Schema documentation for Employee.xsd

december 8, 2018

Table of Contents

Namespace: "employment" 4
Schema(s) 4
Main schema Employee.xsd 4
Element(s) 4
Element employee 4
Element employeeType / control 4
Element controlType / recordId 5
Element controlType / status 6
Element controlType / maintenanceAgency 6
Element maintenanceAgencyType / agencyCode 7
Element maintenanceAgencyType / otherAgencyCode 7
Element maintenanceAgencyType / agencyName 8
Element controlType / languageDeclaration 8
Element languageDeclarationType / language 9
Element languageDeclarationType / script 9
Element controlType / conventionDeclaration 9
Element conventionDeclarationType / code 10
Element conventionDeclarationType / citation 10
Element conventionDeclarationType / description 11
Element controlType / ownValueListDeclaration 11
Element ownValueListDeclarationType / code 12
Element ownValueListDeclarationType / citation 12
Element ownValueListDeclarationType / description 13
Element controlType / maintenanceHistory 13
Element maintenanceHistoryType / maintenanceHistoryEvent 14
Element additionalElements 15
Element ownElement 15
Element ownElement / ownElementDescription 16
Element ownElement / ownElement 16
Element value 17
Element property 18
Element attribute 18
Element ownElementType / ownElement 19
Element AdditionalXMLData 20
Element appendix 21
Element AppendixType / eSignature 22
Element eSignatureComplexType / signature 22
Element employeeType / employeeDescription 23
Element employeeDescriptionType / nextOfKin 25
Element nextOfKinType / nextOfKinFreeText 25
Element nextOfKinType / nextOfKinEntry 26
Element nextOfKinEntryType / name 26
Element nextOfKinEntryType / contactInformationEntries 27
Element contactInformationEntriesType / contactInformationEntry 27
Element contactInformationEntryType / entryDetails 28
Element employeeDescriptionType / adjustments 28
Element adjustmentsType / adjustment 28
Element adjustmentType / description 29
Element adjustmentType / adjustmentPeriod 29
Element adjustmentPeriodType / date 30
Element dateRange 30
Element dateRangeType / startDate 31
Element dateRangeType / endDate 31
Element dateSet 32
Element dateSetType / date 32
Element employeeDescriptionType / salarySeizures 32
Element salarySeizuresType / salarySeizure 33
Element salarySeizureType / description 33
Element salarySeizureType / seizurePeriod 34
Element seizurePeriodType / date 34
Element employeeDescriptionType / taxes 35
Element taxesType / taxEntry 35
Element taxEntryType / taxTable 36

Element taxTableType / description ... 36
Element taxEntryType / optionalTaxes ... 37
Element optionalTaxesType / optionalTaxEntry ... 37
Element employeeDescriptionType / children ... 38
Element childrenType / child ... 38
Element employeeDescriptionType / reassignments ... 39
Element reassignmentsType / reassignment ... 39
Element reassignmentType / description ... 40
Element employeeDescriptionType / disciplinaryActions ... 40
Element disciplinaryActionsType / disciplinaryAction ... 41
Element disciplinaryActionType / description ... 41
Element employeeDescriptionType / workRelatedInjuries ... 41
Element workRelatedInjuriesType / workRelatedInjury ... 42
Element workRelatedInjuryType / description ... 43
Element employeeDescriptionType / rehabilitationArrangements ... 43
Element rehabilitationArrangementsType / rehabilitationArrangement ... 43
Element rehabilitationArrangementType / description ... 44
Element employeeDescriptionType / workRelatedAidGrants ... 44
Element workRelatedAidGrantsType / workRelatedAidGrant ... 45
Element workRelatedAidGrantType / description ... 46
Element employeeDescriptionType / deathRelatedDocuments ... 46
Element deathRelatedDocumentsType / deathRelatedDocument ... 46
Element deathRelatedDocumentType / description ... 47
Element employeeDescriptionType / relations ... 47
Element relationsType / relationToParent ... 48
Element relationToParentType / relationEntry ... 48
Element relationsType / relationToChild ... 49
Element relationToChildType / relationEntry ... 49
Complex Type(s) ... 50
Complex Type employeeType ... 50
Complex Type controlType ... 51
Complex Type maintenanceAgencyType ... 52
Complex Type otherAgencyCodeType ... 52
Complex Type languageDeclarationType ... 53
Complex Type languageType ... 53
Complex Type scriptType ... 54
Complex Type conventionDeclarationType ... 55
Complex Type citationType ... 55
Complex Type ownValueListDeclarationType ... 56
Complex Type maintenanceHistoryType ... 56
Complex Type maintenanceHistoryEventType ... 57
Complex Type additionalElementsType ... 58
Complex Type ownElementType ... 59
Complex Type ExtendingComplexType ... 60
Complex Type AppendixType ... 60
Complex Type eSignatureComplexType ... 61
Complex Type employeeDescriptionType ... 62
Complex Type nextOfKinType ... 64
Complex Type nextOfKinEntryType ... 65
Complex Type contactInformationEntriesType ... 66
Complex Type contactInformationEntryType ... 66
Complex Type adjustmentsType ... 67
Complex Type adjustmentType ... 67
Complex Type adjustmentPeriodType ... 68
Complex Type dateRangeType ... 68
Complex Type dateSetType ... 69
Complex Type salarySeizuresType ... 69
Complex Type salarySeizureType ... 69
Complex Type seizurePeriodType ... 70
Complex Type taxesType ... 70
Complex Type taxEntryType ... 71
Complex Type taxTableType ... 71
Complex Type optionalTaxesType ... 72
Complex Type optionalTaxEntryType ... 72
Complex Type childrenType ... 73
Complex Type childType ... 73
Complex Type reassignmentsType ... 74
Complex Type reassignmentType ... 74
Complex Type disciplinaryActionsType ... 75
Complex Type disciplinaryActionType ... 75
Complex Type workRelatedInjuriesType ... 76
Complex Type workRelatedInjuryType ... 76

