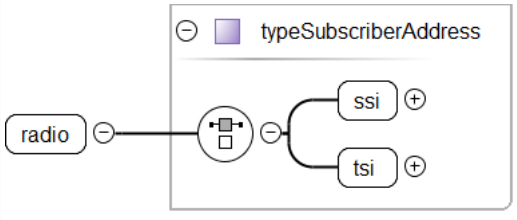
Namespace	DR-GW
Diagram	

Type	typeSubscriberAddress
Properties	content: complex minOccurs: 0 maxOccurs: 7
Model	ssi | tsi
Children	ssi, tsi
Instance	<constitGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </constitGroup>
Source	<xs:element name="constitGroup" type="typeSubscriberAddress" minOccurs="0" maxOccurs="7"/>

Element interfaceGroup / addRadioMemberEvent

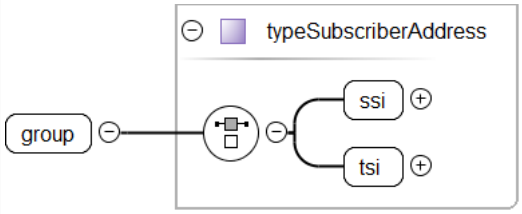
Namespace	DR-GW
Diagram	
Type	typeGroupAddRadioMemberEvent
Type hierarchy	• typeEvent • typeGroupAddRadioMemberEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , group
Children	group, radio, requestId, result
Instance	<addRadioMemberEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <group>{1,1}</group> </addRadioMemberEvent>
Source	<xs:element name="addRadioMemberEvent" type="typeGroupAddRadioMemberEvent"/>

Element typeGroupAddRadioMemberEvent / radio

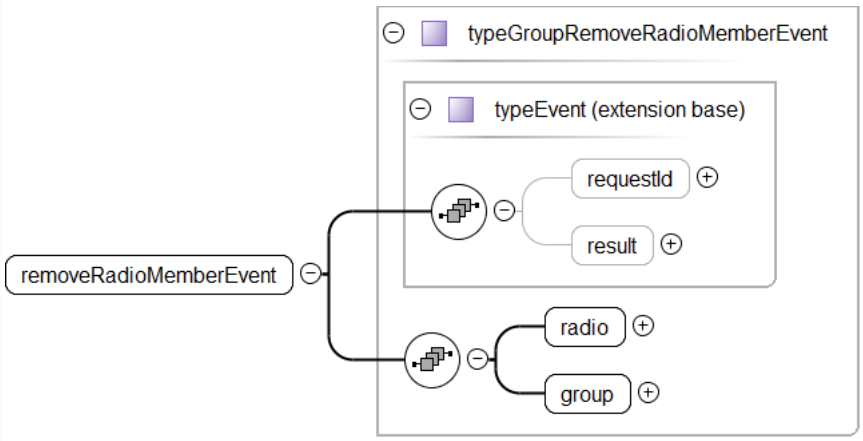
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi

Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

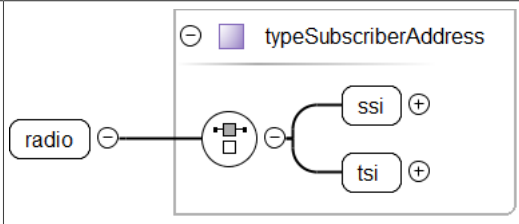
Element typeGroupAddRadioMemberEvent / group

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

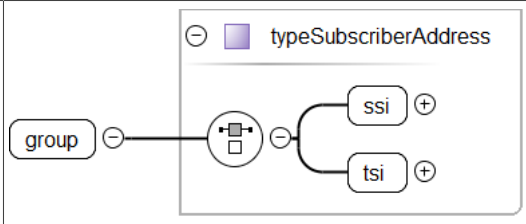
Element interfaceGroup / removeRadioMemberEvent

Namespace	DR-GW
Diagram	
Type	typeGroupRemoveRadioMemberEvent
Type hierarchy	• typeEvent • typeGroupRemoveRadioMemberEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , group
Children	group, radio, requestId, result
Instance	<removeRadioMemberEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <group>{1,1}</group> </removeRadioMemberEvent>
Source	<xs:element name="removeRadioMemberEvent" type="typeGroupRemoveRadioMemberEvent"/>

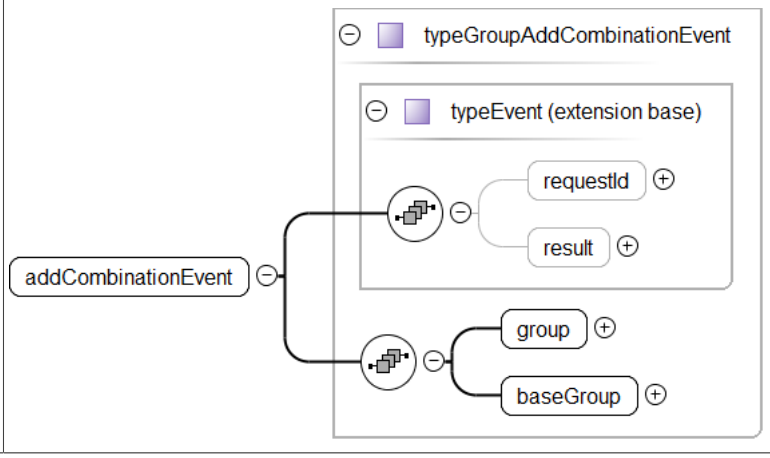
Element typeGroupRemoveRadioMemberEvent / radio

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

Element typeGroupRemoveRadioMemberEvent / group

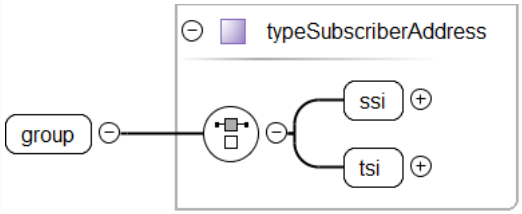
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

Element interfaceGroup / addCombinationEvent

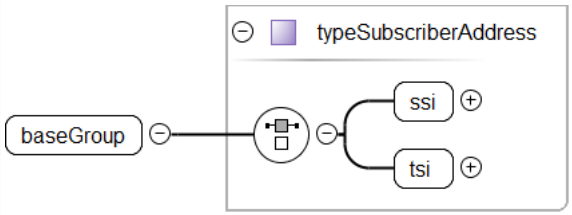
Namespace	DR-GW
Diagram	

Type	typeGroupAddCombinationEvent
Type hierarchy	• typeEvent • typeGroupAddCombinationEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , baseGroup
Children	baseGroup, group, requestId, result
Instance	<addCombinationEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <baseGroup>{1,1}</baseGroup> </addCombinationEvent>
Source	<xs:element name="addCombinationEvent" type="typeGroupAddCombinationEvent"/>

Element typeGroupAddCombinationEvent / group

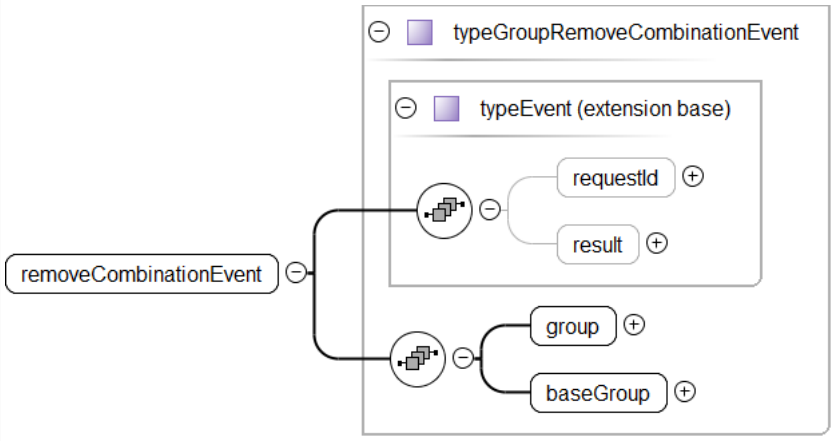
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

Element typeGroupAddCombinationEvent / baseGroup

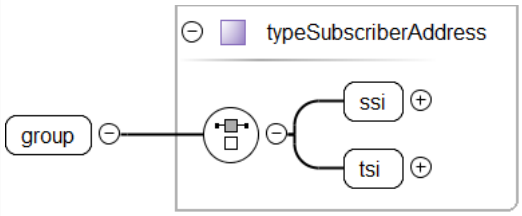
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup>
Source	<xs:element name="baseGroup" type="typeSubscriberAddress"/>

Element interfaceGroup / removeCombinationEvent

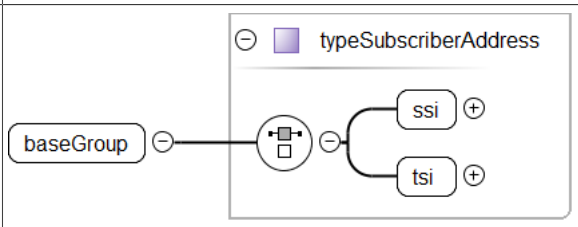
Namespace	DR-GW

Diagram	
Type	typeGroupRemoveCombinationEvent
Type hierarchy	• typeEvent • typeGroupRemoveCombinationEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , baseGroup
Children	baseGroup, group, requestId, result
Instance	<removeCombinationEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <baseGroup>{1,1}</baseGroup> </removeCombinationEvent>
Source	<xs:element name="removeCombinationEvent" type="typeGroupRemoveCombinationEvent"/>

Element typeGroupRemoveCombinationEvent / group

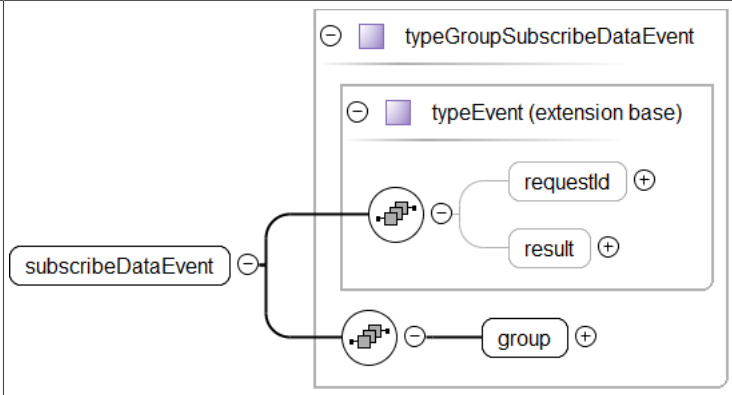
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

Element typeGroupRemoveCombinationEvent / baseGroup

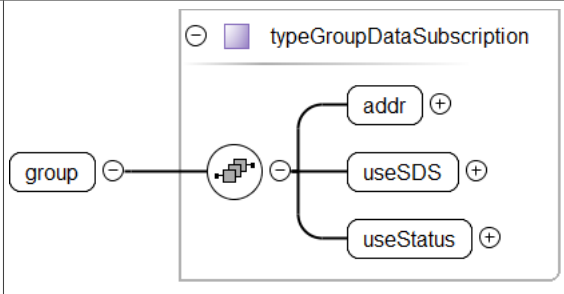
Namespace	DR-GW
Diagram	

Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup>
Source	<xs:element name="baseGroup" type="typeSubscriberAddress"/>

Element interfaceGroup / subscribeDataEvent

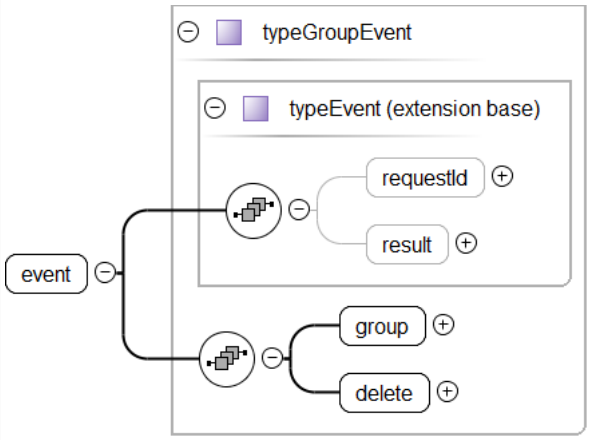
Namespace	DR-GW
Diagram	
Type	typeGroupSubscribeDataEvent
Type hierarchy	• typeEvent • typeGroupSubscribeDataEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group
Children	group, requestId, result
Instance	<subscribeDataEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> </subscribeDataEvent>
Source	<xs:element name="subscribeDataEvent" type="typeGroupSubscribeDataEvent"/>

Element typeGroupSubscribeDataEvent / group

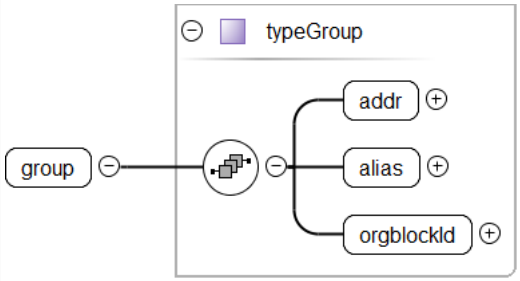
Namespace	DR-GW
Diagram	
Type	typeGroupDataSubscription
Properties	content: complex
Model	addr , useSDS , useStatus
Children	addr, useSDS, useStatus
Instance	<group xmlns="DR-GW"> <addr>{1,1}</addr>

	<useSDS>{1,1}</useSDS> <useStatus>{1,1}</useStatus> </group>
Source	<xs:element name="group" type="typeGroupDataSubscription"/>

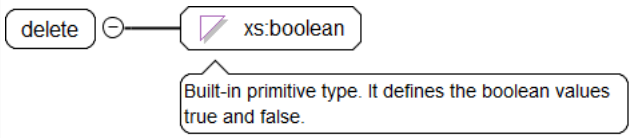
Element interfaceGroup / event

Namespace	DR-GW
Diagram	
Type	typeGroupEvent
Type hierarchy	• typeEvent • typeGroupEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , group , delete
Children	delete, group, requestId, result
Instance	<event xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <group>{1,1}</group> <delete>{1,1}</delete> </event>
Source	<xs:element name="event" type="typeGroupEvent"/>

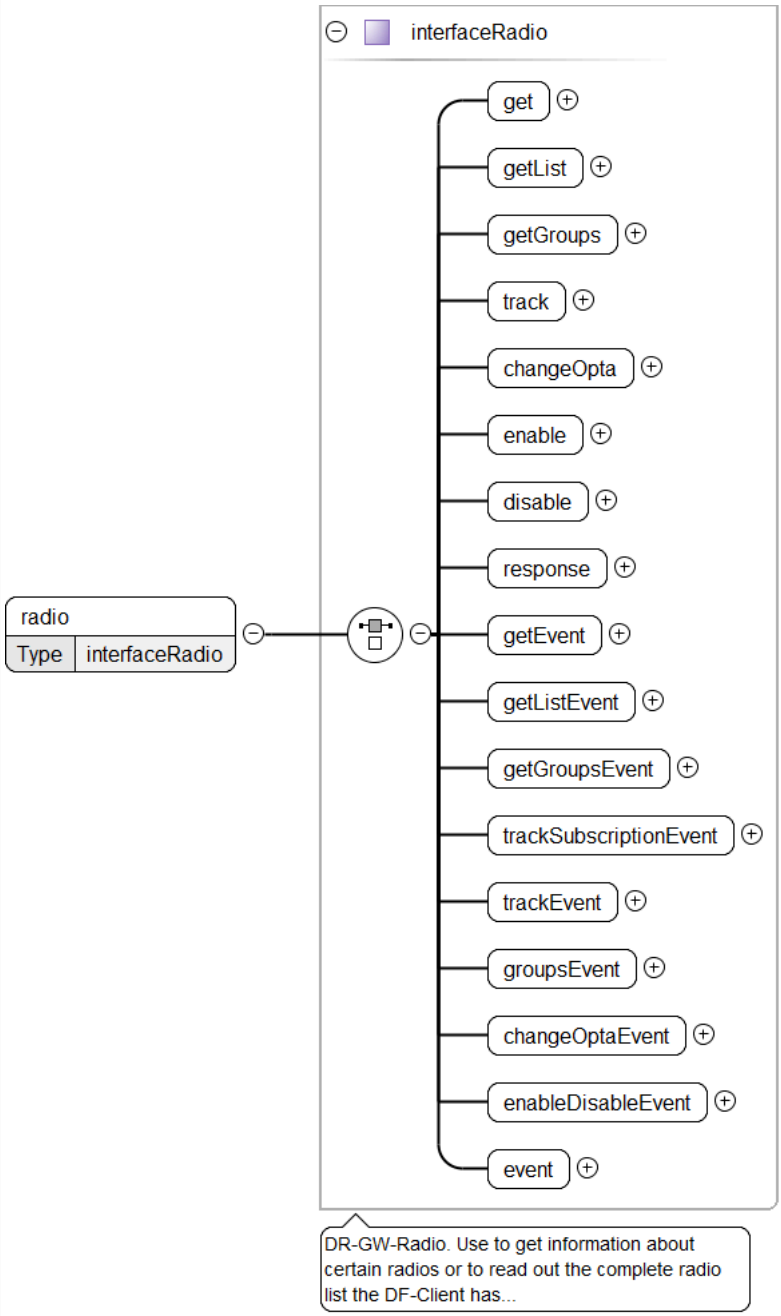
Element typeGroupEvent / group

Namespace	DR-GW
Diagram	
Type	typeGroup
Properties	content: complex
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId
Instance	<group xmlns="DR-GW"> <addr>{1,1}</addr> <alias>{1,1}</alias> <orgblockId>{1,1}</orgblockId> </group>
Source	<xs:element name="group" type="typeGroup"/>

Element `typeGroupEvent / delete`

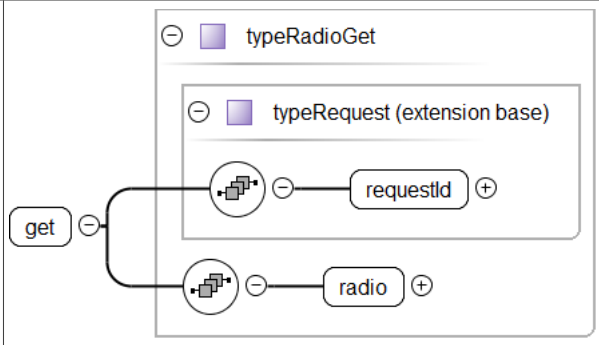
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	`<xs:element name="delete" type="xs:boolean"/>`

Element `drgw / radio`

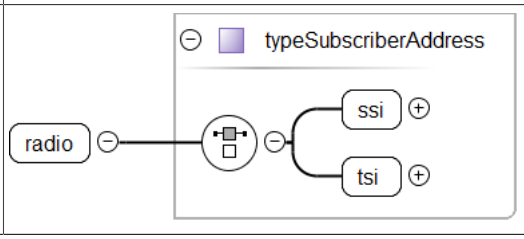
Namespace	DR-GW
Diagram	
Type	interfaceRadio

Properties	content: complex
Model	get | getList | getGroups | track | changeOpta | enable | disable | response | getEvent | getListEvent | getGroupsEvent | trackSubscrip-tionEvent | trackEvent | groupsEvent | changeOptaEvent | enableDisableEvent | event
Children	changeOpta, changeOptaEvent, disable, enable, enableDisableEvent, event, get, getEvent, getGroups, getGroupsEvent, getList, getListEvent, groupsEvent, response, track, trackEvent, trackSubscriptionEvent
Instance	<radio xmlns="DR-GW"> <get>{1,1}</get> <getList>{1,1}</getList> <getGroups>{1,1}</getGroups> <track>{1,1}</track> <changeOpta>{1,1}</changeOpta> <enable>{1,1}</enable> <disable>{1,1}</disable> <response>{1,1}</response> <getEvent>{1,1}</getEvent> <getListEvent>{1,1}</getListEvent> <getGroupsEvent>{1,1}</getGroupsEvent> <trackSubscriptionEvent>{1,1}</trackSubscriptionEvent> <trackEvent>{1,1}</trackEvent> <groupsEvent>{1,1}</groupsEvent> <changeOptaEvent>{1,1}</changeOptaEvent> <enableDisableEvent>{1,1}</enableDisableEvent> <event>{1,1}</event> </radio>
Source	<xs:element name="radio" type="interfaceRadio"/>

Element interfaceRadio / get

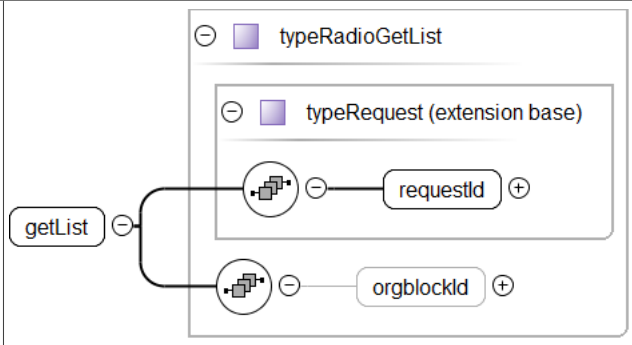
Namespace	DR-GW
Diagram	
Type	typeRadioGet
Type hierar-chy	• typeRequest • typeRadioGet
Properties	content: complex
Model	requestId , radio
Children	radio, requestId
Instance	<get xmlns="DR-GW"> <requestId>{1,1}</requestId> <radio>{1,1}</radio> </get>
Source	<xs:element name="get" type="typeRadioGet"/>

Element typeRadioGet / radio

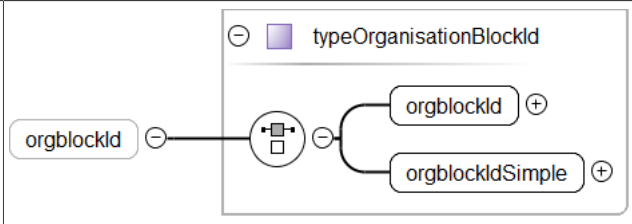
Namespace	DR-GW
Diagram	

Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

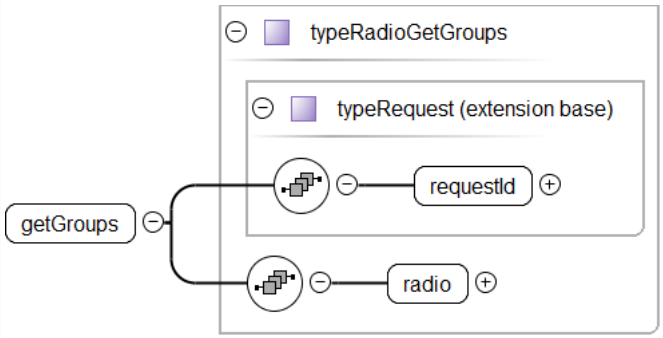
Element interfaceRadio / getList

Namespace	DR-GW
Diagram	
Type	typeRadioGetList
Type hierarchy	• typeRequest • typeRadioGetList
Properties	content: complex
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId
Instance	<getList xmlns="DR-GW"> <requestId>{1,1}</requestId> <orgblockId>{0,1}</orgblockId> </getList>
Source	<xs:element name="getList" type="typeRadioGetList"/>

Element typeRadioGetList / orgblockId

Namespace	DR-GW
Diagram	
Type	typeOrganisationBlockId
Properties	content: complex minOccurs: 0
Model	orgblockId | orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId>
Source	<xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/>

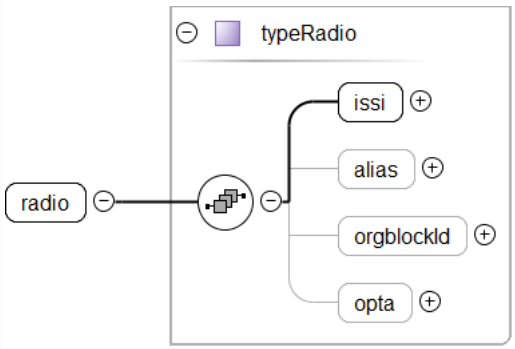
Element `interfaceRadio / getGroups`

Namespace	DR-GW
Diagram	
Type	typeRadioGetGroups
Type hierar-chy	• typeRequest • typeRadioGetGroups
Properties	content: complex
Model	requestId , radio
Children	radio, requestId

Instance	`<getGroups xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <radio>{1,1}</radio>`
	`</getGroups>`

Source	`<xs:element name="getGroups" type="typeRadioGetGroups"/>`

Element `typeRadioGetGroups / radio`

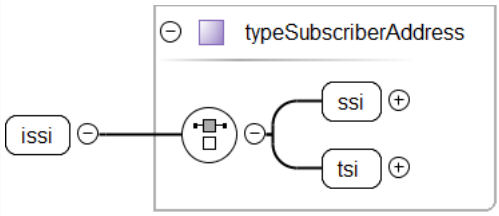
Namespace	DR-GW
Diagram	
Type	typeRadio
Properties	content: complex
Model	issi , alias{0,1} , orgblockId{0,1} , opta{0,1}
Children	alias, issi, opta, orgblockId

Instance	`<radio xmlns="DR-GW">`
	`  <issi>{1,1}</issi>`
	`  <alias>{0,1}</alias>`
	`  <orgblockId>{0,1}</orgblockId>`
	`  <opta>{0,1}</opta>`
	`</radio>`

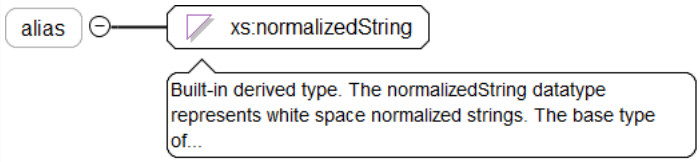
Source	`<xs:element name="radio" type="typeRadio"/>`

Element `typeRadio / issi`

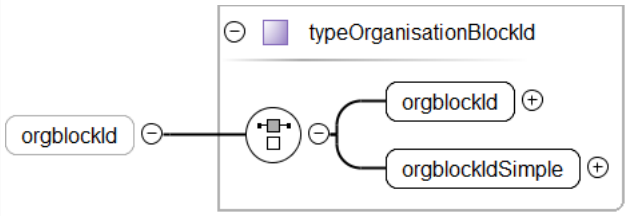
Namespace	DR-GW

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<issi xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </issi>
Source	<xs:element name="issi" type="typeSubscriberAddress"/>

Element typeRadio / alias

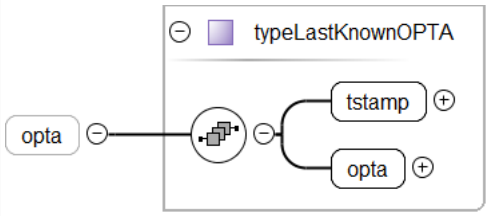
Namespace	DR-GW
Diagram	
Type	xs:normalizedString
Properties	content: simple minOccurs: 0
Source	<xs:element name="alias" type="xs:normalizedString" minOccurs="0"/>

Element typeRadio / orgblockId

Namespace	DR-GW
Diagram	
Type	typeOrganisationBlockId
Properties	content: complex minOccurs: 0
Model	orgblockId | orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId>
Source	<xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/>

Element typeRadio / opta

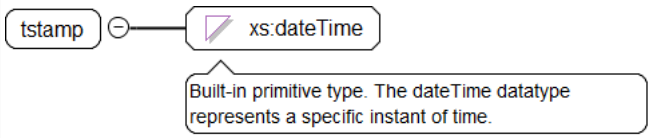
Namespace	DR-GW

Diagram	
Type	typeLastKnownOPTA
Properties	content: complex minOccurs: 0
Model	tstamp , opta
Children	opta, tstamp

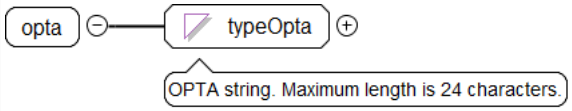
Instance	`<opta xmlns="DR-GW">`
	`  <tstamp>{1,1}</tstamp>`
	`  <opta>{1,1}</opta>`
	`</opta>`

Source	`<xs:element name="opta" type="typeLastKnownOPTA" minOccurs="0"/>`

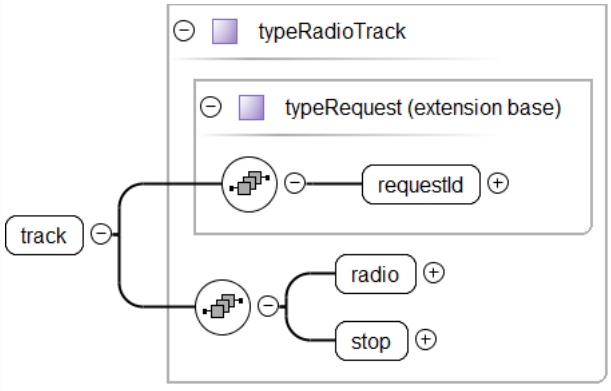
Element typeLastKnownOPTA / tstamp

Namespace	DR-GW
Diagram	
Type	xs:dateTime
Properties	content: simple
Source	`<xs:element name="tstamp" type="xs:dateTime"/>`

Element typeLastKnownOPTA / opta

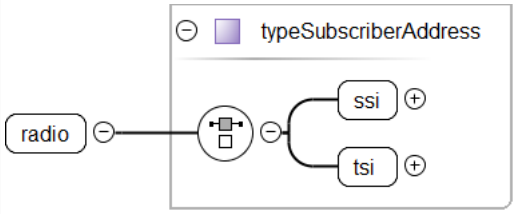
Namespace	DR-GW
Diagram	
Type	typeOpta
Properties	content: simple
Facets	maxLength 24
Source	`<xs:element name="opta" type="typeOpta"/>`

Element interfaceRadio / track

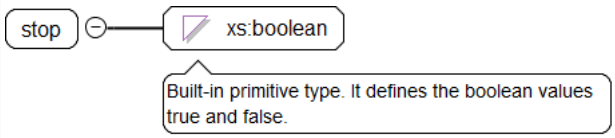
Namespace	DR-GW
Diagram	

Type	typeRadioTrack
Type hierarchy	• typeRequest • typeRadioTrack
Properties	content: complex
Model	requestId , radio , stop
Children	radio, requestId, stop
Instance	<track xmlns="DR-GW"> <requestId>{1,1}</requestId> <radio>{1,1}</radio> <stop>{1,1}</stop> </track>
Source	<xs:element name="track" type="typeRadioTrack"/>

Element typeRadioTrack / radio

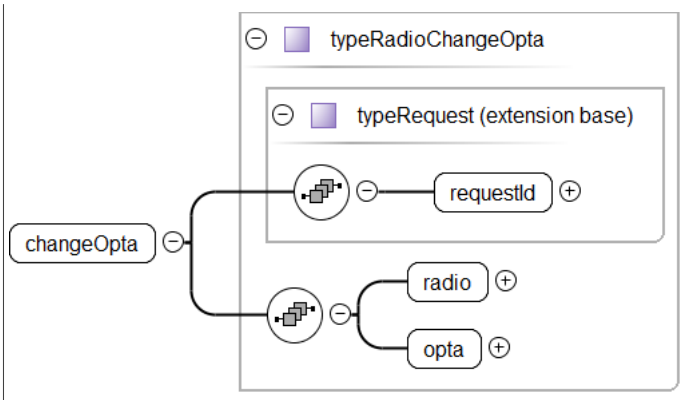
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

Element typeRadioTrack / stop

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="stop" type="xs:boolean"/>

Element interfaceRadio / changeOpta

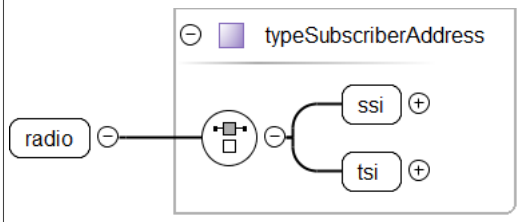
Namespace	DR-GW

Diagram	
Type	typeRadioChangeOpta
Type hierarchy	• typeRequest • typeRadioChangeOpta
Properties	content: complex
Model	requestId , radio , opta
Children	opta, radio, requestId

Instance	`<changeOpta xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <radio>{1,1}</radio>`
	`  <opta>{1,1}</opta>`
	`</changeOpta>`

Source	`<xs:element name="changeOpta" type="typeRadioChangeOpta"/>`

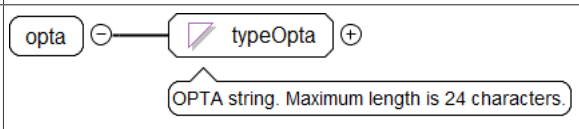
Element typeRadioChangeOpta / radio

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<radio xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</radio>`

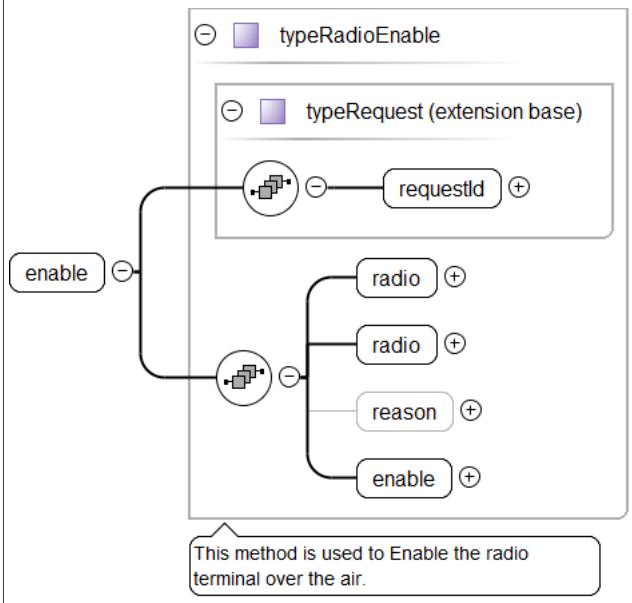
Source	`<xs:element name="radio" type="typeSubscriberAddress"/>`

Element typeRadioChangeOpta / opta

Namespace	DR-GW
Diagram	
Type	typeOpta
Properties	content: simple
Facets	maxLength 24

Source	`<xs:element name="opta" type="typeOpta"/>`

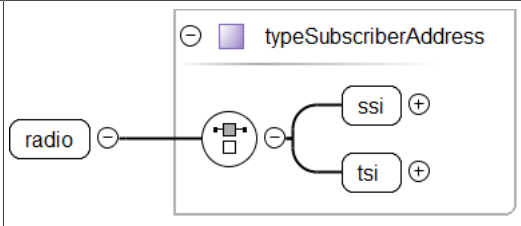
Element `interfaceRadio / enable`

Namespace	DR-GW
Diagram	
Type	typeRadioEnable
Type hierarchy	• typeRequest • typeRadioEnable
Properties	content: complex
Model	requestId , radio , radio , reason{0,1} , enable
Children	enable, radio, reason, requestId

Instance	`<enable xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <radio>{1,1}</radio>`
	`  <radio>{1,1}</radio>`
	`  <reason>{0,1}</reason>`
	`  <enable>{1,1}</enable>`
	`</enable>`

Source	`<xs:element name="enable" type="typeRadioEnable"/>`

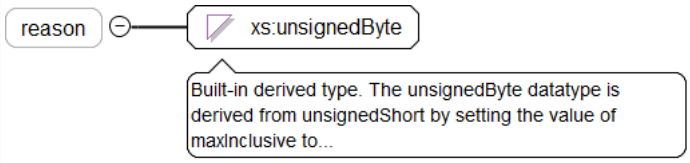
Element `typeRadioEnable / radio`

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

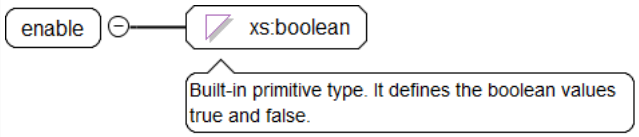
Instance	`<radio xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</radio>`

Source	`<xs:element name="radio" type="typeSubscriberAddress"/>`

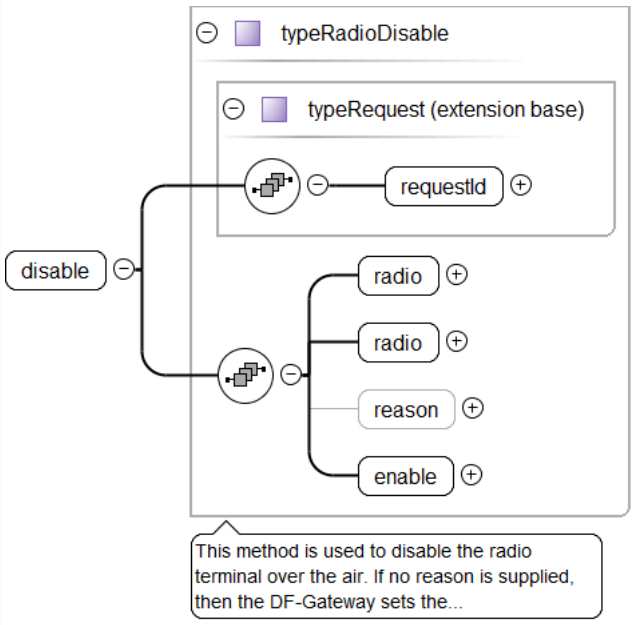
Element typeRadioEnable / reason

Namespace	DR-GW
Diagram	
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0
Source	`<xs:element name="reason" type="xs:unsignedByte" minOccurs="0"/>`

Element typeRadioEnable / enable

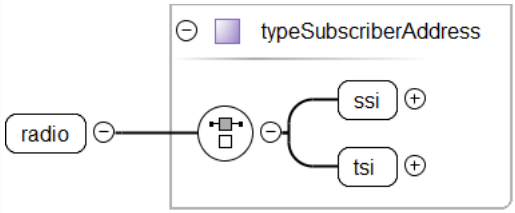
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	`<xs:element name="enable" type="xs:boolean"/>`

Element interfaceRadio / disable

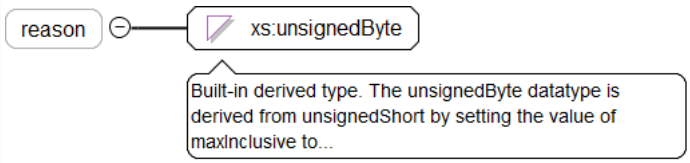
Namespace	DR-GW
Diagram	
Type	typeRadioDisable
Type hierarchy	• typeRequest • typeRadioDisable
Properties	content: complex
Model	requestId , radio , radio , reason{0,1} , enable
Children	enable, radio, reason, requestId

Instance	<disable xmlns="DR-GW"> <requestId>{1,1}</requestId> <radio>{1,1}</radio> <radio>{1,1}</radio> <reason>{0,1}</reason> <enable>{1,1}</enable> </disable>
Source	<xs:element name="disable" type="typeRadioDisable"/>

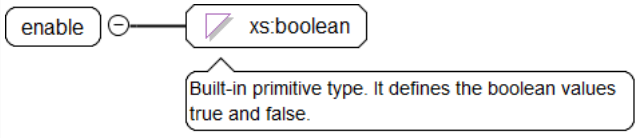
Element typeRadioDisable / radio

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

Element typeRadioDisable / reason

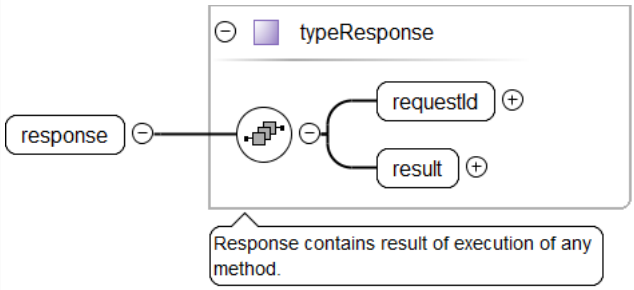
Namespace	DR-GW
Diagram	
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0
Source	<xs:element name="reason" type="xs:unsignedByte" minOccurs="0"/>

Element typeRadioDisable / enable

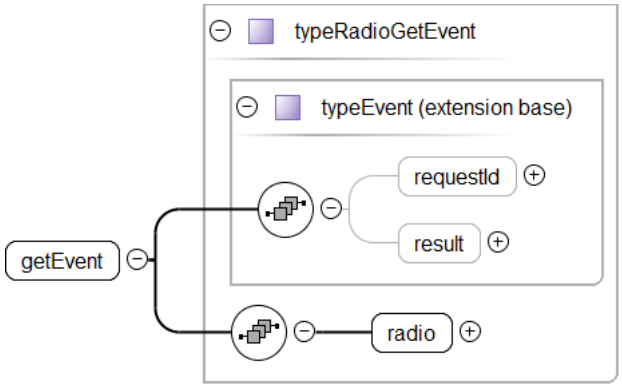
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="enable" type="xs:boolean"/>

Element interfaceRadio / response

Namespace	DR-GW

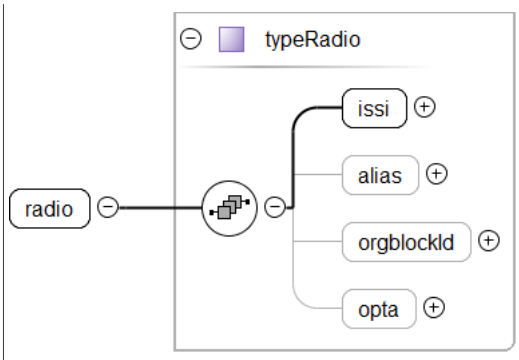
Diagram	
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response>
Source	<xs:element name="response" type="typeResponse"/>

Element interfaceRadio / getEvent

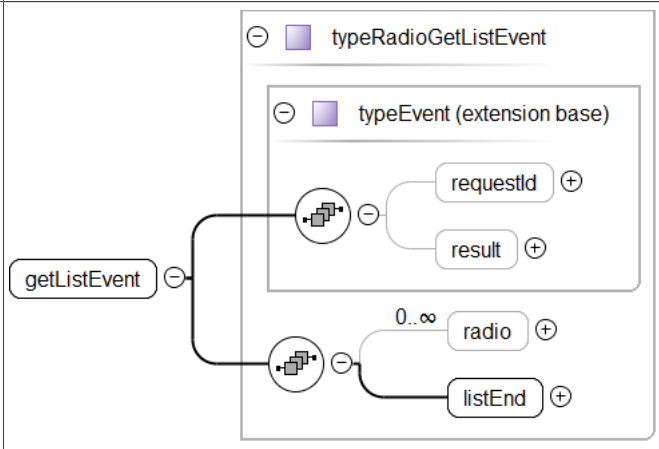
Namespace	DR-GW
Diagram	
Type	typeRadioGetEvent
Type hierarchy	• typeEvent • typeRadioGetEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio
Children	radio, requestId, result
Instance	<getEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> </getEvent>
Source	<xs:element name="getEvent" type="typeRadioGetEvent"/>

Element typeRadioGetEvent / radio

Namespace	DR-GW

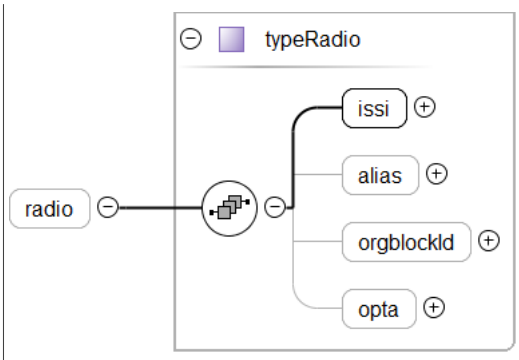
Diagram	
Type	typeRadio
Properties	content: complex
Model	issi , alias{0,1} , orgblockId{0,1} , opta{0,1}
Children	alias, issi, opta, orgblockId
Instance	<radio xmlns="DR-GW"> <issi>{1,1}</issi> <alias>{0,1}</alias> <orgblockId>{0,1}</orgblockId> <opta>{0,1}</opta> </radio>
Source	<xs:element name="radio" type="typeRadio"/>

Element interfaceRadio / getListEvent

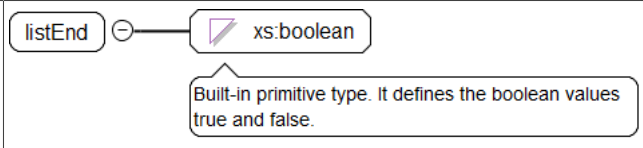
Namespace	DR-GW
Diagram	
Type	typeRadioGetListEvent
Type hierarchy	• typeEvent • typeRadioGetListEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio* , listEnd
Children	listEnd, radio, requestId, result
Instance	<getListEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{0,unbounded}</radio> <listEnd>{1,1}</listEnd> </getListEvent>
Source	<xs:element name="getListEvent" type="typeRadioGetListEvent"/>

Element typeRadioGetListEvent / radio

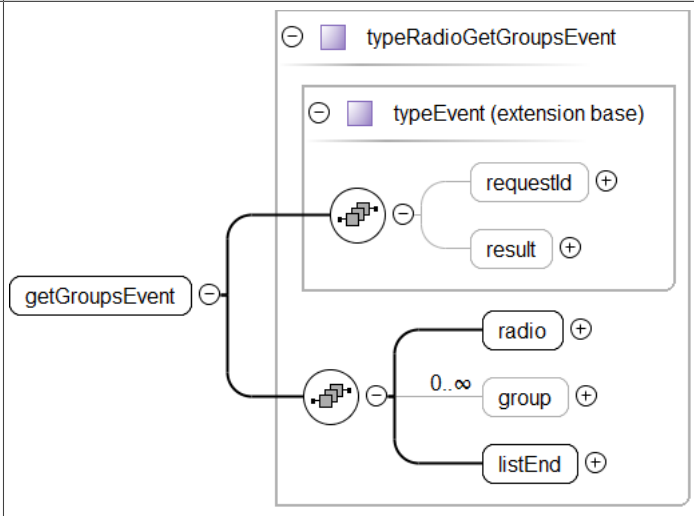
Namespace	DR-GW

Diagram	
Type	typeRadio
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	issi , alias{0,1} , orgblockId{0,1} , opta{0,1}
Children	alias, issi, opta, orgblockId
Instance	<radio xmlns="DR-GW"> <issi>{1,1}</issi> <alias>{0,1}</alias> <orgblockId>{0,1}</orgblockId> <opta>{0,1}</opta> </radio>
Source	<xs:element name="radio" type="typeRadio" minOccurs="0" maxOccurs="unbounded"/>