Complex Type rehabilitationArrangementsType 77
Complex Type rehabilitationArrangementType 77
Complex Type workRelatedAidGrantsType 78
Complex Type workRelatedAidGrantType 78
Complex Type deathRelatedDocumentsType 79
Complex Type deathRelatedDocumentType 79
Complex Type relationsType 80
Complex Type relationToParentType 80
Complex Type relationToChildType 81
Simple Type(s) 82
Simple Type statusType 82
Simple Type agencyCodeType 82
Simple Type otherAgencyCodePrefix 82
Simple Type eventTypeType 83
Simple Type agentTypeType 84
Simple Type entryTypeType 84
Simple Type typeOfRelationType 85
Simple Type ownDate 85
Simple Type typeOfTaxType 86
Namespace: "" 86
Attribute(s) 86
Attribute otherAgencyCodeType / @agencyCodePrefix 86
Attribute otherAgencyCodeType / @otherAgencyCodePrefix 87
Attribute languageType / @languageCode 87
Attribute languageType / @language 87
Attribute scriptType / @scriptCode 87
Attribute scriptType / @script 87
Attribute citationType / @href 88
Attribute maintenanceHistoryEventType / @eventType 88
Attribute maintenanceHistoryEventType / @eventDateTime 88
Attribute maintenanceHistoryEventType / @agentType 89
Attribute maintenanceHistoryEventType / @otherAgentType 89
Attribute maintenanceHistoryEventType / @agent 89
Attribute attribute / @name 89
Attribute attribute / @dataType 90
Attribute attribute / @format 90
Attribute ownElementType / @name 90
Attribute ownElementType / @dataType 90
Attribute ownElementType / @format 90
Attribute eSignatureComplexType / @present 91
Attribute eSignatureComplexType / @dateESignatureIsVerified 91
Attribute AppendixType / @disposable 91
Attribute AppendixType / @name 91
Attribute AppendixType / @description 92
Attribute AppendixType / @fileFormat 92
Attribute AppendixType / @originalFileFormat 92
Attribute AppendixType / @path 92
Attribute AppendixType / @eSignatureHaveExisted 92
Attribute contactInformationEntryType / @entryType 93
Attribute nextOfKinEntryType / @typeOfRelation 93
Attribute nextOfKinEntryType / @otherTypeOfRelation 94
Attribute taxTableType / @table 94
Attribute taxTableType / @column 94
Attribute optionalTaxEntryType / @typeOfTax 94
Attribute optionalTaxEntryType / @otherTypeOfTax 95
Attribute childType / @firstName 95
Attribute childType / @surname 95
Attribute childType / @birthDate 95
Attribute reassignmentType / @href 96
Attribute disciplinaryActionType / @href 96
Attribute workRelatedInjuryType / @href 96
Attribute rehabilitationArrangementType / @href 96
Attribute workRelatedAidGrantType / @href 96
Attribute deathRelatedDocumentType / @href 97
Attribute relationToParentType / relationEntry / @parentOrganisationReferenceCode 97
Attribute relationToParentType / @typeOfRelation 97
Attribute relationToParentType / @href 97
Attribute relationToChildType / relationEntry / @childOrganisationReferenceCode 98
Attribute relationToChildType / @typeOfRelation 98
Attribute relationToChildType / @href 98

Namespace: "employment"

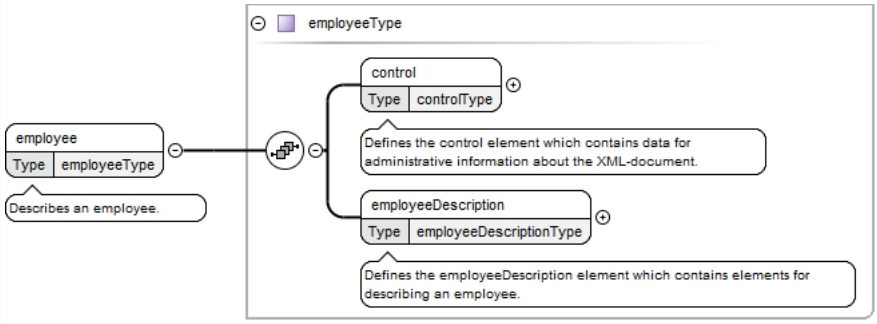
Schema(s)

Main schema `Employee.xsd`

Namespace	employment
Properties	attribute form default: `unqualified` element form default: `qualified`

Element(s)

Element `employee`

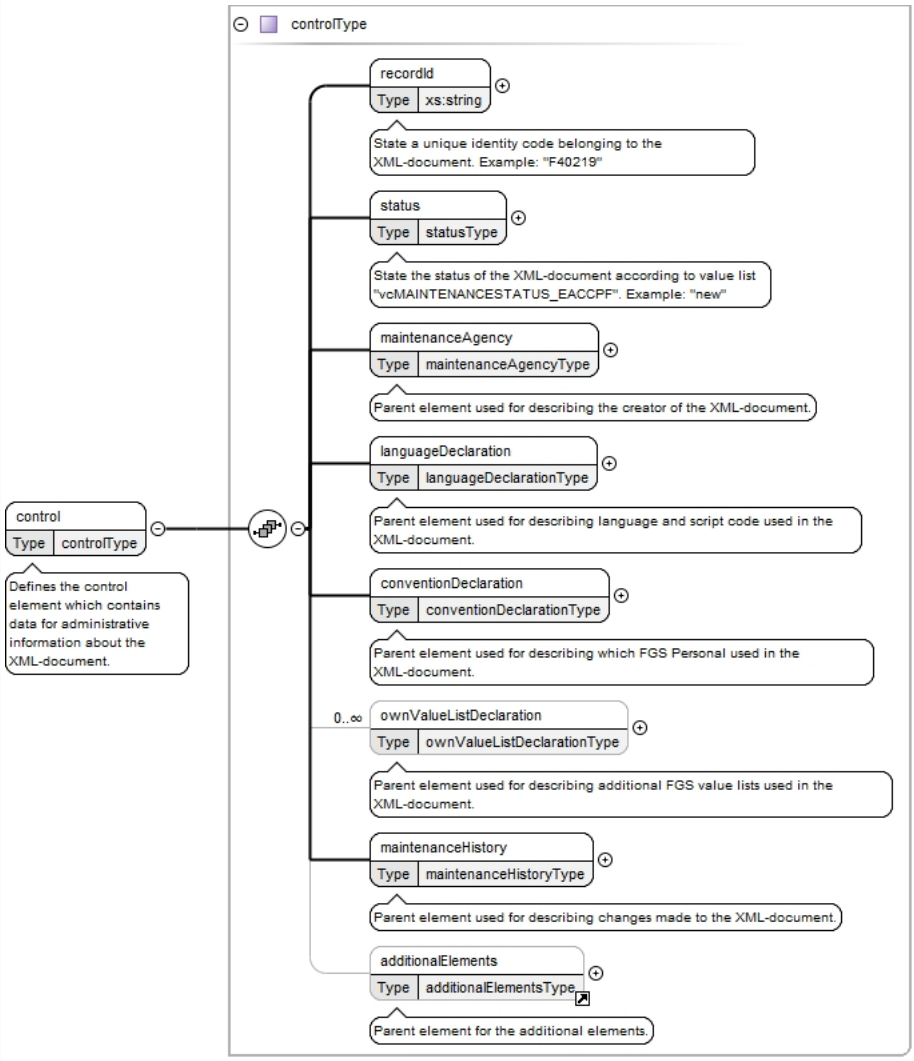
Namespace	employment
Annotations	`Describes an employee.`
Diagram	
Type	employeeType
Properties	content: `complex`
Model	control , employeeDescription
Children	control, employeeDescription

Instance	`<employee xmlns="employment">`
	`  <control>{1,1}</control>`
	`  <employeeDescription>{1,1}</employeeDescription>`
	`</employee>`

Source	`<xs:element name="employee" type="employeeType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describes an employee.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `employeeType / control`

Namespace	employment
Annotations	`Defines the control element which contains data for administrative information about the XML-document.`

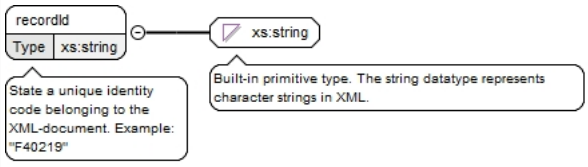
Diagram	
Type	controlType
Properties	content: complex
Model	recordId , status , maintenanceAgency , languageDeclaration , conventionDeclaration , ownValueListDeclaration* , maintenance-History , additionalElements{0,1}
Children	additionalElements, conventionDeclaration, languageDeclaration, maintenanceAgency, maintenanceHistory, ownValueListDeclaration, recordId, status

Instance	`<control xmlns="employment">`
	`  <recordId>{1,1}</recordId>`
	`  <status>{1,1}</status>`
	`  <maintenanceAgency>{1,1}</maintenanceAgency>`
	`  <languageDeclaration>{1,1}</languageDeclaration>`
	`  <conventionDeclaration>{1,1}</conventionDeclaration>`
	`  <ownValueListDeclaration>{0,unbounded}</ownValueListDeclaration>`
	`  <maintenanceHistory>{1,1}</maintenanceHistory>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</control>`

Source	`<xs:element name="control" type="controlType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Defines the control element which contains data for`
	` administrative information about the XML-document.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

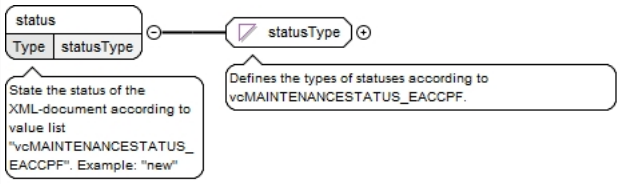
Element `controlType / recordId`

Namespace	employment
Annotations	`State a unique identity code belonging to the XML-document. Example: "F40219"`

Diagram	
Type	xs:string
Properties	content: simple

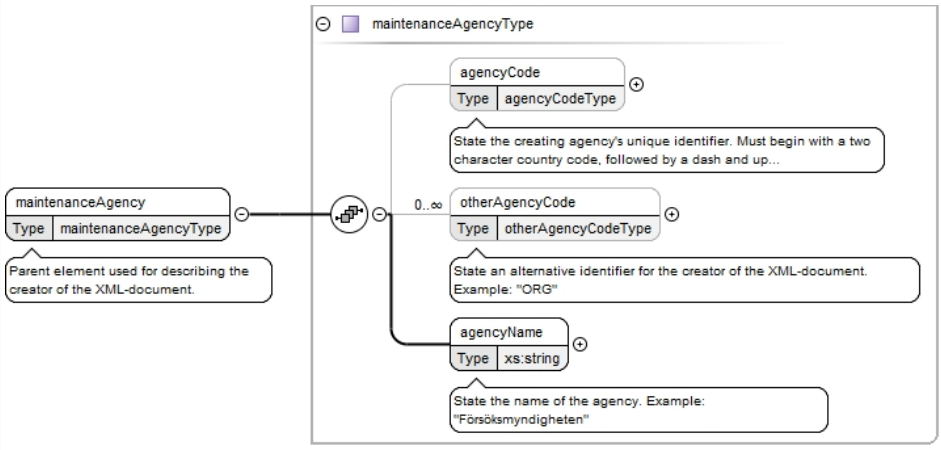
Source	`<xs:element name="recordId" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State a unique identity code belonging to the XML-document. Example: "F40219"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element controlType / status

Namespace	employment
Annotations	State the status of the XML-document according to value list "vcMAINTENANCESTATUS_EACCPF". Example: "new"
Diagram	
Type	statusType
Properties	content: simple
Facets	enumeration: cancelled enumeration: deleted enumeration: deletedReplaced enumeration: derived enumeration: new enumeration: revised

Source	`<xs:element name="status" type="statusType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the status of the XML-document according to value list "vcMAINTENANCESTATUS_EACCPF". Example: "new"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element controlType / maintenanceAgency

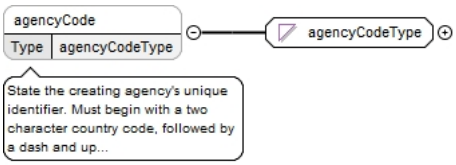
Namespace	employment
Annotations	Parent element used for describing the creator of the XML-document.
Diagram	

Type	maintenanceAgencyType
Properties	content: complex
Model	agencyCode{0,1} , otherAgencyCode* , agencyName
Children	agencyCode, agencyName, otherAgencyCode