Element typeRadioGetListEvent / listEnd

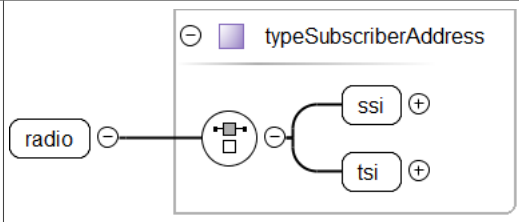
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="listEnd" type="xs:boolean"/>

Element interfaceRadio / getGroupsEvent

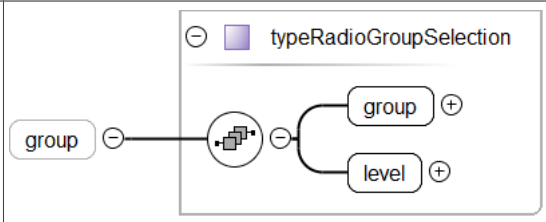
Namespace	DR-GW
Diagram	
Type	typeRadioGetGroupsEvent

Type hierarchy	• typeEvent • typeRadioGetGroupsEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , group* , listEnd
Children	group, listEnd, radio, requestId, result
Instance	<getGroupsEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <group>{0,unbounded}</group> <listEnd>{1,1}</listEnd> </getGroupsEvent>
Source	<xs:element name="getGroupsEvent" type="typeRadioGetGroupsEvent"/>

Element typeRadioGetGroupsEvent / radio

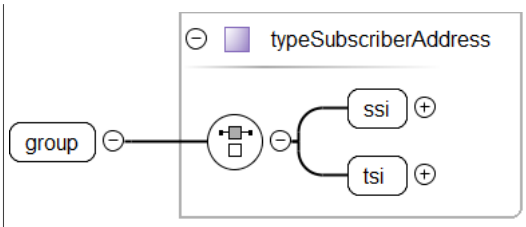
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

Element typeRadioGetGroupsEvent / group

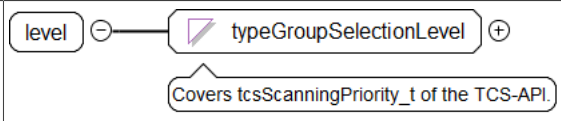
Namespace	DR-GW
Diagram	
Type	typeRadioGroupSelection
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	group , level
Children	group, level
Instance	<group xmlns="DR-GW"> <group>{1,1}</group> <level>{1,1}</level> </group>
Source	<xs:element name="group" type="typeRadioGroupSelection" minOccurs="0" maxOccurs="unbounded"/>

Element typeRadioGroupSelection / group

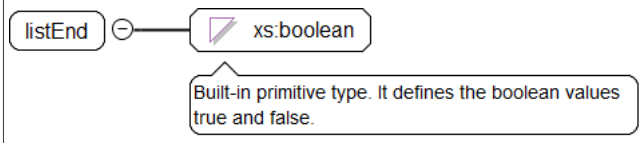
Namespace	DR-GW

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

Element typeRadioGroupSelection / level

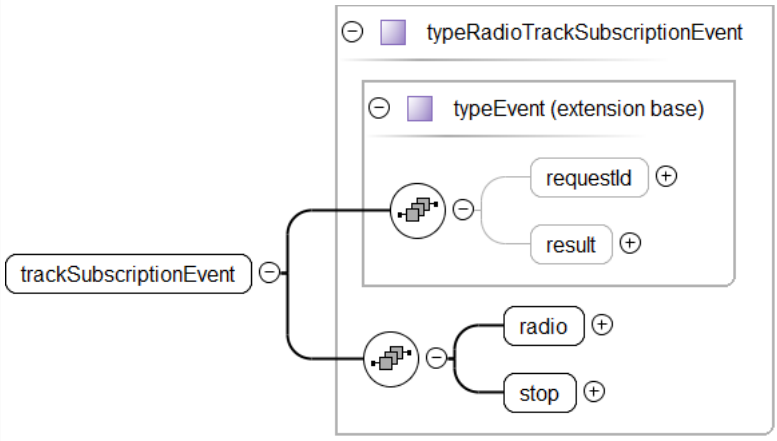
Namespace	DR-GW
Diagram	
Type	typeGroupSelectionLevel
Properties	content: simple
Facets	enumeration notScanned enumeration low enumeration normal enumeration selected enumeration high enumeration background
Source	<xs:element name="level" type="typeGroupSelectionLevel"/>

Element typeRadioGetGroupsEvent / listEnd

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="listEnd" type="xs:boolean"/>

Element interfaceRadio / trackSubscriptionEvent

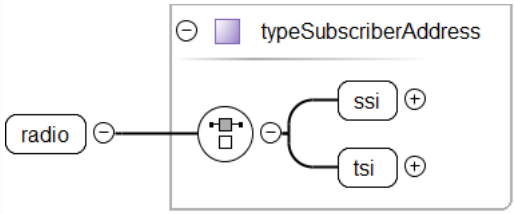
Namespace	DR-GW

Diagram	
Type	typeRadioTrackSubscriptionEvent
Type hierarchy	• typeEvent • typeRadioTrackSubscriptionEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , stop
Children	radio, requestId, result, stop

Instance	`<trackSubscriptionEvent xmlns="DR-GW">`
	`<requestId>{0,1}</requestId>`
	`<result>{0,1}</result>`
	`<radio>{1,1}</radio>`
	`<stop>{1,1}</stop>`
	`</trackSubscriptionEvent>`

Source	`<xs:element name="trackSubscriptionEvent" type="typeRadioTrackSubscriptionEvent"/>`

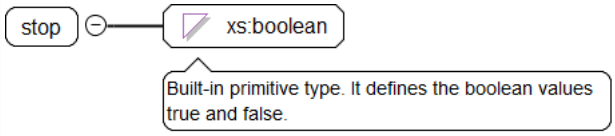
Element typeRadioTrackSubscriptionEvent / radio

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<radio xmlns="DR-GW">`
	`<ssi>{1,1}</ssi>`
	`<tsi>{1,1}</tsi>`
	`</radio>`

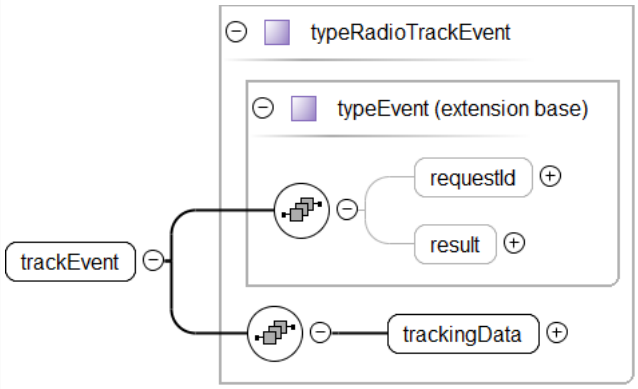
Source	`<xs:element name="radio" type="typeSubscriberAddress"/>`

Element typeRadioTrackSubscriptionEvent / stop

Namespace	DR-GW
Diagram	
Type	xs:boolean

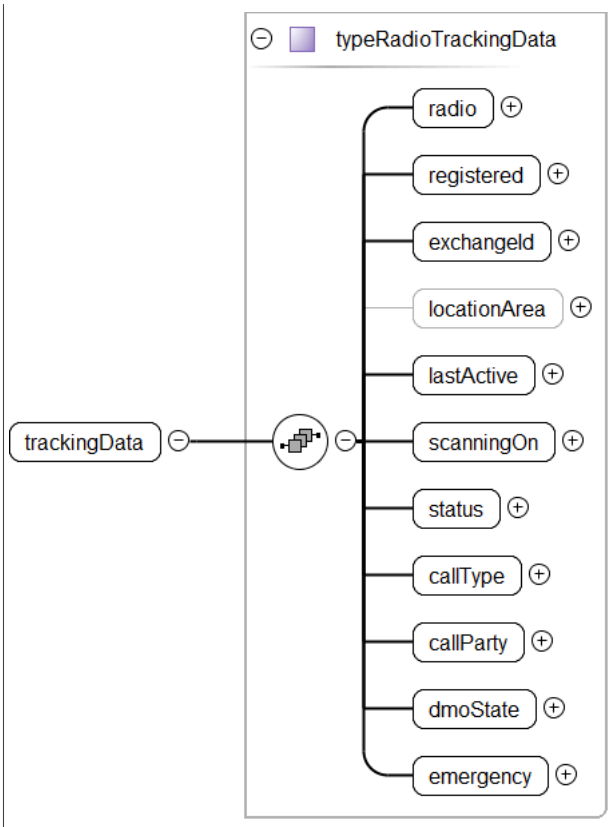
Properties	content: simple
Source	<xs:element name="stop" type="xs:boolean"/>

Element interfaceRadio / trackEvent

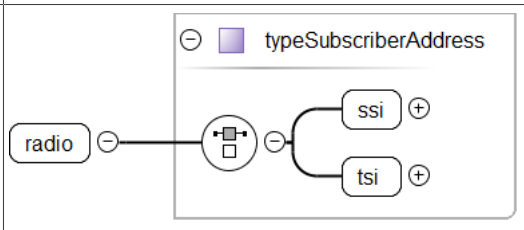
Namespace	DR-GW
Diagram	
Type	typeRadioTrackEvent
Type hierarchy	• typeEvent • typeRadioTrackEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , trackingData
Children	requestId, result, trackingData
Instance	<trackEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <trackingData>{1,1}</trackingData> </trackEvent>
Source	<xs:element name="trackEvent" type="typeRadioTrackEvent"/>

Element typeRadioTrackEvent / trackingData

Namespace	DR-GW

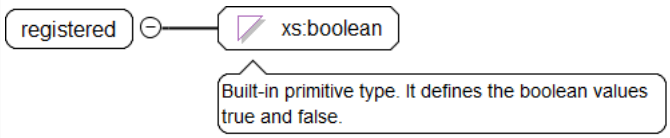
Diagram	
Type	typeRadioTrackingData
Properties	content: complex
Model	radio , registered , exchangeId , locationArea{0,1} , lastActive , scanningOn , status , callType , callParty , dmoState , emergency
Children	callParty, callType, dmoState, emergency, exchangeId, lastActive, locationArea, radio, registered, scanningOn, status
Instance	<trackingData xmlns="DR-GW"> <radio>{1,1}</radio> <registered>{1,1}</registered> <exchangeId>{1,1}</exchangeId> <locationArea>{0,1}</locationArea> <lastActive>{1,1}</lastActive> <scanningOn>{1,1}</scanningOn> <status>{1,1}</status> <callType>{1,1}</callType> <callParty>{1,1}</callParty> <dmoState>{1,1}</dmoState> <emergency>{1,1}</emergency> </trackingData>
Source	<xs:element name="trackingData" type="typeRadioTrackingData"/>

Element typeRadioTrackingData / radio

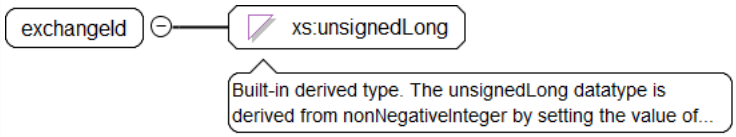
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

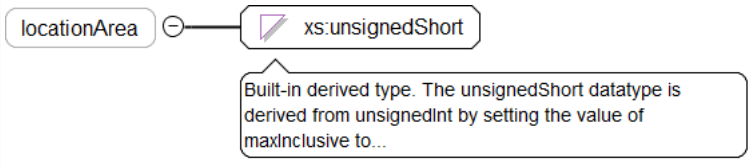
Element typeRadioTrackingData / registered

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="registered" type="xs:boolean"/>

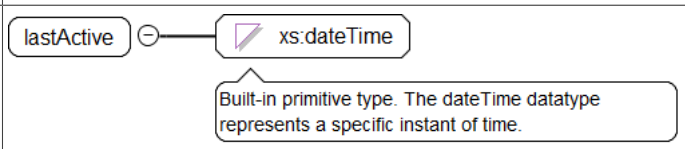
Element typeRadioTrackingData / exchangeId

Namespace	DR-GW
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	<xs:element name="exchangeId" type="xs:unsignedLong"/>

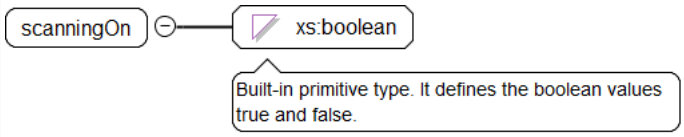
Element typeRadioTrackingData / locationArea

Namespace	DR-GW
Diagram	
Type	xs:unsignedShort
Properties	content: simple minOccurs: 0
Source	<xs:element name="locationArea" type="xs:unsignedShort" minOccurs="0"/>

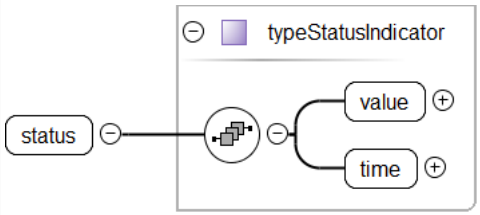
Element typeRadioTrackingData / lastActive

Namespace	DR-GW
Diagram	
Type	xs:dateTime
Properties	content: simple
Source	<xs:element name="lastActive" type="xs:dateTime"/>

Element typeRadioTrackingData / scanningOn

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	`<xs:element name="scanningOn" type="xs:boolean"/>`

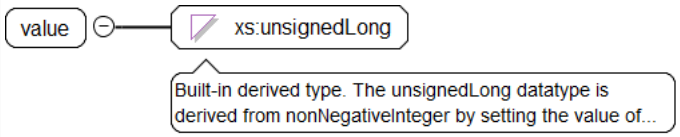
Element typeRadioTrackingData / status

Namespace	DR-GW
Diagram	
Type	typeStatusIndicator
Properties	content: complex
Model	value , time
Children	time, value

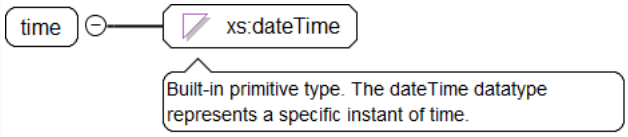
Instance	`<status xmlns="DR-GW">`
	`  <value>{1,1}</value>`
	`  <time>{1,1}</time>`
	`</status>`

Source	`<xs:element name="status" type="typeStatusIndicator"/>`

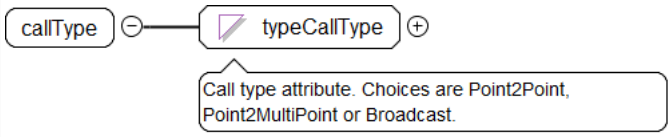
Element typeStatusIndicator / value

Namespace	DR-GW
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	`<xs:element name="value" type="xs:unsignedLong"/>`

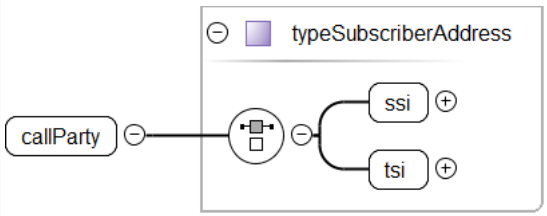
Element typeStatusIndicator / time

Namespace	DR-GW
Diagram	
Type	xs:dateTime
Properties	content: simple
Source	`<xs:element name="time" type="xs:dateTime"/>`

Element typeRadioTrackingData / callType

Namespace	DR-GW
Diagram	
Type	typeCallType
Properties	content: simple
Facets	enumeration p2p enumeration p2mp enumeration bcast
Source	`<xs:element name="callType" type="typeCallType"/>`

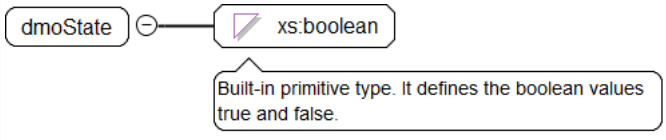
Element typeRadioTrackingData / callParty

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

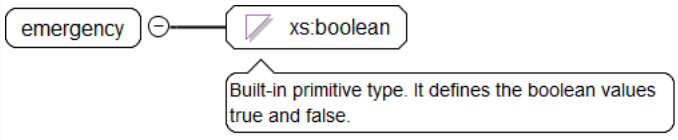
Instance	`<callParty xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</callParty>`

Source	`<xs:element name="callParty" type="typeSubscriberAddress"/>`

Element typeRadioTrackingData / dmoState

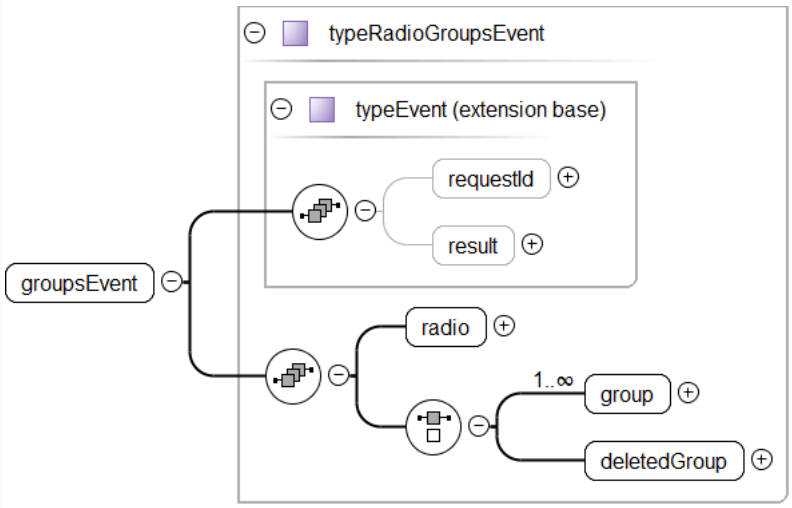
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	`<xs:element name="dmoState" type="xs:boolean"/>`

Element typeRadioTrackingData / emergency

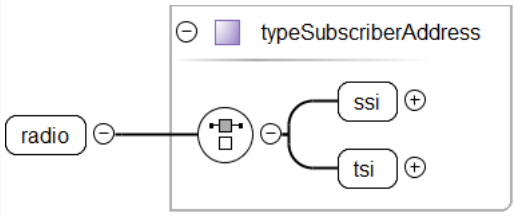
Namespace	DR-GW
Diagram	

Type	xs:boolean
Properties	content: simple
Source	<xs:element name="emergency" type="xs:boolean"/>

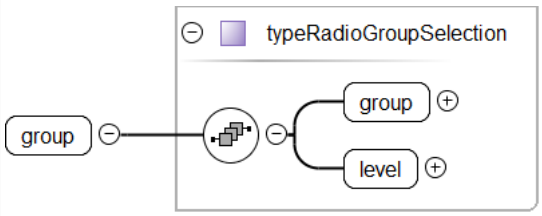
Element interfaceRadio / groupsEvent

Namespace	DR-GW
Diagram	
Type	typeRadioGroupsEvent
Type hierarchy	• typeEvent • typeRadioGroupsEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , (group+ | deletedGroup)
Children	deletedGroup, group, radio, requestId, result
Instance	<groupsEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <group>{1,unbounded}</group> <deletedGroup>{1,1}</deletedGroup> </groupsEvent>
Source	<xs:element name="groupsEvent" type="typeRadioGroupsEvent"/>

Element typeRadioGroupsEvent / radio

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

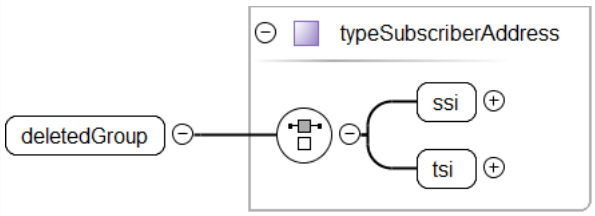
Element `typeRadioGroupsEvent / group`

Namespace	DR-GW
Diagram	
Type	typeRadioGroupSelection
Properties	content: complex maxOccurs: unbounded
Model	group , level
Children	group, level

Instance	`<group xmlns="DR-GW">`
	`  <group>{1,1}</group>`
	`  <level>{1,1}</level>`
	`</group>`

Source	`<xs:element name="group" type="typeRadioGroupSelection" maxOccurs="unbounded"/>`

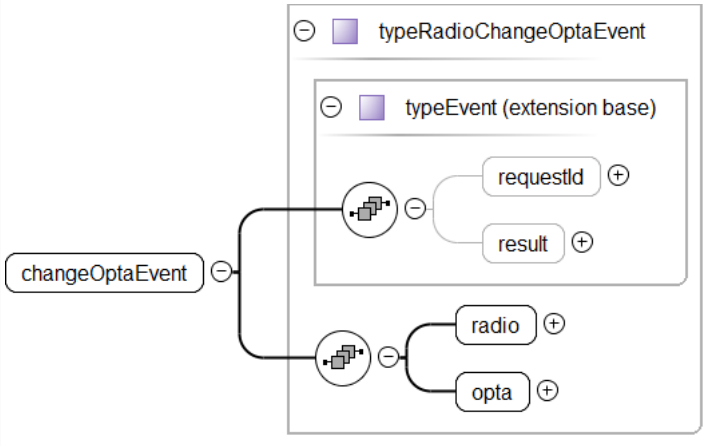
Element `typeRadioGroupsEvent / deletedGroup`

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi

Instance	`<deletedGroup xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</deletedGroup>`

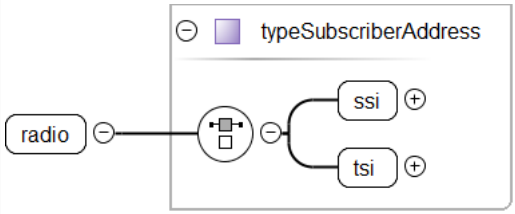
Source	`<xs:element name="deletedGroup" type="typeSubscriberAddress"/>`

Element `interfaceRadio / changeOptaEvent`

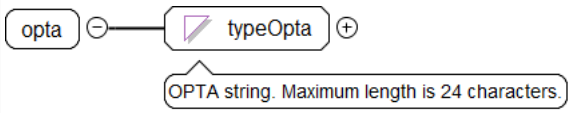
Namespace	DR-GW
Diagram	

Type	typeRadioChangeOptaEvent
Type hierarchy	• typeEvent • typeRadioChangeOptaEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , opta
Children	opta, radio, requestId, result
Instance	<changeOptaEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <opta>{1,1}</opta> </changeOptaEvent>
Source	<xs:element name="changeOptaEvent" type="typeRadioChangeOptaEvent"/>

Element typeRadioChangeOptaEvent / radio

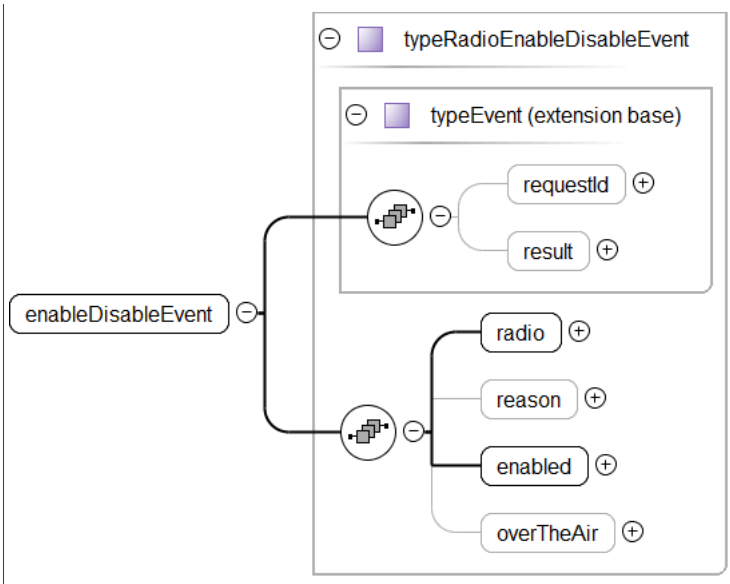
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

Element typeRadioChangeOptaEvent / opta

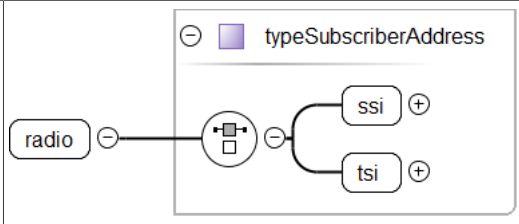
Namespace	DR-GW
Diagram	
Type	typeOpta
Properties	content: simple
Facets	maxLength 24
Source	<xs:element name="opta" type="typeOpta"/>

Element interfaceRadio / enableDisableEvent

Namespace	DR-GW

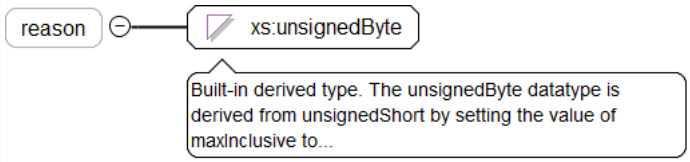
Diagram	
Type	typeRadioEnableDisableEvent
Type hierarchy	• typeEvent • typeRadioEnableDisableEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , reason{0,1} , enabled , overTheAir{0,1}
Children	enabled, overTheAir, radio, reason, requestId, result
Instance	<enableDisableEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <reason>{0,1}</reason> <enabled>{1,1}</enabled> <overTheAir>{0,1}</overTheAir> </enableDisableEvent>
Source	<xs:element name="enableDisableEvent" type="typeRadioEnableDisableEvent"/>

Element typeRadioEnableDisableEvent / radio

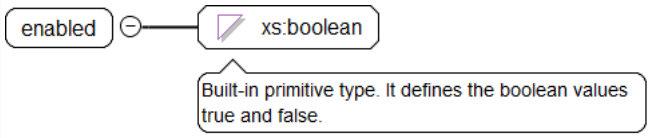
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<radio xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </radio>
Source	<xs:element name="radio" type="typeSubscriberAddress"/>

Element typeRadioEnableDisableEvent / reason

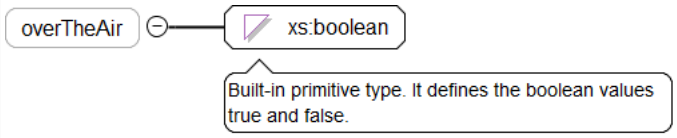
Namespace	DR-GW

Diagram	
Type	xs:unsignedByte
Properties	content: simple minOccurs: 0
Source	`<xs:element name="reason" type="xs:unsignedByte" minOccurs="0"/>`

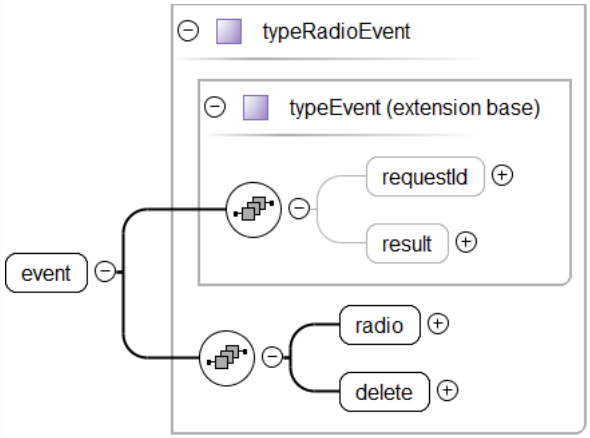
Element typeRadioEnableDisableEvent / enabled

Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	`<xs:element name="enabled" type="xs:boolean"/>`

Element typeRadioEnableDisableEvent / overTheAir

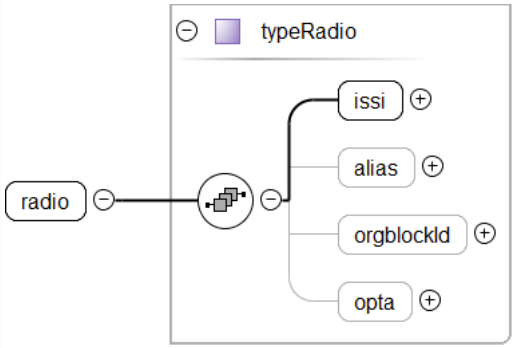
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0
Source	`<xs:element name="overTheAir" type="xs:boolean" minOccurs="0"/>`

Element interfaceRadio / event

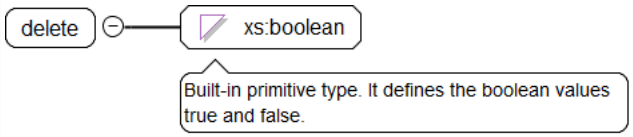
Namespace	DR-GW
Diagram	
Type	typeRadioEvent
Type hierarchy	• typeEvent

	• typeRadioEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , radio , delete
Children	delete, radio, requestId, result
Instance	<event xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <radio>{1,1}</radio> <delete>{1,1}</delete> </event>
Source	<xs:element name="event" type="typeRadioEvent"/>

Element typeRadioEvent / radio

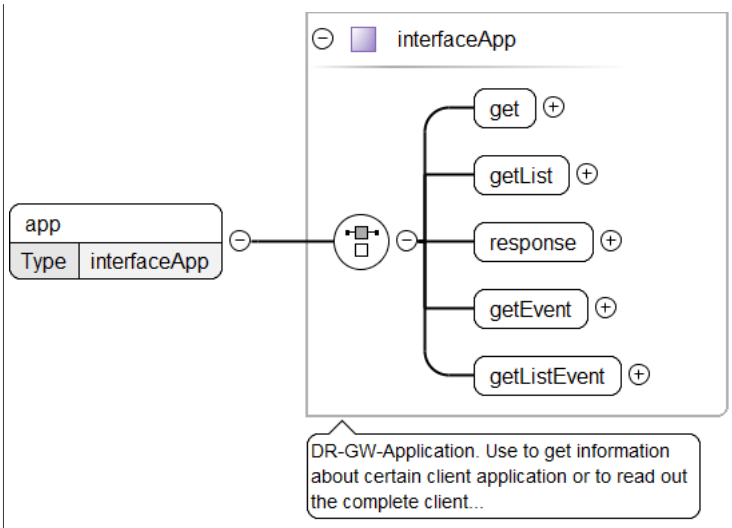
Namespace	DR-GW
Diagram	
Type	typeRadio
Properties	content: complex
Model	issi , alias{0,1} , orgblockId{0,1} , opta{0,1}
Children	alias, issi, opta, orgblockId
Instance	<radio xmlns="DR-GW"> <issi>{1,1}</issi> <alias>{0,1}</alias> <orgblockId>{0,1}</orgblockId> <opta>{0,1}</opta> </radio>
Source	<xs:element name="radio" type="typeRadio"/>

Element typeRadioEvent / delete

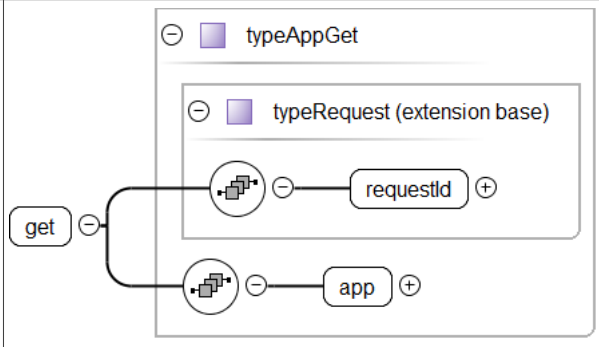
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple
Source	<xs:element name="delete" type="xs:boolean"/>

Element drgw / app

Namespace	DR-GW

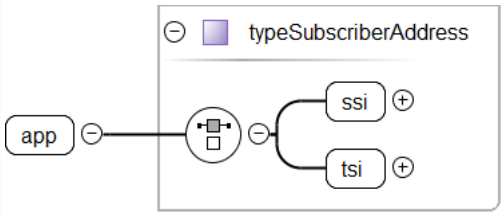
Diagram	
Type	interfaceApp
Properties	content: complex
Model	get | getList | response | getEvent | getListEvent
Children	get, getEvent, getList, getListEvent, response
Instance	`<app xmlns="DR-GW"> <get>{1,1}</get> <getList>{1,1}</getList> <response>{1,1}</response> <getEvent>{1,1}</getEvent> <getListEvent>{1,1}</getListEvent> </app>`
Source	`<xs:element name="app" type="interfaceApp"/>`

Element interfaceApp / get

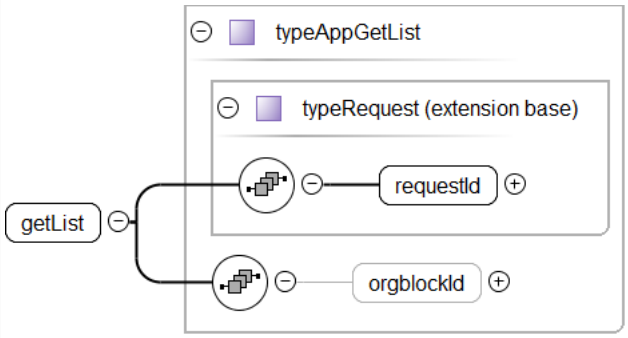
Namespace	DR-GW
Diagram	
Type	typeAppGet
Type hierar-chy	• typeRequest • typeAppGet
Properties	content: complex
Model	requestId , app
Children	app, requestId
Instance	`<get xmlns="DR-GW"> <requestId>{1,1}</requestId> <app>{1,1}</app> </get>`
Source	`<xs:element name="get" type="typeAppGet"/>`

Element typeAppGet / app

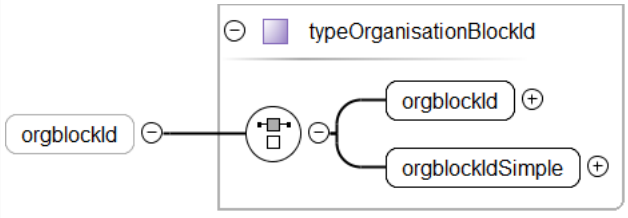
Namespace	DR-GW

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<app xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </app>
Source	<xs:element name="app" type="typeSubscriberAddress"/>

Element interfaceApp / getList

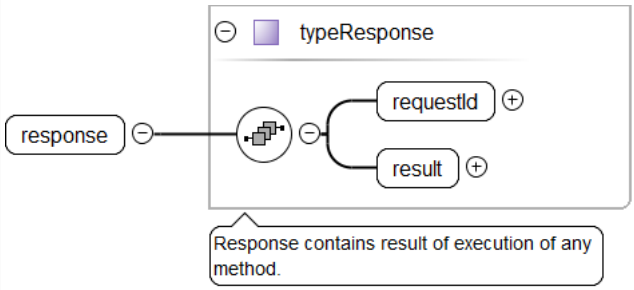
Namespace	DR-GW
Diagram	
Type	typeAppGetList
Type hierarchy	• typeRequest • typeAppGetList
Properties	content: complex
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId
Instance	<getList xmlns="DR-GW"> <requestId>{1,1}</requestId> <orgblockId>{0,1}</orgblockId> </getList>
Source	<xs:element name="getList" type="typeAppGetList"/>

Element typeAppGetList / orgblockId

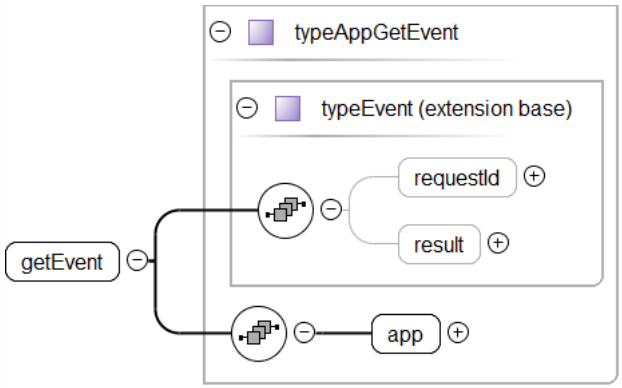
Namespace	DR-GW
Diagram	
Type	typeOrganisationBlockId
Properties	content: complex minOccurs: 0

Model	orgblockId | orgblockIdSimple
Children	orgblockId, orgblockIdSimple
Instance	<orgblockId xmlns="DR-GW"> <orgblockId>{1,1}</orgblockId> <orgblockIdSimple>{1,1}</orgblockIdSimple> </orgblockId>
Source	<xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/>

Element `interfaceApp / response`

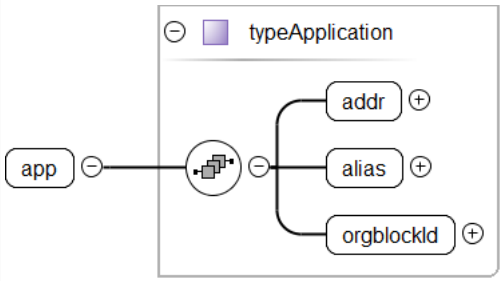
Namespace	DR-GW
Diagram	
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response>
Source	<xs:element name="response" type="typeResponse"/>

Element `interfaceApp / getEvent`

Namespace	DR-GW
Diagram	
Type	typeAppGetEvent
Type hierarchy	• typeEvent • typeAppGetEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , app
Children	app, requestId, result
Instance	<getEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <app>{1,1}</app> </getEvent>

Source	`<xs:element name="getEvent" type="typeAppGetEvent"/>`

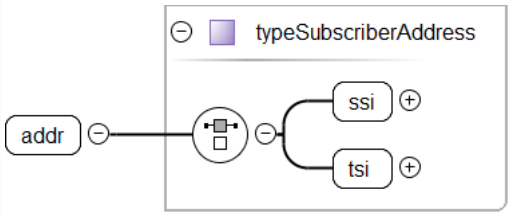
Element `typeAppGetEvent / app`

Namespace	DR-GW
Diagram	
Type	typeApplication
Properties	content: `complex` maxOccurs: `1`
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId

Instance	`<app xmlns="DR-GW">`
	`  <addr>{1,1}</addr>`
	`  <alias>{1,1}</alias>`
	`  <orgblockId>{1,1}</orgblockId>`
	`</app>`

Source	`<xs:element name="app" type="typeApplication" maxOccurs="1"/>`

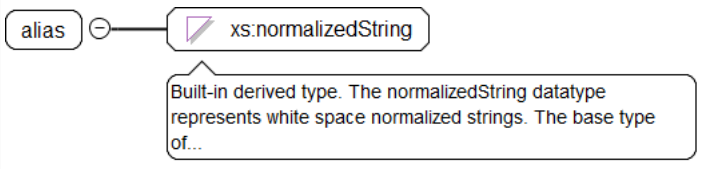
Element `typeApplication / addr`

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: `complex`
Model	ssi | tsi
Children	ssi, tsi

Instance	`<addr xmlns="DR-GW">`
	`  <ssi>{1,1}</ssi>`
	`  <tsi>{1,1}</tsi>`
	`</addr>`

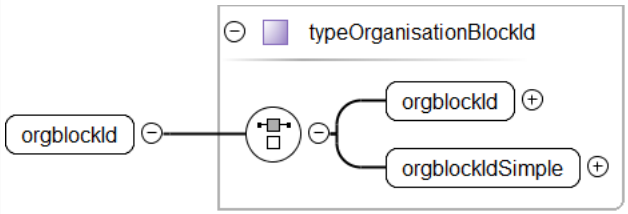
Source	`<xs:element name="addr" type="typeSubscriberAddress"/>`

Element `typeApplication / alias`

Namespace	DR-GW
Diagram	
Type	xs:normalizedString
Properties	content: `simple`

Source	`<xs:element name="alias" type="xs:normalizedString"/>`

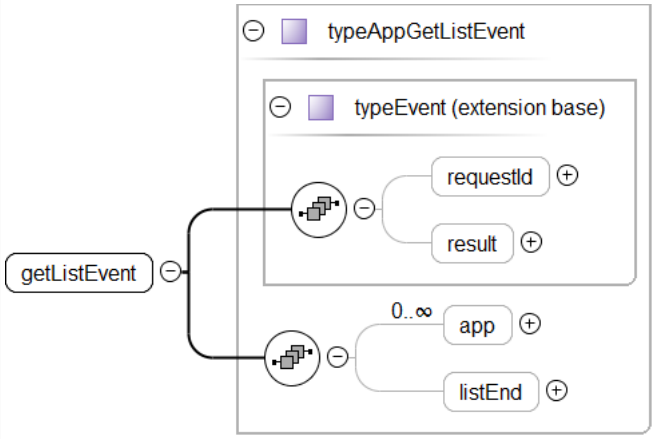
Element `typeApplication / orgblockId`

Namespace	DR-GW
Diagram	
Type	typeOrganisationBlockId
Properties	content: `complex`
Model	orgblockId | orgblockIdSimple
Children	orgblockId, orgblockIdSimple

Instance	`<orgblockId xmlns="DR-GW">`
	`  <orgblockId>{1,1}</orgblockId>`
	`  <orgblockIdSimple>{1,1}</orgblockIdSimple>`
	`</orgblockId>`

Source	`<xs:element name="orgblockId" type="typeOrganisationBlockId"/>`

Element `interfaceApp / getListEvent`

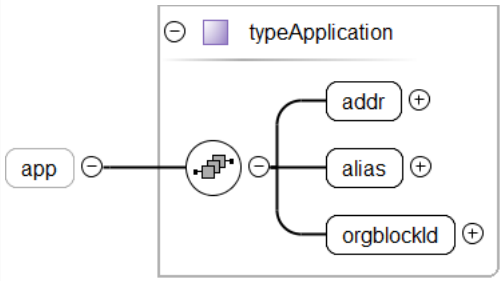
Namespace	DR-GW
Diagram	
Type	typeAppGetListEvent
Type hierarchy	• typeEvent • typeAppGetListEvent
Properties	content: `complex`
Model	requestId{0,1} , result{0,1} , app* , listEnd{0,1}
Children	app, listEnd, requestId, result

Instance	`<getListEvent xmlns="DR-GW">`
	`  <requestId>{0,1}</requestId>`
	`  <result>{0,1}</result>`
	`  <app>{0,unbounded}</app>`
	`  <listEnd>{0,1}</listEnd>`
	`</getListEvent>`

Source	`<xs:element name="getListEvent" type="typeAppGetListEvent"/>`

Element `typeAppGetListEvent / app`

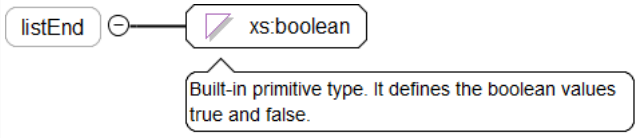
Namespace	DR-GW

Diagram	
Type	typeApplication
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId

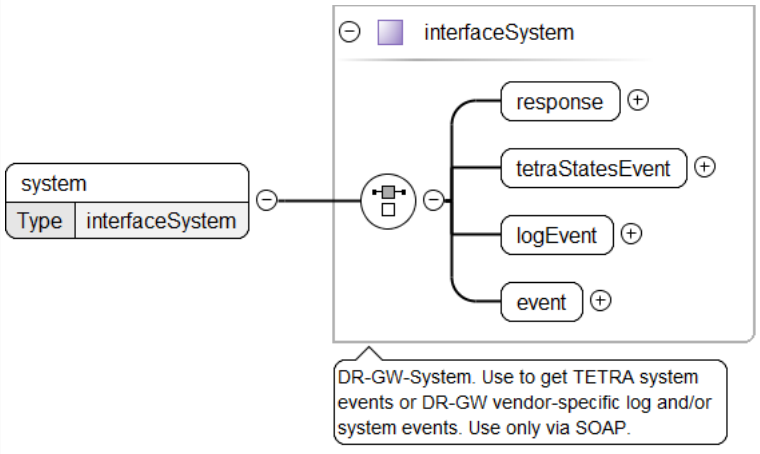
Instance	`<app xmlns="DR-GW">`
	`  <addr>{1,1}</addr>`
	`  <alias>{1,1}</alias>`
	`  <orgblockId>{1,1}</orgblockId>`
	`</app>`

Source	`<xs:element name="app" type="typeApplication" minOccurs="0" maxOccurs="unbounded"/>`

Element typeAppGetListEvent / listEnd

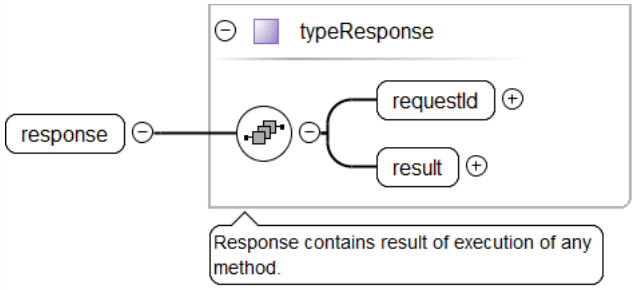
Namespace	DR-GW
Diagram	
Type	xs:boolean
Properties	content: simple minOccurs: 0
Source	`<xs:element name="listEnd" type="xs:boolean" minOccurs="0"/>`

Element drgw / system

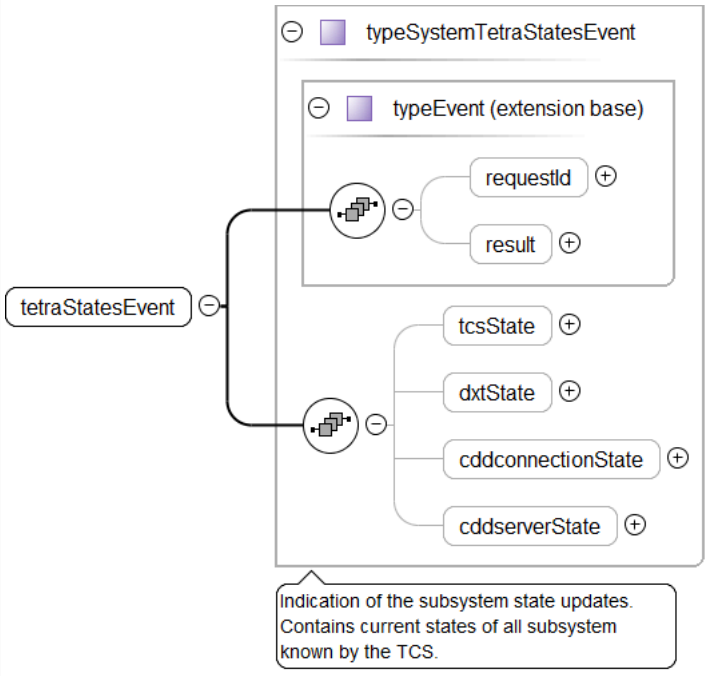
Namespace	DR-GW
Diagram	
Type	interfaceSystem
Properties	content: complex
Model	response | tetraStatesEvent | logEvent | event
Children	event, logEvent, response, tetraStatesEvent

Instance	<system xmlns="DR-GW"> <response>{1,1}</response> <tetraStatesEvent>{1,1}</tetraStatesEvent> <logEvent>{1,1}</logEvent> <event>{1,1}</event> </system>
Source	<xs:element name="system" type="interfaceSystem"/>

Element interfaceSystem / response

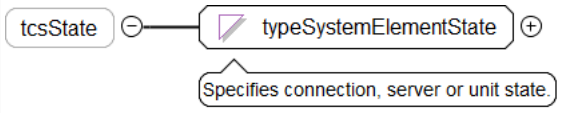
Namespace	DR-GW
Diagram	
Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response>
Source	<xs:element name="response" type="typeResponse"/>

Element interfaceSystem / tetraStatesEvent

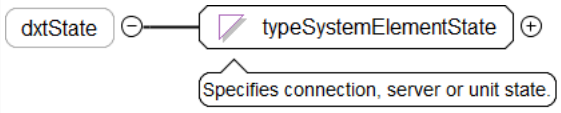
Namespace	DR-GW
Diagram	
Type	typeSystemTetraStatesEvent
Type hierarchy	• typeEvent • typeSystemTetraStatesEvent
Properties	content: complex

Model	requestId{0,1} , result{0,1} , tcsState{0,1} , dxtState{0,1} , cddconnectionState{0,1} , cddserverState{0,1}
Children	cddconnectionState, cddserverState, dxtState, requestId, result, tcsState
Instance	<tetraStatesEvent xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <tcsState>{0,1}</tcsState> <dxtState>{0,1}</dxtState> <cddconnectionState>{0,1}</cddconnectionState> <cddserverState>{0,1}</cddserverState> </tetraStatesEvent>
Source	<xs:element name="tetraStatesEvent" type="typeSystemTetraStatesEvent"/>

Element typeSystemTetraStatesEvent / tcsState

Namespace	DR-GW
Diagram	
Type	typeSystemElementState
Properties	content: simple minOccurs: 0
Facets	enumeration unknown — Unknown state. enumeration ok — Connection or server is working. enumeration n_Ok — Connection or server is not working.
Source	<xs:element name="tcsState" type="typeSystemElementState" minOccurs="0"/>

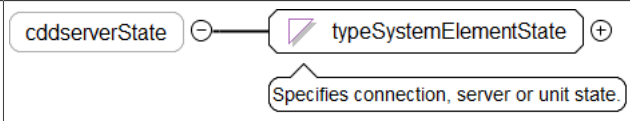
Element typeSystemTetraStatesEvent / dxtState

Namespace	DR-GW
Diagram	
Type	typeSystemElementState
Properties	content: simple minOccurs: 0
Facets	enumeration unknown — Unknown state. enumeration ok — Connection or server is working. enumeration n_Ok — Connection or server is not working.
Source	<xs:element name="dxtState" type="typeSystemElementState" minOccurs="0"/>

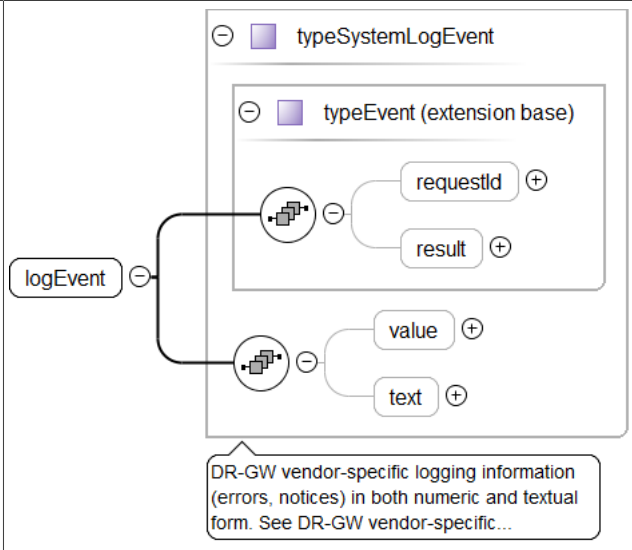
Element typeSystemTetraStatesEvent / cddconnectionState

Namespace	DR-GW
Diagram	
Type	typeSystemElementState
Properties	content: simple minOccurs: 0
Facets	enumeration unknown — Unknown state. enumeration ok — Connection or server is working. enumeration n_Ok — Connection or server is not working.
Source	<xs:element name="cddconnectionState" type="typeSystemElementState" minOccurs="0"/>

Element typeSystemTetraStatesEvent / cddserverState

Namespace	DR-GW
Diagram	
Type	typeSystemElementState
Properties	content: simple minOccurs: 0
Facets	enumeration unknown Unknown state. enumeration ok Connection or server is working. enumeration n_Ok Connection or server is not working.
Source	`<xs:element name="cddserverState" type="typeSystemElementState" minOccurs="0"/>`

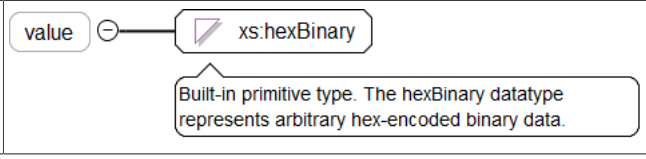
Element interfaceSystem / logEvent

Namespace	DR-GW
Diagram	
Type	typeSystemLogEvent
Type hierarchy	• typeEvent • typeSystemLogEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , value{0,1} , text{0,1}
Children	requestId, result, text, value

Instance	`<logEvent xmlns="DR-GW">`
	`  <requestId>{0,1}</requestId>`
	`  <result>{0,1}</result>`
	`  <value>{0,1}</value>`
	`  <text>{0,1}</text>`
	`</logEvent>`

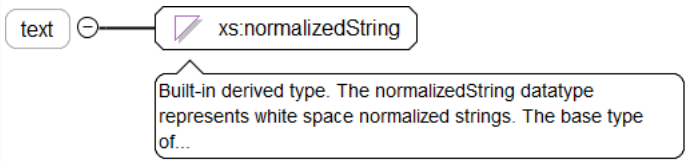
Source	`<xs:element name="logEvent" type="typeSystemLogEvent"/>`

Element typeSystemLogEvent / value

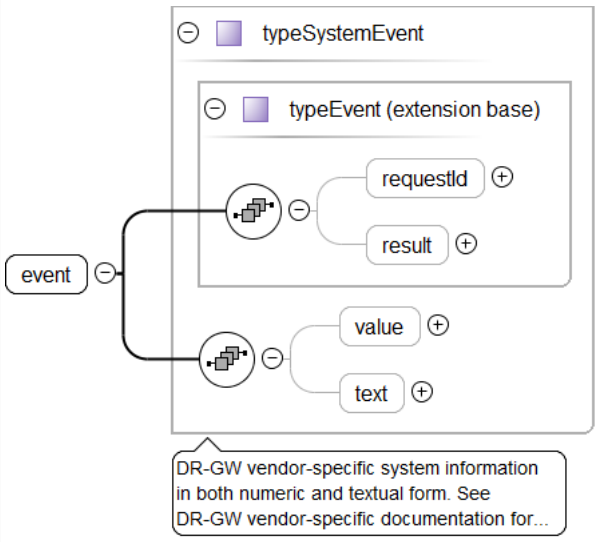
Namespace	DR-GW
Diagram	

Type	xs:hexBinary
Properties	content: simple minOccurs: 0
Source	`<xs:element name="value" type="xs:hexBinary" minOccurs="0"/>`

Element typeSystemLogEvent / text

Namespace	DR-GW
Diagram	
Type	xs:normalizedString
Properties	content: simple minOccurs: 0
Source	`<xs:element name="text" type="xs:normalizedString" minOccurs="0"/>`

Element interfaceSystem / event

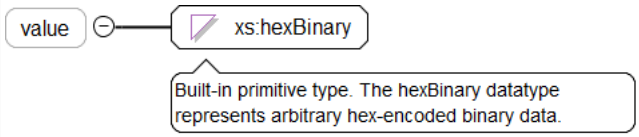
Namespace	DR-GW
Diagram	
Type	typeSystemEvent
Type hierarchy	• typeEvent • typeSystemEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , value{0,1} , text{0,1}
Children	requestId, result, text, value

Instance	`<event xmlns="DR-GW">`
	`  <requestId>{0,1}</requestId>`
	`  <result>{0,1}</result>`
	`  <value>{0,1}</value>`
	`  <text>{0,1}</text>`
	`</event>`

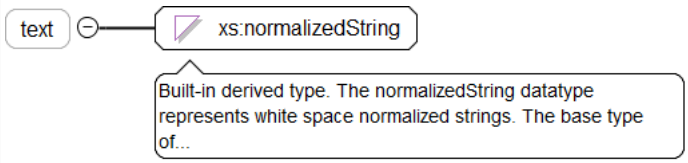
Source	`<xs:element name="event" type="typeSystemEvent"/>`

Element typeSystemEvent / value

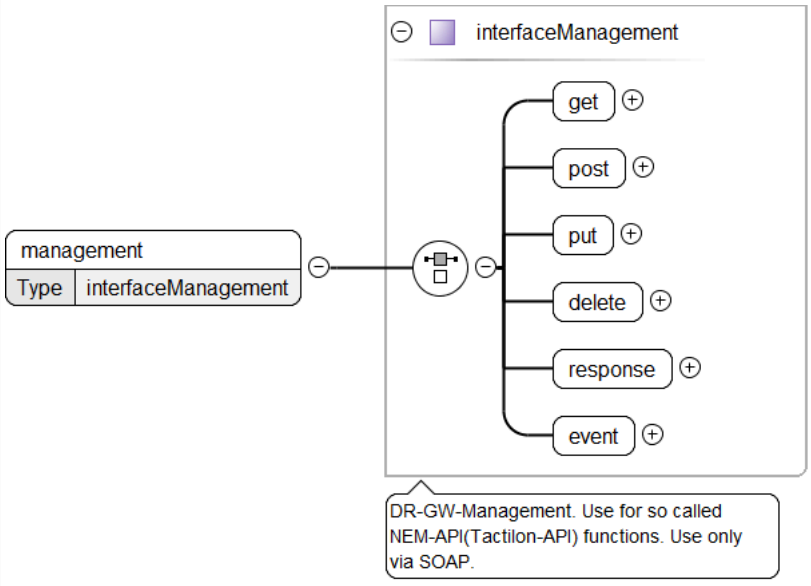
Namespace	DR-GW

Diagram	
Type	xs:hexBinary
Properties	content: simple minOccurs: 0
Source	`<xs:element name="value" type="xs:hexBinary" minOccurs="0"/>`

Element typeSystemEvent / text

Namespace	DR-GW
Diagram	
Type	xs:normalizedString
Properties	content: simple minOccurs: 0
Source	`<xs:element name="text" type="xs:normalizedString" minOccurs="0"/>`

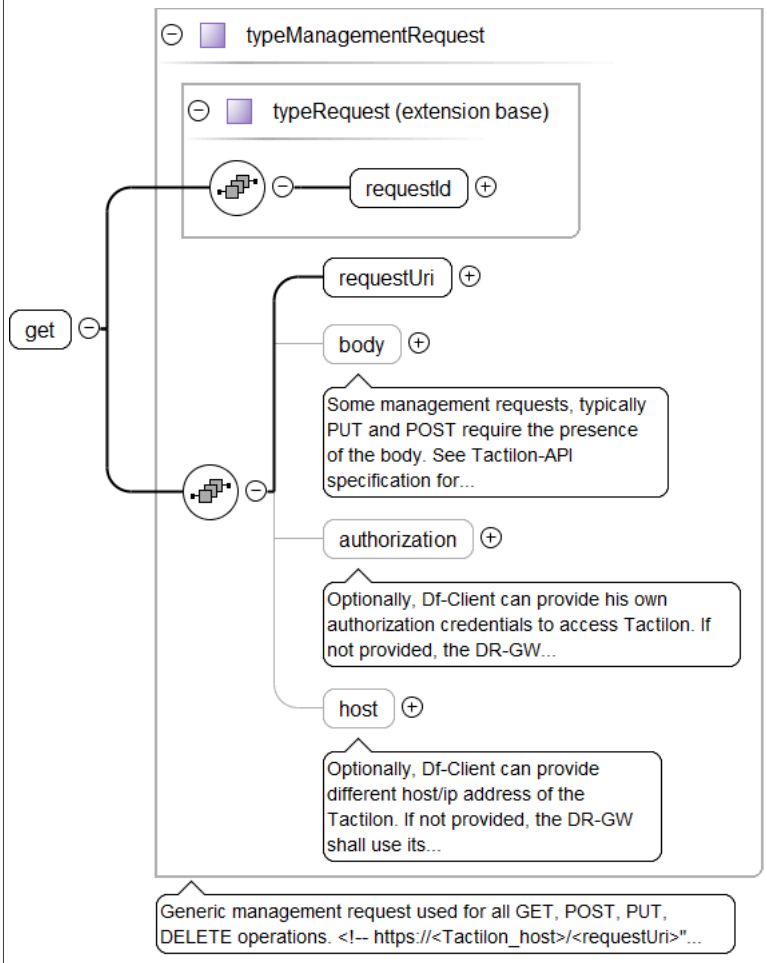
Element drgw / management

Namespace	DR-GW
Diagram	
Type	interfaceManagement
Properties	content: complex
Model	get | post | put | delete | response | event
Children	delete, event, get, post, put, response

Instance	`<management xmlns="DR-GW">`
	`<get>{1,1}</get>`
	`<post>{1,1}</post>`
	`<put>{1,1}</put>`
	`<delete>{1,1}</delete>`
	`<response>{1,1}</response>`
	`<event>{1,1}</event>`
	`</management>`

Source	`<xs:element name="management" type="interfaceManagement"/>`

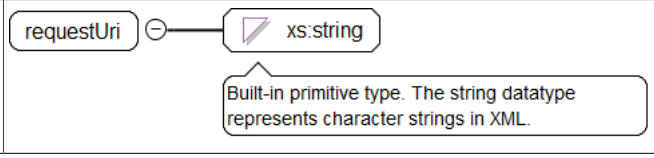
Element `interfaceManagement / get`

Namespace	DR-GW
Diagram	
Type	typeManagementRequest
Type hierar-chy	• typeRequest • typeManagementRequest
Properties	content: complex
Model	requestId , requestUri , body{0,1} , authorization{0,1} , host{0,1}
Children	authorization, body, host, requestId, requestUri

Instance	`<get xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <requestUri>{1,1}</requestUri>`
	`  <body>{0,1}</body>`
	`  <authorization>{0,1}</authorization>`
	`  <host>{0,1}</host>`
	`</get>`

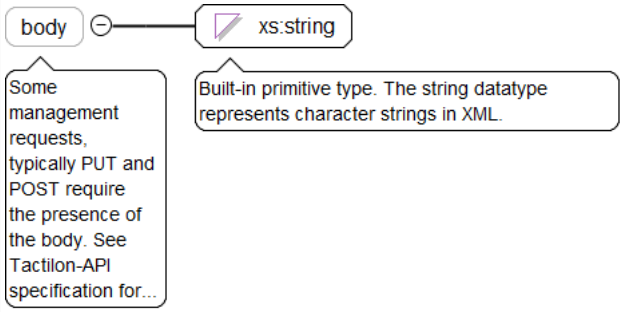
Source	`<xs:element name="get" type="typeManagementRequest"/>`

Element `typeManagementRequest / requestUri`

Namespace	DR-GW
Diagram	

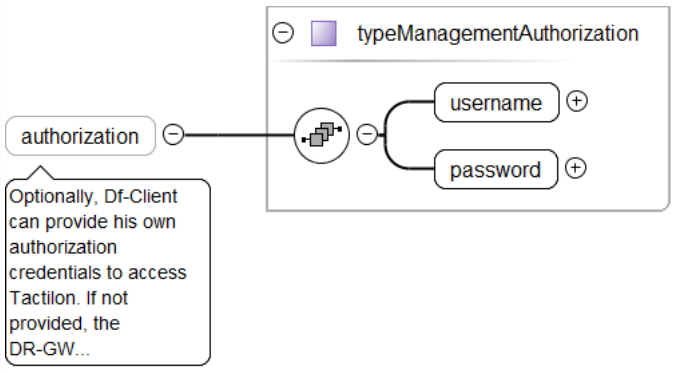
Type	xs:string
Properties	content: simple
Source	`<xs:element name="requestUri" type="xs:string"/>`

Element typeManagementRequest / body

Namespace	DR-GW
Annotations	Some management requests, typically PUT and POST require the presence of the body. See Tactilon-API specification for details on how to construct the body.
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="body" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation>Some management requests, typically PUT and POST require the presence of the body. See Tactilon-API specification for details on how to construct the body.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element typeManagementRequest / authorization

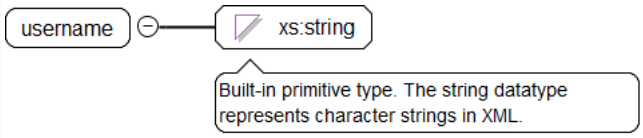
Namespace	DR-GW
Annotations	Optionally, Df-Client can provide his own authorization credentials to access Tactilon. If not provided, the DR-GW shall use its own, globally configured.
Diagram	
Type	typeManagementAuthorization
Properties	content: complex minOccurs: 0
Model	username , password
Children	password, username

Instance	`<authorization xmlns="DR-GW">`
	`  <username>{1,1}</username>`
	`  <password>{1,1}</password>`
	`</authorization>`

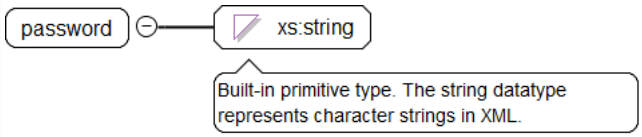
Source	`<xs:element name="authorization" type="typeManagementAuthorization" minOccurs="0">`

```
  <xs:annotation>
    <xs:documentation>Optionally, Df-Client can provide his own authorization credentials to access
 Tactilon. If not provided, the DR-GW shall use its own, globally configured.</xs:documentation>
  </xs:annotation>
</xs:element>
```

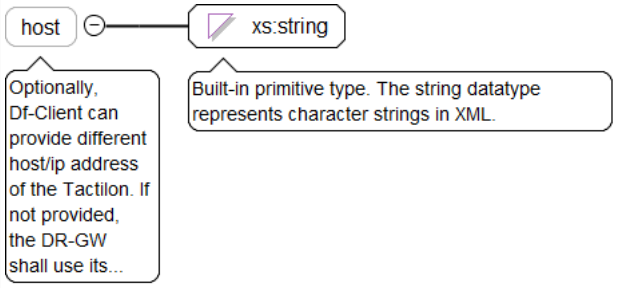
Element typeManagementAuthorization / username

Namespace	DR-GW
Diagram	
Type	xs:string
Properties	content: simple
Source	`<xs:element name="username" type="xs:string"/>`

Element typeManagementAuthorization / password

Namespace	DR-GW
Diagram	
Type	xs:string
Properties	content: simple
Source	`<xs:element name="password" type="xs:string"/>`

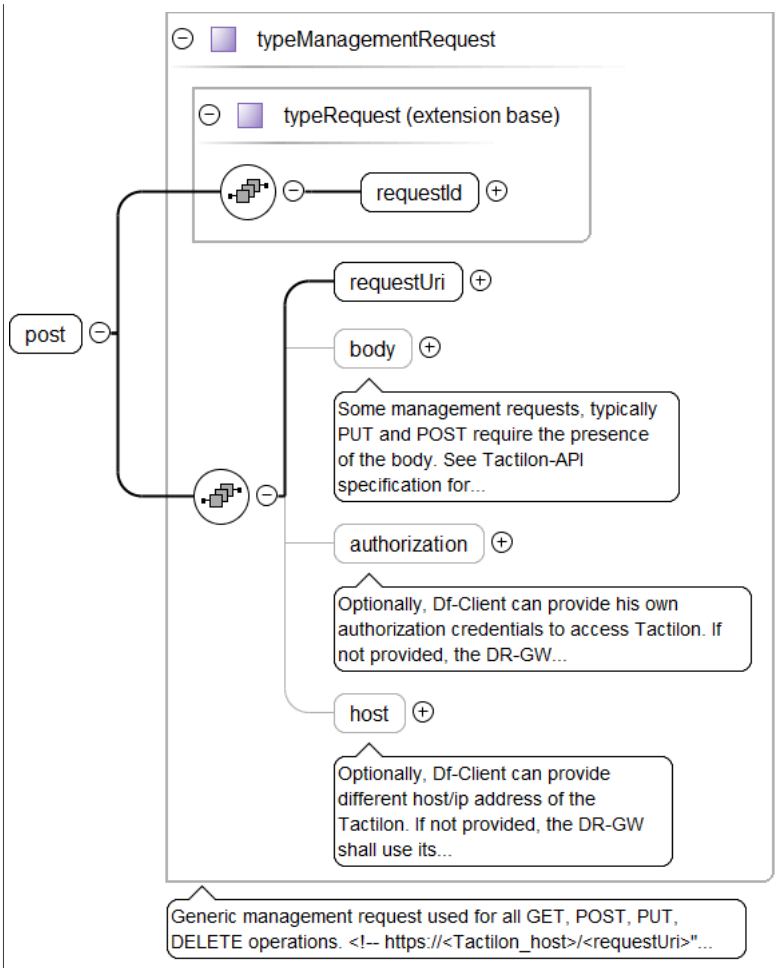
Element typeManagementRequest / host

Namespace	DR-GW
Annotations	Optionally, Df-Client can provide different host/ip address of the Tactilon. If not provided, the DR-GW shall use its own, globally configured.
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="host" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation>Optionally, Df-Client can provide different host/ip address of the Tactilon. If not provided, the DR-GW shall use its own, globally configured.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element interfaceManagement / post

Namespace	DR-GW

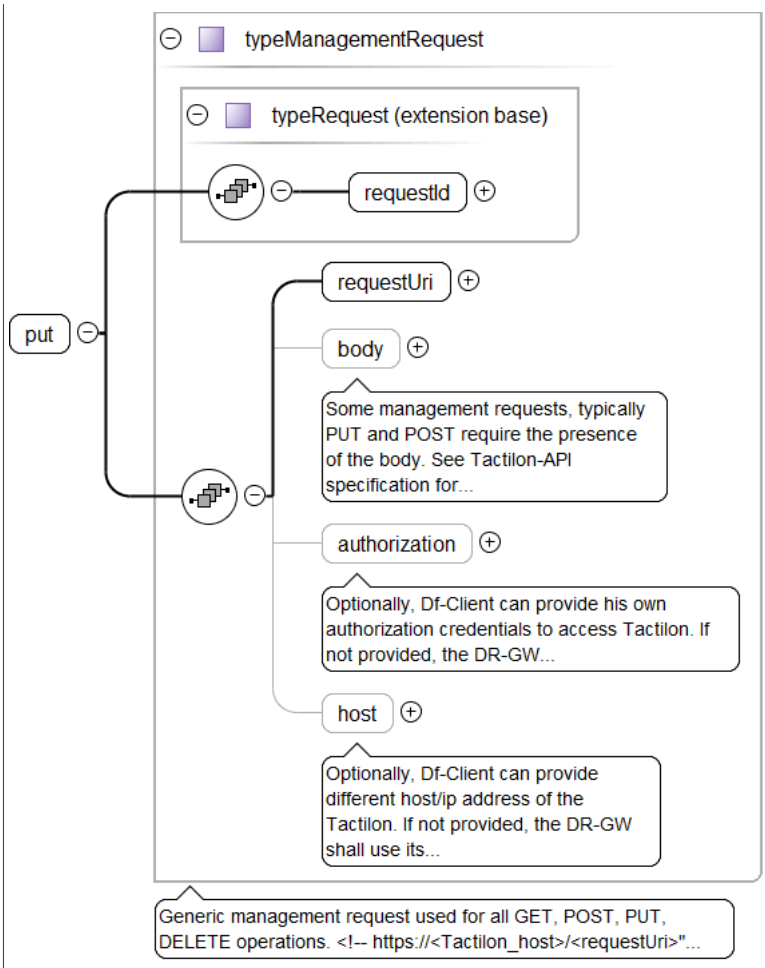
Diagram	
Type	typeManagementRequest
Type hierar-chy	• typeRequest • typeManagementRequest
Properties	content: complex
Model	requestId , requestUri , body{0,1} , authorization{0,1} , host{0,1}
Children	authorization, body, host, requestId, requestUri

Instance	`<post xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <requestUri>{1,1}</requestUri>`
	`  <body>{0,1}</body>`
	`  <authorization>{0,1}</authorization>`
	`  <host>{0,1}</host>`
	`</post>`

Source	`<xs:element name="post" type="typeManagementRequest"/>`

Element interfaceManagement / put

Namespace	DR-GW

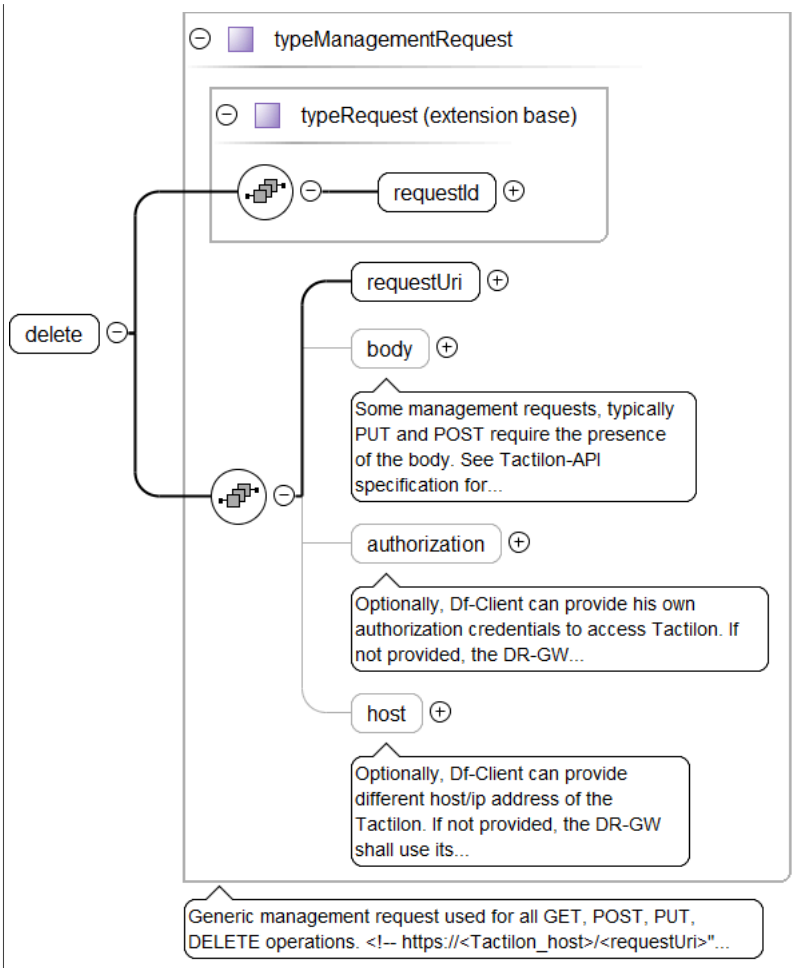
Diagram	
Type	typeManagementRequest
Type hierar-chy	• typeRequest • typeManagementRequest
Properties	content: complex
Model	requestId , requestUri , body{0,1} , authorization{0,1} , host{0,1}
Children	authorization, body, host, requestId, requestUri

Instance	`<put xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <requestUri>{1,1}</requestUri>`
	`  <body>{0,1}</body>`
	`  <authorization>{0,1}</authorization>`
	`  <host>{0,1}</host>`
	`</put>`

Source	`<xs:element name="put" type="typeManagementRequest"/>`

Element interfaceManagement / delete

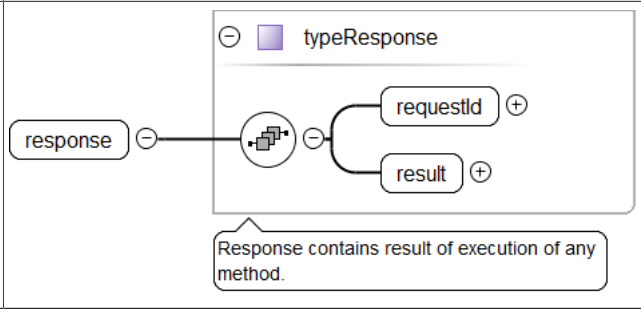
Namespace	DR-GW

Diagram	
Type	typeManagementRequest
Type hierar-chy	• typeRequest • typeManagementRequest
Properties	content: complex
Model	requestId , requestUri , body{0,1} , authorization{0,1} , host{0,1}
Children	authorization, body, host, requestId, requestUri

Instance	`<delete xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <requestUri>{1,1}</requestUri>`
	`  <body>{0,1}</body>`
	`  <authorization>{0,1}</authorization>`
	`  <host>{0,1}</host>`
	`</delete>`

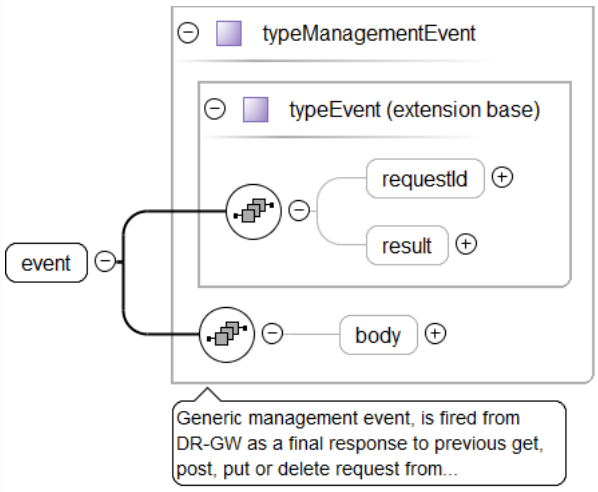
Source	`<xs:element name="delete" type="typeManagementRequest"/>`

Element `interfaceManagement / response`

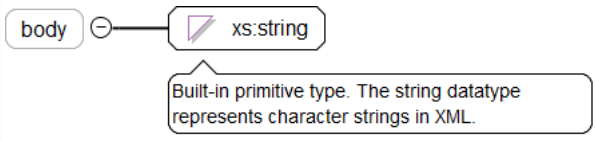
Namespace	DR-GW
Diagram	

Type	typeResponse
Properties	content: complex
Model	requestId , result
Children	requestId, result
Instance	<response xmlns="DR-GW"> <requestId>{1,1}</requestId> <result>{1,1}</result> </response>
Source	<xs:element name="response" type="typeResponse"/>

Element interfaceManagement / event

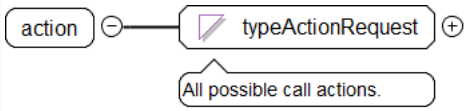
Namespace	DR-GW
Diagram	
Type	typeManagementEvent
Type hierarchy	• typeEvent • typeManagementEvent
Properties	content: complex
Model	requestId{0,1} , result{0,1} , body{0,1}
Children	body, requestId, result
Instance	<event xmlns="DR-GW"> <requestId>{0,1}</requestId> <result>{0,1}</result> <body>{0,1}</body> </event>
Source	<xs:element name="event" type="typeManagementEvent"/>

Element typeManagementEvent / body

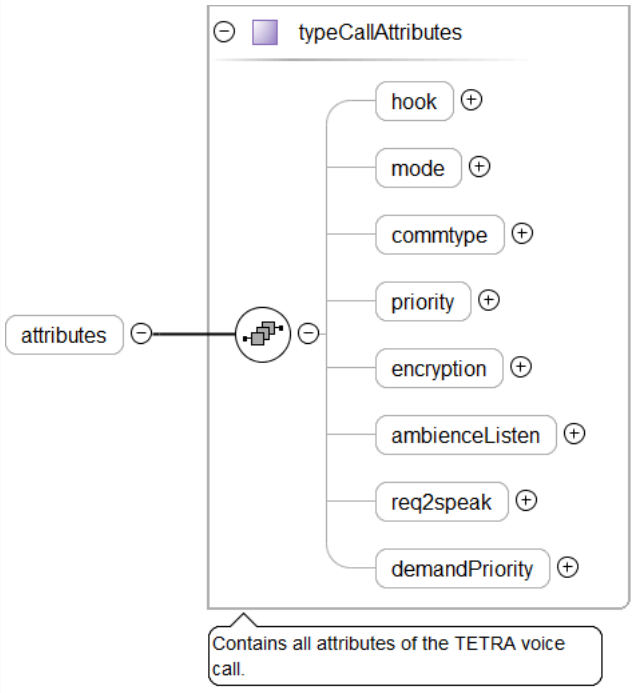
Namespace	DR-GW
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0
Source	<xs:element name="body" type="xs:string" minOccurs="0"/>

Element typeCallRequest / action

Namespace	DR-GW

Diagram			
Type	typeActionRequest		
Properties	content:	simple	
Facets	enumeration	setup	This method is used to initiate a new call setup. For a call setup to be successful it is required that the resources have been reserved prior this method call.
	enumeration	connect	This method is used to connect an incoming call.
	enumeration	hold	This method requests to put an individual call to hold.
	enumeration	unhold	This method is a request for resuming an individual call from hold.
	enumeration	disconnect	This method is used to disconnect a call.
	enumeration	transfer	This method is used to transfer an individual call to a new recipient.
	enumeration	releasecall	This method is used to release radio subscriber's individual call.
Source	<xs:element name="action" type="typeActionRequest"/>		

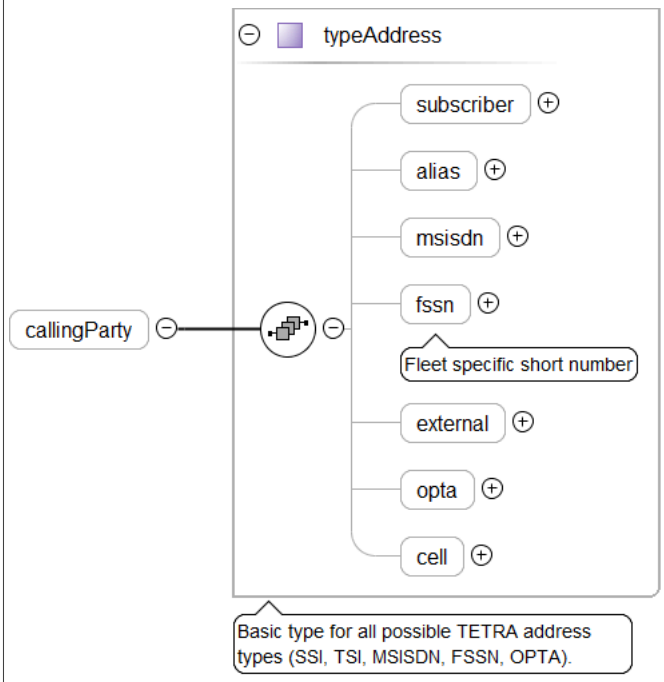
Element typeCallRequest / attributes

Namespace	DR-GW
Diagram	
Type	typeCallAttributes
Properties	content: complex minOccurs: 0
Model	hook{0,1} , mode{0,1} , commtype{0,1} , priority{0,1} , encryption{0,1} , ambienceListen{0,1} , req2speak{0,1} , demandPriority{0,1}
Children	ambienceListen, commtype, demandPriority, encryption, hook, mode, priority, req2speak
Instance	<attributes xmlns="DR-GW"> <hook>{0,1}</hook>

```
  <mode>{0,1}</mode>
  <commtype>{0,1}</commtype>
  <priority>{0,1}</priority>
  <encryption>{0,1}</encryption>
  <ambienceListen>{0,1}</ambienceListen>
  <req2speak>{0,1}</req2speak>
  <demandPriority>{0,1}</demandPriority>
</attributes>
```

Source	`<xs:element name="attributes" type="typeCallAttributes" minOccurs="0"/>`

Element `typeCallRequest / callingParty`

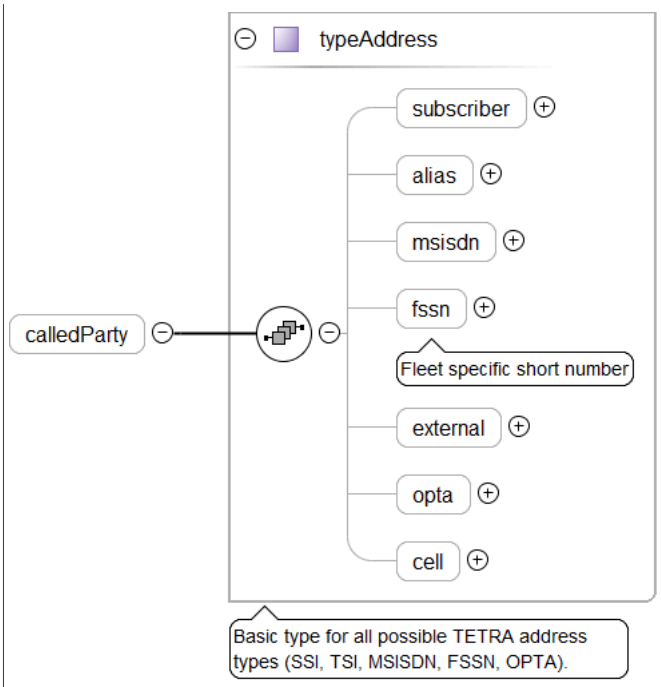
Namespace	DR-GW
Diagram	
Type	typeAddress
Properties	content: `complex` minOccurs: `0`
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber

Instance	`<callingParty xmlns="DR-GW">`
	`  <subscriber>{0,1}</subscriber>`
	`  <alias>{0,1}</alias>`
	`  <msisdn>{0,1}</msisdn>`
	`  <fssn>{0,1}</fssn>`
	`  <external>{0,1}</external>`
	`  <opta>{0,1}</opta>`
	`  <cell>{0,1}</cell>`
	`</callingParty>`

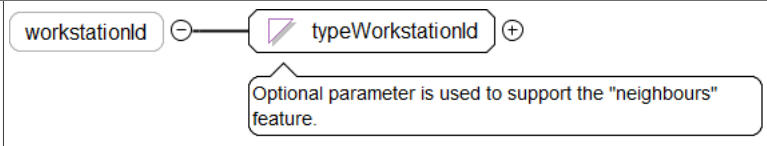
Source	`<xs:element name="callingParty" type="typeAddress" minOccurs="0"/>`

Element `typeCallRequest / calledParty`

Namespace	DR-GW

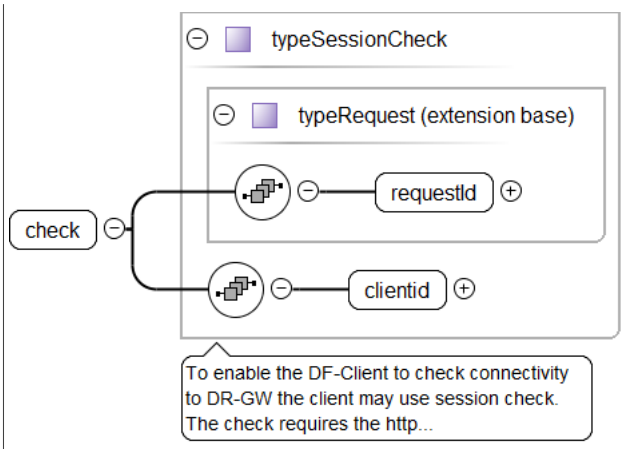
Diagram	
Type	typeAddress
Properties	content: complex minOccurs: 0
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Instance	<calledParty xmlns="DR-GW"> <subscriber>{0,1}</subscriber> <alias>{0,1}</alias> <msisdn>{0,1}</msisdn> <fssn>{0,1}</fssn> <external>{0,1}</external> <opta>{0,1}</opta> <cell>{0,1}</cell> </calledParty>
Source	<xs:element name="calledParty" type="typeAddress" minOccurs="0"/>

Element typeCallRequest / workstationId

Namespace	DR-GW
Diagram	
Type	typeWorkstationId
Properties	content: simple minOccurs: 0
Source	<xs:element name="workstationId" type="typeWorkstationId" minOccurs="0"/>

Element check

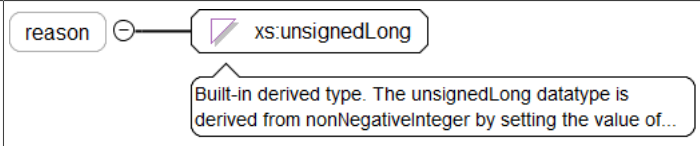
Namespace	DR-GW

Diagram	
Type	typeSessionCheck
Type hierarchy	• typeRequest • typeSessionCheck
Properties	content: complex
Model	requestId , clientid
Children	clientid, requestId

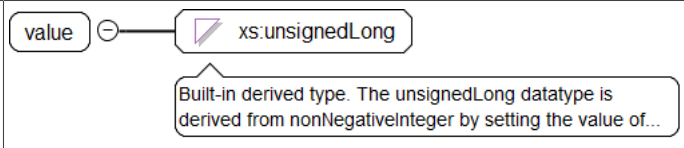
Instance	`<check xmlns="DR-GW">`
	`  <requestId>{1,1}</requestId>`
	`  <clientid>{1,1}</clientid>`
	`</check>`

Source	`<xs:element name="check" type="typeSessionCheck"/>`

Element typeSessionLogoutEvent / reason

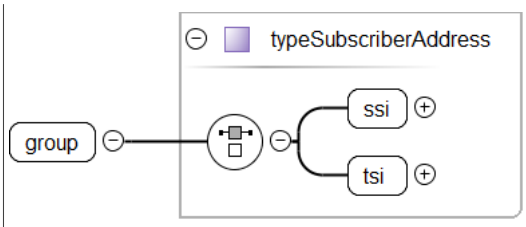
Namespace	DR-GW
Diagram	
Type	xs:unsignedLong
Properties	content: simple minOccurs: 0
Source	`<xs:element name="reason" type="xs:unsignedLong" minOccurs="0"/>`

Element typeSdsValidity / value

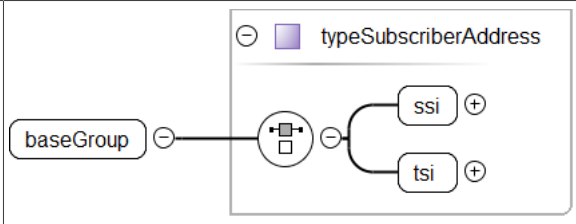
Namespace	DR-GW
Diagram	
Type	xs:unsignedLong
Properties	content: simple
Source	`<xs:element name="value" type="xs:unsignedLong"/>`

Element typeGroupGetCombinationsEvent / group

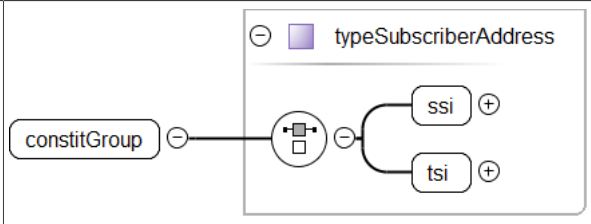
Namespace	DR-GW

Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<group xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </group>
Source	<xs:element name="group" type="typeSubscriberAddress"/>

Element typeGroupGetCombinationsEvent / baseGroup

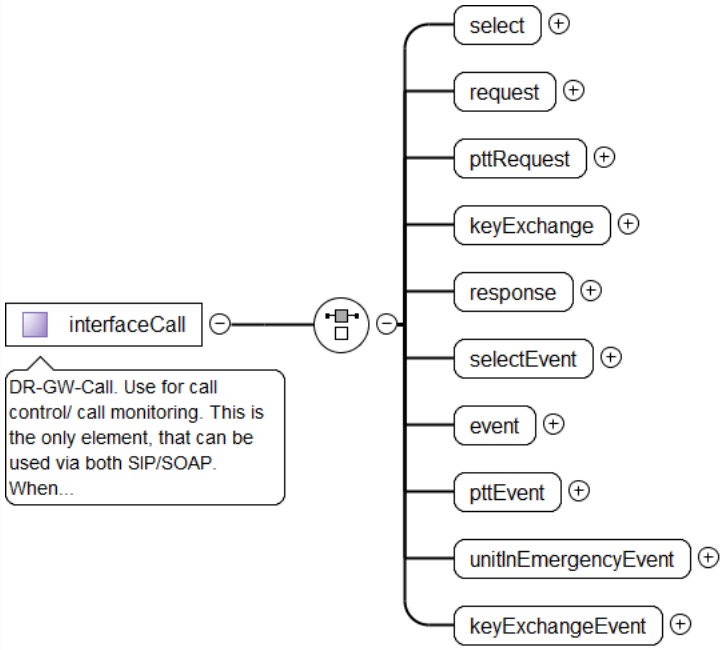
Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex
Model	ssi | tsi
Children	ssi, tsi
Instance	<baseGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </baseGroup>
Source	<xs:element name="baseGroup" type="typeSubscriberAddress"/>