Instance	`<maintenanceAgency xmlns="employment">`
	`  <agencyCode>{0,1}</agencyCode>`
	`  <otherAgencyCode agencyCodePrefix="" otherAgencyCodePrefix="">{0,unbounded}</otherAgencyCode>`
	`  <agencyName>{1,1}</agencyName>`
	`</maintenanceAgency>`

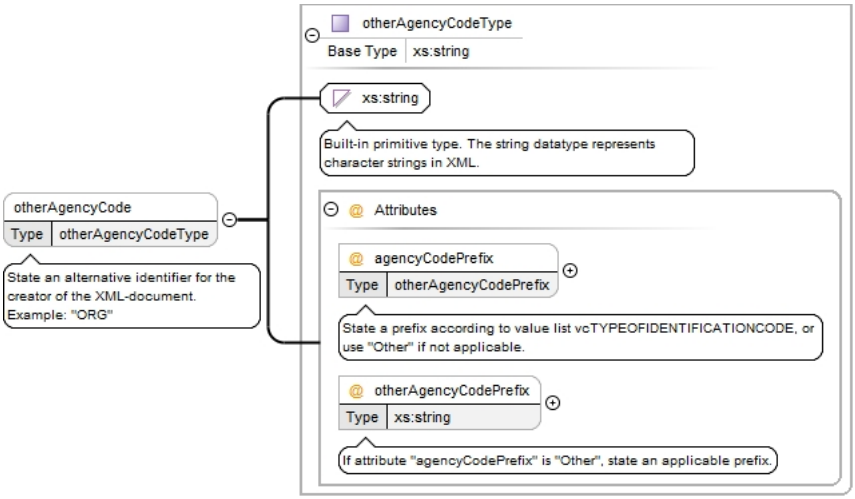
Source	`<xs:element name="maintenanceAgency" type="maintenanceAgencyType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element used for describing the creator of the XML-document.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `maintenanceAgencyType / agencyCode`

Namespace	employment
Annotations	State the creating agency's unique identifier. Must begin with a two character country code, followed by a dash and up to 10 characters. Example: "SE-A:ARA/123
Diagram	
Type	agencyCodeType
Properties	content: simple minOccurs: 0
Facets	pattern [A-Z]{2}\-[a-zA-Z0-9/:\-]{1,10}

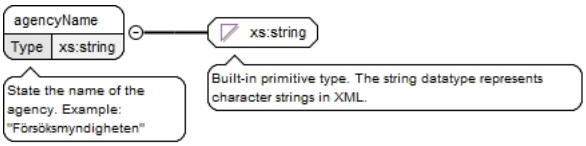
Source	`<xs:element name="agencyCode" type="agencyCodeType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the creating agency's unique identifier. Must begin with a two character country code, followed by a dash and up to 10 characters. Example: "SE-A:ARA/123</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `maintenanceAgencyType / otherAgencyCode`

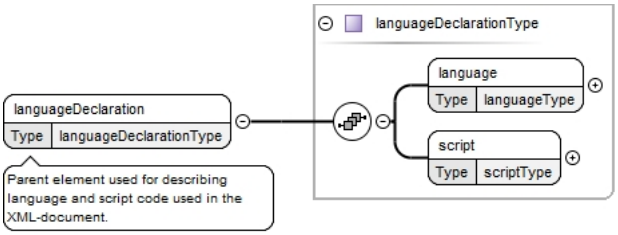
Namespace	employment
Annotations	State an alternative identifier for the creator of the XML-document. Example: "ORG"
Diagram	
Type	otherAgencyCodeType

Properties	content:	complex		
	minOccurs:	0		
	maxOccurs:	unbounded		
Attributes	**QName**	**Type**	**Use**	
	agencyCodePrefix	otherAgencyCodePrefix	optional	
		State a prefix according to value list vcTYPEOFIDENTIFICATIONCODE, or use "Other" if not applicable.		
	otherAgencyCodePrefix	xs:string	optional	
		If attribute "agencyCodePrefix" is "Other", state an applicable prefix.		
Source	<xs:element name="otherAgencyCode" minOccurs="0" maxOccurs="unbounded" type="otherAgencyCodeType"> <xs:annotation> <xs:documentation xml:lang="en">State an alternative identifier for the creator of the XML-document. Example: "ORG"</xs:documentation> </xs:annotation> </xs:element>			

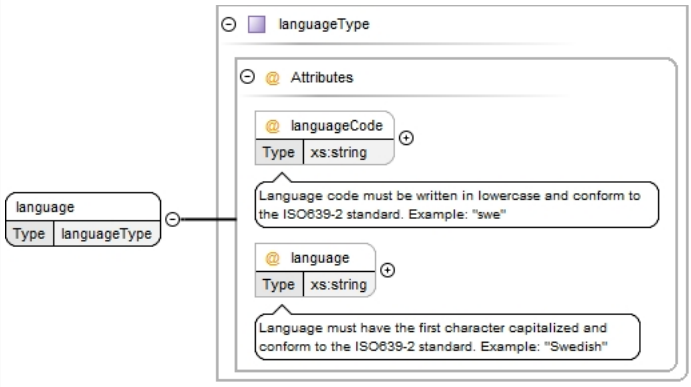
Element maintenanceAgencyType / agencyName

Namespace	employment
Annotations	State the name of the agency. Example: "Försöksmyndigheten"
Diagram	
Type	xs:string
Properties	content: simple
Source	<xs:element name="agencyName" type="xs:string"> <xs:annotation> <xs:documentation xml:lang="en">State the name of the agency. Example: "Försöksmyndigheten"</xs:documentation> </xs:annotation> </xs:element>

Element controlType / languageDeclaration

Namespace	employment
Annotations	Parent element used for describing language and script code used in the XML-document.
Diagram	
Type	languageDeclarationType
Properties	content: complex
Model	language , script
Children	language, script
Instance	<languageDeclaration xmlns="employment"> <language language="" languageCode="">{1,1}</language> <script script="" scriptCode="">{1,1}</script> </languageDeclaration>
Source	<xs:element name="languageDeclaration" type="languageDeclarationType"> <xs:annotation> <xs:documentation xml:lang="en">Parent element used for describing language and script code used in the XML-document.</xs:documentation> </xs:annotation> </xs:element>