Element typeGroupGetCombinationsEvent / constitGroup

Namespace	DR-GW
Diagram	
Type	typeSubscriberAddress
Properties	content: complex maxOccurs: 7
Model	ssi | tsi
Children	ssi, tsi
Instance	<constitGroup xmlns="DR-GW"> <ssi>{1,1}</ssi> <tsi>{1,1}</tsi> </constitGroup>
Source	<xs:element name="constitGroup" type="typeSubscriberAddress" maxOccurs="7"/>

Complex Type(s)

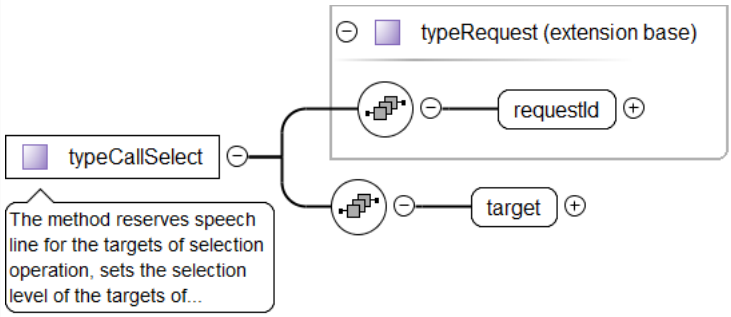
Complex Type `interfaceCall`

Namespace	DR-GW
Annotations	`DR-GW-Call. Use for call control/ call monitoring. This is the only element, that can be used via both SIP/SOAP. When used via SOAP it can only be used for call event monitoring.`
Diagram	
Used by	Element drgw/call
Model	select | request | pttRequest | keyExchange | response | selectEvent | event | pttEvent | unitInEmergencyEvent | keyExchangeEvent
Children	event, keyExchange, keyExchangeEvent, pttEvent, pttRequest, request, response, select, selectEvent, unitInEmergencyEvent

Source	`<xs:complexType name="interfaceCall">`
	`<xs:annotation>`
	`<xs:documentation>DR-GW-Call. Use for call control/ call monitoring. This is the only element, that can be used via both SIP/SOAP. When used via SOAP it can only be used for call event monitoring.</xs:documentation>`
	`</xs:annotation>`
	`<xs:choice>`
	`<xs:element name="select" type="typeCallSelect"/>`
	`<xs:element name="request" type="typeCallEvent"/>`
	`<xs:element name="pttRequest" type="typeCallPTTRequest"/>`
	`<xs:element name="keyExchange" type="typeCallKeyExchange"/>`
	`<xs:element name="response" type="typeResponse"/>`
	`<xs:element name="selectEvent" type="typeCallSelectEvent"/>`
	`<xs:element name="event" type="typeCallEvent"/>`
	`<xs:element name="pttEvent" type="typeCallPTTEvent"/>`
	`<xs:element name="unitInEmergencyEvent" type="typeCallUnitInEmergencyEvent"/>`
	`<xs:element name="keyExchangeEvent" type="typeCallKeyExchangeEvent"/>`
	`</xs:choice>`
	`</xs:complexType>`

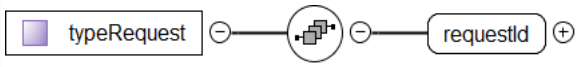
Complex Type `typeCallSelect`

Namespace	DR-GW
Annotations	`The method reserves speech line for the targets of selection operation, sets the selection level of the targets of selection operation to given values.`

Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeCallSelect
Used by	Element interfaceCall/select
Model	requestId , target
Children	requestId, target

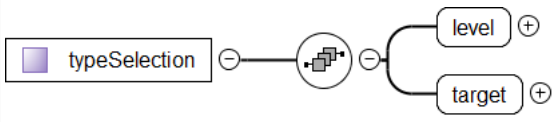
Source	`<xs:complexType name="typeCallSelect">`
	`  <xs:annotation>`
	`  <xs:documentation>The method reserves speech line for the targets of selection operation, sets the selection level of the targets of selection operation to given values.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="target" type="typeSelection"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Complex Type `typeRequest`

Namespace	DR-GW
Diagram	
Used by	Complex Types typeAppGet, typeAppGetList, typeCallKeyExchange, typeCallPTTRequest, typeCallRequest, typeCallSelect, typeGroupAddCombination, typeGroupAddRadioMember, typeGroupGet, typeGroupGetAppMembers, typeGroupGetCombinations, typeGroupGetList, typeGroupGetRadioMembers, typeGroupRemoveCombination, typeGroupRemoveRadioMember, typeGroupSubscribeData, typeGroupTrack, typeManagementRequest, typeOrgGet, typeOrgGetList, typeRadioChangeOpta, typeRadioDisable, typeRadioEnable, typeRadioGet, typeRadioGetGroups, typeRadioGetList, typeRadioTrack, typeSdsSend, typeSdsSendReport, typeSessionCheck, typeSessionLogin, typeSessionLogout, typeSessionSupervise, typeStatusSend
Model	requestId
Children	requestId

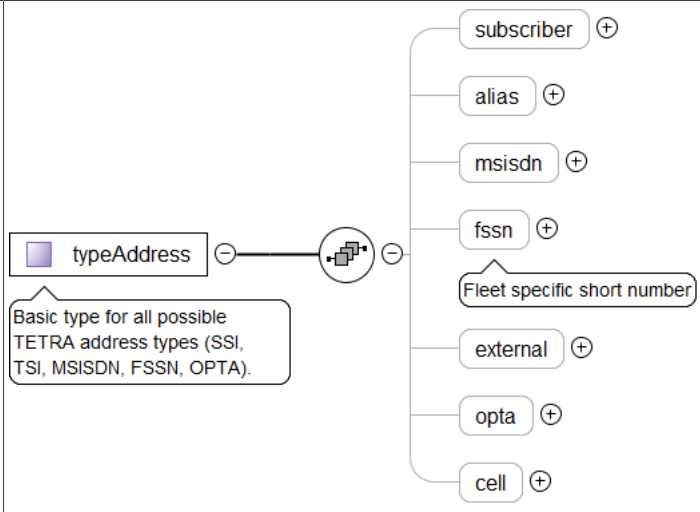
Source	`<xs:complexType name="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="requestId" type="xs:long"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `typeSelection`

Namespace	DR-GW
Annotations	
Diagram	
Used by	Elements typeCallSelect/target, typeCallSelectEvent/target

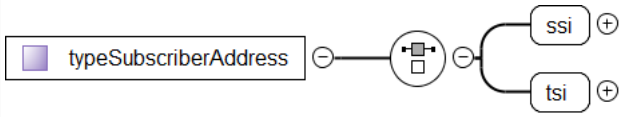
Model	level , target
Children	level, target
Source	<xs:complexType name="typeSelection"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="level" type="typeSelectionLevel"/> <xs:element name="target" type="typeAddress"/> </xs:sequence> </xs:complexType>

Complex Type `typeAddress`

Namespace	DR-GW
Annotations	Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).
Diagram	
Used by	Elements: typeCallEvent/calledParty, typeCallEvent/callingParty, typeCallPTTRequest/talkingParty, typeCallRequest/calledParty, typeCallRequest/callingParty, typeCallUnitInEmergencyEvent/unitInEmg, typeSds/forward, typeSds/source, typeSds/target, typeSdsReportEvent/source, typeSdsReportEvent/target, typeSdsSendReport/target, typeSelection/target, typeStatus/source, typeStatus/target, typeTxGranted/talkingParty
Model	subscriber{0,1} , alias{0,1} , msisdn{0,1} , fssn{0,1} , external{0,1} , opta{0,1} , cell{0,1}
Children	alias, cell, external, fssn, msisdn, opta, subscriber
Source	<xs:complexType name="typeAddress"> <xs:annotation> <xs:documentation>Basic type for all possible TETRA address types (SSI, TSI, MSISDN, FSSN, OPTA).</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="subscriber" type="typeSubscriberAddress" minOccurs="0"/> <xs:element name="alias" type="xs:normalizedString" minOccurs="0"/> <xs:element name="msisdn" type="typeDialString" minOccurs="0"/> <xs:element name="fssn" type="xs:unsignedLong" minOccurs="0"> <xs:annotation> <xs:documentation>Fleet specific short number</xs:documentation> </xs:annotation> </xs:element> <xs:element name="external" type="typeExternal" minOccurs="0"/> <xs:element name="opta" type="typeOpta" minOccurs="0"/> <xs:element name="cell" type="xs:short" minOccurs="0"/> </xs:sequence> </xs:complexType>

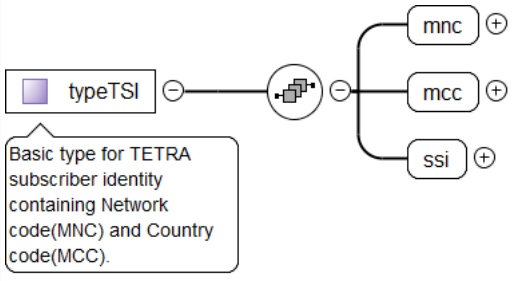
Complex Type `typeSubscriberAddress`

Namespace	DR-GW
Annotations	

Diagram	
Used by	Elements: typeAddress/subscriber, typeAppGet/app, typeApplication/addr, typeCallUnitInEmergencyEvent/group, typeGroup/addr, typeGroupAddCombination/baseGroup, typeGroupAddCombination/group, typeGroupAddCombinationEvent/baseGroup, typeGroupAddCombinationEvent/group, typeGroupAddRadioMember/group, typeGroupAddRadioMember/radio, typeGroupAddRadioMemberEvent/group, typeGroupAddRadioMemberEvent/radio, typeGroupAppMemberEvent/app, typeGroupAppMemberEvent/group, typeGroupCombinationEvent/baseGroup, typeGroupCombinationEvent/constitGroup, typeGroupCombinationEvent/group, typeGroupDataSubscription/addr, typeGroupGet/group, typeGroupGetAppMembers/group, typeGroupGetAppMembersEvent/app, typeGroupGetCombinations/group, typeGroupGetCombinationsEvent/baseGroup, typeGroupGetCombinationsEvent/constitGroup, typeGroupGetCombinationsEvent/group, typeGroupGetRadioMembers/group, typeGroupGetRadioMembersEvent/group, typeGroupGetRadioMembersEvent/radio, typeGroupRadioMemberEvent/group, typeGroupRadioMemberEvent/radio, typeGroupRemoveCombination/baseGroup, typeGroupRemoveCombination/group, typeGroupRemoveCombinationEvent/baseGroup, typeGroupRemoveCombinationEvent/group, typeGroupRemoveRadioMember/group, typeGroupRemoveRadioMember/radio, typeGroupRemoveRadioMemberEvent/group, typeGroupRemoveRadioMemberEvent/radio, typeGroupTrack/group, typeGroupTrackSubscriptionEvent/group, typeRadio/issi, typeRadioChangeOpta/radio, typeRadioChangeOptaEvent/radio, typeRadioDisable/radio, typeRadioEnable/radio, typeRadioEnableDisableEvent/radio, typeRadioGet/radio, typeRadioGetGroupsEvent/radio, typeRadioGroupSelection/group, typeRadioGroupsEvent/deletedGroup, typeRadioGroupsEvent/radio, typeRadioTrack/radio, typeRadioTrackSubscriptionEvent/radio, typeRadioTrackingData/callParty, typeRadioTrackingData/radio, typeSds/e2eegroup
Model	ssi | tsi
Children	ssi, tsi

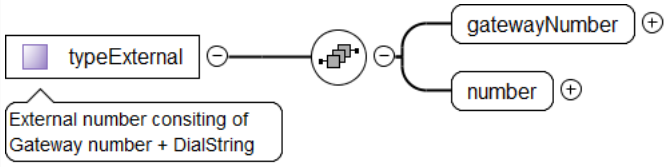
Source	`<xs:complexType name="typeSubscriberAddress">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:choice>`
	`  <xs:element name="ssi" type="xs:unsignedLong"/>`
	`  <xs:element name="tsi" type="typeTSI"/>`
	`  </xs:choice>`
	`</xs:complexType>`

Complex Type `typeTSI`

Namespace	DR-GW
Annotations	Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).
Diagram	
Used by	Element: typeSubscriberAddress/tsi
Model	mnc , mcc , ssi
Children	mcc, mnc, ssi

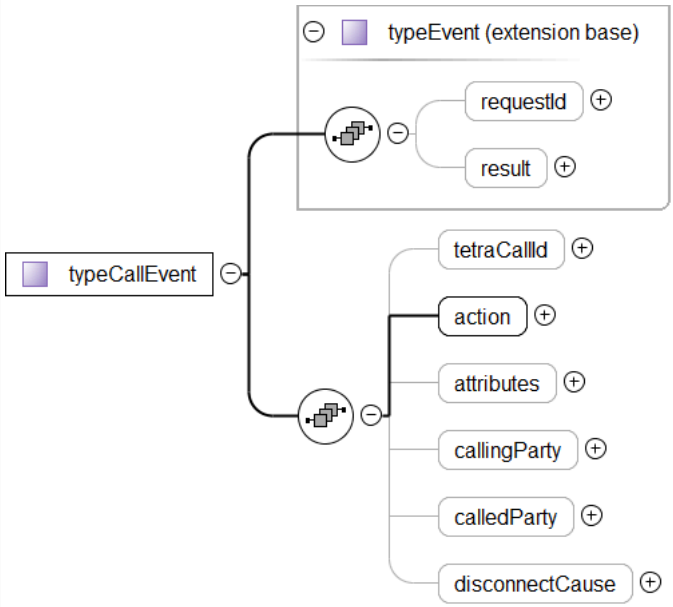
Source	`<xs:complexType name="typeTSI">`
	`  <xs:annotation>`
	`  <xs:documentation>Basic type for TETRA subscriber identity containing Network code(MNC) and Country code(MCC).</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="mnc" type="xs:unsignedShort"/>`
	`  <xs:element name="mcc" type="xs:unsignedShort"/>`
	`  <xs:element name="ssi" type="xs:unsignedLong"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `typeExternal`

Namespace	DR-GW
Annotations	External number consiting of Gateway number + DialString
Diagram	
Used by	Element typeAddress/external
Model	gatewayNumber , number
Children	gatewayNumber, number

Source	`<xs:complexType name="typeExternal">`
	`  <xs:annotation>`
	`  <xs:documentation>External number consiting of Gateway number + DialString</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="gatewayNumber" type="xs:unsignedLong"/>`
	`  <xs:element name="number" type="typeDialString"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

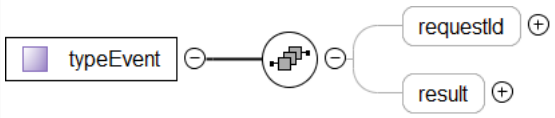
Complex Type `typeCallEvent`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeCallEvent
Used by	Elements interfaceCall/event, interfaceCall/request
Model	requestId{0,1} , result{0,1} , tetraCallId{0,1} , action , attributes{0,1} , callingParty{0,1} , calledParty{0,1} , disconnectCause{0,1}
Children	action, attributes, calledParty, callingParty, disconnectCause, requestId, result, tetraCallId

Source	`<xs:complexType name="typeCallEvent">`
	`  <xs:annotation>`
	`  <xs:documentation>`
	`  </xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`

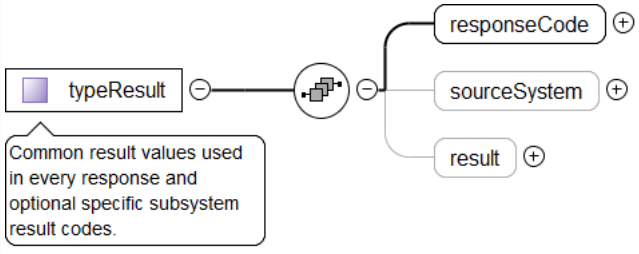
```
      <xs:sequence>
        <xs:element name="tetraCallId" type="typeTetraCallId" minOccurs="0"/>
        <xs:element name="action" type="typeActionEvent"/>
        <xs:element name="attributes" type="typeCallAttributes" minOccurs="0"/>
        <xs:element name="callingParty" type="typeAddress" minOccurs="0"/>
        <xs:element name="calledParty" type="typeAddress" minOccurs="0"/>
        <xs:element name="disconnectCause" type="typeDisconnectCause" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeEvent`

Namespace	DR-GW
Diagram	
Used by	Complex Types: typeAppGetEvent, typeAppGetListEvent, typeCallEvent, typeCallKeyExchangeEvent, typeCallPTTEvent, typeCallSelectEvent, typeCallUnitInEmergencyEvent, typeGroupAddCombinationEvent, typeGroupAddRadioMemberEvent, typeGroupAppMemberEvent, typeGroupCombinationEvent, typeGroupEvent, typeGroupGetAppMembersEvent, typeGroupGetCombinationsEvent, typeGroupGetEvent, typeGroupGetListEvent, typeGroupGetRadioMembersEvent, typeGroupRadioMemberEvent, typeGroupRemoveCombinationEvent, typeGroupRemoveRadioMemberEvent, typeGroupSubscribeDataEvent, typeGroupTrackSubscriptionEvent, typeManagementEvent, typeOrgEvent, typeOrgGetEvent, typeOrgGetListEvent, typeRadioChangeOptaEvent, typeRadioEnableDisableEvent, typeRadioEvent, typeRadioGetEvent, typeRadioGetGroupsEvent, typeRadioGetListEvent, typeRadioGroupsEvent, typeRadioTrackEvent, typeRadioTrackSubscriptionEvent, typeSdsReceiveEvent, typeSdsReportEvent, typeSdsSendEvent, typeSessionLoginEvent, typeSessionLogoutEvent, typeSessionSuperviseEvent, typeStatusReceiveEvent, typeStatusSendEvent, typeSystemEvent, typeSystemLogEvent, typeSystemTetraStatesEvent
Model	requestId{0,1} , result{0,1}
Children	requestId, result

Source	`<xs:complexType name="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="requestId" type="xs:unsignedLong" minOccurs="0"/>`
	`  <xs:element name="result" type="typeResult" minOccurs="0"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

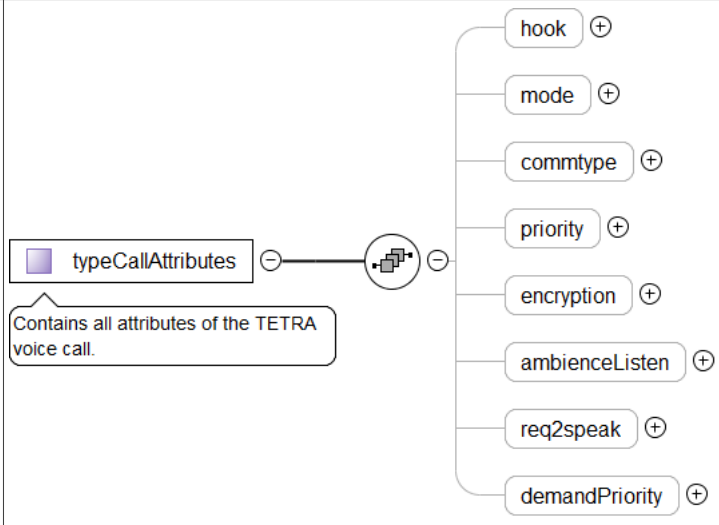
Complex Type `typeResult`

Namespace	DR-GW
Annotations	Common result values used in every response and optional specific subsystem result codes.
Diagram	
Used by	Elements: typeEvent/result, typeResponse/result
Model	responseCode , sourceSystem{0,1} , result{0,1}
Children	responseCode, result, sourceSystem

Source	`<xs:complexType name="typeResult">`
	`  <xs:annotation>`
	`  <xs:documentation>Common result values used in every response and optional specific subsystem result codes.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="responseCode" type="typeResponseCode"/>`
	`  <xs:element name="sourceSystem" type="typeSourceSystem" minOccurs="0"/>`
	`  <xs:element name="result" type="xs:unsignedLong" minOccurs="0"/>`
	`  </xs:sequence>`

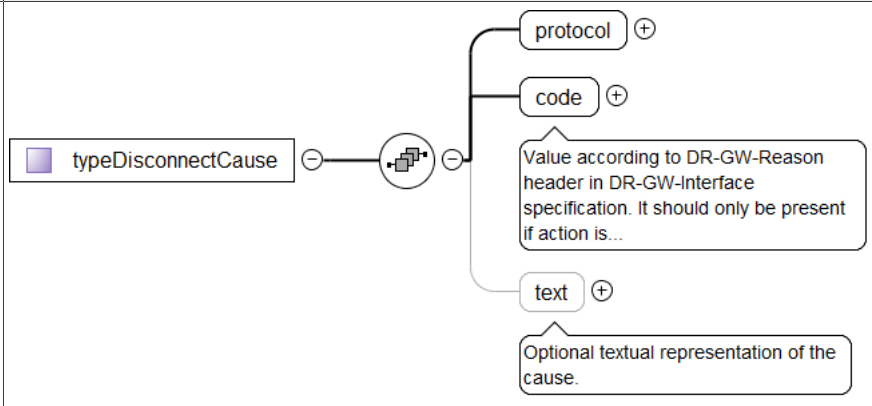
```
</xs:complexType>
```

Complex Type `typeCallAttributes`

Namespace	DR-GW
Annotations	Contains all attributes of the TETRA voice call.
Diagram	
Used by	Elements typeCallEvent/attributes, typeCallPTTRequest/attributes, typeCallRequest/attributes
Model	hook{0,1} , mode{0,1} , commtype{0,1} , priority{0,1} , encryption{0,1} , ambienceListen{0,1} , req2speak{0,1} , demandPriority{0,1}
Children	ambienceListen, commtype, demandPriority, encryption, hook, mode, priority, req2speak

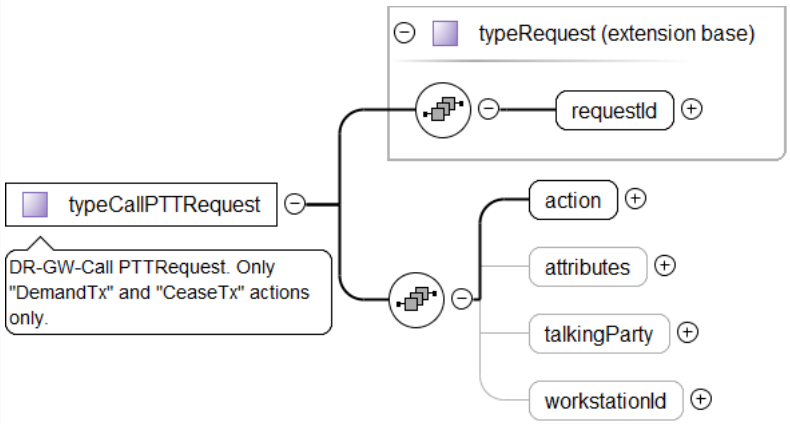
Source	`<xs:complexType name="typeCallAttributes">`
	`<xs:annotation>`
	`<xs:documentation>Contains all attributes of the TETRA voice call.</xs:documentation>`
	`</xs:annotation>`
	`<xs:sequence>`
	`<xs:element name="hook" type="xs:boolean" minOccurs="0"/>`
	`<xs:element name="mode" type="typeCallMode" minOccurs="0"/>`
	`<xs:element name="commtype" type="typeCallType" minOccurs="0"/>`
	`<xs:element name="priority" type="xs:unsignedByte" default="1" minOccurs="0"/>`
	`<xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/>`
	`<xs:element name="ambienceListen" type="xs:boolean" default="0" minOccurs="0"/>`
	`<xs:element name="req2speak" type="xs:boolean" default="1" minOccurs="0"/>`
	`<xs:element name="demandPriority" type="typeTxDemandPriority" default="normal" minOccurs="0"/>`
	`</xs:sequence>`
	`</xs:complexType>`

Complex Type `typeDisconnectCause`

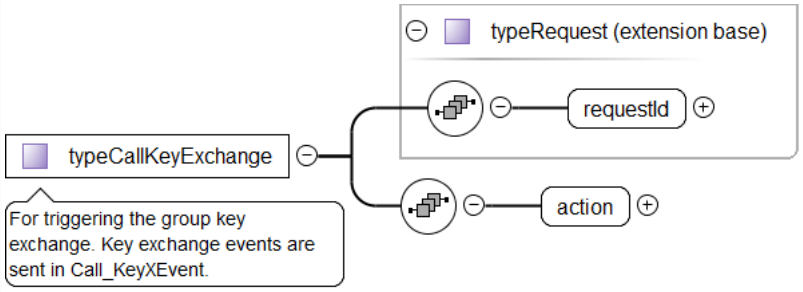
Namespace	DR-GW
Diagram	
Used by	Element typeCallEvent/disconnectCause
Model	protocol , code , text{0,1}
Children	code, protocol, text

Source	<xs:complexType name="typeDisconnectCause"> <xs:sequence> <xs:element name="protocol"> <xs:simpleType> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="DR-GW"/> <xs:enumeration value="TCS-API"/> </xs:restriction> </xs:simpleType> </xs:element> <xs:element name="code" type="xs:unsignedInt"> <xs:annotation> <xs:documentation>Value according to DR-GW-Reason header in DR-GW-Interface specification. It should only be present if action is "disconnected" and holds the reason for call disconnection.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="text" minOccurs="0"> <xs:annotation> <xs:documentation>Optional textual representation of the cause.</xs:documentation> </xs:annotation> <xs:simpleType> <xs:restriction base="xs:normalizedString"> <xs:maxLength value="80"/> </xs:restriction> </xs:simpleType> </xs:element> </xs:sequence> </xs:complexType>

Complex Type `typeCallPTTRequest`

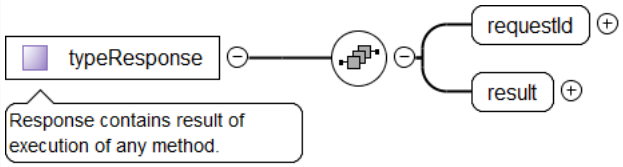
Namespace	DR-GW
Annotations	DR-GW-Call PTTRequest. Only "DemandTx" and "CeaseTx" actions only.
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeCallPTTRequest
Used by	Element interfaceCall/pttRequest
Model	requestId , action , attributes{0,1} , talkingParty{0,1} , workstationId{0,1}
Children	action, attributes, requestId, talkingParty, workstationId
Source	<xs:complexType name="typeCallPTTRequest"> <xs:annotation> <xs:documentation>DR-GW-Call PTTRequest. Only "DemandTx" and "CeaseTx" actions only.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="action" type="typeActionPTTRequest"/> <xs:element name="attributes" type="typeCallAttributes" minOccurs="0"/> <xs:element name="talkingParty" type="typeAddress" minOccurs="0"/> <xs:element name="workstationId" type="typeWorkstationId" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeCallKeyExchange`

Namespace	DR-GW
Annotations	`For triggering the group key exchange. Key exchange events are sent in Call_KeyXEvent.`
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeCallKeyExchange
Used by	Element interfaceCall/keyExchange
Model	requestId , action
Children	action, requestId

Source	`<xs:complexType name="typeCallKeyExchange">`
	`  <xs:annotation>`
	`  <xs:documentation>For triggering the group key exchange. Key exchange events are sent in Call_KeyXEvent.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="action" type="typeKeyExchangeAction"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

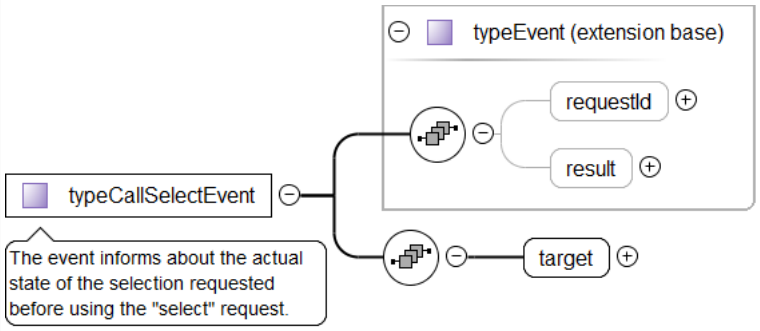
Complex Type `typeResponse`

Namespace	DR-GW
Annotations	`Response contains result of execution of any method.`
Diagram	
Used by	Elements interfaceApp/response, interfaceCall/response, interfaceGroup/response, interfaceManagement/response, interfaceOrg/response, interfaceRadio/response, interfaceSds/response, interfaceSession/response, interfaceStatus/response, interfaceSystem/response
Model	requestId , result
Children	requestId, result

Source	`<xs:complexType name="typeResponse">`
	`  <xs:annotation>`
	`  <xs:documentation>Response contains result of execution of any method.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="requestId" type="xs:unsignedLong"/>`
	`  <xs:element name="result" type="typeResult"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

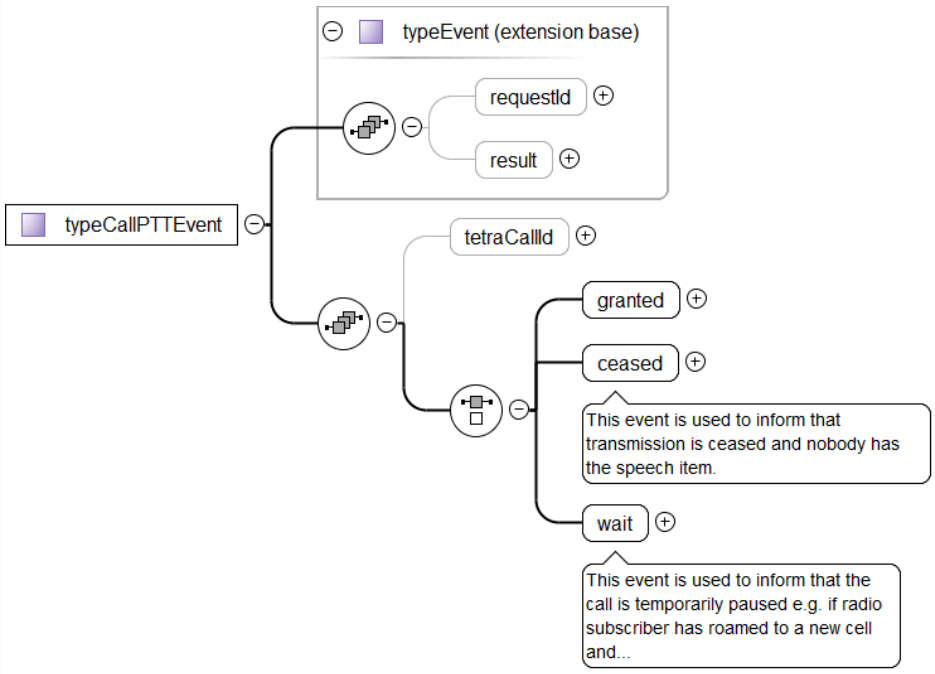
Complex Type `typeCallSelectEvent`

Namespace	DR-GW
Annotations	`The event informs about the actual state of the selection requested before using the "select" request.`

Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeCallSelectEvent
Used by	Element interfaceCall/selectEvent
Model	requestId{0,1} , result{0,1} , target
Children	requestId, result, target

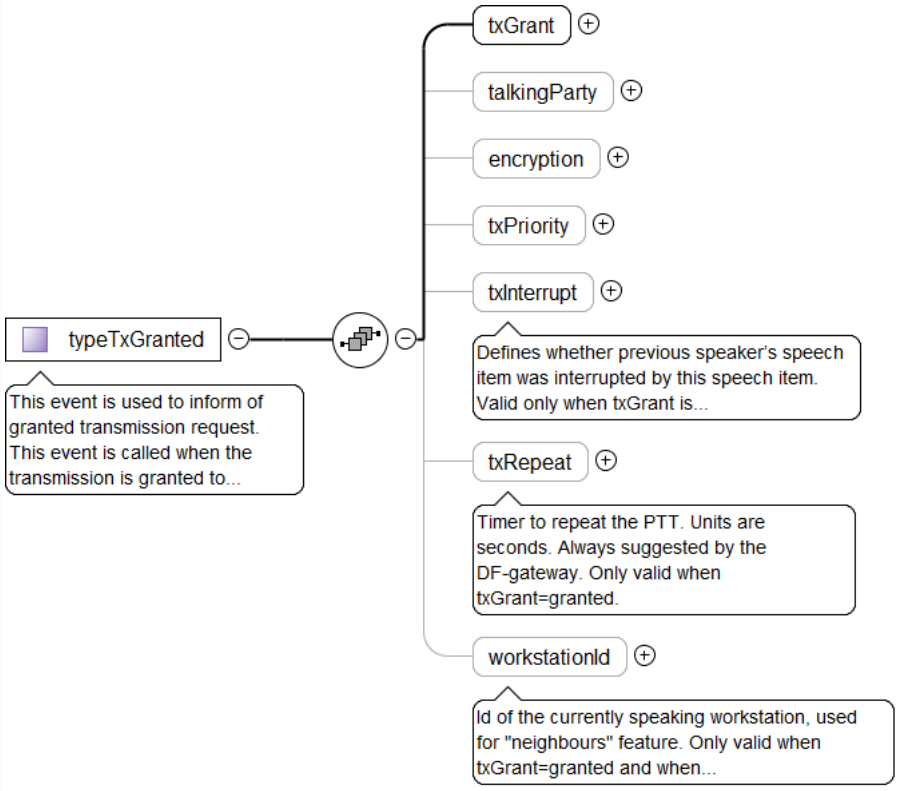
Source	`<xs:complexType name="typeCallSelectEvent">`
	`  <xs:annotation>`
	`  <xs:documentation>The event informs about the actual state of the selection requested before using the "select" request.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="target" type="typeSelection"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Complex Type `typeCallPTTEvent`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeCallPTTEvent

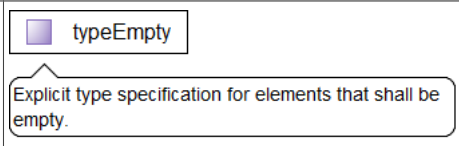
Used by	Element interfaceCall/pttEvent
Model	requestId{0,1} , result{0,1} , tetraCallId{0,1} , (granted | ceased | wait)
Children	ceased, granted, requestId, result, tetraCallId, wait
Source	<xs:complexType name="typeCallPTTEvent"> <xs:annotation> <xs:documentation> </xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="tetraCallId" type="typeTetraCallId" minOccurs="0"/> <xs:choice> <xs:element name="granted" type="typeTxGranted"/> <xs:element name="ceased" type="typeEmpty"> <xs:annotation> <xs:documentation>This event is used to inform that transmission is ceased and nobody has the speech item.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="wait" type="typeEmpty"> <xs:annotation> <xs:documentation>This event is used to inform that the call is temporarily paused e.g. if radio subscriber has roamed to a new cell and there are no free resources available.</xs:documentation> </xs:annotation> </xs:element> </xs:choice> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type typeTxGranted

Namespace	DR-GW
Annotations	This event is used to inform of granted transmission request. This event is called when the transmission is granted to client, to another user or the request has been queued.
Diagram	
Used by	Element typeCallPTTEvent/granted
Model	txGrant , talkingParty{0,1} , encryption{0,1} , txPriority{0,1} , txInterrupt{0,1} , txRepeat{0,1} , workstationId{0,1}

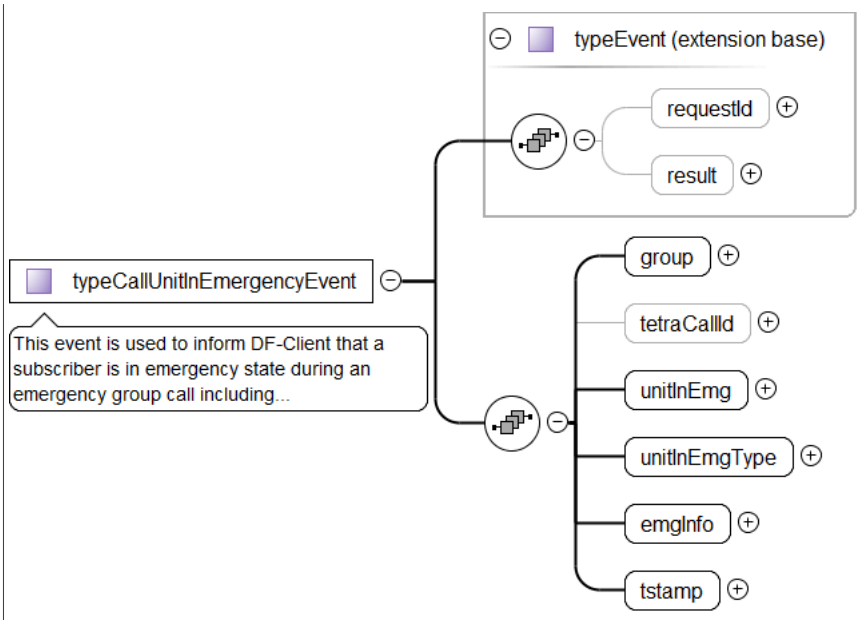
Children	encryption, talkingParty, txGrant, txInterrupt, txPriority, txRepeat, workstationId
Source	<xs:complexType name="typeTxGranted"> <xs:annotation> <xs:documentation>This event is used to inform of granted transmission request. This event is called when the transmission is granted to client, to another user or the request has been queued.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="txGrant" type="typeTxGrant"/> <xs:element name="talkingParty" type="typeAddress" minOccurs="0"/> <xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/> <xs:element name="txPriority" type="typeTxPriority" minOccurs="0" default="normal"/> <xs:element name="txInterrupt" type="xs:boolean" default="false" minOccurs="0"> <xs:annotation> <xs:documentation>Defines whether previous speaker's speech item was interrupted by this speech item. Valid only when txGrant is granted2another.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="txRepeat" type="xs:unsignedLong" minOccurs="0" default="0"> <xs:annotation> <xs:documentation>Timer to repeat the PTT. Units are seconds. Always suggested by the DF-gateway. Only valid when txGrant=granted.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="workstationId" type="xs:normalizedString" minOccurs="0"> <xs:annotation> <xs:documentation>Id of the currently speaking workstation, used for "neighbours" feature. Only valid when txGrant=granted and when supplied by the DF-client in PTT request.</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType>

Complex Type `typeEmpty`

Namespace	DR-GW
Annotations	Explicit type specification for elements that shall be empty.
Diagram	
Used by	Elements typeCallPTTEvent/ceased, typeCallPTTEvent/wait
Source	<xs:complexType name="typeEmpty"> <xs:annotation> <xs:documentation>Explicit type specification for elements that shall be empty.</xs:documentation> </xs:annotation> </xs:complexType>

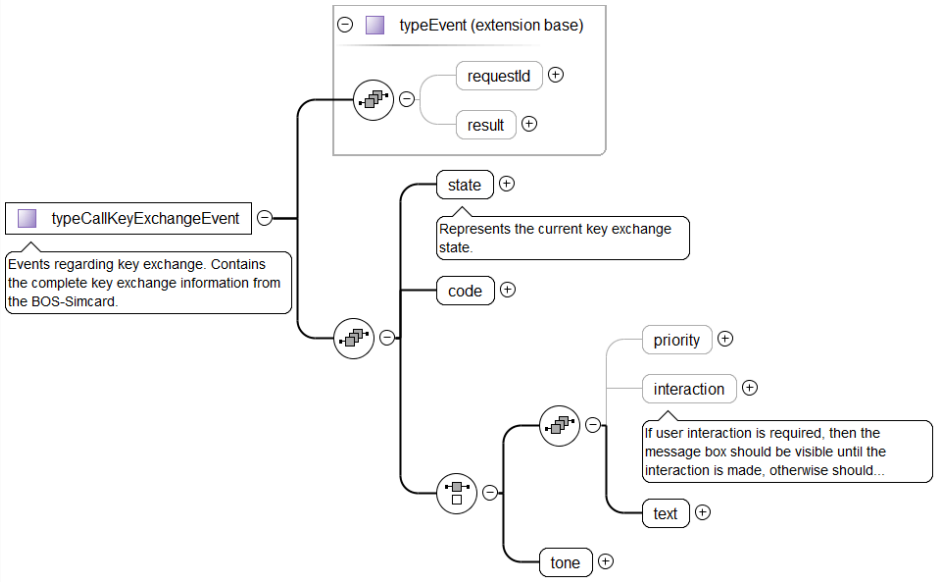
Complex Type `typeCallUnitInEmergencyEvent`

Namespace	DR-GW
Annotations	This event is used to inform DF-Client that a subscriber is in emergency state during an emergency group call including the ending of its emergency situation. Also queuing of emergency speech item requests is indicated using this event. Event is fired every time the TETRA system informs the Gateway that subscriber's emergency information is changed. For example, based on this information TCS Client could use pre-emptive speech item to request the current speaker to stop in order to let the unit in emergency to speak.

Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeCallUnitInEmergencyEvent
Used by	Element interfaceCall/unitInEmergencyEvent
Model	requestId{0,1} , result{0,1} , group , tetraCallId{0,1} , unitInEmg , unitInEmgType , emgInfo , tstamp
Children	emgInfo, group, requestId, result, tetraCallId, tstamp, unitInEmg, unitInEmgType
Source	<xs:complexType name="typeCallUnitInEmergencyEvent"> <xs:annotation> <xs:documentation>This event is used to inform DF-Client that a subscriber is in emergency state during an emergency group call including the ending of its emergency situation. Also queuing of emergency speech item requests is indicated using this event. Event is fired every time the TETRA system informs the Gateway that subscriber's emergency information is changed. For example, based on this information TCS Client could use pre-emptive speech item to request the current speaker to stop in order to let the unit in emergency to speak.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="tetraCallId" type="typeTetraCallId" minOccurs="0"/> <xs:element name="unitInEmg" type="typeAddress"/> <xs:element name="unitInEmgType" type="typeUnitInEmergencyType"/> <xs:element name="emgInfo" type="typeEmergencyInfo"/> <xs:element name="tstamp" type="xs:dateTime"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

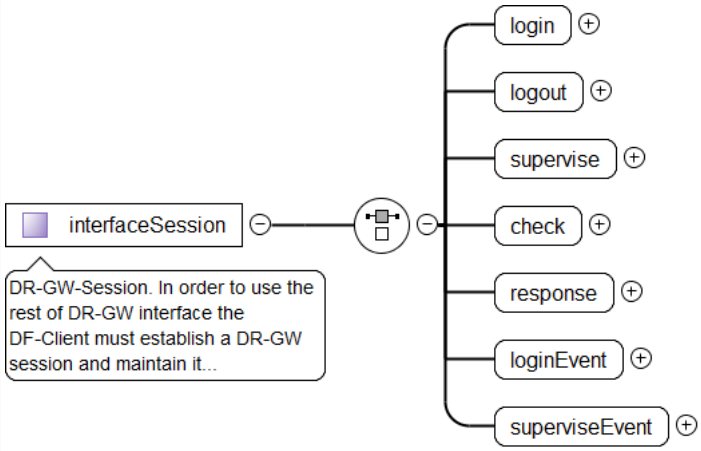
Complex Type `typeCallKeyExchangeEvent`

Namespace	DR-GW
Annotations	Events regarding key exchange. Contains the complete key exchange information from the BOS-Simcard.

Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeCallKeyExchangeEvent
Used by	Element interfaceCall/keyExchangeEvent
Model	requestId{0,1} , result{0,1} , state , code , ((priority{0,1} , interaction{0,1} , text) | tone)
Children	code, interaction, priority, requestId, result, state, text, tone
Source	<xs:complexType name="typeCallKeyExchangeEvent"> <xs:annotation> <xs:documentation>Events regarding key exchange. Contains the complete key exchange information from the BOS-Simcard.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="state" type="typeKeyExchangeState"> <xs:annotation> <xs:documentation>Represents the current key exchange state.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="code" type="typeKeyExchangeCode"/> <xs:choice> <xs:sequence> <xs:element name="priority" type="typeKeyExchangeTextPriority" minOccurs="0" default="normal"/> <xs:element name="interaction" type="xs:boolean" minOccurs="0" default="false"> <xs:annotation> <xs:documentation>If user interaction is required, then the message box should be visible until the interaction is made, otherwise should be hidden after delay.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="text" type="typeKeyExchangeText"/> </xs:sequence> <xs:element name="tone" type="xs:boolean" fixed="true"/> </xs:choice> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

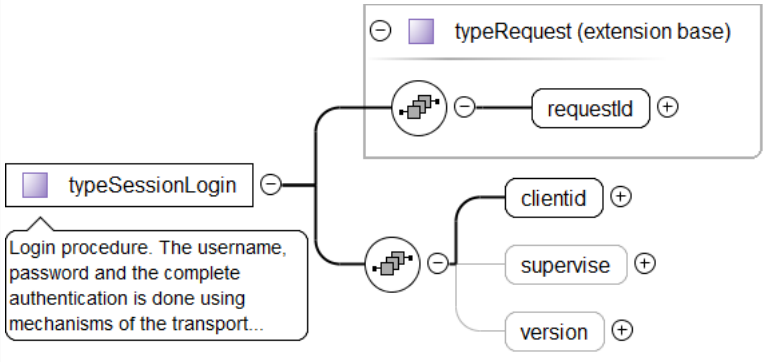
Complex Type `interfaceSession`

Namespace	DR-GW
Annotations	DR-GW-Session. In order to use the rest of DR-GW interface the DF-Client must establish a DR-GW session and maintain it using supervise in a timely manner. Use only via SOAP.

Diagram	
Used by	Element drgw/session
Model	login | logout | supervise | check | response | loginEvent | superviseEvent
Children	check, login, loginEvent, logout, response, supervise, superviseEvent

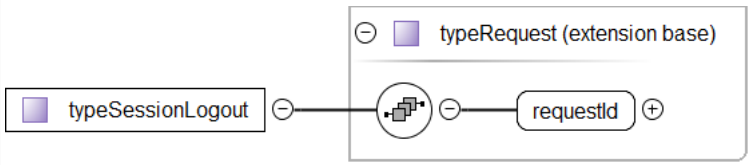
Source	`<xs:complexType name="interfaceSession">`
	`  <xs:annotation>`
	`  <xs:documentation>DR-GW-Session. In order to use the rest of DR-GW interface the DF-Client must establish a DR-GW session and maintain it using supervise in a timely manner. Use only via SOAP.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:choice>`
	`  <xs:element name="login" type="typeSessionLogin"/>`
	`  <xs:element name="logout" type="typeSessionLogout"/>`
	`  <xs:element name="supervise" type="typeSessionSupervise"/>`
	`  <xs:element name="check" type="typeSessionCheck"/>`
	`  <xs:element name="response" type="typeResponse"/>`
	`  <xs:element name="loginEvent" type="typeSessionLoginEvent"/>`
	`  <xs:element name="superviseEvent" type="typeSessionSuperviseEvent"/>`
	`  </xs:choice>`
	`</xs:complexType>`

Complex Type `typeSessionLogin`

Namespace	DR-GW
Annotations	`Login procedure. The username, password and the complete authentication is done using mechanisms of the transport protocol, digest access authentication on HTTP.`
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeSessionLogin
Used by	Element interfaceSession/login
Model	requestId , clientid , supervise{0,1} , version{0,1}
Children	clientid, requestId, supervise, version
Source	`<xs:complexType name="typeSessionLogin">` `  <xs:annotation>`

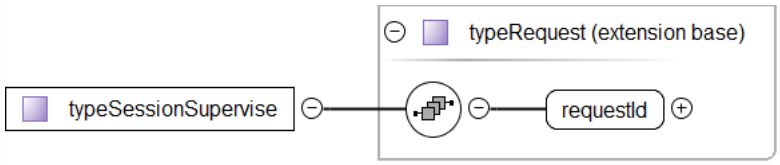
```
    <xs:documentation>Login procedure. The username, password and the complete authentication
 is done using mechanisms of the transport protocol, digest access authentication on HTTP.</
xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="typeRequest">
      <xs:sequence>
        <xs:element name="clientid" type="xs:string"/>
        <xs:element name="supervise" type="typeSuperviseTimeout" default="60" minOccurs="0"/>
        <xs:element name="version" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeSessionLogout`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeSessionLogout
Used by	Element interfaceSession/logout
Model	requestId
Children	requestId

Source	`<xs:complexType name="typeSessionLogout">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest"/>`
	`  </xs:complexContent>`
	`</xs:complexType>`

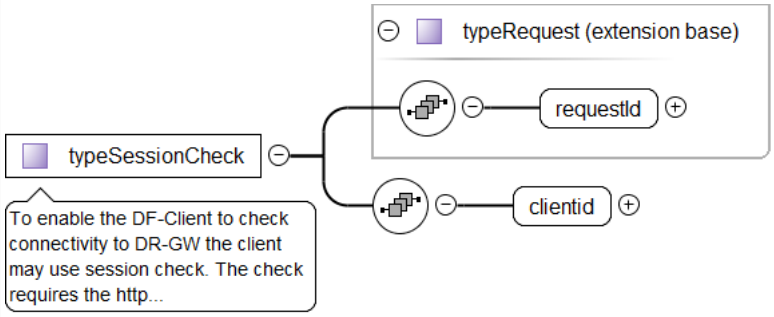
Complex Type `typeSessionSupervise`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeSessionSupervise
Used by	Element interfaceSession/supervise
Model	requestId
Children	requestId

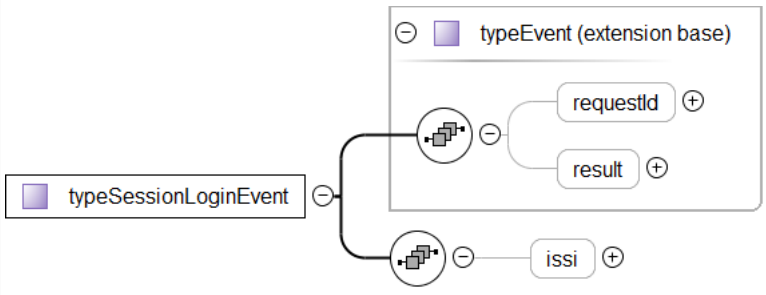
Source	`<xs:complexType name="typeSessionSupervise">`
	`  <xs:annotation>`
	`  <xs:documentation>`
	`  </xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest"/>`
	`  </xs:complexContent>`

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

Complex Type `typeSessionCheck`

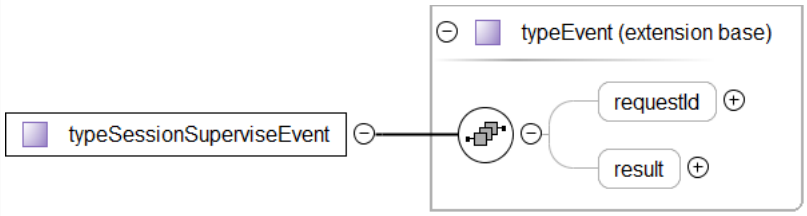
Namespace	DR-GW
Annotations	To enable the DF-Client to check connectivity to DR-GW the client may use session check. The check requires the http authentication as it would be needed for a session login. Once the session check is OK, the client should be able to login later on. There is no resource allocation associated with the session check.
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeSessionCheck
Used by	Elements check, interfaceSession/check
Model	requestId , clientid
Children	clientid, requestId
Source	<xs:complexType name="typeSessionCheck"> <xs:annotation> <xs:documentation>To enable the DF-Client to check connectivity to DR-GW the client may use session check. The check requires the http authentication as it would be needed for a session login. Once the session check is OK, the client should be able to login later on. There is no resource allocation associated with the session check.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="clientid" type="xs:string"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeSessionLoginEvent`

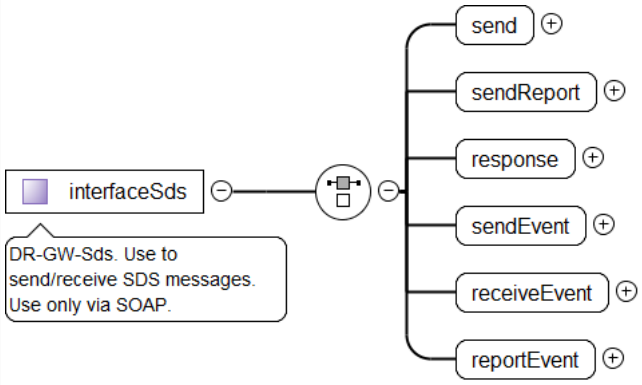
Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeSessionLoginEvent
Used by	Element interfaceSession/loginEvent

Model	requestId{0,1} , result{0,1} , issi{0,1}
Children	issi, requestId, result
Source	<xs:complexType name="typeSessionLoginEvent"> <xs:annotation> <xs:documentation> </xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="issi" type="xs:string" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeSessionSuperviseEvent`

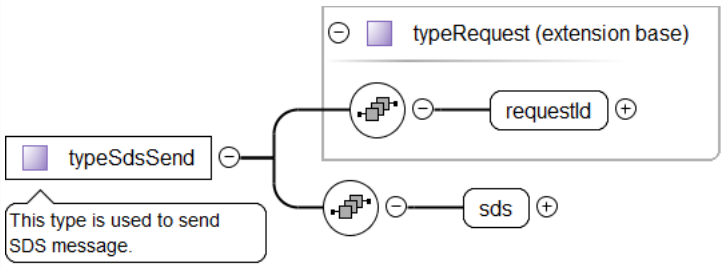
Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierar-chy	• typeEvent • typeSessionSuperviseEvent
Used by	Element interfaceSession/superviseEvent
Model	requestId{0,1} , result{0,1}
Children	requestId, result
Source	<xs:complexType name="typeSessionSuperviseEvent"> <xs:annotation> <xs:documentation> </xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"/> </xs:complexContent> </xs:complexType>

Complex Type `interfaceSds`

Namespace	DR-GW
Annotations	DR-GW-Sds. Use to send/receive SDS messages. Use only via SOAP.
Diagram	
Used by	Element drgw/sds

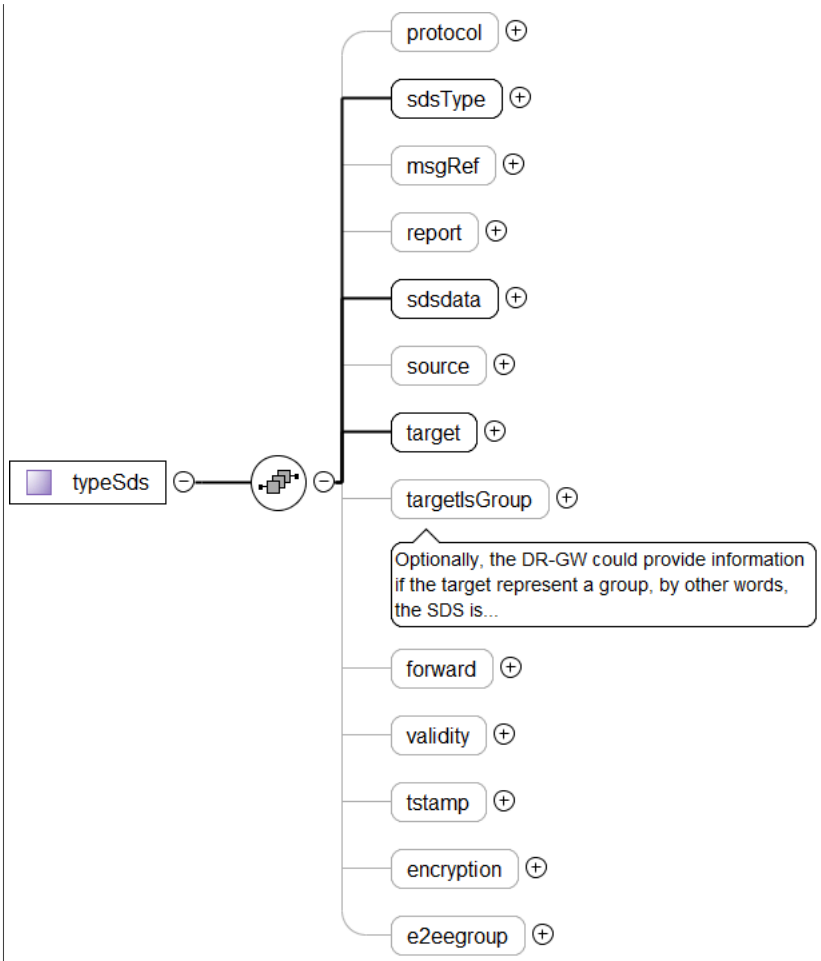
Model	send | sendReport | response | sendEvent | receiveEvent | reportEvent
Children	receiveEvent, reportEvent, response, send, sendEvent, sendReport
Source	<xs:complexType name="interfaceSds"> <xs:annotation> <xs:documentation>DR-GW-Sds. Use to send/receive SDS messages. Use only via SOAP.</ xs:documentation> </xs:annotation> <xs:choice> <xs:element name="send" type="typeSdsSend"/> <xs:element name="sendReport" type="typeSdsSendReport"/> <xs:element name="response" type="typeResponse"/> <xs:element name="sendEvent" type="typeSdsSendEvent"/> <xs:element name="receiveEvent" type="typeSdsReceiveEvent"/> <xs:element name="reportEvent" type="typeSdsReportEvent"/> </xs:choice> </xs:complexType>

Complex Type typeSdsSend

Namespace	DR-GW
Annotations	This type is used to send SDS message.
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeSdsSend
Used by	Element interfaceSds/send
Model	requestId , sds
Children	requestId, sds
Source	<xs:complexType name="typeSdsSend"> <xs:annotation> <xs:documentation>This type is used to send SDS message.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="sds" type="typeSds"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

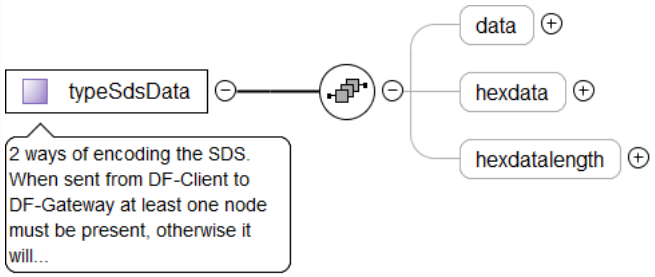
Complex Type typeSds

Namespace	DR-GW
Annotations	

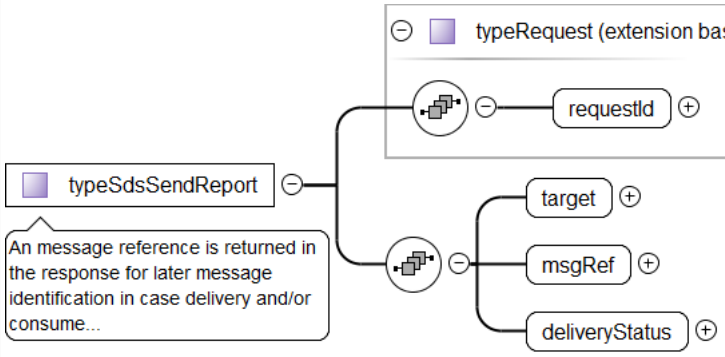
Diagram	
Used by	Elements typeSdsReceiveEvent/sds, typeSdsSend/sds, typeSdsSendEvent/sds
Model	protocol{0,1} , sdsType , msgRef{0,1} , report{0,1} , sdsdata , source{0,1} , target , targetIsGroup{0,1} , forward{0,1} , validity{0,1} , tstamp{0,1} , encryption{0,1} , e2eegroup{0,1}
Children	e2eegroup, encryption, forward, msgRef, protocol, report, sdsType, sdsdata, source, target, targetIsGroup, tstamp, validity
Source	<xs:complexType name="typeSds"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="protocol" type="xs:unsignedByte" minOccurs="0"/> <xs:element name="sdsType" type="typeSdsType"/> <xs:element name="msgRef" type="xs:unsignedByte" minOccurs="0"/> <xs:element name="report" type="typeReport" default="none" minOccurs="0"/> <xs:element name="sdsdata" type="typeSdsData"/> <xs:element name="source" type="typeAddress" minOccurs="0"/> <xs:element name="target" type="typeAddress"/> <xs:element name="targetIsGroup" type="xs:boolean" minOccurs="0"> <xs:annotation> <xs:documentation>Optionally, the DR-GW could provide information if the target represent a group, by other words, the SDS is group-addressed. Also the Df-Client can provide this information to help DR-GW decide which resource to use for actual sending.</xs:documentation> </xs:annotation> </xs:element> <xs:element name="forward" type="typeAddress" minOccurs="0"/> <xs:element name="validity" type="xs:unsignedByte" minOccurs="0"/> <xs:element name="tstamp" type="xs:dateTime" minOccurs="0"/> <xs:element name="encryption" type="xs:boolean" default="true" minOccurs="0"/> <xs:element name="e2eegroup" type="typeSubscriberAddress" minOccurs="0"/> </xs:sequence> </xs:complexType>

Complex Type `typeSdsData`

Namespace	DR-GW
Annotations	2 ways of encoding the SDS. When sent from DF-Client to DF-Gateway at least

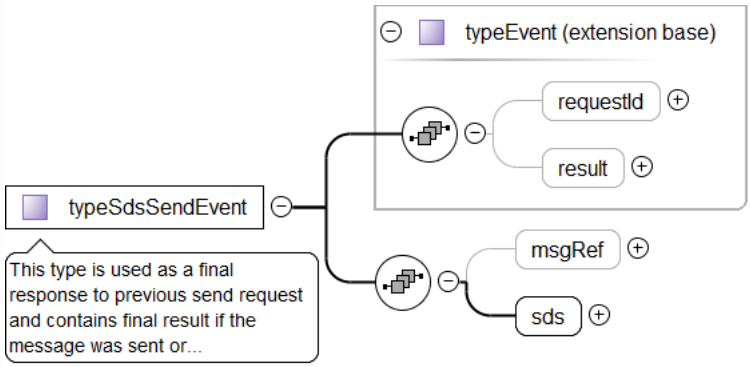
	one node must be present, otherwise it will be discarded as not valid. When sent from DF-Gateway to DF-Client both nodes must be present, as it is unclear if the DF-Client supports the encoding inside raw "hexdata", so the readable decoded content must be present to. The default charset used within the "data" node is ISO-8859-15.
Diagram	
Used by	Element typeSds/sdsdata
Model	data{0,1} , hexdata{0,1} , hexdatalength{0,1}
Children	data, hexdata, hexdatalength
Source	<xs:complexType name="typeSdsData"> <xs:annotation> <xs:documentation>2 ways of encoding the SDS. When sent from DF-Client to DF-Gateway at least one node must be present, otherwise it will be discarded as not valid. When sent from DF-Gateway to DF-Client both nodes must be present, as it is unclear if the DF-Client supports the encoding inside raw "hexdata", so the readable decoded content must be present to. The default charset used within the "data" node is ISO-8859-15.</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="data" type="xs:string" minOccurs="0"/> <xs:element name="hexdata" type="xs:hexBinary" minOccurs="0"/> <xs:element name="hexdatalength" type="xs:integer" minOccurs="0"/> </xs:sequence> </xs:complexType>

Complex Type typeSdsSendReport

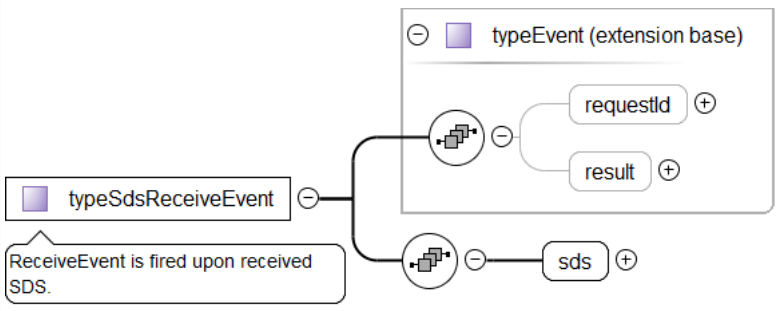
Namespace	DR-GW
Annotations	An message reference is returned in the response for later message identification in case delivery and/or consume requests were asked. Is only valid for SDS-TL.
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeSdsSendReport
Used by	Element interfaceSds/sendReport
Model	requestId , target , msgRef , deliveryStatus
Children	deliveryStatus, msgRef, requestId, target
Source	<xs:complexType name="typeSdsSendReport"> <xs:annotation> <xs:documentation>An message reference is returned in the response for later message identification in case delivery and/or consume requests were asked. Is only valid for SDS-TL.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence>

```
        <xs:element name="target" type="typeAddress"/>
        <xs:element name="msgRef" type="xs:unsignedByte"/>
        <xs:element name="deliveryStatus" type="xs:unsignedByte"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeSdsSendEvent`

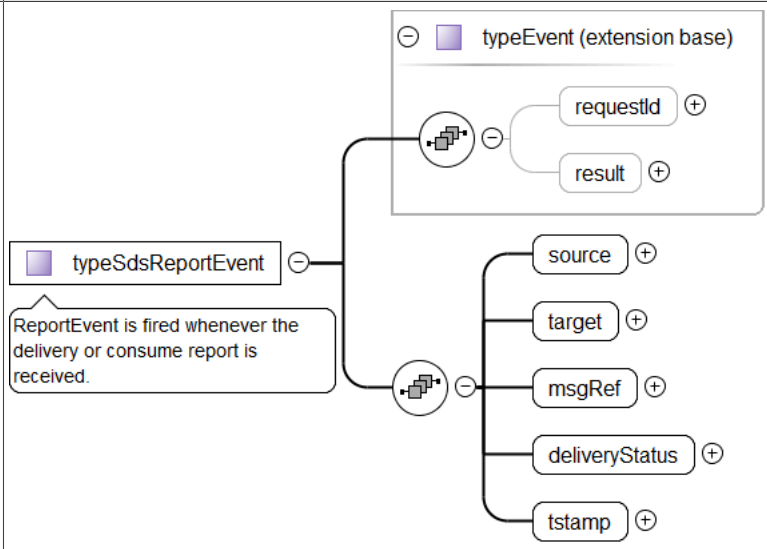
Namespace	DR-GW
Annotations	`This type is used as a final response to previous send request and contains final result if the message was sent or not. A message reference is returned in the response for later message identification in case delivery and/or consume requests were asked. Is valid only for SDS-TL.`
Diagram	
Type	extension of typeEvent
Type hierar-chy	• typeEvent • typeSdsSendEvent
Used by	Element interfaceSds/sendEvent
Model	requestId{0,1} , result{0,1} , msgRef{0,1} , sds
Children	msgRef, requestId, result, sds
Source	`<xs:complexType name="typeSdsSendEvent">` `  <xs:annotation>` ` <xs:documentation>This type is used as a final response to previous send request and contains final result if the message was sent or not. A message reference is returned in the response for later message identification in case delivery and/or consume requests were asked. Is valid only for SDS-TL.</xs:documentation>` ` </xs:annotation>` ` <xs:complexContent>` ` <xs:extension base="typeEvent">` ` <xs:sequence>` ` <xs:element name="msgRef" type="xs:unsignedByte" minOccurs="0" default="0"/>` ` <xs:element name="sds" type="typeSds"/>` ` </xs:sequence>` ` </xs:extension>` ` </xs:complexContent>` `</xs:complexType>`

Complex Type `typeSdsReceiveEvent`

Namespace	DR-GW
Annotations	`ReceiveEvent is fired upon received SDS.`
Diagram	

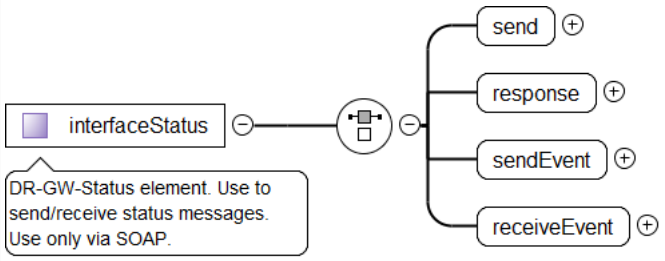
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeSdsReceiveEvent
Used by	Element interfaceSds/receiveEvent
Model	requestId{0,1} , result{0,1} , sds
Children	requestId, result, sds
Source	<xs:complexType name="typeSdsReceiveEvent"> <xs:annotation> <xs:documentation>ReceiveEvent is fired upon received SDS.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="sds" type="typeSds"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type typeSdsReportEvent

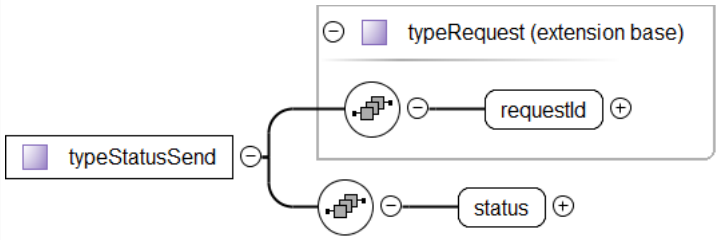
Namespace	DR-GW
Annotations	ReportEvent is fired whenever the delivery or consume report is received.
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeSdsReportEvent
Used by	Element interfaceSds/reportEvent
Model	requestId{0,1} , result{0,1} , source , target , msgRef , deliveryStatus , tstamp
Children	deliveryStatus, msgRef, requestId, result, source, target, tstamp
Source	<xs:complexType name="typeSdsReportEvent"> <xs:annotation> <xs:documentation>ReportEvent is fired whenever the delivery or consume report is received.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="source" type="typeAddress"/> <xs:element name="target" type="typeAddress"/> <xs:element name="msgRef" type="xs:unsignedByte"/> <xs:element name="deliveryStatus" type="xs:unsignedByte"/> <xs:element name="tstamp" type="xs:dateTime"/> </xs:sequence> </xs:extension>

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

Complex Type `interfaceStatus`

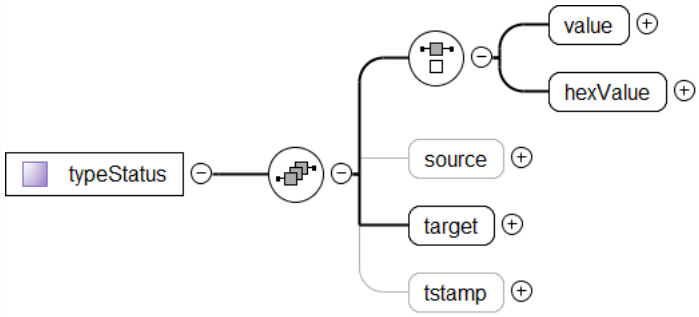
Namespace	DR-GW
Annotations	`DR-GW-Status element. Use to send/receive status messages. Use only via SOAP.`
Diagram	
Used by	Element drgw/status
Model	send | response | sendEvent | receiveEvent
Children	receiveEvent, response, send, sendEvent
Source	<xs:complexType name="interfaceStatus"> <xs:annotation> <xs:documentation>DR-GW-Status element. Use to send/receive status messages. Use only via SOAP.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="send" type="typeStatusSend"/> <xs:element name="response" type="typeResponse"/> <xs:element name="sendEvent" type="typeStatusSendEvent"/> <xs:element name="receiveEvent" type="typeStatusReceiveEvent"/> </xs:choice> </xs:complexType>

Complex Type `typeStatusSend`

Namespace	DR-GW
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeStatusSend
Used by	Element interfaceStatus/send
Model	requestId , status
Children	requestId, status
Source	<xs:complexType name="typeStatusSend"> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="status" type="typeStatus"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

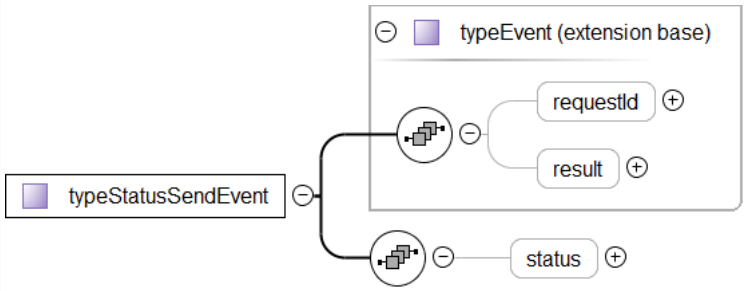
Complex Type `typeStatus`

Namespace	DR-GW

Diagram	
Used by	Elements typeStatusReceiveEvent/status, typeStatusSend/status, typeStatusSendEvent/status
Model	(value | hexValue) , source{0,1} , target , tstamp{0,1}
Children	hexValue, source, target, tstamp, value

Source	`<xs:complexType name="typeStatus">`
	`  <xs:sequence>`
	`  <xs:choice>`
	`  <xs:element name="value" type="xs:unsignedShort"/>`
	`  <xs:element name="hexValue" type="xs:hexBinary"/>`
	`  </xs:choice>`
	`  <xs:element name="source" type="typeAddress" minOccurs="0"/>`
	`  <xs:element name="target" type="typeAddress"/>`
	`  <xs:element name="tstamp" type="xs:dateTime" minOccurs="0"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

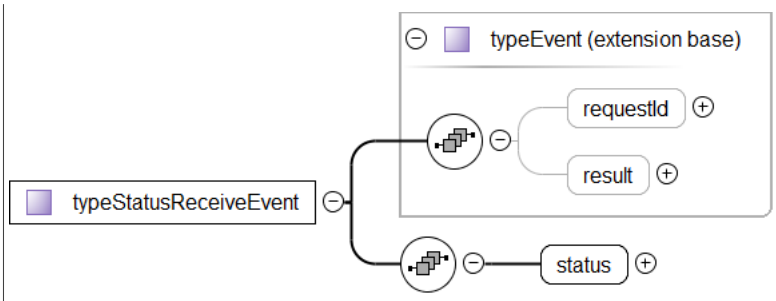
Complex Type `typeStatusSendEvent`

Namespace	DR-GW
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeStatusSendEvent
Used by	Element interfaceStatus/sendEvent
Model	requestId{0,1} , result{0,1} , status{0,1}
Children	requestId, result, status

Source	`<xs:complexType name="typeStatusSendEvent">`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="status" type="typeStatus" minOccurs="0"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

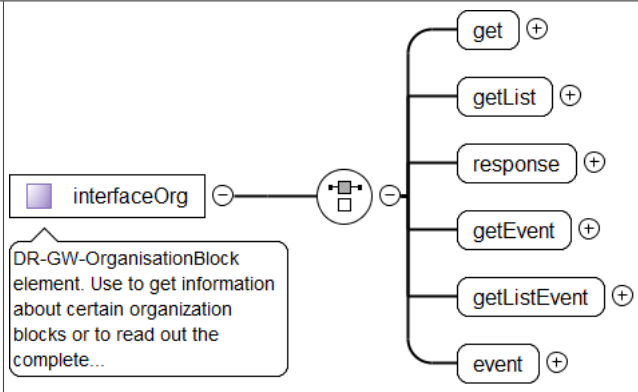
Complex Type `typeStatusReceiveEvent`

Namespace	DR-GW

Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeStatusReceiveEvent
Used by	Element interfaceStatus/receiveEvent
Model	requestId{0,1} , result{0,1} , status
Children	requestId, result, status

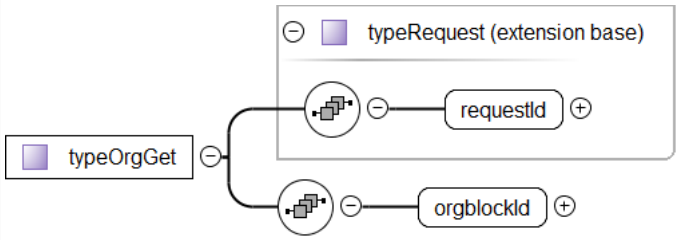
Source	`<xs:complexType name="typeStatusReceiveEvent">`
	`<xs:complexContent>`
	`<xs:extension base="typeEvent">`
	`<xs:sequence>`
	`<xs:element name="status" type="typeStatus"/>`
	`</xs:sequence>`
	`</xs:extension>`
	`</xs:complexContent>`
	`</xs:complexType>`

Complex Type `interfaceOrg`

Namespace	DR-GW
Annotations	DR-GW-OrganisationBlock element. Use to get information about certain organization blocks or to read out the complete organization schema the DF-Client has rights for. Use only via SOAP.
Diagram	
Used by	Element drgw/org
Model	get | getList | response | getEvent | getListEvent | event
Children	event, get, getEvent, getList, getListEvent, response

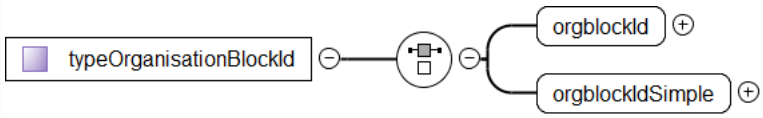
Source	`<xs:complexType name="interfaceOrg">`
	`<xs:annotation>`
	`<xs:documentation>DR-GW-OrganisationBlock element. Use to get information about certain organization blocks or to read out the complete organization schema the DF-Client has rights for. Use only via SOAP.</xs:documentation>`
	`</xs:annotation>`
	`<xs:choice>`
	`<xs:element name="get" type="typeOrgGet"/>`
	`<xs:element name="getList" type="typeOrgGetList"/>`
	`<xs:element name="response" type="typeResponse"/>`
	`<xs:element name="getEvent" type="typeOrgGetEvent"/>`
	`<xs:element name="getListEvent" type="typeOrgGetListEvent"/>`
	`<xs:element name="event" type="typeOrgEvent"/>`
	`</xs:choice>`
	`</xs:complexType>`

Complex Type `typeOrgGet`

Namespace	DR-GW
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeOrgGet
Used by	Element interfaceOrg/get
Model	requestId , orgblockId
Children	orgblockId, requestId

Source	`<xs:complexType name="typeOrgGet">`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="orgblockId" type="typeOrganisationBlockId"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

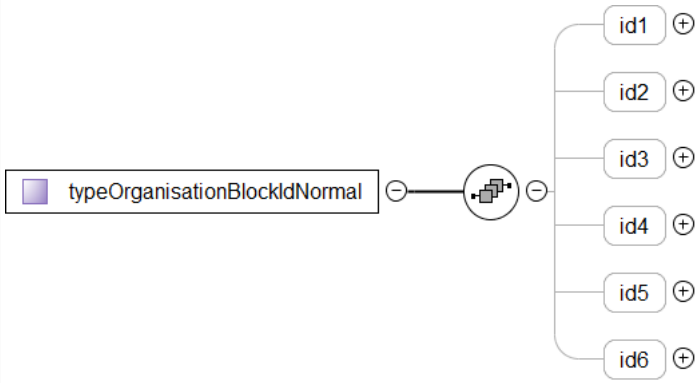
Complex Type `typeOrganisationBlockId`

Namespace	DR-GW
Annotations	
Diagram	
Used by	Elements typeAppGetList/orgblockId, typeApplication/orgblockId, typeGroup/orgblockId, typeGroupGetList/orgblockId, typeOrgGet/orgblockId, typeOrgGetList/orgblockId, typeOrganisationBlock/orgblockId, typeRadio/orgblockId, typeRadioGetList/orgblockId
Model	orgblockId | orgblockIdSimple
Children	orgblockId, orgblockIdSimple

Source	`<xs:complexType name="typeOrganisationBlockId">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:choice>`
	`  <xs:element name="orgblockId" type="typeOrganisationBlockIdNormal"/>`
	`  <xs:element name="orgblockIdSimple" type="typeOrganisationBlockIdSimple"/>`
	`  </xs:choice>`
	`</xs:complexType>`

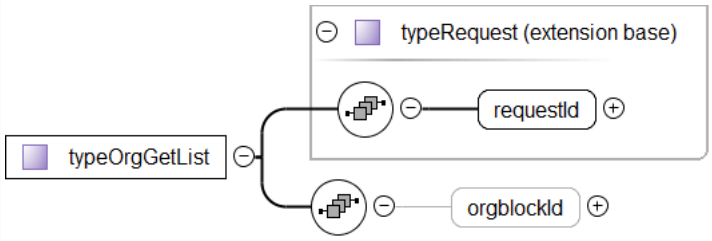
Complex Type `typeOrganisationBlockIdNormal`

Namespace	DR-GW
Annotations	

Diagram	
Used by	Element typeOrganisationBlockId/orgblockId
Model	id1{0,1} , id2{0,1} , id3{0,1} , id4{0,1} , id5{0,1} , id6{0,1}
Children	id1, id2, id3, id4, id5, id6

Source	`<xs:complexType name="typeOrganisationBlockIdNormal">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="id1" type="xs:unsignedShort" minOccurs="0"/>`
	`  <xs:element name="id2" type="xs:unsignedShort" minOccurs="0"/>`
	`  <xs:element name="id3" type="xs:unsignedShort" minOccurs="0"/>`
	`  <xs:element name="id4" type="xs:unsignedShort" minOccurs="0"/>`
	`  <xs:element name="id5" type="xs:unsignedShort" minOccurs="0"/>`
	`  <xs:element name="id6" type="xs:unsignedShort" minOccurs="0"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

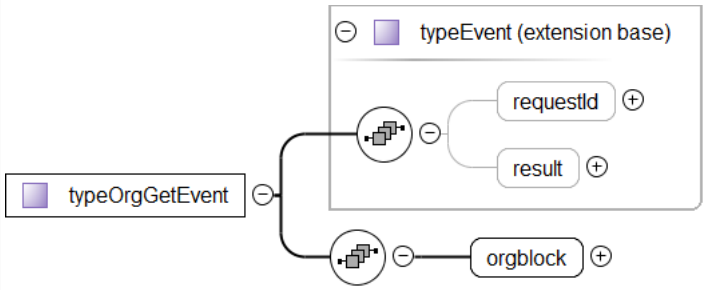
Complex Type `typeOrgGetList`

Namespace	DR-GW
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeOrgGetList
Used by	Element interfaceOrg/getList
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId

Source	`<xs:complexType name="typeOrgGetList">`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

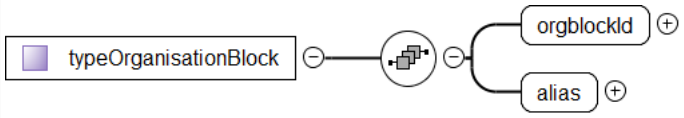
Complex Type `typeOrgGetEvent`

Namespace	DR-GW

Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeOrgGetEvent
Used by	Element interfaceOrg/getEvent
Model	requestId{0,1} , result{0,1} , orgblock
Children	orgblock, requestId, result

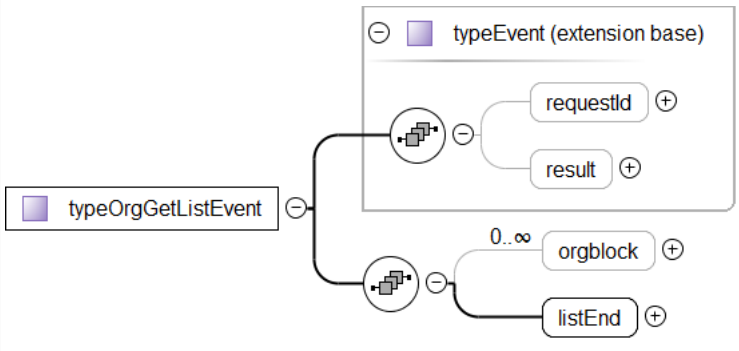
Source	`<xs:complexType name="typeOrgGetEvent">`
	`<xs:complexContent>`
	`<xs:extension base="typeEvent">`
	`<xs:sequence>`
	`<xs:element name="orgblock" type="typeOrganisationBlock"/>`
	`</xs:sequence>`
	`</xs:extension>`
	`</xs:complexContent>`
	`</xs:complexType>`

Complex Type `typeOrganisationBlock`

Namespace	DR-GW
Annotations	
Diagram	
Used by	Elements typeOrgEvent/orgblock, typeOrgGetEvent/orgblock, typeOrgGetListEvent/orgblock
Model	orgblockId , alias
Children	alias, orgblockId

Source	`<xs:complexType name="typeOrganisationBlock">`
	`<xs:annotation>`
	`<xs:documentation/>`
	`</xs:annotation>`
	`<xs:sequence>`
	`<xs:element name="orgblockId" type="typeOrganisationBlockId"/>`
	`<xs:element name="alias" type="xs:normalizedString"/>`
	`</xs:sequence>`
	`</xs:complexType>`

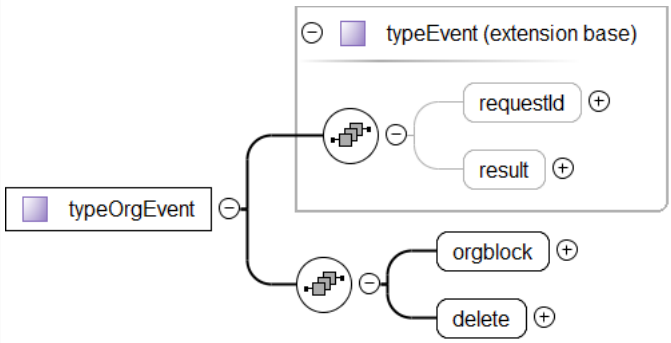
Complex Type `typeOrgGetListEvent`

Namespace	DR-GW
Diagram	

Type	extension of typeEvent
Type hierarchy	• typeEvent • typeOrgGetListEvent
Used by	Element interfaceOrg/getListEvent
Model	requestId{0,1} , result{0,1} , orgblock* , listEnd
Children	listEnd, orgblock, requestId, result

Source	`<xs:complexType name="typeOrgGetListEvent">`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="orgblock" type="typeOrganisationBlock" minOccurs="0"`
	`  maxOccurs="unbounded"/>`
	`  <xs:element name="listEnd" type="xs:boolean"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

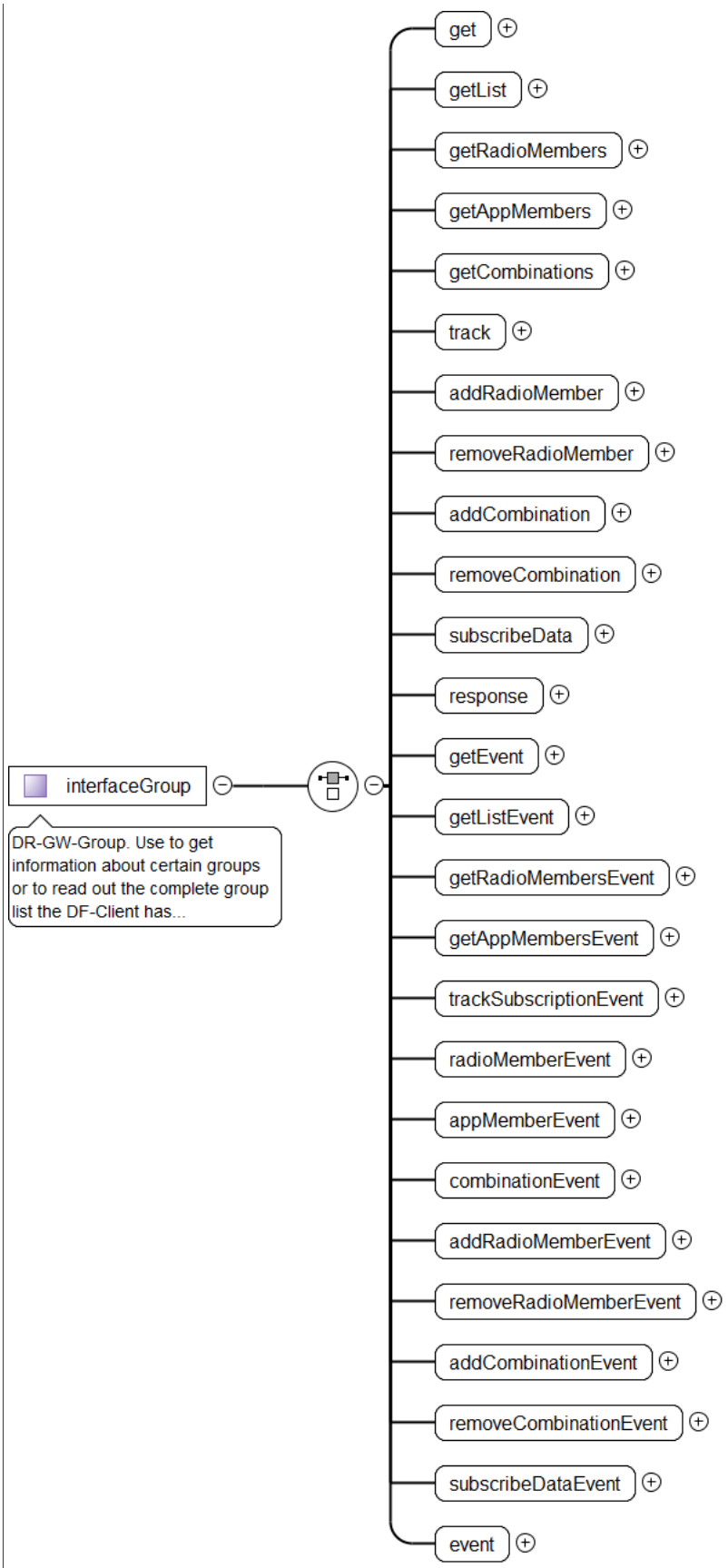
Complex Type `typeOrgEvent`

Namespace	DR-GW
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeOrgEvent
Used by	Element interfaceOrg/event
Model	requestId{0,1} , result{0,1} , orgblock , delete
Children	delete, orgblock, requestId, result

Source	`<xs:complexType name="typeOrgEvent">`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="orgblock" type="typeOrganisationBlock"/>`
	`  <xs:element name="delete" type="xs:boolean"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

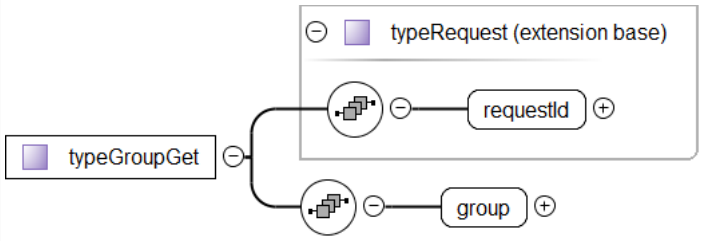
Complex Type `interfaceGroup`

Namespace	DR-GW
Annotations	`DR-GW-Group. Use to get information about certain groups or to read out the complete group list the DF-Client has rights for. There are also many other methods to execute over a certain group. See each method description. Use only via SOAP.`

Diagram	
Used by	Element drgw/group
Model	get | getList | getRadioMembers | getAppMembers | getCombinations | track | addRadioMember | removeRadioMember | addCombination | removeCombination | subscribeData | response | getEvent | getListEvent | getRadioMembersEvent | getAppMembersEvent |

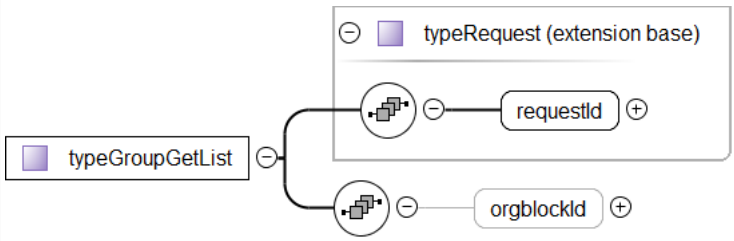
	trackSubscriptionEvent | radioMemberEvent | appMemberEvent | combinationEvent | addRadioMemberEvent | removeRadioMemberEvent | addCombinationEvent | removeCombinationEvent | subscribeDataEvent | event
Children	addCombination, addCombinationEvent, addRadioMember, addRadioMemberEvent, appMemberEvent, combinationEvent, event, get, getAppMembers, getAppMembersEvent, getCombinations, getEvent, getList, getListEvent, getRadioMembers, getRadioMembersEvent, radioMemberEvent, removeCombination, removeCombinationEvent, removeRadioMember, removeRadioMemberEvent, response, subscribeData, subscribeDataEvent, track, trackSubscriptionEvent
Source	<xs:complexType name="interfaceGroup"> <xs:annotation> <xs:documentation>DR-GW-Group. Use to get information about certain groups or to read out the complete group list the DF-Client has rights for. There are also many other methods to execute over a certain group. See each method description. Use only via SOAP.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="get" type="typeGroupGet"/> <xs:element name="getList" type="typeGroupGetList"/> <xs:element name="getRadioMembers" type="typeGroupGetRadioMembers"/> <xs:element name="getAppMembers" type="typeGroupGetAppMembers"/> <xs:element name="getCombinations" type="typeGroupGetCombinations"/> <xs:element name="track" type="typeGroupTrack"/> <xs:element name="addRadioMember" type="typeGroupAddRadioMember"/> <xs:element name="removeRadioMember" type="typeGroupRemoveRadioMember"/> <xs:element name="addCombination" type="typeGroupAddCombination"/> <xs:element name="removeCombination" type="typeGroupRemoveCombination"/> <xs:element name="subscribeData" type="typeGroupSubscribeData"/> <xs:element name="response" type="typeResponse"/> <xs:element name="getEvent" type="typeGroupGetEvent"/> <xs:element name="getListEvent" type="typeGroupGetListEvent"/> <xs:element name="getRadioMembersEvent" type="typeGroupGetRadioMembersEvent"/> <xs:element name="getAppMembersEvent" type="typeGroupGetAppMembersEvent"/> <xs:element name="trackSubscriptionEvent" type="typeGroupTrackSubscriptionEvent"/> <xs:element name="radioMemberEvent" type="typeGroupRadioMemberEvent"/> <xs:element name="appMemberEvent" type="typeGroupAppMemberEvent"/> <xs:element name="combinationEvent" type="typeGroupCombinationEvent"/> <xs:element name="addRadioMemberEvent" type="typeGroupAddRadioMemberEvent"/> <xs:element name="removeRadioMemberEvent" type="typeGroupRemoveRadioMemberEvent"/> <xs:element name="addCombinationEvent" type="typeGroupAddCombinationEvent"/> <xs:element name="removeCombinationEvent" type="typeGroupRemoveCombinationEvent"/> <xs:element name="subscribeDataEvent" type="typeGroupSubscribeDataEvent"/> <xs:element name="event" type="typeGroupEvent"/> </xs:choice> </xs:complexType>

Complex Type `typeGroupGet`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeGroupGet
Used by	Element interfaceGroup/get
Model	requestId , group
Children	group, requestId
Source	<xs:complexType name="typeGroupGet"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> </xs:sequence> </xs:extension>

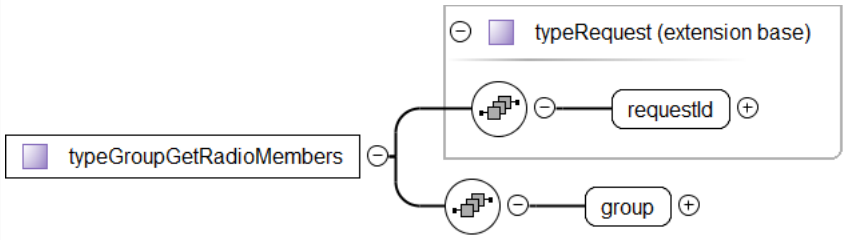
```
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeGroupGetList`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeGroupGetList
Used by	Element interfaceGroup/getList
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId

Source	`<xs:complexType name="typeGroupGetList">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

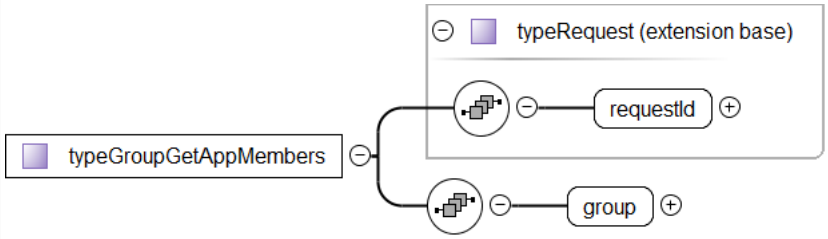
Complex Type `typeGroupGetRadioMembers`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeGroupGetRadioMembers
Used by	Element interfaceGroup/getRadioMembers
Model	requestId , group
Children	group, requestId

Source	`<xs:complexType name="typeGroupGetRadioMembers">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="group" type="typeSubscriberAddress"/>`
	`  </xs:sequence>`

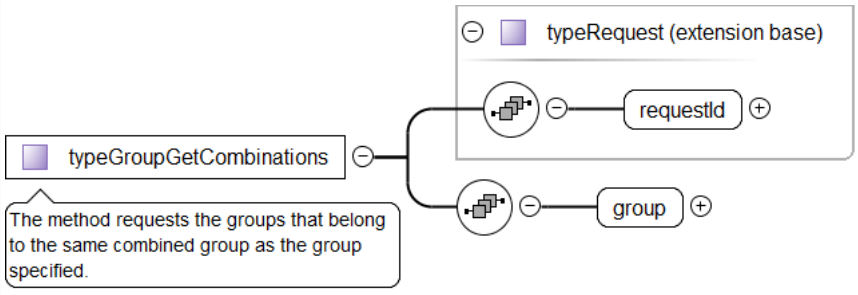
```
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeGroupGetAppMembers`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeGroupGetAppMembers
Used by	Element interfaceGroup/getAppMembers
Model	requestId , group
Children	group, requestId

Source	`<xs:complexType name="typeGroupGetAppMembers">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="group" type="typeSubscriberAddress"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

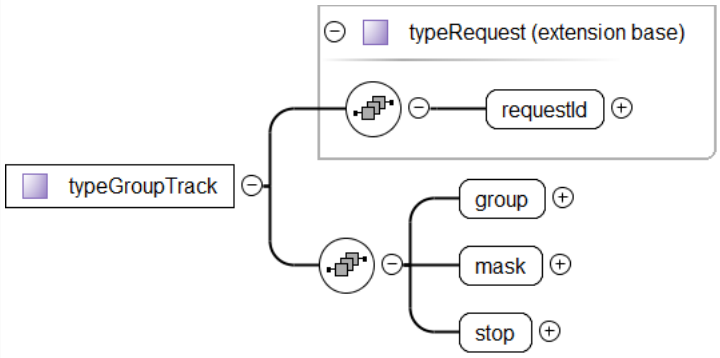
Complex Type `typeGroupGetCombinations`

Namespace	DR-GW
Annotations	The method requests the groups that belong to the same combined group as the group specified.
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeGroupGetCombinations
Used by	Element interfaceGroup/getCombinations
Model	requestId , group
Children	group, requestId

Source	`<xs:complexType name="typeGroupGetCombinations">`
	`  <xs:annotation>`
	`  <xs:documentation>The method requests the groups that belong to the same combined group as the group specified.</xs:documentation>`
	`  </xs:annotation>`

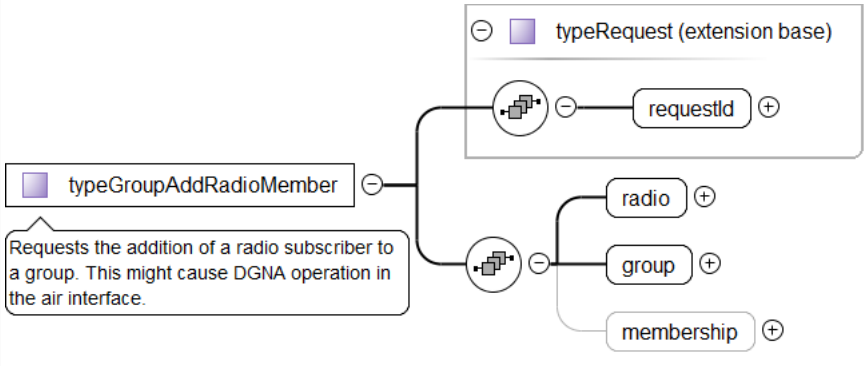
```
  <xs:complexContent>
    <xs:extension base="typeRequest">
      <xs:sequence>
        <xs:element name="group" type="typeSubscriberAddress"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeGroupTrack`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeGroupTrack
Used by	Element interfaceGroup/track
Model	requestId , group , mask , stop
Children	group, mask, requestId, stop

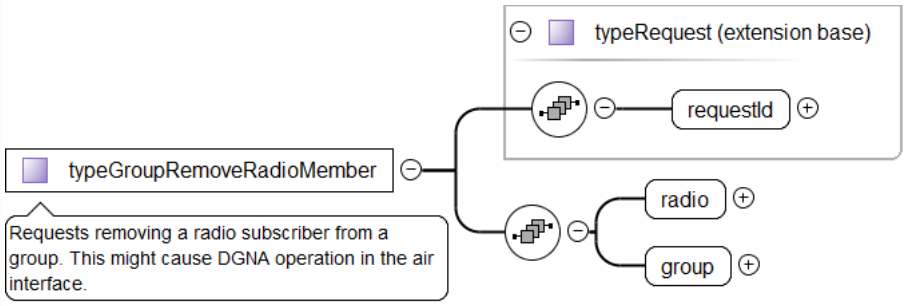
Source	<pre><xs:complexType name="typeGroupTrack">
	  <xs:annotation>
	    <xs:documentation/>
	  </xs:annotation>
	  <xs:complexContent>
	    <xs:extension base="typeRequest">
	      <xs:sequence>
	        <xs:element name="group" type="typeSubscriberAddress"/>
	        <xs:element name="mask" type="typeGroupTrackingMask"/>
	        <xs:element name="stop" type="xs:boolean"/>
	      </xs:sequence>
	    </xs:extension>
	  </xs:complexContent>
	</xs:complexType></pre>

Complex Type `typeGroupAddRadioMember`

Namespace	DR-GW
Annotations	Requests the addition of a radio subscriber to a group. This might cause DGNA operation in the air interface.
Diagram	

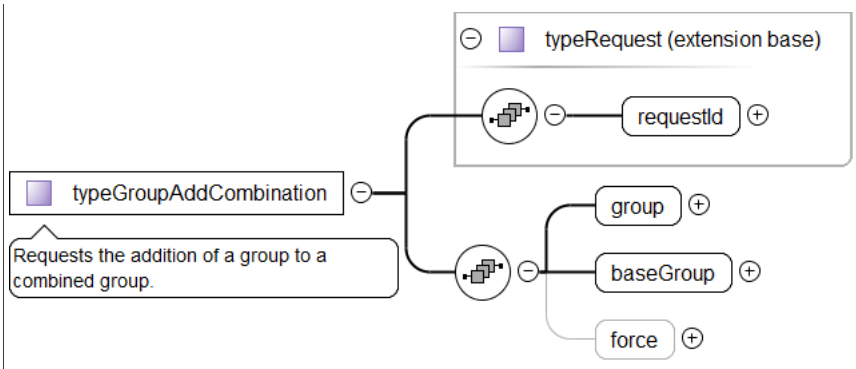
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeGroupAddRadioMember
Used by	Element interfaceGroup/addRadioMember
Model	requestId , radio , group , membership{0,1}
Children	group, membership, radio, requestId
Source	<xs:complexType name="typeGroupAddRadioMember"> <xs:annotation> <xs:documentation>Requests the addition of a radio subscriber to a group. This might cause DGNA operation in the air interface.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="membership" type="typeMembershipType" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeGroupRemoveRadioMember`

Namespace	DR-GW
Annotations	Requests removing a radio subscriber from a group. This might cause DGNA operation in the air interface.
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeGroupRemoveRadioMember
Used by	Element interfaceGroup/removeRadioMember
Model	requestId , radio , group
Children	group, radio, requestId
Source	<xs:complexType name="typeGroupRemoveRadioMember"> <xs:annotation> <xs:documentation>Requests removing a radio subscriber from a group. This might cause DGNA operation in the air interface.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="group" type="typeSubscriberAddress"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

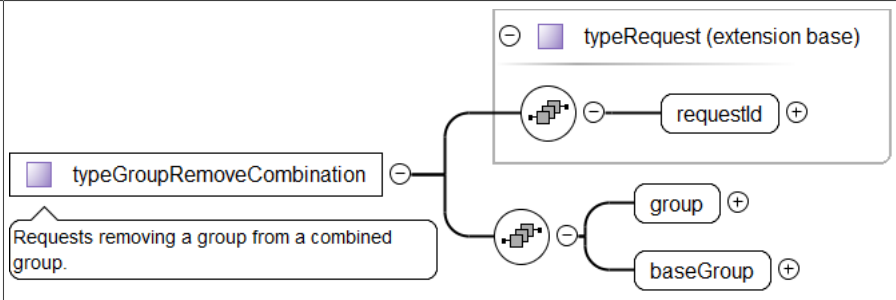
Complex Type `typeGroupAddCombination`

Namespace	DR-GW
Annotations	Requests the addition of a group to a combined group.

Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeGroupAddCombination
Used by	Element interfaceGroup/addCombination
Model	requestId , group , baseGroup , force{0,1}
Children	baseGroup, force, group, requestId

Source	`<xs:complexType name="typeGroupAddCombination">`
	`<xs:annotation>`
	`<xs:documentation>Requests the addition of a group to a combined group.</xs:documentation>`
	`</xs:annotation>`
	`<xs:complexContent>`
	`<xs:extension base="typeRequest">`
	`<xs:sequence>`
	`<xs:element name="group" type="typeSubscriberAddress"/>`
	`<xs:element name="baseGroup" type="typeSubscriberAddress"/>`
	`<xs:element name="force" type="xs:boolean" minOccurs="0" default="true"/>`
	`</xs:sequence>`
	`</xs:extension>`
	`</xs:complexContent>`
	`</xs:complexType>`

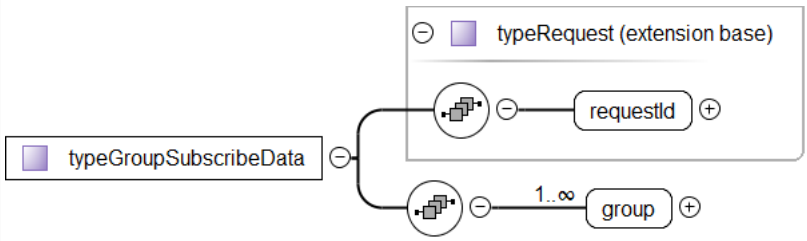
Complex Type `typeGroupRemoveCombination`

Namespace	DR-GW
Annotations	`Requests removing a group from a combined group.`
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeGroupRemoveCombination
Used by	Element interfaceGroup/removeCombination
Model	requestId , group , baseGroup
Children	baseGroup, group, requestId

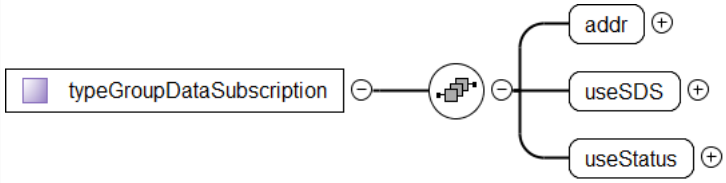
Source	`<xs:complexType name="typeGroupRemoveCombination">`
	`<xs:annotation>`
	`<xs:documentation>Requests removing a group from a combined group.</xs:documentation>`
	`</xs:annotation>`
	`<xs:complexContent>`
	`<xs:extension base="typeRequest">`
	`<xs:sequence>`
	`<xs:element name="group" type="typeSubscriberAddress"/>`
	`<xs:element name="baseGroup" type="typeSubscriberAddress"/>`

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

Complex Type `typeGroupSubscribeData`

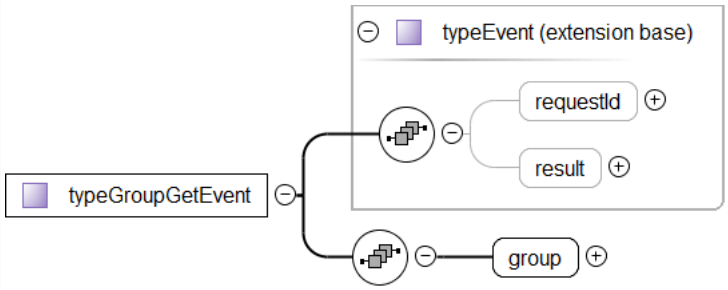
Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeGroupSubscribeData
Used by	Element interfaceGroup/subscribeData
Model	requestId , group+
Children	group, requestId
Source	<xs:complexType name="typeGroupSubscribeData"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="group" type="typeGroupDataSubscription" maxOccurs="unbounded"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeGroupDataSubscription`

Namespace	DR-GW
Annotations	
Diagram	
Used by	Elements typeGroupSubscribeData/group, typeGroupSubscribeDataEvent/group
Model	addr , useSDS , useStatus
Children	addr, useSDS, useStatus
Source	<xs:complexType name="typeGroupDataSubscription"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="addr" type="typeSubscriberAddress"/> <xs:element name="useSDS" type="xs:boolean"/> <xs:element name="useStatus" type="xs:boolean"/> </xs:sequence> </xs:complexType>

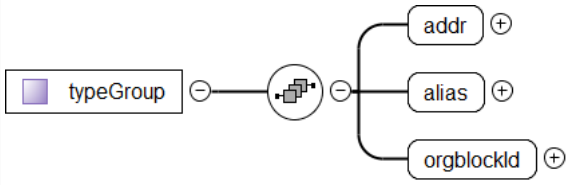
Complex Type `typeGroupGetEvent`

Namespace	DR-GW

Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupGetEvent
Used by	Element interfaceGroup/getEvent
Model	requestId{0,1} , result{0,1} , group
Children	group, requestId, result

Source	`<xs:complexType name="typeGroupGetEvent">`
	`<xs:annotation>`
	`<xs:documentation/>`
	`</xs:annotation>`
	`<xs:complexContent>`
	`<xs:extension base="typeEvent">`
	`<xs:sequence>`
	`<xs:element name="group" type="typeGroup"/>`
	`</xs:sequence>`
	`</xs:extension>`
	`</xs:complexContent>`
	`</xs:complexType>`

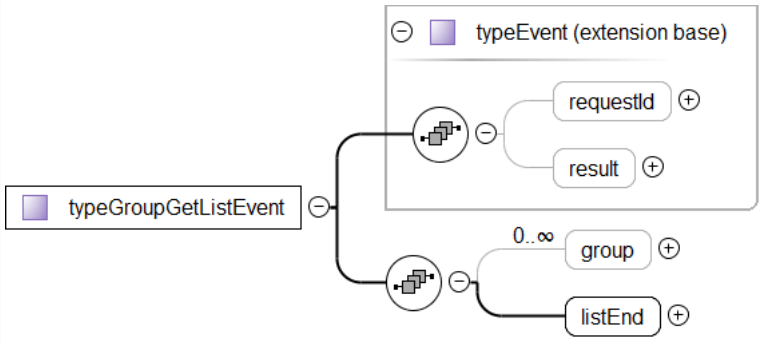
Complex Type `typeGroup`

Namespace	DR-GW
Annotations	
Diagram	
Used by	Elements typeGroupEvent/group, typeGroupGetEvent/group, typeGroupGetListEvent/group
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId

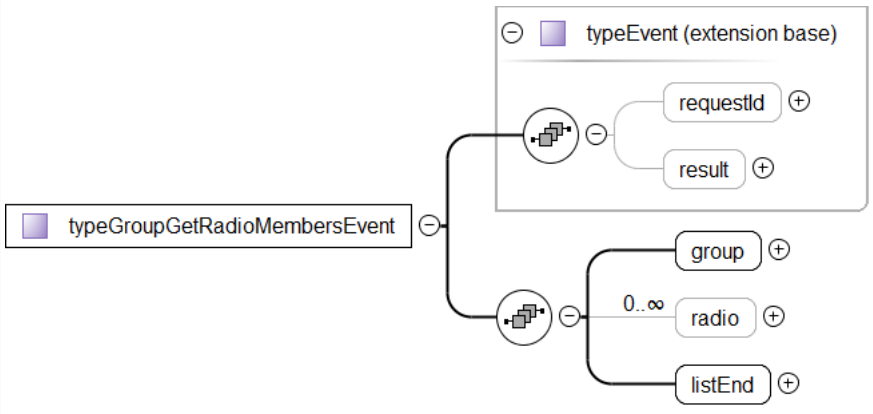
Source	`<xs:complexType name="typeGroup">`
	`<xs:annotation>`
	`<xs:documentation/>`
	`</xs:annotation>`
	`<xs:sequence>`
	`<xs:element name="addr" type="typeSubscriberAddress"/>`
	`<xs:element name="alias" type="xs:normalizedString"/>`
	`<xs:element name="orgblockId" type="typeOrganisationBlockId"/>`
	`</xs:sequence>`
	`</xs:complexType>`

Complex Type `typeGroupGetListEvent`

Namespace	DR-GW
Annotations	

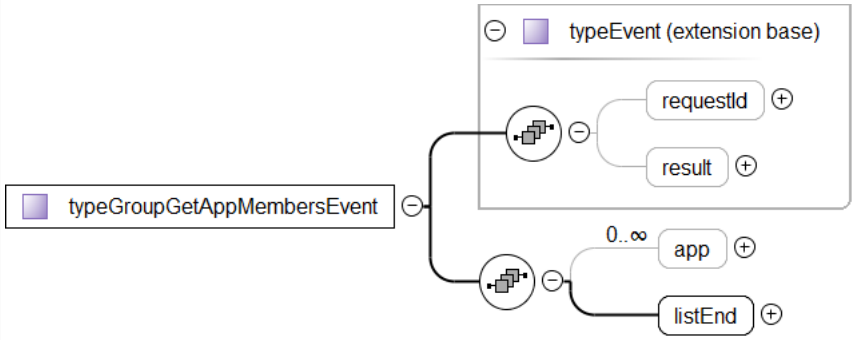
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupGetListEvent
Used by	Element interfaceGroup/getListEvent
Model	requestId{0,1} , result{0,1} , group* , listEnd
Children	group, listEnd, requestId, result
Source	<xs:complexType name="typeGroupGetListEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeGroup" minOccurs="0" maxOccurs="unbounded"/> <xs:element name="listEnd" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeGroupGetRadioMembersEvent`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupGetRadioMembersEvent
Used by	Element interfaceGroup/getRadioMembersEvent
Model	requestId{0,1} , result{0,1} , group , radio* , listEnd
Children	group, listEnd, radio, requestId, result
Source	<xs:complexType name="typeGroupGetRadioMembersEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent>

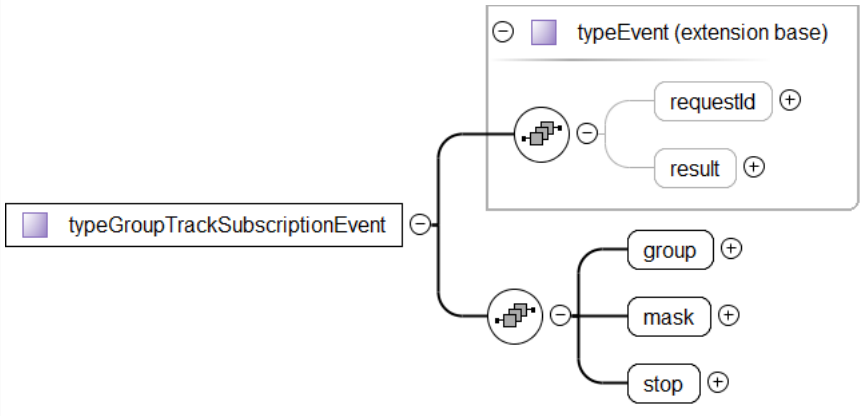
```
    <xs:extension base="typeEvent">
      <xs:sequence>
        <xs:element name="group" type="typeSubscriberAddress"/>
        <xs:element name="radio" type="typeSubscriberAddress" minOccurs="0" maxOccurs="unbounded"/>
        <xs:element name="listEnd" type="xs:boolean"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeGroupGetAppMembersEvent`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupGetAppMembersEvent
Used by	Element interfaceGroup/getAppMembersEvent
Model	requestId{0,1} , result{0,1} , app* , listEnd
Children	app, listEnd, requestId, result

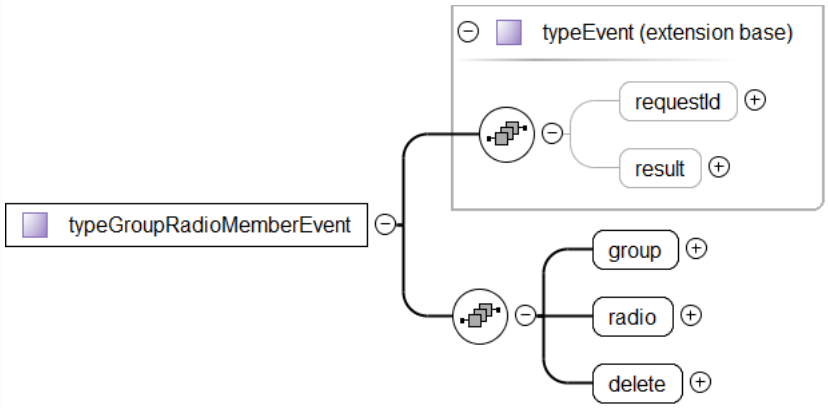
Source	`<xs:complexType name="typeGroupGetAppMembersEvent">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="app" type="typeSubscriberAddress" minOccurs="0" maxOccurs="unbounded"/>`
	`  <xs:element name="listEnd" type="xs:boolean"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Complex Type `typeGroupTrackSubscriptionEvent`

Namespace	DR-GW
Annotations	
Diagram	

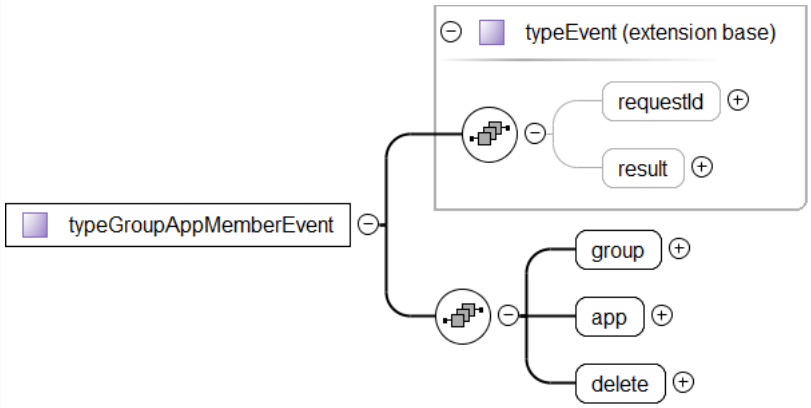
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupTrackSubscriptionEvent
Used by	Element interfaceGroup/trackSubscriptionEvent
Model	requestId{0,1} , result{0,1} , group , mask , stop
Children	group, mask, requestId, result, stop
Source	<xs:complexType name="typeGroupTrackSubscriptionEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="mask" type="typeGroupTrackingMask"/> <xs:element name="stop" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type typeGroupRadioMemberEvent

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupRadioMemberEvent
Used by	Element interfaceGroup/radioMemberEvent
Model	requestId{0,1} , result{0,1} , group , radio , delete
Children	delete, group, radio, requestId, result
Source	<xs:complexType name="typeGroupRadioMemberEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="delete" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

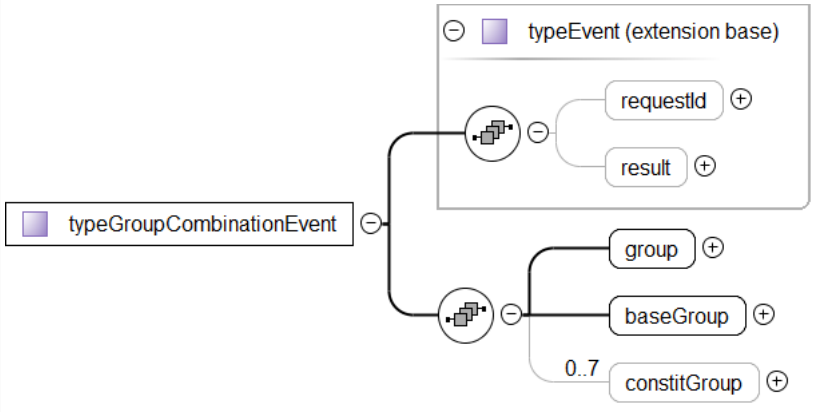
Complex Type typeGroupAppMemberEvent

Namespace	DR-GW
Annotations	

Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupAppMemberEvent
Used by	Element interfaceGroup/appMemberEvent
Model	requestId{0,1} , result{0,1} , group , app , delete
Children	app, delete, group, requestId, result

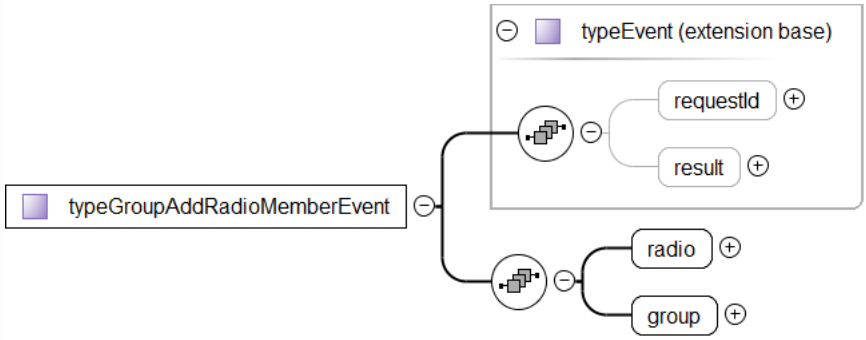
Source	`<xs:complexType name="typeGroupAppMemberEvent">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="group" type="typeSubscriberAddress"/>`
	`  <xs:element name="app" type="typeSubscriberAddress"/>`
	`  <xs:element name="delete" type="xs:boolean"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Complex Type `typeGroupCombinationEvent`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupCombinationEvent
Used by	Element interfaceGroup/combinationEvent
Model	requestId{0,1} , result{0,1} , group , baseGroup , constitGroup{0,7}
Children	baseGroup, constitGroup, group, requestId, result
Source	`<xs:complexType name="typeGroupCombinationEvent">`

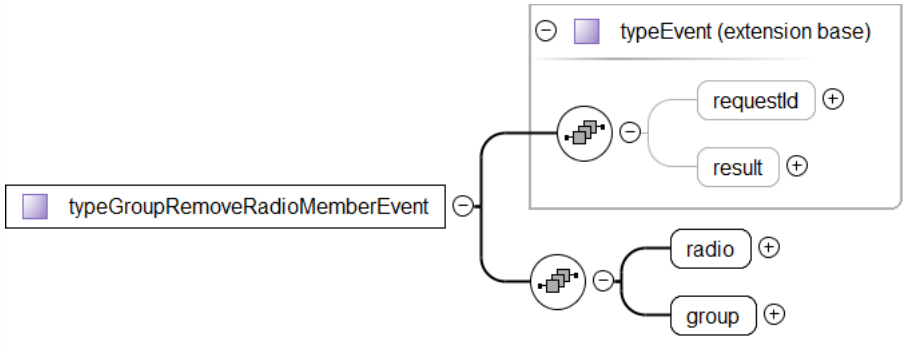
```
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="typeEvent">
      <xs:sequence>
        <xs:element name="group" type="typeSubscriberAddress"/>
        <xs:element name="baseGroup" type="typeSubscriberAddress"/>
        <xs:element name="constitGroup" type="typeSubscriberAddress" minOccurs="0" maxOccurs="7"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeGroupAddRadioMemberEvent`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupAddRadioMemberEvent
Used by	Element interfaceGroup/addRadioMemberEvent
Model	requestId{0,1} , result{0,1} , radio , group
Children	group, radio, requestId, result

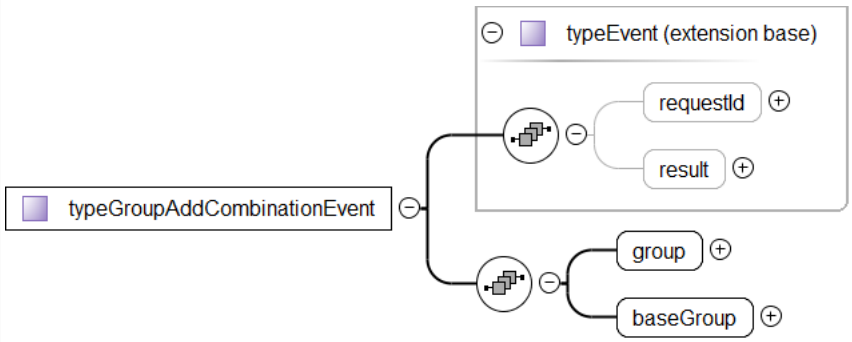
Source	`<xs:complexType name="typeGroupAddRadioMemberEvent">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="radio" type="typeSubscriberAddress"/>`
	`  <xs:element name="group" type="typeSubscriberAddress"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Complex Type `typeGroupRemoveRadioMemberEvent`

Namespace	DR-GW
Annotations	
Diagram	

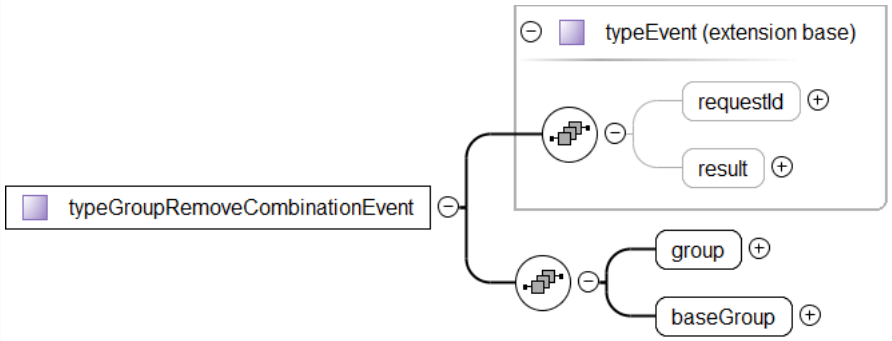
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupRemoveRadioMemberEvent
Used by	Element interfaceGroup/removeRadioMemberEvent
Model	requestId{0,1} , result{0,1} , radio , group
Children	group, radio, requestId, result
Source	<xs:complexType name="typeGroupRemoveRadioMemberEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="group" type="typeSubscriberAddress"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type typeGroupAddCombinationEvent

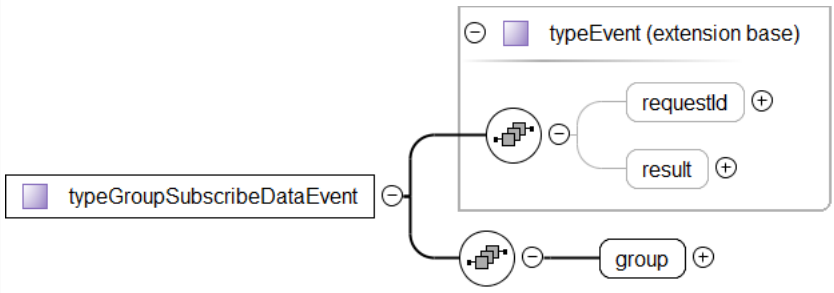
Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupAddCombinationEvent
Used by	Element interfaceGroup/addCombinationEvent
Model	requestId{0,1} , result{0,1} , group , baseGroup
Children	baseGroup, group, requestId, result
Source	<xs:complexType name="typeGroupAddCombinationEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="baseGroup" type="typeSubscriberAddress"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type typeGroupRemoveCombinationEvent

Namespace	DR-GW
Annotations	

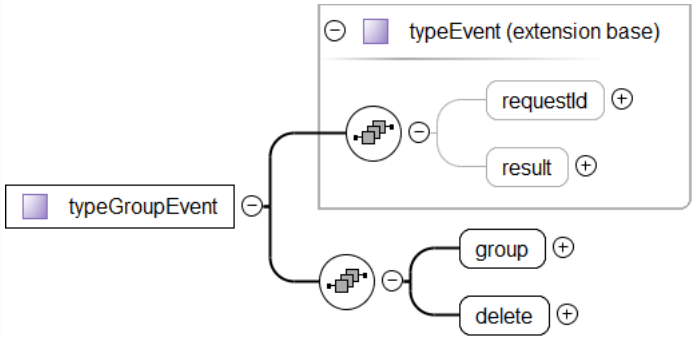
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupRemoveCombinationEvent
Used by	Element interfaceGroup/removeCombinationEvent
Model	requestId{0,1} , result{0,1} , group , baseGroup
Children	baseGroup, group, requestId, result
Source	<xs:complexType name="typeGroupRemoveCombinationEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="baseGroup" type="typeSubscriberAddress"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeGroupSubscribeDataEvent`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupSubscribeDataEvent
Used by	Element interfaceGroup/subscribeDataEvent
Model	requestId{0,1} , result{0,1} , group
Children	group, requestId, result
Source	<xs:complexType name="typeGroupSubscribeDataEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeGroupDataSubscription"/> </xs:sequence> </xs:extension>

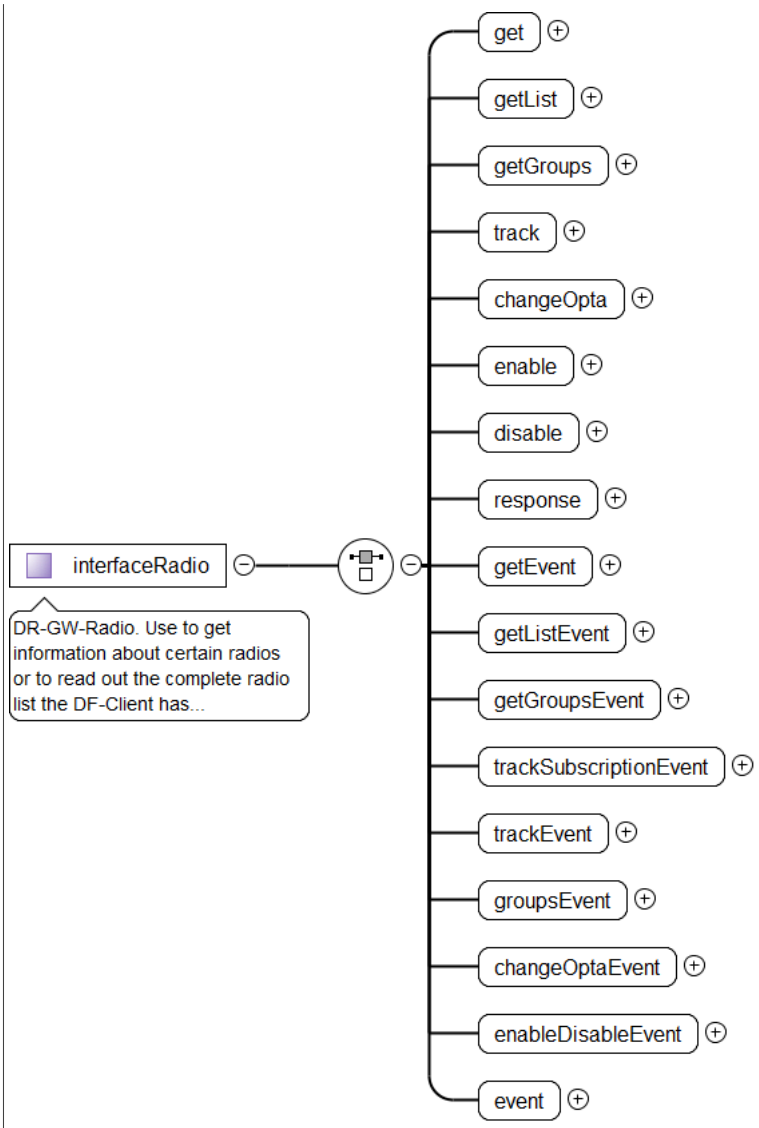
```
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeGroupEvent`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupEvent
Used by	Element interfaceGroup/event
Model	requestId{0,1} , result{0,1} , group , delete
Children	delete, group, requestId, result
Source	<xs:complexType name="typeGroupEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeGroup"/> <xs:element name="delete" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

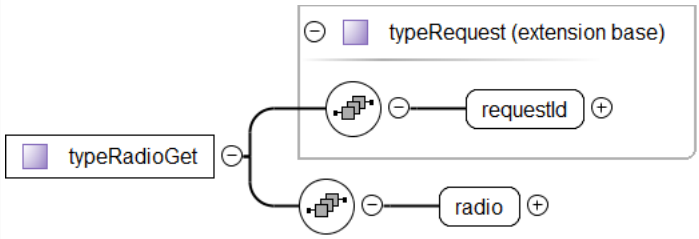
Complex Type `interfaceRadio`

Namespace	DR-GW
Annotations	DR-GW-Radio. Use to get information about certain radios or to read out the complete radio list the DF-Client has rights for. There are also many other methods to execute over a certain radio. See each method description. Use only via SOAP.

Diagram	
Used by	Element drgw/radio
Model	get | getList | getGroups | track | changeOpta | enable | disable | response | getEvent | getListEvent | getGroupsEvent | trackSubscrip-tionEvent | trackEvent | groupsEvent | changeOptaEvent | enableDisableEvent | event
Children	changeOpta, changeOptaEvent, disable, enable, enableDisableEvent, event, get, getEvent, getGroups, getGroupsEvent, getList, getListEvent, groupsEvent, response, track, trackEvent, trackSubscriptionEvent
Source	<xs:complexType name="interfaceRadio"> <xs:annotation> <xs:documentation>DR-GW-Radio. Use to get information about certain radios or to read out the complete radio list the DF-Client has rights for. There are also many other methods to execute over a certain radio. See each method description. Use only via SOAP.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="get" type="typeRadioGet"/> <xs:element name="getList" type="typeRadioGetList"/> <xs:element name="getGroups" type="typeRadioGetGroups"/> <xs:element name="track" type="typeRadioTrack"/> <xs:element name="changeOpta" type="typeRadioChangeOpta"/> <xs:element name="enable" type="typeRadioEnable"/> <xs:element name="disable" type="typeRadioDisable"/> <xs:element name="response" type="typeResponse"/> <xs:element name="getEvent" type="typeRadioGetEvent"/> <xs:element name="getListEvent" type="typeRadioGetListEvent"/> <xs:element name="getGroupsEvent" type="typeRadioGetGroupsEvent"/> <xs:element name="trackSubscriptionEvent" type="typeRadioTrackSubscriptionEvent"/> <xs:element name="trackEvent" type="typeRadioTrackEvent"/> <xs:element name="groupsEvent" type="typeRadioGroupsEvent"/> <xs:element name="changeOptaEvent" type="typeRadioChangeOptaEvent"/> <xs:element name="enableDisableEvent" type="typeRadioEnableDisableEvent"/> <xs:element name="event" type="typeRadioEvent"/>

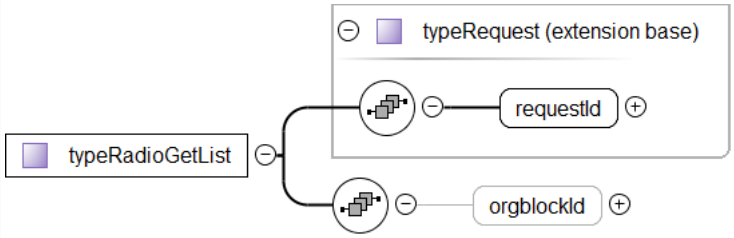
```
  </xs:choice>
</xs:complexType>
```

Complex Type `typeRadioGet`

Namespace	DR-GW
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeRadioGet
Used by	Element interfaceRadio/get
Model	requestId , radio
Children	radio, requestId

Source	`<xs:complexType name="typeRadioGet">`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="radio" type="typeSubscriberAddress"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

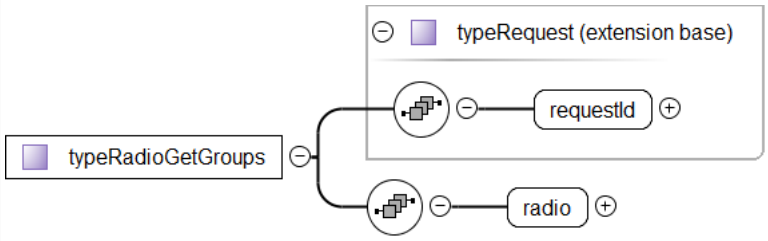
Complex Type `typeRadioGetList`

Namespace	DR-GW
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeRadioGetList
Used by	Element interfaceRadio/getList
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId

Source	`<xs:complexType name="typeRadioGetList">`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

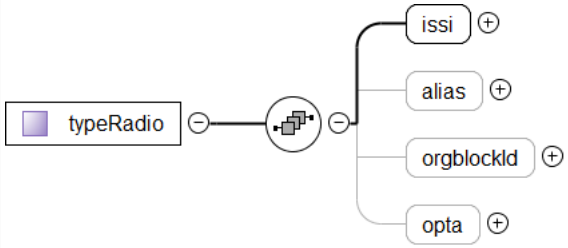
Complex Type `typeRadioGetGroups`

Namespace	DR-GW

Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeRadioGetGroups
Used by	Element interfaceRadio/getGroups
Model	requestId , radio
Children	radio, requestId

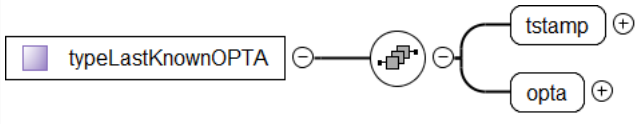
Source	`<xs:complexType name="typeRadioGetGroups">`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="radio" type="typeRadio"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Complex Type `typeRadio`

Namespace	DR-GW
Annotations	
Diagram	
Used by	Elements typeRadioEvent/radio, typeRadioGetEvent/radio, typeRadioGetGroups/radio, typeRadioGetListEvent/radio
Model	issi , alias{0,1} , orgblockId{0,1} , opta{0,1}
Children	alias, issi, opta, orgblockId

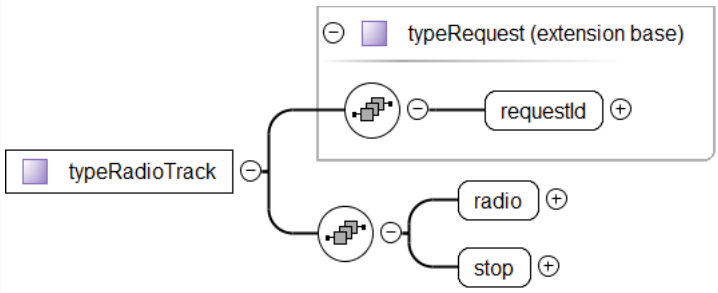
Source	`<xs:complexType name="typeRadio">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="issi" type="typeSubscriberAddress"/>`
	`  <xs:element name="alias" type="xs:normalizedString" minOccurs="0"/>`
	`  <xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/>`
	`  <xs:element name="opta" type="typeLastKnownOPTA" minOccurs="0"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `typeLastKnownOPTA`

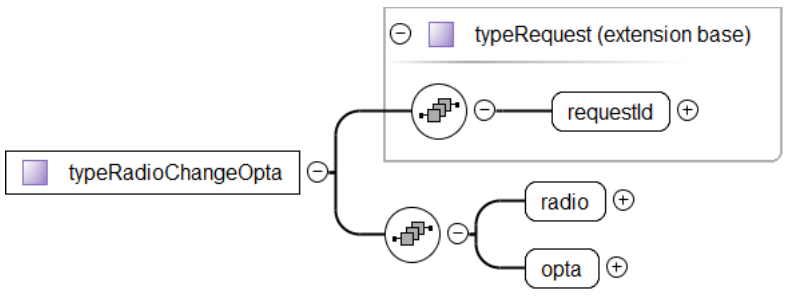
Namespace	DR-GW
Annotations	
Diagram	

Used by	Element typeRadio/opta
Model	tstamp , opta
Children	opta, tstamp
Source	<xs:complexType name="typeLastKnownOPTA"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="tstamp" type="xs:dateTime"/> <xs:element name="opta" type="typeOpta"/> </xs:sequence> </xs:complexType>

Complex Type `typeRadioTrack`

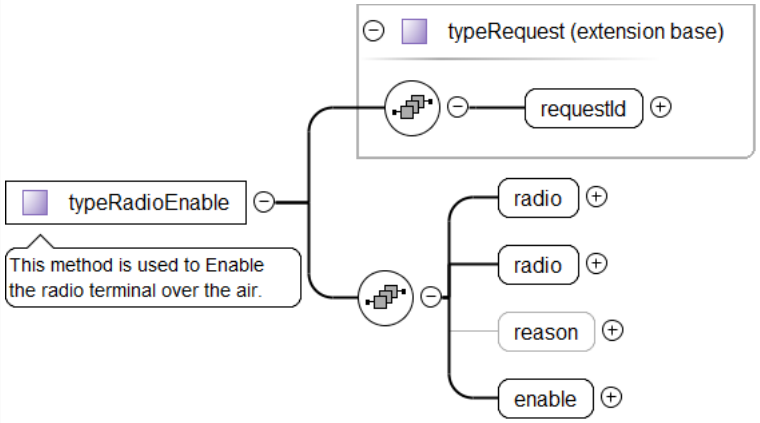
Namespace	DR-GW
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeRadioTrack
Used by	Element interfaceRadio/track
Model	requestId , radio , stop
Children	radio, requestId, stop
Source	<xs:complexType name="typeRadioTrack"> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="stop" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeRadioChangeOpta`

Namespace	DR-GW
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeRadioChangeOpta
Used by	Element interfaceRadio/changeOpta

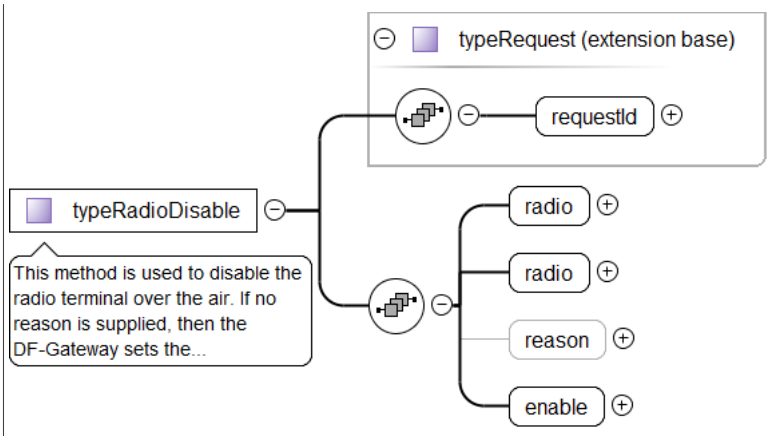
Model	requestId , radio , opta
Children	opta, radio, requestId
Source	<xs:complexType name="typeRadioChangeOpta"> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="opta" type="typeOpta"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeRadioEnable`

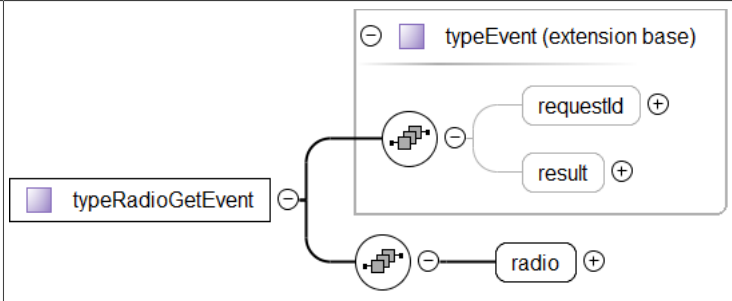
Namespace	DR-GW
Annotations	This method is used to Enable the radio terminal over the air.
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeRadioEnable
Used by	Element interfaceRadio/enable
Model	requestId , radio , radio , reason{0,1} , enable
Children	enable, radio, reason, requestId
Source	<xs:complexType name="typeRadioEnable"> <xs:annotation> <xs:documentation>This method is used to Enable the radio terminal over the air.</ xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="reason" type="xs:unsignedByte" minOccurs="0"/> <xs:element name="enable" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeRadioDisable`

Namespace	DR-GW
Annotations	This method is used to disable the radio terminal over the air. If no reason is supplied, then the DF-Gateway sets the default reason. There is no default reason value, it depends on the DF-Gateway configuration what reason is used when no reason is supplied by DF-Client. See TCS API Description for all possible reasons for disabling.

Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeRadioDisable
Used by	Element interfaceRadio/disable
Model	requestId , radio , radio , reason{0,1} , enable
Children	enable, radio, reason, requestId
Source	`<xs:complexType name="typeRadioDisable">` `  <xs:annotation>` ` <xs:documentation>This method is used to disable the radio terminal over the air. If no reason is supplied, then the DF-Gateway sets the default reason. There is no default reason value, it depends on the DF-Gateway configuration what reason is used when no reason is supplied by DF-Client. See TCS API Description for all possible reasons for disabling.</xs:documentation>` ` </xs:annotation>` ` <xs:complexContent>` ` <xs:extension base="typeRequest">` ` <xs:sequence>` ` <xs:element name="radio" type="typeSubscriberAddress"/>` ` <xs:element name="radio" type="typeSubscriberAddress"/>` ` <xs:element name="reason" type="xs:unsignedByte" minOccurs="0"/>` ` <xs:element name="enable" type="xs:boolean"/>` ` </xs:sequence>` ` </xs:extension>` ` </xs:complexContent>` `</xs:complexType>`

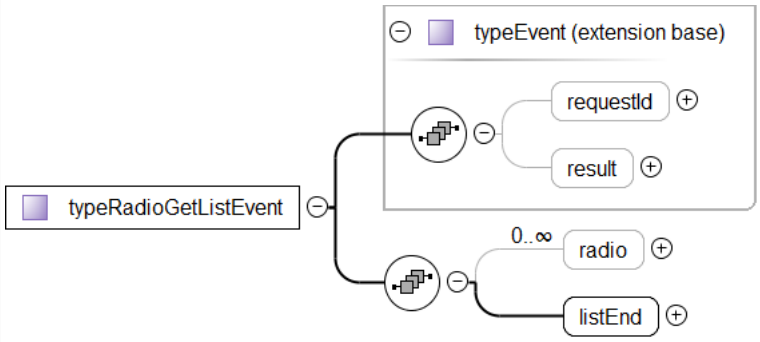
Complex Type `typeRadioGetEvent`

Namespace	DR-GW
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeRadioGetEvent
Used by	Element interfaceRadio/getEvent
Model	requestId{0,1} , result{0,1} , radio
Children	radio, requestId, result

Source	`<xs:complexType name="typeRadioGetEvent">`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`

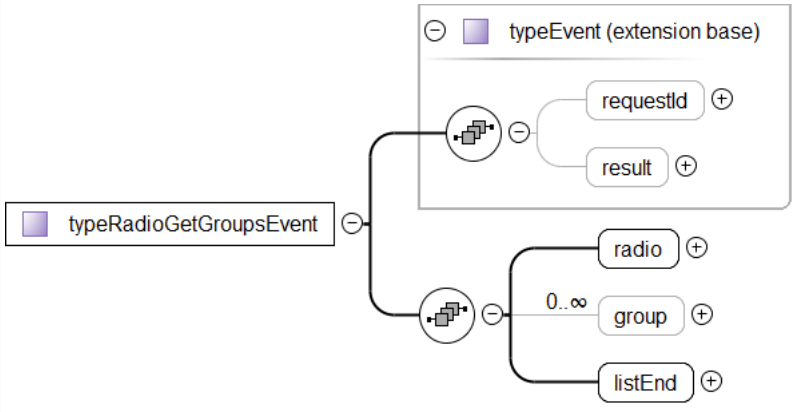
```
        <xs:element name="radio" type="typeRadio"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeRadioGetListEvent`