Element `languageDeclarationType / language`

Namespace	employment
Diagram	
Type	languageType
Properties	content: `complex`

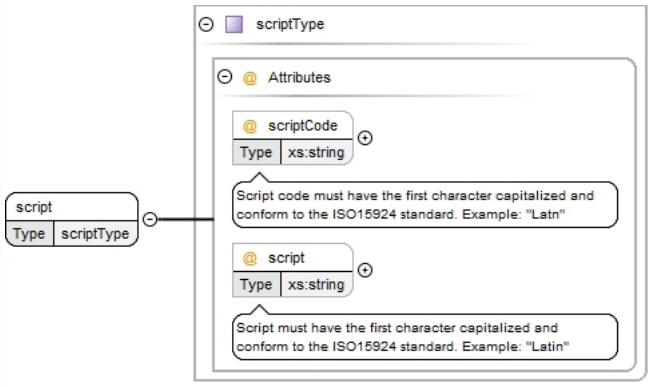
Attributes

QName	Type	Use	
language	xs:string	optional	
	Language must have the first character capitalized and conform to the ISO639-2 standard. Example: "Swedish"		
languageCode	xs:string	optional	
	Language code must be written in lowercase and conform to the ISO639-2 standard. Example: "swe"		

Source

```
<xs:element name="language" type="languageType"/>
```

Element `languageDeclarationType / script`

Namespace	employment
Diagram	
Type	scriptType
Properties	content: `complex`

Attributes

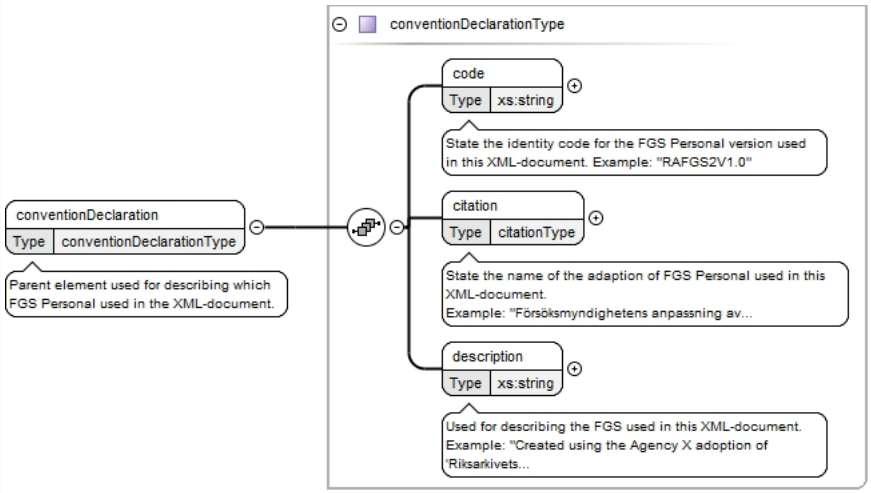
QName	Type	Use	
script	xs:string	optional	
	Script must have the first character capitalized and conform to the ISO15924 standard. Example: "Latin"		
scriptCode	xs:string	optional	
	Script code must have the first character capitalized and conform to the ISO15924 standard. Example: "Latn"		

Source

```
<xs:element name="script" type="scriptType"/>
```

Element `controlType / conventionDeclaration`

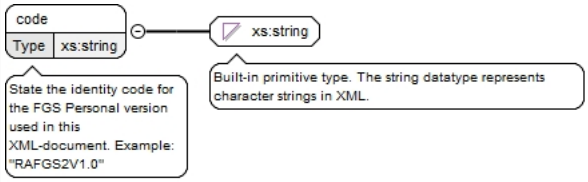
Namespace	employment

Annotations	Parent element used for describing which FGS Personal used in the XML-document.
Diagram	
Type	conventionDeclarationType
Properties	content: complex
Model	code , citation , description
Children	citation, code, description

Instance	`<conventionDeclaration xmlns="employment">`
	`  <code>{1,1}</code>`
	`  <citation href="">{1,1}</citation>`
	`  <description>{1,1}</description>`
	`</conventionDeclaration>`

Source	`<xs:element name="conventionDeclaration" type="conventionDeclarationType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element used for describing which FGS Personal used in the XML-document.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

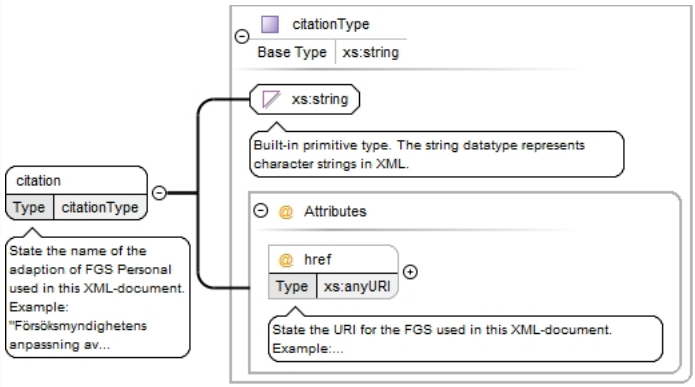
Element conventionDeclarationType / code

Namespace	employment
Annotations	State the identity code for the FGS Personal version used in this XML-document. Example: "RAFGS2V1.0"
Diagram	
Type	xs:string
Properties	content: simple

Source	`<xs:element name="code" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the identity code for the FGS Personal version used in this XML-document. Example: "RAFGS2V1.0"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element conventionDeclarationType / citation

Namespace	employment
Annotations	State the name of the adaption of FGS Personal used in this XML-document. Example: "Försöksmyndighetens anpassning av FGS Personal för överföring av personalinformation till det egna E-arktivet"

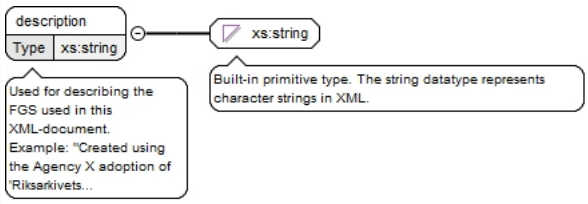
Diagram	
Type	citationType
Properties	content: `complex`
Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional `State the URI for the FGS used in this XML-document. Example: "https://myndighet.se/Försöksmyndighetens_FGS_Personal_V_20181224.pdf"`

Source	`<xs:element name="citation" type="citationType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the name of the adaption of FGS Personal used in this XML-document.</xs:documentation>`
	`  <xs:documentation xml:lang="en">Example: "Försöksmyndighetens anpassning av FGS Personal för överföring av personalinformation till det egna E-arktivet"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `conventionDeclarationType / description`

Namespace	employment

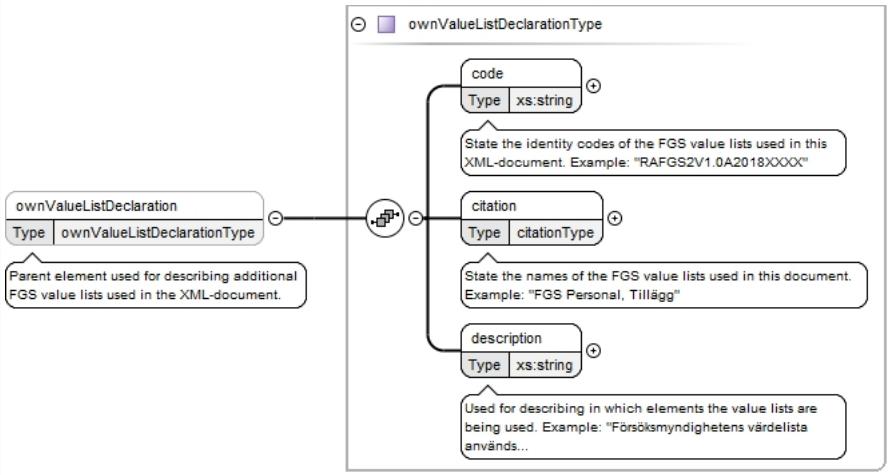
Annotations	`Used for describing the FGS used in this XML-document.`
	
	`Example: "Created using the Agency X adoption of 'Riksarkivets anpassning av FGS Personal'"`

Diagram	
Type	xs:string
Properties	content: `simple`

Source	`<xs:element name="description" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for describing the FGS used in this XML-document.</xs:documentation>`
	`  <xs:documentation xml:lang="en">Example: "Created using the Agency X adoption of 'Riksarkivets anpassning av FGS Personal'"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `controlType / ownValueListDeclaration`

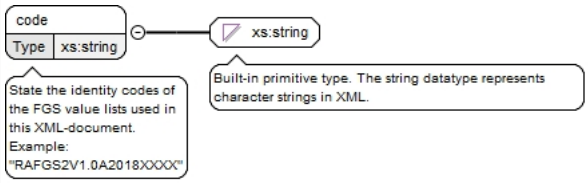
Namespace	employment
Annotations	`Parent element used for describing additional FGS value lists used in the XML-document.`

Diagram	
Type	ownValueListDeclarationType
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	code , citation , description
Children	citation, code, description

Instance	`<ownValueListDeclaration xmlns="employment">`
	`  <code>{1,1}</code>`
	`  <citation href="">{1,1}</citation>`
	`  <description>{1,1}</description>`
	`</ownValueListDeclaration>`

Source	`<xs:element name="ownValueListDeclaration" type="ownValueListDeclarationType" minOccurs="0" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element used for describing additional FGS value lists used in the XML-document.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

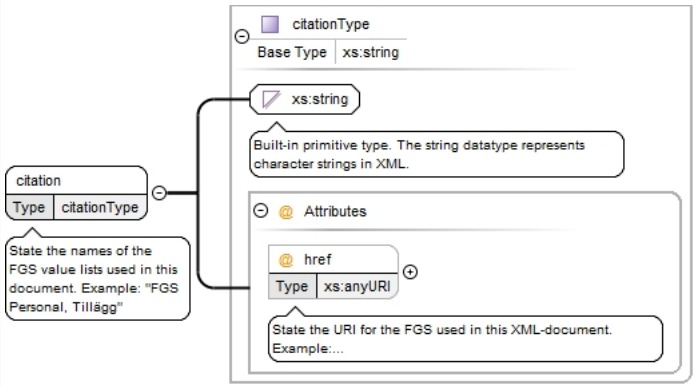
Element ownValueListDeclarationType / code

Namespace	employment
Annotations	State the identity codes of the FGS value lists used in this XML-document. Example: "RAFGS2V1.0A2018XXXX"
Diagram	
Type	xs:string
Properties	content: simple

Source	`<xs:element name="code" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the identity codes of the FGS value lists used in this XML-document. Example: "RAFGS2V1.0A2018XXXX"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

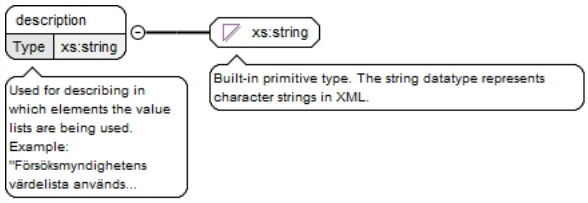
Element ownValueListDeclarationType / citation

Namespace	employment
Annotations	State the names of the FGS value lists used in this document. Example: "FGS Personal, Tillägg"

Diagram	
Type	citationType
Properties	content: `complex`
Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional `State the URI for the FGS used in this XML-document. Example: "https://myndighet.se/Försöksmyndighetens_FGS_Personal_V_20181224.pdf"`

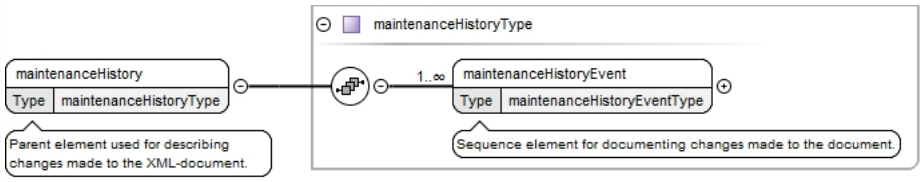
Source	`<xs:element name="citation" type="citationType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the names of the FGS value lists used in this document. Example: "FGS Personal, Tillägg"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `ownValueListDeclarationType / description`

Namespace	employment
Annotations	`Used for describing in which elements the value lists are being used. Example: "Försöksmyndighetens värdelista används i elementet AnhörigFält"`
Diagram	
Type	xs:string
Properties	content: `simple`