Namespace	DR-GW
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeRadioGetListEvent
Used by	Element interfaceRadio/getListEvent
Model	requestId{0,1} , result{0,1} , radio* , listEnd
Children	listEnd, radio, requestId, result

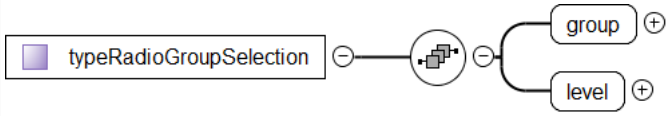
Source	`<xs:complexType name="typeRadioGetListEvent">`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="radio" type="typeRadio" minOccurs="0" maxOccurs="unbounded"/>`
	`  <xs:element name="listEnd" type="xs:boolean"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Complex Type `typeRadioGetGroupsEvent`

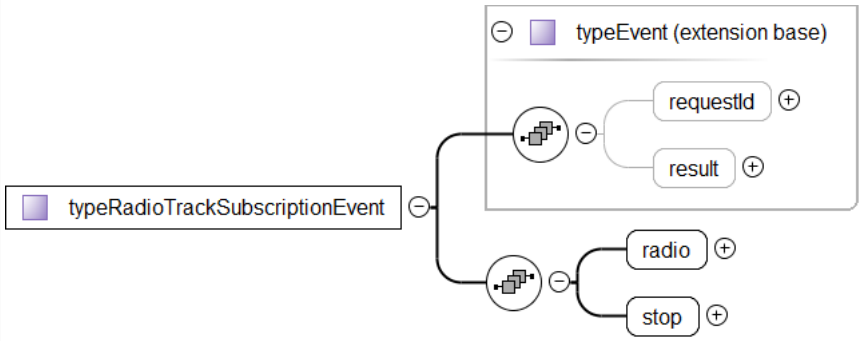
Namespace	DR-GW
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeRadioGetGroupsEvent
Used by	Element interfaceRadio/getGroupsEvent
Model	requestId{0,1} , result{0,1} , radio , group* , listEnd
Children	group, listEnd, radio, requestId, result

Source	<xs:complexType name="typeRadioGetGroupsEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="group" type="typeRadioGroupSelection" minOccurs="0" maxOccurs="unbounded"/ > <xs:element name="listEnd" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

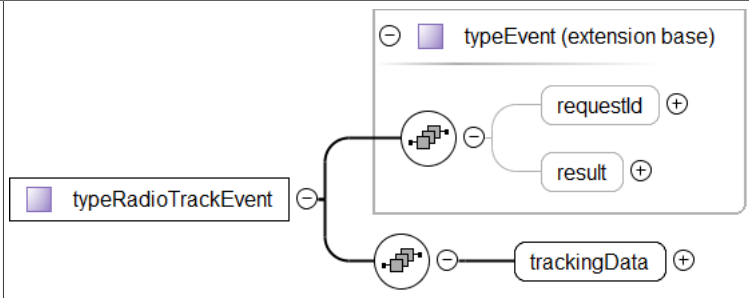
Complex Type `typeRadioGroupSelection`

Namespace	DR-GW
Annotations	
Diagram	
Used by	Elements typeRadioGetGroupsEvent/group, typeRadioGroupsEvent/group
Model	group , level
Children	group, level
Source	<xs:complexType name="typeRadioGroupSelection"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:element name="level" type="typeGroupSelectionLevel"/> </xs:sequence> </xs:complexType>

Complex Type `typeRadioTrackSubscriptionEvent`

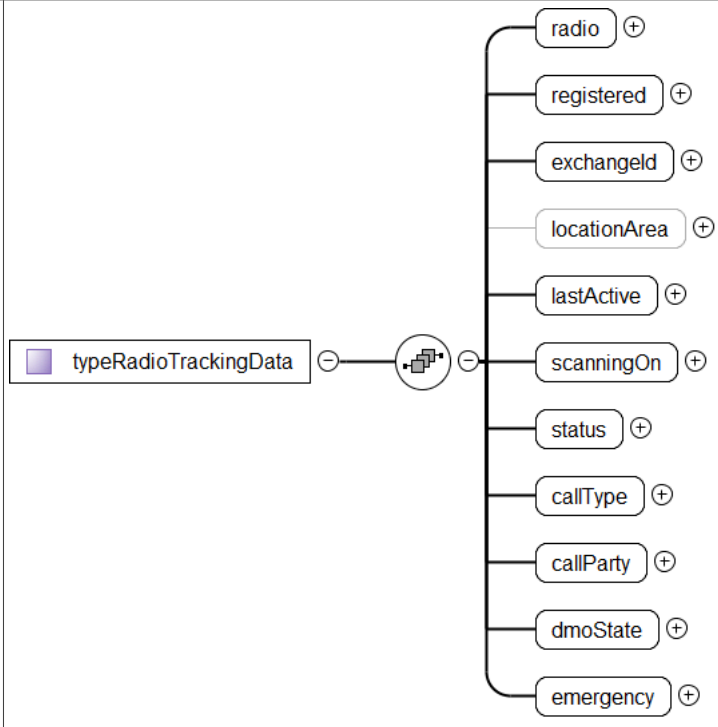
Namespace	DR-GW
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeRadioTrackSubscriptionEvent
Used by	Element interfaceRadio/trackSubscriptionEvent
Model	requestId{0,1} , result{0,1} , radio , stop
Children	radio, requestId, result, stop
Source	<xs:complexType name="typeRadioTrackSubscriptionEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="stop" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeRadioTrackEvent`

Namespace	DR-GW
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeRadioTrackEvent
Used by	Element interfaceRadio/trackEvent
Model	requestId{0,1} , result{0,1} , trackingData
Children	requestId, result, trackingData

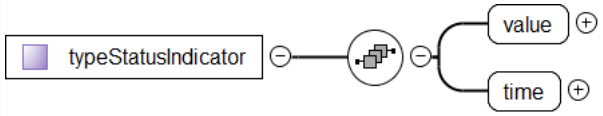
Source	`<xs:complexType name="typeRadioTrackEvent">`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="trackingData" type="typeRadioTrackingData"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Complex Type `typeRadioTrackingData`

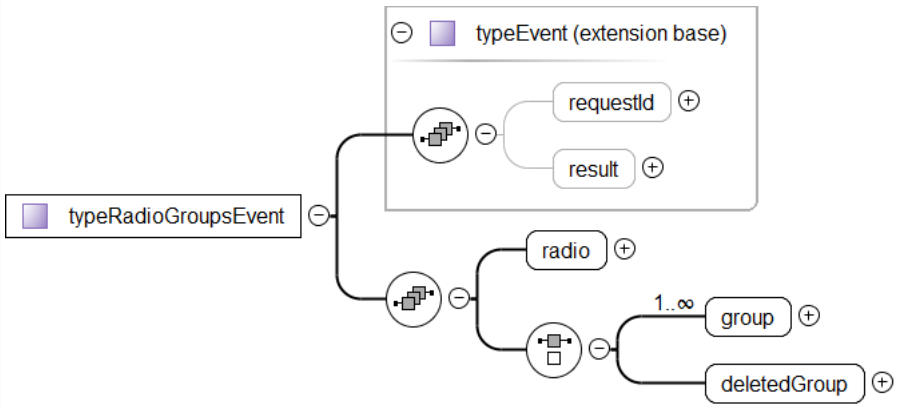
Namespace	DR-GW
Annotations	
Diagram	
Used by	Element typeRadioTrackEvent/trackingData
Model	radio , registered , exchangeId , locationArea{0,1} , lastActive , scanningOn , status , callType , callParty , dmoState , emergency

Children	callParty, callType, dmoState, emergency, exchangeId, lastActive, locationArea, radio, registered, scanningOn, status
Source	<xs:complexType name="typeRadioTrackingData"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="registered" type="xs:boolean"/> <xs:element name="exchangeId" type="xs:unsignedLong"/> <xs:element name="locationArea" type="xs:unsignedShort" minOccurs="0"/> <xs:element name="lastActive" type="xs:dateTime"/> <xs:element name="scanningOn" type="xs:boolean"/> <xs:element name="status" type="typeStatusIndicator"/> <xs:element name="callType" type="typeCallType"/> <xs:element name="callParty" type="typeSubscriberAddress"/> <xs:element name="dmoState" type="xs:boolean"/> <xs:element name="emergency" type="xs:boolean"/> </xs:sequence> </xs:complexType>

Complex Type `typeStatusIndicator`

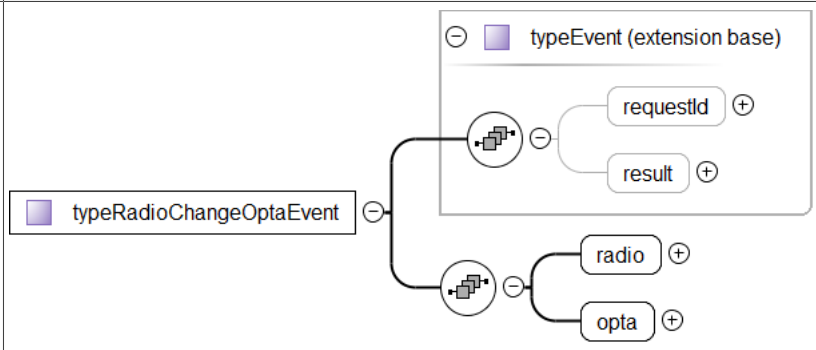
Namespace	DR-GW
Annotations	
Diagram	
Used by	Element typeRadioTrackingData/status
Model	value , time
Children	time, value
Source	<xs:complexType name="typeStatusIndicator"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:sequence> <xs:element name="value" type="xs:unsignedLong"/> <xs:element name="time" type="xs:dateTime"/> </xs:sequence> </xs:complexType>

Complex Type `typeRadioGroupsEvent`

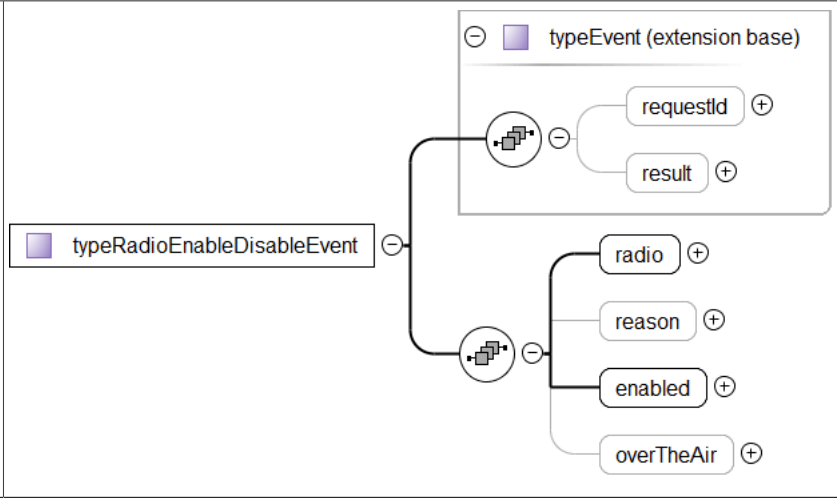
Namespace	DR-GW
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeRadioGroupsEvent
Used by	Element interfaceRadio/groupsEvent
Model	requestId{0,1} , result{0,1} , radio , (group+ | deletedGroup)
Children	deletedGroup, group, radio, requestId, result

Source	<xs:complexType name="typeRadioGroupsEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:choice> <xs:element name="group" type="typeRadioGroupSelection" maxOccurs="unbounded"/> <xs:element name="deletedGroup" type="typeSubscriberAddress"/> </xs:choice> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeRadioChangeOptaEvent`

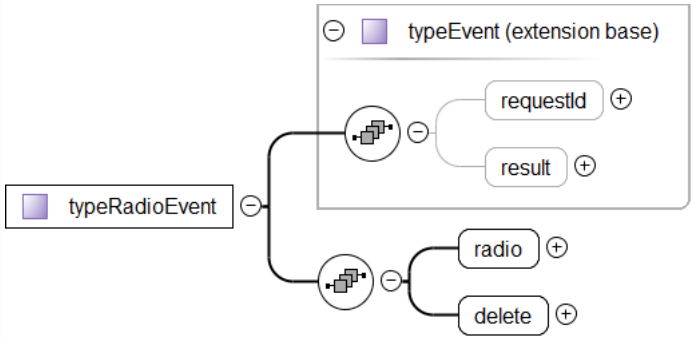
Namespace	DR-GW
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeRadioChangeOptaEvent
Used by	Element interfaceRadio/changeOptaEvent
Model	requestId{0,1} , result{0,1} , radio , opta
Children	opta, radio, requestId, result
Source	<xs:complexType name="typeRadioChangeOptaEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="opta" type="typeOpta"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeRadioEnableDisableEvent`

Namespace	DR-GW
Diagram	

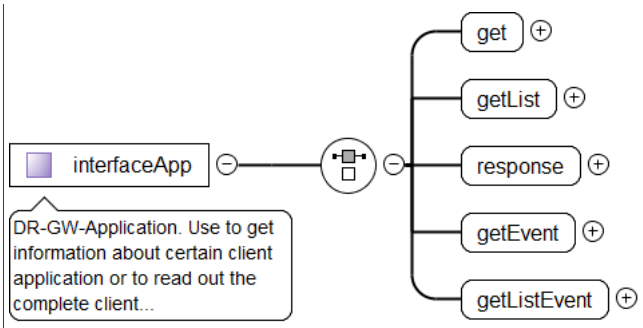
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeRadioEnableDisableEvent
Used by	Element interfaceRadio/enableDisableEvent
Model	requestId{0,1} , result{0,1} , radio , reason{0,1} , enabled , overTheAir{0,1}
Children	enabled, overTheAir, radio, reason, requestId, result
Source	<xs:complexType name="typeRadioEnableDisableEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeSubscriberAddress"/> <xs:element name="reason" type="xs:unsignedByte" minOccurs="0"/> <xs:element name="enabled" type="xs:boolean"/> <xs:element name="overTheAir" type="xs:boolean" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeRadioEvent`

Namespace	DR-GW
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeRadioEvent
Used by	Element interfaceRadio/event
Model	requestId{0,1} , result{0,1} , radio , delete
Children	delete, radio, requestId, result
Source	<xs:complexType name="typeRadioEvent"> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="radio" type="typeRadio"/> <xs:element name="delete" type="xs:boolean"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

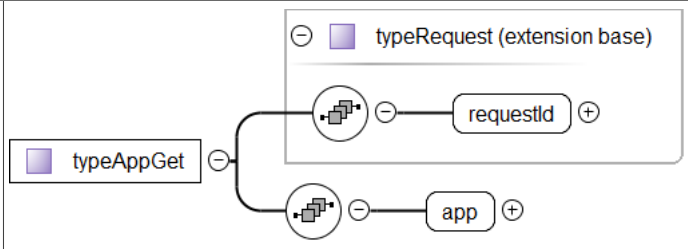
Complex Type `interfaceApp`

Namespace	DR-GW
Annotations	DR-GW-Application. Use to get information about certain client application or to read out the complete client application list the DF-Client has rights for. Use only via SOAP.

Diagram	
Used by	Element drgw/app
Model	get | getList | response | getEvent | getListEvent
Children	get, getEvent, getList, getListEvent, response

Source	`<xs:complexType name="interfaceApp">`
	`  <xs:annotation>`
	`  <xs:documentation>DR-GW-Application. Use to get information about certain client application or to read out the complete client application list the DF-Client has rights for. Use only via SOAP.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:choice>`
	`  <xs:element name="get" type="typeAppGet"/>`
	`  <xs:element name="getList" type="typeAppGetList"/>`
	`  <xs:element name="response" type="typeResponse"/>`
	`  <xs:element name="getEvent" type="typeAppGetEvent"/>`
	`  <xs:element name="getListEvent" type="typeAppGetListEvent"/>`
	`  </xs:choice>`
	`</xs:complexType>`

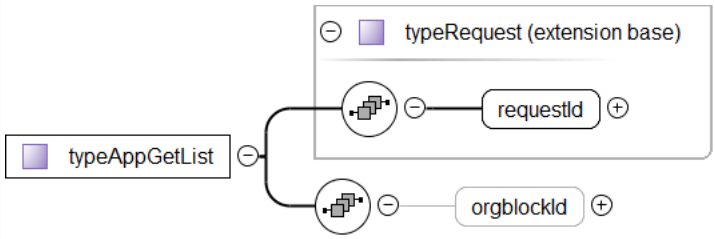
Complex Type `typeAppGet`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeAppGet
Used by	Element interfaceApp/get
Model	requestId , app
Children	app, requestId

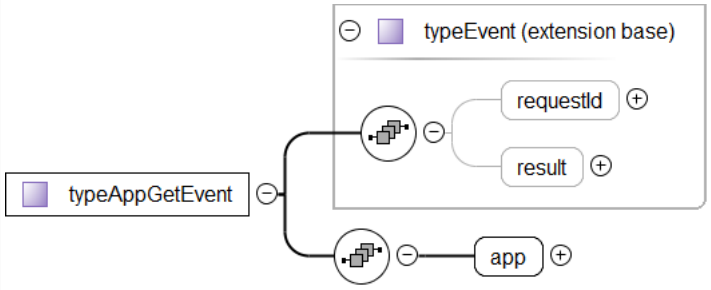
Source	`<xs:complexType name="typeAppGet">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeRequest">`
	`  <xs:sequence>`
	`  <xs:element name="app" type="typeSubscriberAddress"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Complex Type `typeAppGetList`

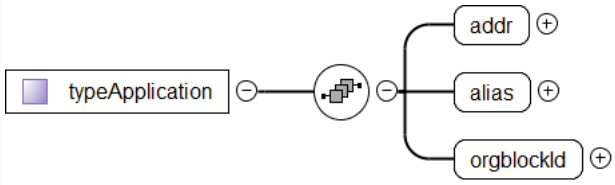
Namespace	DR-GW

Annotations	
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeAppGetList
Used by	Element interfaceApp/getList
Model	requestId , orgblockId{0,1}
Children	orgblockId, requestId
Source	<xs:complexType name="typeAppGetList"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeRequest"> <xs:sequence> <xs:element name="orgblockId" type="typeOrganisationBlockId" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeAppGetEvent`

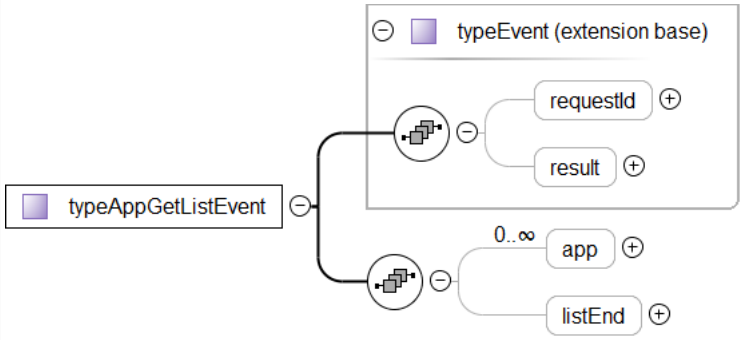
Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeAppGetEvent
Used by	Element interfaceApp/getEvent
Model	requestId{0,1} , result{0,1} , app
Children	app, requestId, result
Source	<xs:complexType name="typeAppGetEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="app" type="typeApplication" maxOccurs="1"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeApplication`

Namespace	DR-GW
Annotations	
Diagram	
Used by	Elements typeAppGetEvent/app, typeAppGetListEvent/app
Model	addr , alias , orgblockId
Children	addr, alias, orgblockId

Source	`<xs:complexType name="typeApplication">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="addr" type="typeSubscriberAddress"/>`
	`  <xs:element name="alias" type="xs:normalizedString"/>`
	`  <xs:element name="orgblockId" type="typeOrganisationBlockId"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

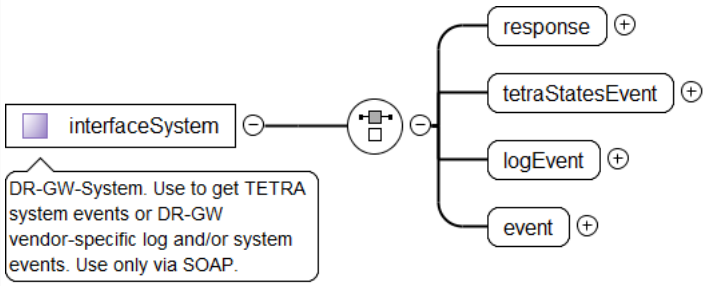
Complex Type `typeAppGetListEvent`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeAppGetListEvent
Used by	Element interfaceApp/getListEvent
Model	requestId{0,1} , result{0,1} , app* , listEnd{0,1}
Children	app, listEnd, requestId, result

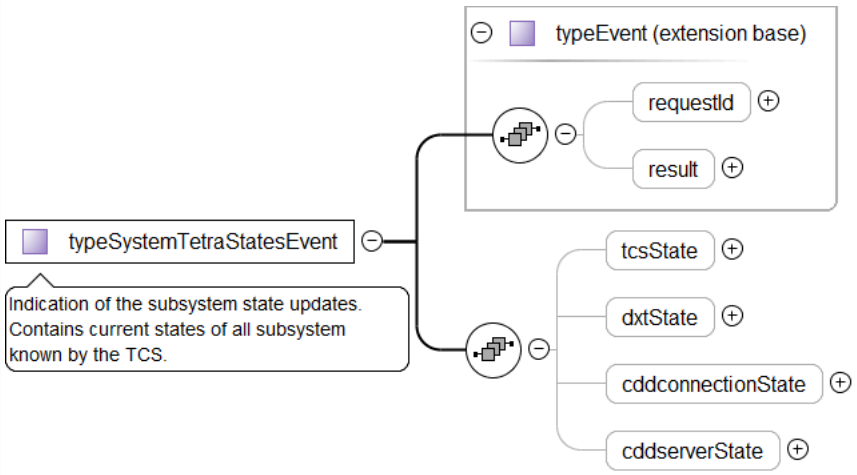
Source	`<xs:complexType name="typeAppGetListEvent">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="app" type="typeApplication" minOccurs="0" maxOccurs="unbounded"/>`
	`  <xs:element name="listEnd" type="xs:boolean" minOccurs="0"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

Complex Type `interfaceSystem`

Namespace	DR-GW

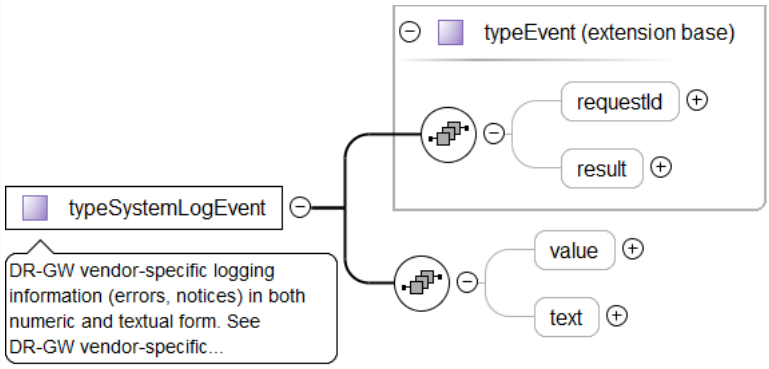
Annotations	DR-GW-System. Use to get TETRA system events or DR-GW vendor-specific log and/or system events. Use only via SOAP.
Diagram	
Used by	Element drgw/system
Model	response | tetraStatesEvent | logEvent | event
Children	event, logEvent, response, tetraStatesEvent
Source	<xs:complexType name="interfaceSystem"> <xs:annotation> <xs:documentation>DR-GW-System. Use to get TETRA system events or DR-GW vendor-specific log and/or system events. Use only via SOAP.</xs:documentation> </xs:annotation> <xs:choice> <xs:element name="response" type="typeResponse"/> <xs:element name="tetraStatesEvent" type="typeSystemTetraStatesEvent"/> <xs:element name="logEvent" type="typeSystemLogEvent"/> <xs:element name="event" type="typeSystemEvent"/> </xs:choice> </xs:complexType>

Complex Type typeSystemTetraStatesEvent

Namespace	DR-GW
Annotations	Indication of the subsystem state updates. Contains current states of all subsystem known by the TCS.
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeSystemTetraStatesEvent
Used by	Element interfaceSystem/tetraStatesEvent
Model	requestId{0,1} , result{0,1} , tcsState{0,1} , dxtState{0,1} , cddconnectionState{0,1} , cddserverState{0,1}
Children	cddconnectionState, cddserverState, dxtState, requestId, result, tcsState
Source	<xs:complexType name="typeSystemTetraStatesEvent"> <xs:annotation> <xs:documentation>Indication of the subsystem state updates. Contains current states of all subsystem known by the TCS.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence>

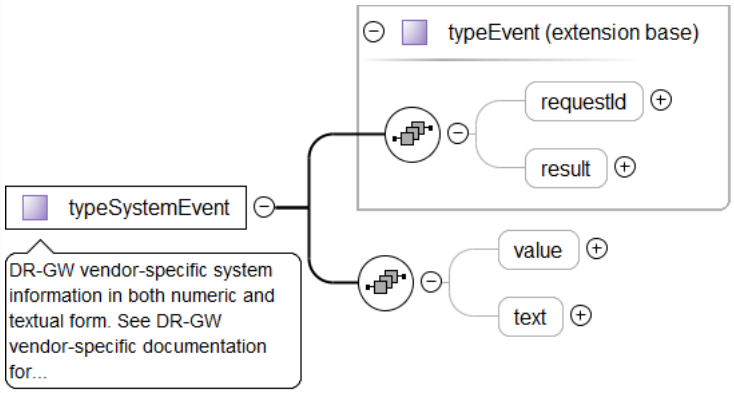
```
        <xs:element name="tcsState" type="typeSystemElementState" minOccurs="0"/>
        <xs:element name="dxtState" type="typeSystemElementState" minOccurs="0"/>
        <xs:element name="cddconnectionState" type="typeSystemElementState" minOccurs="0"/>
        <xs:element name="cddserverState" type="typeSystemElementState" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeSystemLogEvent`

Namespace	DR-GW
Annotations	DR-GW vendor-specific logging information (errors, notices) in both numeric and textual form. See DR-GW vendor-specific documentation for the details.
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeSystemLogEvent
Used by	Element interfaceSystem/logEvent
Model	requestId{0,1} , result{0,1} , value{0,1} , text{0,1}
Children	requestId, result, text, value
Source	<xs:complexType name="typeSystemLogEvent"> <xs:annotation> <xs:documentation>DR-GW vendor-specific logging information (errors, notices) in both numeric and textual form. See DR-GW vendor-specific documentation for the details.</xs:documentation> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="value" type="xs:hexBinary" minOccurs="0"/> <xs:element name="text" type="xs:normalizedString" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

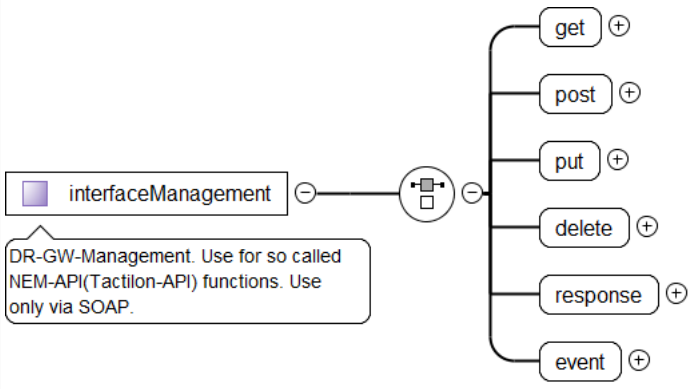
Complex Type `typeSystemEvent`

Namespace	DR-GW
Annotations	DR-GW vendor-specific system information in both numeric and textual form. See DR-GW vendor-specific documentation for the details.

Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeSystemEvent
Used by	Element interfaceSystem/event
Model	requestId{0,1} , result{0,1} , value{0,1} , text{0,1}
Children	requestId, result, text, value

Source	`<xs:complexType name="typeSystemEvent">`
	`  <xs:annotation>`
	`  <xs:documentation>DR-GW vendor-specific system information in both numeric and textual form. See DR-GW vendor-specific documentation for the details.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`
	`  <xs:sequence>`
	`  <xs:element name="value" type="xs:hexBinary" minOccurs="0"/>`
	`  <xs:element name="text" type="xs:normalizedString" minOccurs="0"/>`
	`  </xs:sequence>`
	`  </xs:extension>`
	`  </xs:complexContent>`
	`</xs:complexType>`

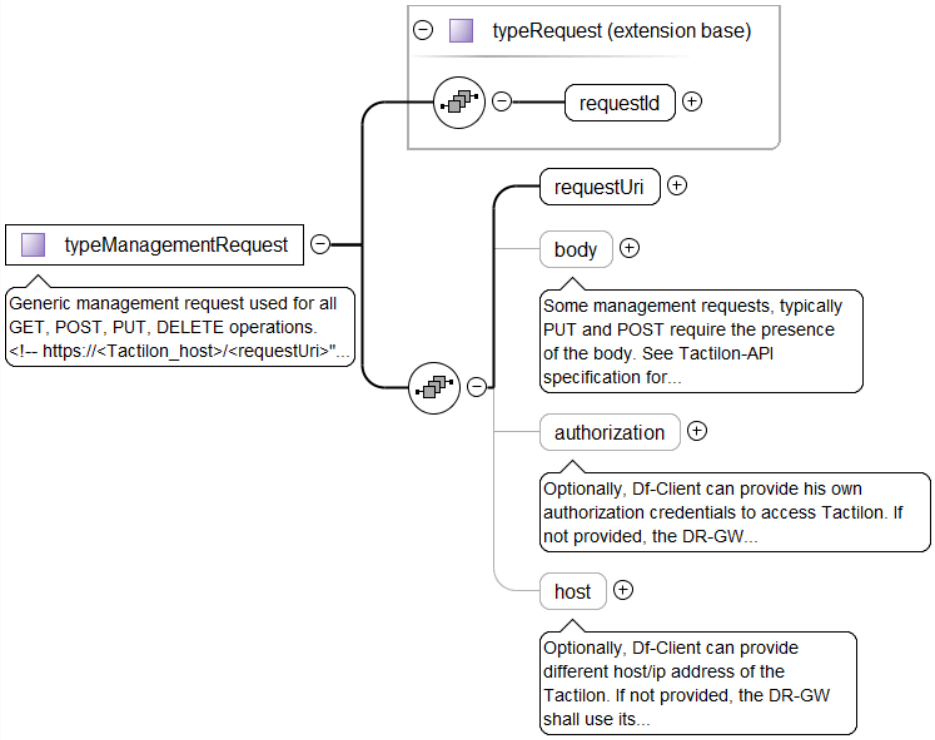
Complex Type `interfaceManagement`

Namespace	DR-GW
Annotations	`DR-GW-Management. Use for so called NEM-API(Tactilon-API) functions. Use only via SOAP.`
Diagram	
Used by	Element drgw/management
Model	get | post | put | delete | response | event
Children	delete, event, get, post, put, response

Source	`<xs:complexType name="interfaceManagement">`
	`  <xs:annotation>`
	`  <xs:documentation>DR-GW-Management. Use for so called NEM-API(Tactilon-API) functions. Use only via SOAP.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:choice>`
	`  <xs:element name="get" type="typeManagementRequest"/>`
	`  <xs:element name="post" type="typeManagementRequest"/>`

```
    <xs:element name="put" type="typeManagementRequest"/>
    <xs:element name="delete" type="typeManagementRequest"/>
    <xs:element name="response" type="typeResponse"/>
    <xs:element name="event" type="typeManagementEvent"/>
  </xs:choice>
</xs:complexType>
```

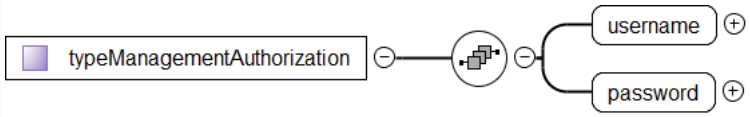
Complex Type `typeManagementRequest`

Namespace	DR-GW
Annotations	`Generic management request used for all GET, POST, PUT, DELETE operations.`
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeManagementRequest
Used by	Elements: interfaceManagement/delete, interfaceManagement/get, interfaceManagement/post, interfaceManagement/put
Model	requestId , requestUri , body{0,1} , authorization{0,1} , host{0,1}
Children	authorization, body, host, requestId, requestUri
Source	

```
<xs:complexType name="typeManagementRequest">
  <xs:annotation>
    <xs:documentation>Generic management request used for all GET, POST, PUT, DELETE operations.
      <!-- https://<Tactilon_host>/<requestUri>" -->
    </xs:documentation>
  </xs:annotation>
  <xs:complexContent>
    <xs:extension base="typeRequest">
      <xs:sequence>
        <xs:element name="requestUri" type="xs:string"/>
        <xs:element name="body" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Some management requests, typically PUT and POST require the
 presence of the body. See Tactilon-API specification for details on how to construct the body.</
xs:documentation>
          </xs:annotation>
        </xs:element>
        <xs:element name="authorization" type="typeManagementAuthorization" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Optionally, Df-Client can provide his own authorization credentials
 to access Tactilon. If not provided, the DR-GW shall use its own, globally configured.</
xs:documentation>
          </xs:annotation>
        </xs:element>
```

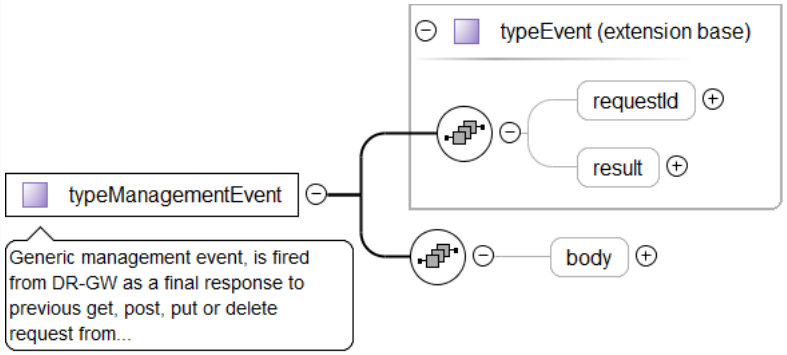
```
        <xs:element name="host" type="xs:string" minOccurs="0">
          <xs:annotation>
            <xs:documentation>Optionally, Df-Client can provide different host/ip address of the
 Tactilon. If not provided, the DR-GW shall use its own, globally configured.</xs:documentation>
          </xs:annotation>
        </xs:element>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeManagementAuthorization`

Namespace	DR-GW
Annotations	
Diagram	
Used by	Element typeManagementRequest/authorization
Model	username , password
Children	password, username

Source	`<xs:complexType name="typeManagementAuthorization">`
	`  <xs:annotation>`
	`  <xs:documentation>`
	`  </xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:element name="username" type="xs:string"/>`
	`  <xs:element name="password" type="xs:string"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

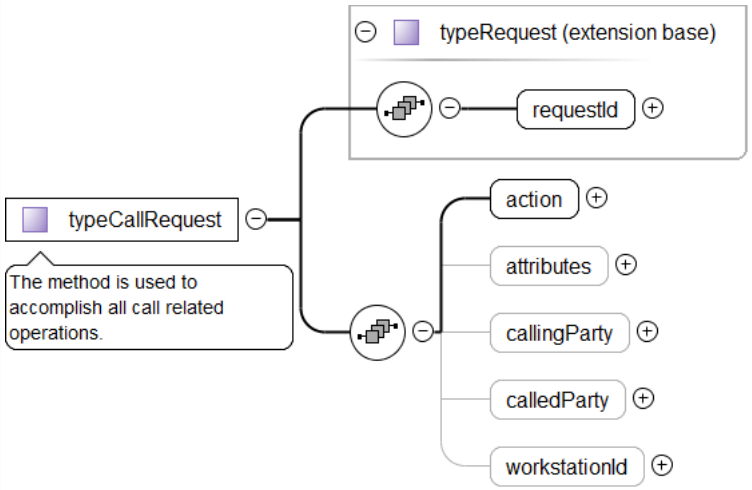
Complex Type `typeManagementEvent`

Namespace	DR-GW
Annotations	Generic management event, is fired from DR-GW as a final response to previous get, post, put or delete request from Df-Client and contains the actual response from Tactilon.
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeManagementEvent
Used by	Element interfaceManagement/event
Model	requestId{0,1} , result{0,1} , body{0,1}
Children	body, requestId, result

Source	`<xs:complexType name="typeManagementEvent">`
	`  <xs:annotation>`
	`  <xs:documentation>Generic management event, is fired from DR-GW as a final response to previous`
	` get, post, put or delete request from Df-Client and contains the actual response from Tactilon.</`
	`xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexContent>`
	`  <xs:extension base="typeEvent">`