Source	`<xs:element name="description" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for describing in which elements the value lists are being used. Example: "Försöksmyndighetens värdelista används i elementet AnhörigFält"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `controlType / maintenanceHistory`

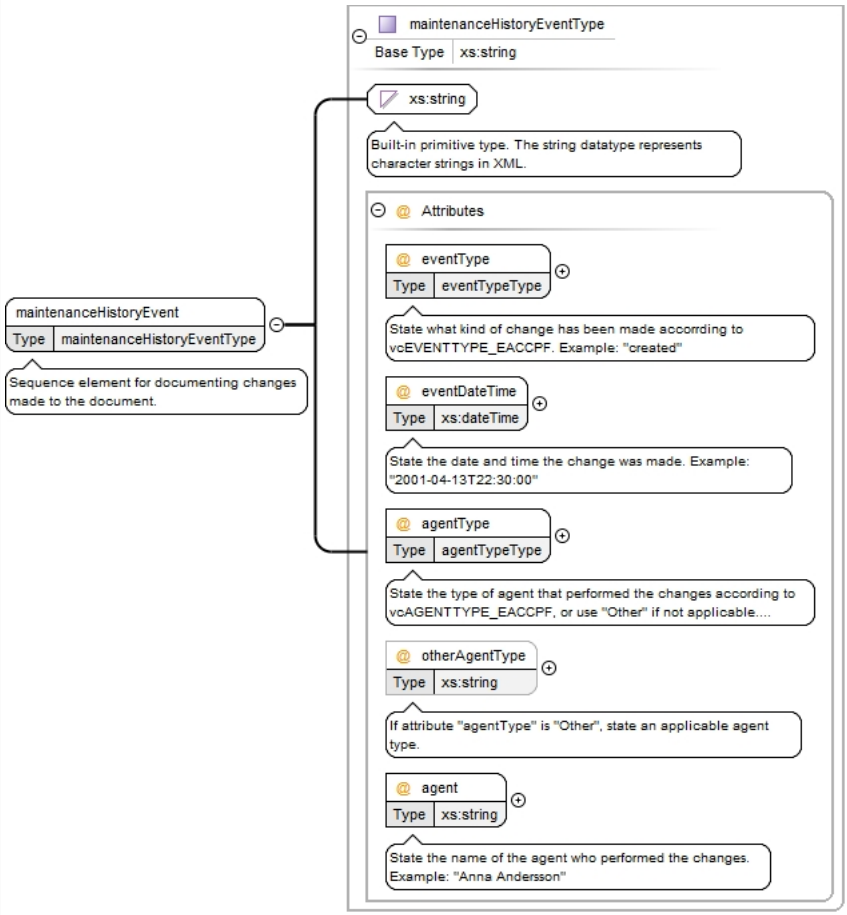
Namespace	employment
Annotations	`Parent element used for describing changes made to the XML-document.`
Diagram	
Type	maintenanceHistoryType
Properties	content: `complex`
Model	maintenanceHistoryEvent+

Children	maintenanceHistoryEvent

Instance	`<maintenanceHistory xmlns="employment">`
	`  <maintenanceHistoryEvent agent="" agentType="" eventDateTime="" eventType="" otherAgentType="">{1,unbounded}</maintenanceHistoryEvent>`
	`</maintenanceHistory>`

Source	`<xs:element name="maintenanceHistory" type="maintenanceHistoryType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element used for describing changes made to the XML-document.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `maintenanceHistoryType / maintenanceHistoryEvent`

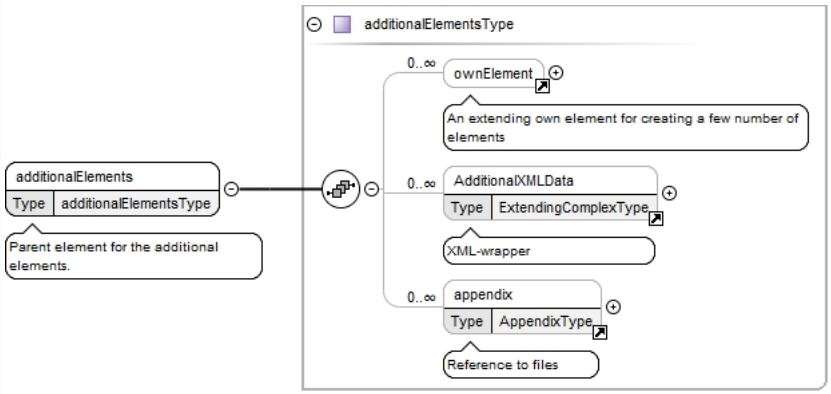
Namespace	employment
Annotations	Sequence element for documenting changes made to the document.
Diagram	
Type	maintenanceHistoryEventType
Properties	content: `complex` minOccurs: `1` maxOccurs: `unbounded`
Attributes	**QName** / **Type** / **Use** **agent** / xs:string / required State the name of the agent who performed the changes. Example: "Anna Andersson" **agentType** / agentTypeType / required State the type of agent that performed the changes according to vcAGENTTYPE_EACCPF, or use "Other" if not applicable. Example: "human" **eventDateTime** / xs:dateTime / required State the date and time the change was made. Example: "2001-04-13T22:30:00"

	QName	Type	Use	
	eventType	eventTypeType	required	
		State what kind of change has been made accorrding to vcEVENTTYPE_EACCPF. Example: "created"		
	otherAgentType	xs:string	optional	
		If attribute "agentType" is "Other", state an applicable agent type.		