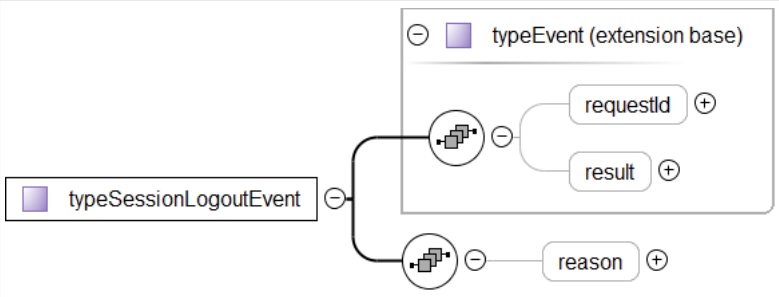
```
      <xs:sequence>
        <xs:element name="body" type="xs:string" minOccurs="0"/>
      </xs:sequence>
    </xs:extension>
  </xs:complexContent>
</xs:complexType>
```

Complex Type `typeCallRequest`

Namespace	DR-GW
Annotations	`The method is used to accomplish all call related operations.`
Diagram	
Type	extension of typeRequest
Type hierarchy	• typeRequest • typeCallRequest
Model	requestId , action , attributes{0,1} , callingParty{0,1} , calledParty{0,1} , workstationId{0,1}
Children	action, attributes, calledParty, callingParty, requestId, workstationId

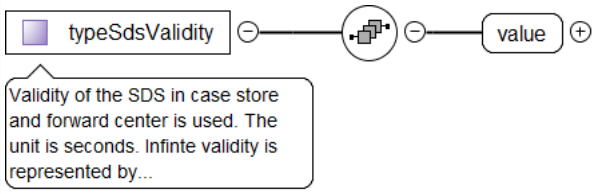
Source	<pre><xs:complexType name="typeCallRequest">
	  <xs:annotation>
	    <xs:documentation>The method is used to accomplish all call related operations.</
	xs:documentation>
	  </xs:annotation>
	  <xs:complexContent>
	    <xs:extension base="typeRequest">
	      <xs:sequence>
	        <xs:element name="action" type="typeActionRequest"/>
	        <xs:element name="attributes" type="typeCallAttributes" minOccurs="0"/>
	        <xs:element name="callingParty" type="typeAddress" minOccurs="0"/>
	        <xs:element name="calledParty" type="typeAddress" minOccurs="0"/>
	        <xs:element name="workstationId" type="typeWorkstationId" minOccurs="0"/>
	      </xs:sequence>
	    </xs:extension>
	  </xs:complexContent>
	</xs:complexType></pre>

Complex Type `typeSessionLogoutEvent`

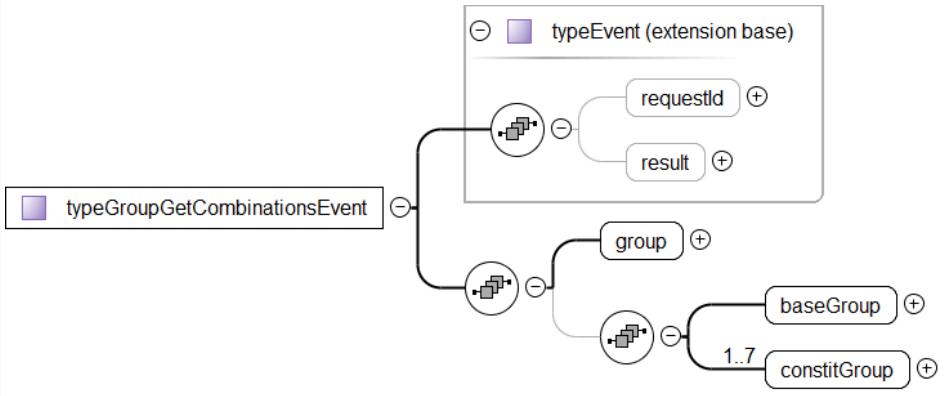
Namespace	DR-GW
Annotations	
Diagram	

Type	extension of typeEvent
Type hierarchy	• typeEvent • typeSessionLogoutEvent
Model	requestId{0,1} , result{0,1} , reason{0,1}
Children	reason, requestId, result
Source	<xs:complexType name="typeSessionLogoutEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="reason" type="xs:unsignedLong" minOccurs="0"/> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Complex Type `typeSdsValidity`

Namespace	DR-GW
Annotations	Validity of the SDS in case store and forward center is used. The unit is seconds. Infinte validity is represented by 0xFFFFFFFF
Diagram	
Model	value
Children	value
Source	<xs:complexType name="typeSdsValidity"> <xs:annotation> <xs:documentation>Validity of the SDS in case store and forward center is used. The unit is seconds. Infinte validity is represented by 0xFFFFFFFF</xs:documentation> </xs:annotation> <xs:sequence> <xs:element name="value" type="xs:unsignedLong"/> </xs:sequence> </xs:complexType>

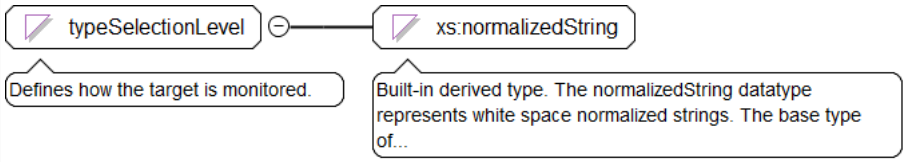
Complex Type `typeGroupGetCombinationsEvent`

Namespace	DR-GW
Annotations	
Diagram	
Type	extension of typeEvent
Type hierarchy	• typeEvent • typeGroupGetCombinationsEvent

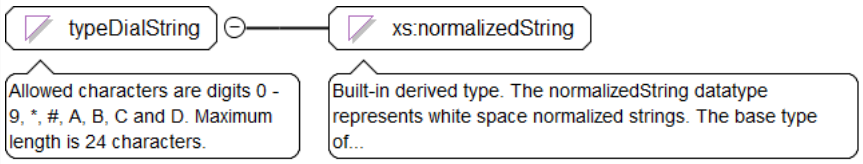
Model	requestId{0,1} , result{0,1} , group , baseGroup , constitGroup{1,7}
Children	baseGroup, constitGroup, group, requestId, result
Source	<xs:complexType name="typeGroupGetCombinationsEvent"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:complexContent> <xs:extension base="typeEvent"> <xs:sequence> <xs:element name="group" type="typeSubscriberAddress"/> <xs:sequence minOccurs="0"> <xs:element name="baseGroup" type="typeSubscriberAddress"/> <xs:element name="constitGroup" type="typeSubscriberAddress" maxOccurs="7"/> </xs:sequence> </xs:sequence> </xs:extension> </xs:complexContent> </xs:complexType>

Simple Type(s)

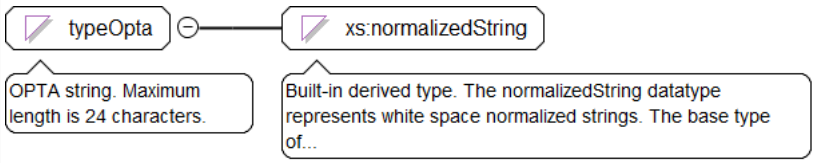
Simple Type `typeSelectionLevel`

Namespace	DR-GW		
Annotations	Defines how the target is monitored.		
Diagram			
Type	restriction of xs:normalizedString		
Facets	enumeration	no	No selection. Used to remove selection.
	enumeration	event	Event monitoring.
	enumeration	audio	Audio monitoring.
	enumeration	use	Selection level use.
	enumeration	a_use	Selection level active use.
Used by	Element	typeSelection/level	
Source	<xs:simpleType name="typeSelectionLevel"> <xs:annotation> <xs:documentation>Defines how the target is monitored.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="no"> <xs:annotation> <xs:documentation>No selection. Used to remove selection.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="event"> <xs:annotation> <xs:documentation>Event monitoring.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="audio"> <xs:annotation> <xs:documentation>Audio monitoring.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="use"> <xs:annotation> <xs:documentation>Selection level use.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="a_use"> <xs:annotation> <xs:documentation>Selection level active use.</xs:documentation> </xs:annotation> </xs:enumeration> </xs:restriction> </xs:simpleType>		

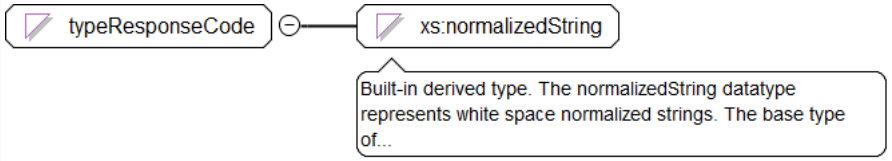
Simple Type `typeDialString`

Namespace	DR-GW
Annotations	Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.
Diagram	
Type	restriction of xs:normalizedString
Facets	maxLength 24
Used by	Elements typeAddress/msisdn, typeExternal/number
Source	<xs:simpleType name="typeDialString"> <xs:annotation> <xs:documentation>Allowed characters are digits 0 - 9, *, #, A, B, C and D. Maximum length is 24 characters.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:maxLength value="24"/> </xs:restriction> </xs:simpleType>

Simple Type `typeOpta`

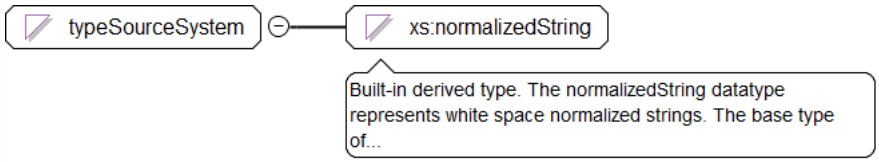
Namespace	DR-GW
Annotations	OPTA string. Maximum length is 24 characters.
Diagram	
Type	restriction of xs:normalizedString
Facets	maxLength 24
Used by	Elements typeAddress/opta, typeLastKnownOPTA/opta, typeRadioChangeOpta/opta, typeRadioChangeOptaEvent/opta
Source	<xs:simpleType name="typeOpta"> <xs:annotation> <xs:documentation>OPTA string. Maximum length is 24 characters.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:maxLength value="24"/> </xs:restriction> </xs:simpleType>

Simple Type `typeResponseCode`

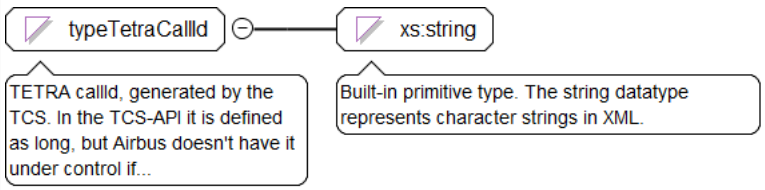
Namespace	DR-GW
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration success enumeration final_response_pending enumeration error enumeration not_authorized_error enumeration temporary_failure

	enumeration	subscription_failed
Used by	Element	typeResult/responseCode
Source	<xs:simpleType name="typeResponseCode"> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="success"/> <xs:enumeration value="final_response_pending"/> <xs:enumeration value="error"/> <xs:enumeration value="not_authorized_error"/> <xs:enumeration value="temporary_failure"/> <xs:enumeration value="subscription_failed"/> </xs:restriction> </xs:simpleType>	

Simple Type typeSourceSystem

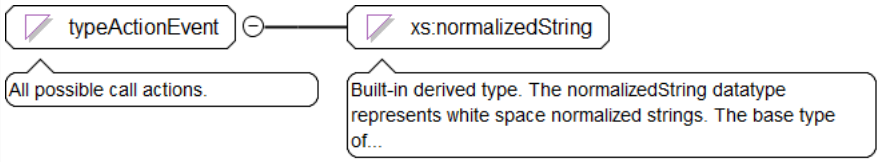
Namespace	DR-GW
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration DR-GW enumeration TCS-API enumeration TETRA enumeration NEM-API
Used by	Element typeResult/sourceSystem
Source	<xs:simpleType name="typeSourceSystem"> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="DR-GW"/> <xs:enumeration value="TCS-API"/> <xs:enumeration value="TETRA"/> <xs:enumeration value="NEM-API"/> </xs:restriction> </xs:simpleType>

Simple Type typeTetraCallId

Namespace	DR-GW
Annotations	TETRA callId, generated by the TCS. In the TCS-API it is defined as long, but Airbus doesn't have it under control if it can only be unsigned, therefore modified to string in DR-GW.
Diagram	
Type	xs:string
Used by	Elements typeCallEvent/tetraCallId, typeCallPTTEvent/tetraCallId, typeCallUnitInEmergencyEvent/tetraCallId
Source	<xs:simpleType name="typeTetraCallId"> <xs:annotation> <xs:documentation>TETRA callId, generated by the TCS. In the TCS-API it is defined as long, but Airbus doesn't have it under control if it can only be unsigned, therefore modified to string in DR-GW.</xs:documentation> </xs:annotation> <xs:restriction base="xs:string"/> </xs:simpleType>

Simple Type typeActionEvent

Namespace	DR-GW

Annotations	All possible call actions.
Diagram	
Type	restriction of xs:normalizedString

Facets

enumeration	incoming	This event fired when there is an incoming call. This is the first indication of a new incoming call.
enumeration	connected	This event is used to inform that call has been connected and call setup is finished.
enumeration	held	This event is used to inform TCS Client that individual call was put to hold.
enumeration	resumed	This event is used to inform that individual call has been taken from hold.
enumeration	disconnected	This event is used to inform that the call was disconnected.
enumeration	transferred	This event is a response to transfer method call and indicates the result of the request.

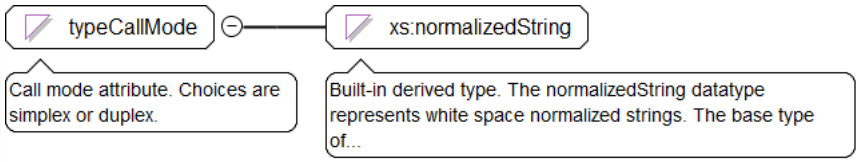
Used by	Element	typeCallEvent/action

Source

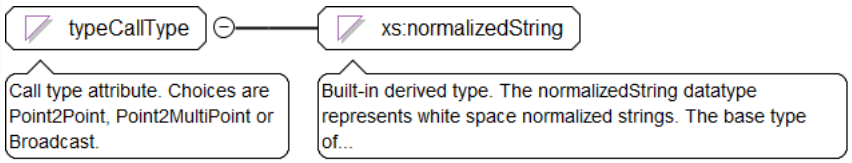
```
<xs:simpleType name="typeActionEvent">
  <xs:annotation>
    <xs:documentation>All possible call actions.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="incoming">
      <xs:annotation>
        <xs:documentation>This event fired when there is an incoming call. This is the first
 indication of a new incoming call.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="connected">
      <xs:annotation>
        <xs:documentation>This event is used to inform that call has been connected and call setup
 is finished.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="held">
      <xs:annotation>
        <xs:documentation>This event is used to inform TCS Client that individual call was put to
 hold.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="resumed">
      <xs:annotation>
        <xs:documentation>This event is used to inform that individual call has been taken from
 hold.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="disconnected">
      <xs:annotation>
        <xs:documentation>This event is used to inform that the call was disconnected.</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="transferred">
      <xs:annotation>
        <xs:documentation>This event is a response to transfer method call and indicates the result
 of the request.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

Simple Type typeCallMode

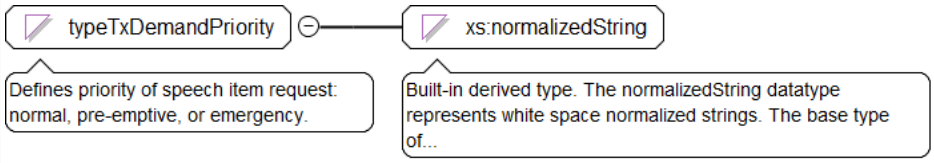
Namespace	DR-GW

Annotations	Call mode attribute. Choices are simplex or duplex.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration simplex enumeration duplex
Used by	Element typeCallAttributes/mode
Source	<xs:simpleType name="typeCallMode"> <xs:annotation> <xs:documentation>Call mode attribute. Choices are simplex or duplex.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="simplex"/> <xs:enumeration value="duplex"/> </xs:restriction> </xs:simpleType>

Simple Type typeCallType

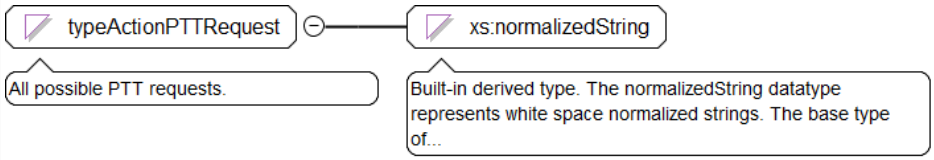
Namespace	DR-GW
Annotations	Call type attribute. Choices are Point2Point, Point2MultiPoint or Broadcast.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration p2p enumeration p2mp enumeration bcast
Used by	Elements typeCallAttributes/commtype, typeRadioTrackingData/callType
Source	<xs:simpleType name="typeCallType"> <xs:annotation> <xs:documentation>Call type attribute. Choices are Point2Point, Point2MultiPoint or Broadcast.</ xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="p2p"/> <xs:enumeration value="p2mp"/> <xs:enumeration value="bcast"/> </xs:restriction> </xs:simpleType>

Simple Type typeTxDemandPriority

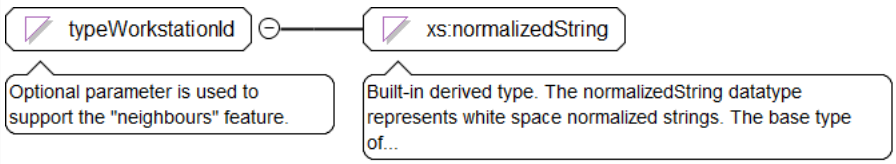
Namespace	DR-GW
Annotations	Defines priority of speech item request: normal, pre-emptive, or emergency.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration normal enumeration preemptive

	enumeration	emergency
Used by	Element	typeCallAttributes/demandPriority
Source	<xs:simpleType name="typeTxDemandPriority"> <xs:annotation> <xs:documentation>Defines priority of speech item request: normal, pre-emptive, or emergency.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="normal"/> <xs:enumeration value="preemptive"/> <xs:enumeration value="emergency"/> </xs:restriction> </xs:simpleType>	

Simple Type typeActionPTTRequest

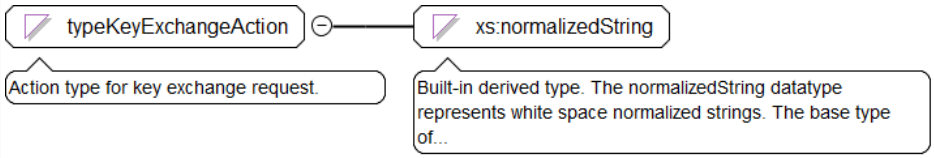
Namespace	DR-GW		
Annotations	All possible PTT requests.		
Diagram			
Type	restriction of xs:normalizedString		
Facets	enumeration	demandtx	This method can be used to request a speech item for a connected call.
	enumeration	ceasetx	This method is used to inform the system that the speech item is not needed any more.
Used by	Element	typeCallPTTRequest/action	
Source	<xs:simpleType name="typeActionPTTRequest"> <xs:annotation> <xs:documentation>All possible PTT requests.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="demandtx"> <xs:annotation> <xs:documentation>This method can be used to request a speech item for a connected call.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="ceasetx"> <xs:annotation> <xs:documentation>This method is used to inform the system that the speech item is not needed any more.</xs:documentation> </xs:annotation> </xs:enumeration> </xs:restriction> </xs:simpleType>		

Simple Type typeWorkstationId

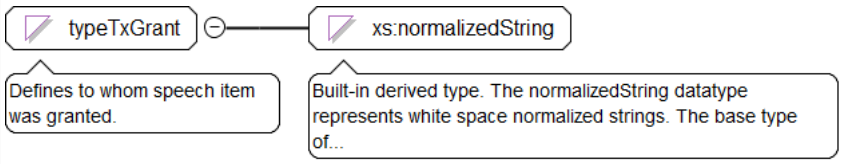
Namespace	DR-GW	
Annotations	Optional parameter is used to support the "neighbours" feature.	
Diagram		
Type	xs:normalizedString	
Used by	Elements	typeCallPTTRequest/workstationId, typeCallRequest/workstationId
Source	<xs:simpleType name="typeWorkstationId"> <xs:annotation>	

```
    <xs:documentation>Optional parameter is used to support the "neighbours" feature.</
xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString"/>
</xs:simpleType>
```

Simple Type `typeKeyExchangeAction`

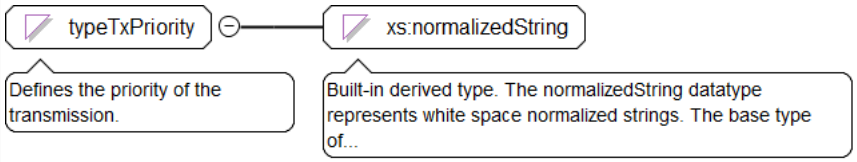
Namespace	DR-GW
Annotations	Action type for key exchange request.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration: start enumeration: stop
Used by	Element: typeCallKeyExchange/action
Source	<xs:simpleType name="typeKeyExchangeAction"> <xs:annotation> <xs:documentation>Action type for key exchange request.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="start"/> <xs:enumeration value="stop"/> </xs:restriction> </xs:simpleType>

Simple Type `typeTxGrant`

Namespace	DR-GW
Annotations	Defines to whom speech item was granted.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration: granted enumeration: notGranted enumeration: queued enumeration: granted2another
Used by	Element: typeTxGranted/txGrant
Source	<xs:simpleType name="typeTxGrant"> <xs:annotation> <xs:documentation>Defines to whom speech item was granted.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="granted"/> <xs:enumeration value="notGranted"/> <xs:enumeration value="queued"/> <xs:enumeration value="granted2another"/> </xs:restriction> </xs:simpleType>

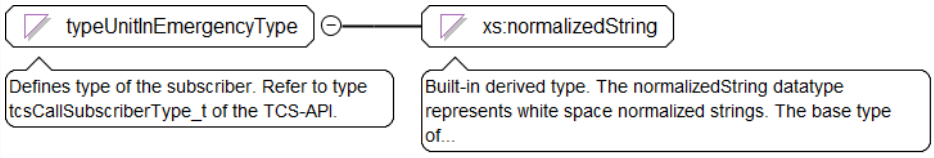
Simple Type `typeTxPriority`

Namespace	DR-GW
Annotations	Defines the priority of the transmission.

Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `normal` enumeration `emergency`
Used by	Element typeTxGranted/txPriority

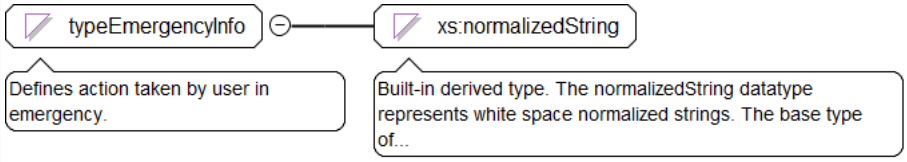
Source	`<xs:simpleType name="typeTxPriority">`
	`  <xs:annotation>`
	`  <xs:documentation>Defines the priority of the transmission.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="normal"/>`
	`  <xs:enumeration value="emergency"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `typeUnitInEmergencyType`

Namespace	DR-GW
Annotations	Defines type of the subscriber. Refer to type tcsCallSubscriberType_t of the TCS-API.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `dummy` enumeration `ms` enumeration `g4wif` enumeration `external` enumeration `ws`
Used by	Element typeCallUnitInEmergencyEvent/unitInEmgType

Source	`<xs:simpleType name="typeUnitInEmergencyType">`
	`  <xs:annotation>`
	`  <xs:documentation>Defines type of the subscriber. Refer to type tcsCallSubscriberType_t of the TCS-API.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="dummy"/>`
	`  <xs:enumeration value="ms"/>`
	`  <xs:enumeration value="g4wif"/>`
	`  <xs:enumeration value="external"/>`
	`  <xs:enumeration value="ws"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `typeEmergencyInfo`

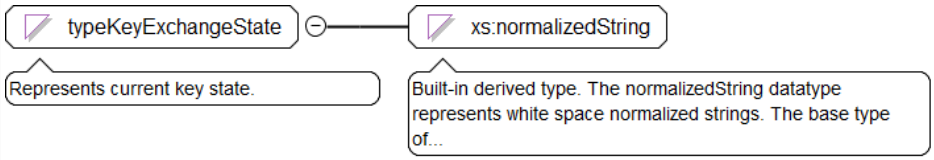
Namespace	DR-GW
Annotations	Defines action taken by user in emergency.
Diagram	
Type	restriction of xs:normalizedString

Facets	enumeration	addTx
	enumeration	add
	enumeration	ceased
	enumeration	demandTx
	enumeration	removed
	enumeration	emergencyCallDisconnected
Used by	Element	typeCallUnitInEmergencyEvent/emgInfo

Source

```
<xs:simpleType name="typeEmergencyInfo">
  <xs:annotation>
    <xs:documentation>Defines action taken by user in emergency.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="addTx"/>
    <xs:enumeration value="add"/>
    <xs:enumeration value="ceased"/>
    <xs:enumeration value="demandTx"/>
    <xs:enumeration value="removed"/>
    <xs:enumeration value="emergencyCallDisconnected"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type typeKeyExchangeState

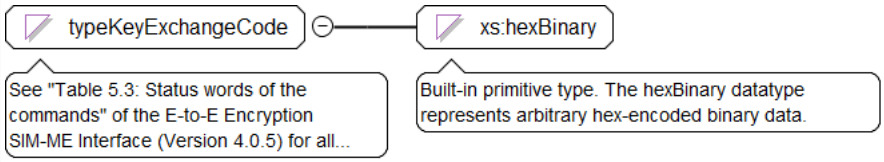
Namespace	DR-GW		
Annotations	Represents current key state.		
Diagram			
Type	restriction of xs:normalizedString		
Facets	enumeration	keyValid	current key is valid, no user action required.
	enumeration	keyInvalid	Key invalid, user must request key exchange.
	enumeration	keyExchangeInProgress	Key exchange in progress, user may abort exchange or wait until it gets finished.
Used by	Element	typeCallKeyExchangeEvent/state	

Source

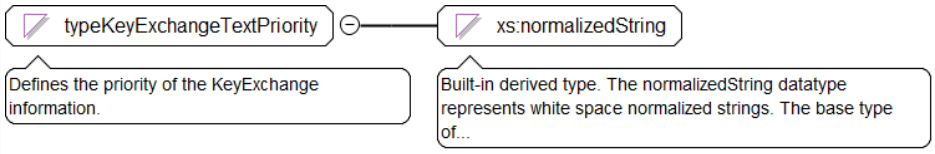
```
<xs:simpleType name="typeKeyExchangeState">
  <xs:annotation>
    <xs:documentation>Represents current key state.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="keyValid">
      <xs:annotation>
        <xs:documentation>current key is valid, no user action required.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="keyInvalid">
      <xs:annotation>
        <xs:documentation>Key invalid, user must request key exchange.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="keyExchangeInProgress">
      <xs:annotation>
        <xs:documentation>Key exchange in progress, user may abort exchange or wait until it gets
 finished.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

Simple Type typeKeyExchangeCode

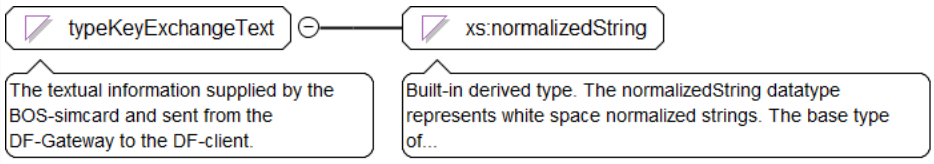
Namespace	DR-GW
Annotations	See "Table 5.3: Status words of the commands" of the E-to-E Encryption

	SIM-ME Interface (Version 4.0.5) for all possible code values.
Diagram	
Type	restriction of xs:hexBinary
Facets	length 2
Used by	Element typeCallKeyExchangeEvent/code
Source	<xs:simpleType name="typeKeyExchangeCode"> <xs:annotation> <xs:documentation>See "Table 5.3: Status words of the commands" of the E-to-E Encryption SIM-ME Interface (Version 4.0.5) for all possible code values.</xs:documentation> </xs:annotation> <xs:restriction base="xs:hexBinary"> <xs:length value="2"/> </xs:restriction> </xs:simpleType>

Simple Type typeKeyExchangeTextPriority

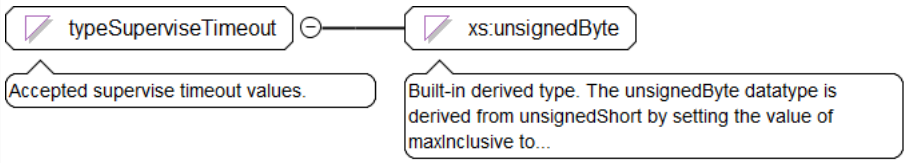
Namespace	DR-GW
Annotations	Defines the priority of the KeyExchange information.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration normal enumeration high
Used by	Element typeCallKeyExchangeEvent/priority
Source	<xs:simpleType name="typeKeyExchangeTextPriority"> <xs:annotation> <xs:documentation>Defines the priority of the KeyExchange information.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="normal"/> <xs:enumeration value="high"/> </xs:restriction> </xs:simpleType>

Simple Type typeKeyExchangeText

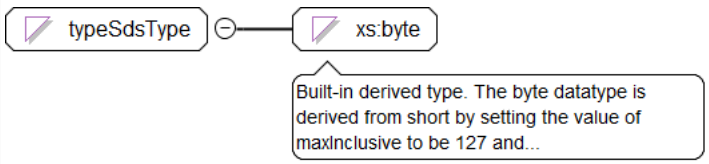
Namespace	DR-GW
Annotations	The textual information supplied by the BOS-simcard and sent from the DF-Gateway to the DF-client.
Diagram	
Type	restriction of xs:normalizedString
Facets	maxLength 100
Used by	Element typeCallKeyExchangeEvent/text
Source	<xs:simpleType name="typeKeyExchangeText"> <xs:annotation> <xs:documentation>The textual information supplied by the BOS-simcard and sent from the DF-Gateway to the DF-client.</xs:documentation> </xs:annotation>

```
  <xs:restriction base="xs:normalizedString">
    <xs:maxLength value="100"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeSuperviseTimeout`

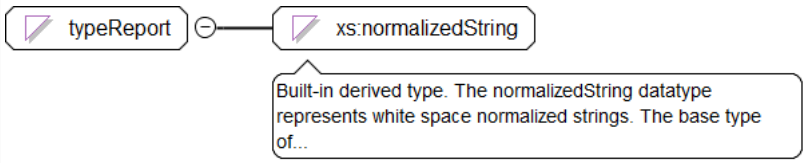
Namespace	DR-GW
Annotations	Accepted supervise timeout values.
Diagram	
Type	restriction of xs:unsignedByte
Facets	enumeration 20 enumeration 30 enumeration 60
Used by	Element typeSessionLogin/supervise
Source	<xs:simpleType name="typeSuperviseTimeout"> <xs:annotation> <xs:documentation>Accepted supervise timeout values.</xs:documentation> </xs:annotation> <xs:restriction base="xs:unsignedByte"> <xs:enumeration value="20"/> <xs:enumeration value="30"/> <xs:enumeration value="60"/> </xs:restriction> </xs:simpleType>

Simple Type `typeSdsType`

Namespace	DR-GW
Annotations	
Diagram	
Type	restriction of xs:byte
Facets	enumeration 0 SDS1. enumeration 1 SDS2. enumeration 2 SDS3. enumeration 3 SDS4. enumeration 4 SDS-TL. enumeration 5 Status.
Used by	Element typeSds/sdsType
Source	<xs:simpleType name="typeSdsType"> <xs:annotation> <xs:documentation/> </xs:annotation> <xs:restriction base="xs:byte"> <xs:enumeration value="0"> <xs:annotation> <xs:documentation>SDS1.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="1"> <xs:annotation> <xs:documentation>SDS2.</xs:documentation> </xs:annotation>

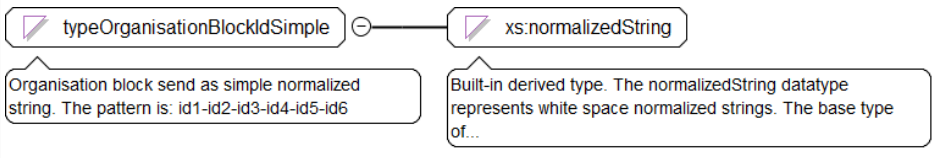
```
    </xs:enumeration>
    <xs:enumeration value="2">
      <xs:annotation>
        <xs:documentation>SDS3.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="3">
      <xs:annotation>
        <xs:documentation>SDS4.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="4">
      <xs:annotation>
        <xs:documentation>SDS-TL.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="5">
      <xs:annotation>
        <xs:documentation>Status.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeReport`

Namespace	DR-GW
Annotations	
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `none` enumeration `delivery` enumeration `consume` enumeration `both`
Used by	Element typeSds/report

Source	`<xs:simpleType name="typeReport">`
	`  <xs:annotation>`
	`  <xs:documentation/>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="none"/>`
	`  <xs:enumeration value="delivery"/>`
	`  <xs:enumeration value="consume"/>`
	`  <xs:enumeration value="both"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

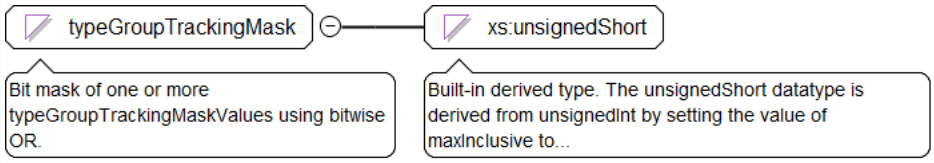
Simple Type `typeOrganisationBlockIdSimple`

Namespace	DR-GW
Annotations	`Organisation block send as simple normalized string. The pattern is: id1-id2-id3-id4-id5-id6`
Diagram	
Type	restriction of xs:normalizedString

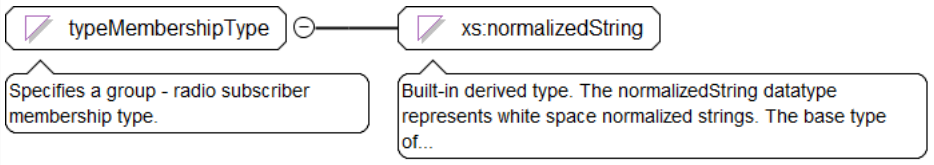
Facets	pattern `(([0-9]|[1-9]\d{0,3}|`
	`[1-5]\d{4}|6[0-4]\d{3}|`
	`65[0-4]\d{2}|655[0-2]\d|`
	`6553[0-5])-){0,5}([0-9]|`
	`[1-9]\d{0,3}|[1-5]\d{4}|`

	6[0-4]\d{3}|65[0-4]\d{2}| 655[0-2]\d|6553[0-5])
Used by	Element typeOrganisationBlockId/orgblockIdSimple
Source	<xs:simpleType name="typeOrganisationBlockIdSimple"> <xs:annotation> <xs:documentation>Organisation block send as simple normalized string. The pattern is: id1-id2-id3-id4-id5-id6</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:pattern value="(([0-9]|[1-9]\d{0,3}|[1-5]\d{4}|6[0-4]\d{3}|65[0-4]\d{2}|655[0-2]\d|6553[0-5])-){0,5}([0-9]|[1-9]\d{0,3}|[1-5]\d{4}|6[0-4]\d{3}|65[0-4]\d{2}|655[0-2]\d|6553[0-5])"/> </xs:restriction> </xs:simpleType>

Simple Type typeGroupTrackingMask

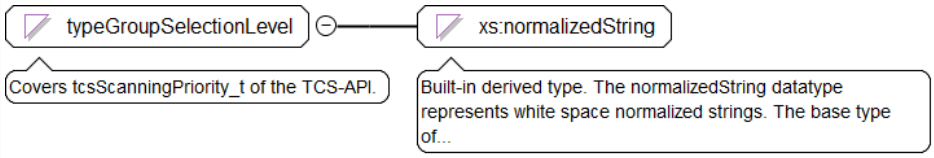
Namespace	DR-GW
Annotations	Bit mask of one or more typeGroupTrackingMaskValues using bitwise OR.
Diagram	
Type	xs:unsignedShort
Used by	Elements typeGroupTrack/mask, typeGroupTrackSubscriptionEvent/mask
Source	<xs:simpleType name="typeGroupTrackingMask"> <xs:annotation> <xs:documentation>Bit mask of one or more typeGroupTrackingMaskValues using bitwise OR.</xs:documentation> </xs:annotation> <xs:restriction base="xs:unsignedShort"/> </xs:simpleType>

Simple Type typeMembershipType

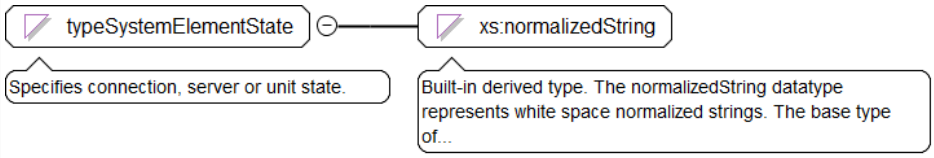
Namespace	DR-GW	
Annotations	Specifies a group - radio subscriber membership type.	
Diagram		
Type	restriction of xs:normalizedString	
Facets	enumeration	unknown
	enumeration	permanent
	enumeration	visiting
Used by	Element typeGroupAddRadioMember/membership	
Source	<xs:simpleType name="typeMembershipType"> <xs:annotation> <xs:documentation>Specifies a group - radio subscriber membership type.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="unknown"/> <xs:enumeration value="permanent"/> <xs:enumeration value="visiting"/> </xs:restriction> </xs:simpleType>	

Simple Type typeGroupSelectionLevel

Namespace	DR-GW

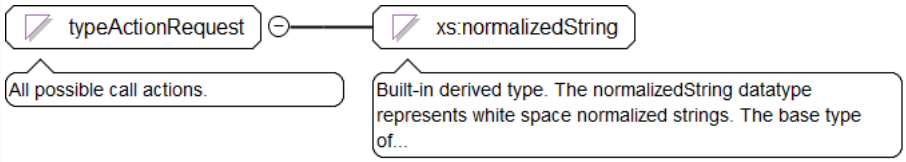
Annotations	Covers tcsScanningPriority_t of the TCS-API.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration notScanned enumeration low enumeration normal enumeration selected enumeration high enumeration background
Used by	Element typeRadioGroupSelection/level
Source	<xs:simpleType name="typeGroupSelectionLevel"> <xs:annotation> <xs:documentation>Covers tcsScanningPriority_t of the TCS-API.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="notScanned"/> <xs:enumeration value="low"/> <xs:enumeration value="normal"/> <xs:enumeration value="selected"/> <xs:enumeration value="high"/> <xs:enumeration value="background"/> </xs:restriction> </xs:simpleType>

Simple Type typeSystemElementState

Namespace	DR-GW
Annotations	Specifies connection, server or unit state.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration unknown Unknown state. enumeration ok Connection or server is working. enumeration n_Ok Connection or server is not working.
Used by	Elements typeSystemTetraStatesEvent/cddconnectionState, typeSystemTetraStatesEvent/cddserverState, typeSystemTetraStatesEvent/dxtState, typeSystemTetraStatesEvent/tcsState
Source	<xs:simpleType name="typeSystemElementState"> <xs:annotation> <xs:documentation>Specifies connection, server or unit state.</xs:documentation> </xs:annotation> <xs:restriction base="xs:normalizedString"> <xs:enumeration value="unknown"> <xs:annotation> <xs:documentation>Unknown state.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="ok"> <xs:annotation> <xs:documentation>Connection or server is working.</xs:documentation> </xs:annotation> </xs:enumeration> <xs:enumeration value="n_Ok"> <xs:annotation> <xs:documentation>Connection or server is not working.</xs:documentation> </xs:annotation> </xs:enumeration>

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

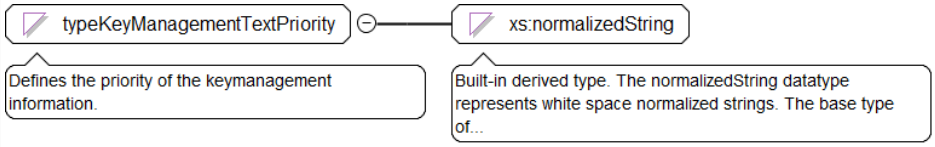
Simple Type `typeActionRequest`

Namespace	DR-GW
Annotations	`All possible call actions.`
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `setup` — `This method is used to initiate a new call setup. For a call setup to be successful it is required that the resources have been reserved prior this method call.` enumeration `connect` — `This method is used to connect an incoming call.` enumeration `hold` — `This method requests to put an individual call to hold.` enumeration `unhold` — `This method is a request for resuming an individual call from hold.` enumeration `disconnect` — `This method is used to disconnect a call.` enumeration `transfer` — `This method is used to transfer an individual call to a new recipient.` enumeration `releasecall` — `This method is used to release radio subscriber's individual call.`
Used by	Element typeCallRequest/action
Source	see below

```
<xs:simpleType name="typeActionRequest">
  <xs:annotation>
    <xs:documentation>All possible call actions.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:normalizedString">
    <xs:enumeration value="setup">
      <xs:annotation>
        <xs:documentation>This method is used to initiate a new call setup. For a call setup to
 be successful it is required that the resources have been reserved prior this method call.</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="connect">
      <xs:annotation>
        <xs:documentation>This method is used to connect an incoming call.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="hold">
      <xs:annotation>
        <xs:documentation>This method requests to put an individual call to hold.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="unhold">
      <xs:annotation>
        <xs:documentation>This method is a request for resuming an individual call from hold.</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="disconnect">
      <xs:annotation>
        <xs:documentation>This method is used to disconnect a call.</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="transfer">
      <xs:annotation>
        <xs:documentation>This method is used to transfer an individual call to a new recipient.</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
```

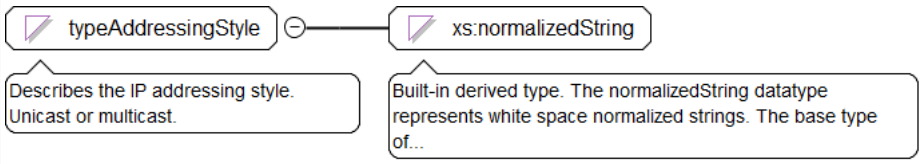
```
    <xs:enumeration value="releasecall">
      <xs:annotation>
        <xs:documentation>This method is used to release radio subscriber's individual call.</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeKeyManagementTextPriority`

Namespace	DR-GW
Annotations	Defines the priority of the keymanagement information.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `normal` enumeration `high`

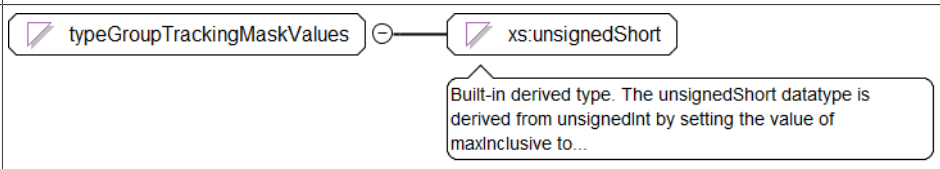
Source	`<xs:simpleType name="typeKeyManagementTextPriority">`
	`  <xs:annotation>`
	`  <xs:documentation>Defines the priority of the keymanagement information.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="normal"/>`
	`  <xs:enumeration value="high"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `typeAddressingStyle`

Namespace	DR-GW
Annotations	Describes the IP addressing style. Unicast or multicast.
Diagram	
Type	restriction of xs:normalizedString
Facets	enumeration `ucast` enumeration `mcast`

Source	`<xs:simpleType name="typeAddressingStyle">`
	`  <xs:annotation>`
	`  <xs:documentation>Describes the IP addressing style. Unicast or multicast.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:normalizedString">`
	`  <xs:enumeration value="ucast"/>`
	`  <xs:enumeration value="mcast"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `typeGroupTrackingMaskValues`

Namespace	DR-GW
Annotations	
Diagram	
Type	restriction of xs:unsignedShort

Facets	enumeration	0	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_BASIC_C
	enumeration	1	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_RS_ADD_REMOVE_C
	enumeration	2	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_G4WIF_ADD_REMOVE_C
	enumeration	4	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_WSUSER_ADD_REMOVE_C
	enumeration	8	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_CBR_REMOVE_C
	enumeration	16	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_GROUP_ADD_REMOVE_C
	enumeration	65535	TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_ALL_C

Source

```
<xs:simpleType name="typeGroupTrackingMaskValues">
  <xs:annotation>
    <xs:documentation/>
  </xs:annotation>
  <xs:restriction base="xs:unsignedShort">
    <xs:enumeration value="0">
      <xs:annotation>
        <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_BASIC_C</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="1">
      <xs:annotation>
        <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_RS_ADD_REMOVE_C</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="2">
      <xs:annotation>
        <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_G4WIF_ADD_REMOVE_C</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="4">
      <xs:annotation>
        <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_WSUSER_ADD_REMOVE_C</
xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="8">
      <xs:annotation>
        <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_CBR_REMOVE_C</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="16">
      <xs:annotation>
        <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_GROUP_ADD_REMOVE_C</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
    <xs:enumeration value="65535">
      <xs:annotation>
        <xs:documentation>TCS_GROUP_SUBSCRIPTION_MASK_VALUES_T_ALL_C</xs:documentation>
      </xs:annotation>
    </xs:enumeration>
  </xs:restriction>
</xs:simpleType>
```

Attribute(s)

Attribute `drgw / @version`

Namespace	DR-GW	
Properties	use:	required
	fixed:	2.0
Used by	Element	drgw
Source	`<xs:attribute name="version" fixed="2.0" use="required"/>`	