Source

```
<xs:element name="maintenanceHistoryEvent" type="maintenanceHistoryEventType" minOccurs="1"
 maxOccurs="unbounded">
  <xs:annotation>
    <xs:documentation xml:lang="en">Sequence element for documenting changes made to the document.</
xs:documentation>
  </xs:annotation>
</xs:element>
```

Element additionalElements

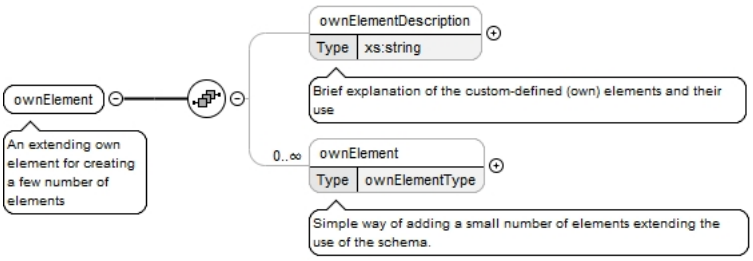
Namespace	employment
Annotations	Parent element for the additional elements.
Diagram	
Type	additionalElementsType
Properties	content: complex
Used by	Complex Types: adjustmentType, childType, contactInformationEntryType, controlType, deathRelatedDocumentType, disciplinaryActionType, employeeDescriptionType, nextOfKinEntryType, optionalTaxEntryType, reassignmentType, rehabilitationArrangementType, salarySeizureType, taxTableType, workRelatedAidGrantType, workRelatedInjuryType
Model	ownElement* , AdditionalXMLData* , appendix*
Children	AdditionalXMLData, appendix, ownElement

Instance	`<additionalElements xmlns="employment">`
	`  <ownElement>{0,unbounded}</ownElement>`
	`  <AdditionalXMLData>{0,unbounded}</AdditionalXMLData>`
	`  <appendix description="" disposable="" eSignatureHaveExisted="" fileFormat="" name="" originalFileFormat="" path=`
	`appendix>`
	`</additionalElements>`

Source	`<xs:element name="additionalElements" type="additionalElementsType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for the additional elements.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element ownElement

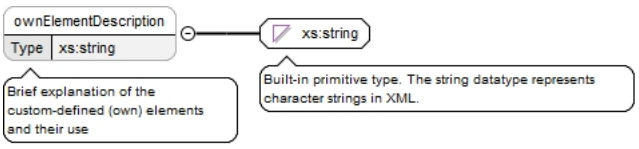
Namespace	employment
Annotations	An extending own element for creating a few number of elements

Diagram	
Properties	content: complex
Used by	Complex Type additionalElementsType
Model	ownElementDescription{0,1} , ownElement*
Children	ownElement, ownElementDescription

Instance	`<ownElement xmlns="employment">`
	`  <ownElementDescription>{0,1}</ownElementDescription>`
	`  <ownElement dataType="" format="" name="">{0,unbounded}</ownElement>`
	`</ownElement>`

Source	`<xs:element name="ownElement">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="sv">An extending own element for creating a few number of elements</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element name="ownElementDescription" minOccurs="0" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Brief explanation of the custom-defined (own) elements and their use</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element name="ownElement" type="ownElementType" minOccurs="0" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Simple way of adding a small number of elements extending the use of the schema.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

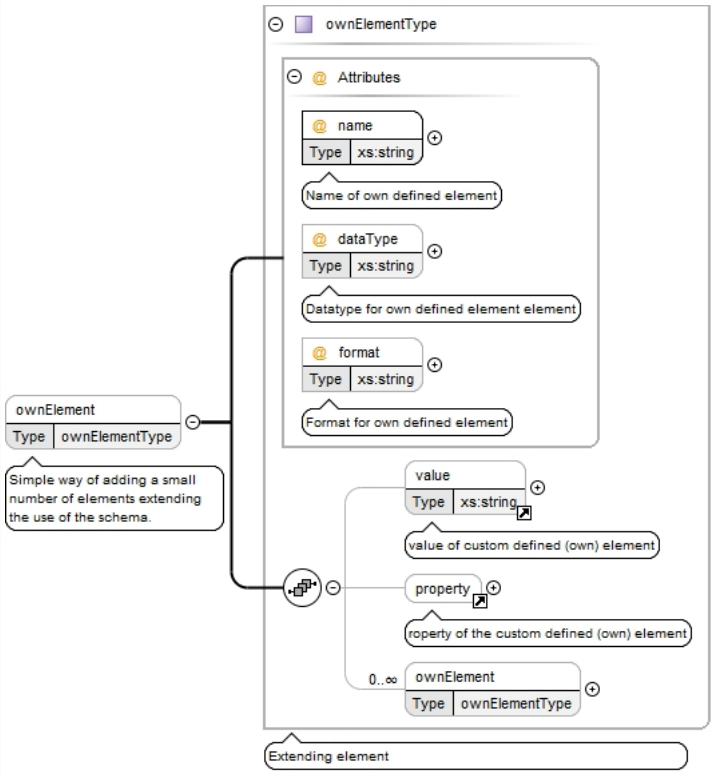
Element ownElement / ownElementDescription

Namespace	employment
Annotations	Brief explanation of the custom-defined (own) elements and their use
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="ownElementDescription" minOccurs="0" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Brief explanation of the custom-defined (own) elements and their use</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element ownElement / ownElement

Namespace	employment
Annotations	Simple way of adding a small number of elements extending the use of the schema.

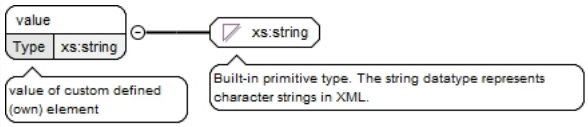
Diagram	
Type	ownElementType
Properties	content: `complex` minOccurs: `0` maxOccurs: `unbounded`
Model	value{0,1} , property{0,1} , ownElement*
Children	ownElement, property, value

Instance	`<ownElement dataType="" format="" name="" xmlns="employment">`
	`  <value>{0,1}</value>`
	`  <property>{0,1}</property>`
	`  <ownElement dataType="" format="" name="">{0,unbounded}</ownElement>`
	`</ownElement>`

Attributes	**QName** / **Type** / **Use**
	dataType / xs:string / optional
	`Datatype for own defined element element`
	format / xs:string / optional
	`Format for own defined element`
	name / xs:string / required
	`Name of own defined element`

Source	`<xs:element name="ownElement" type="ownElementType" minOccurs="0" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Simple way of adding a small number of elements extending the use of the schema.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

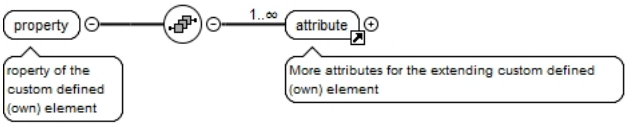
Element `value`

Namespace	employment
Annotations	`value of custom defined (own) element`
Diagram	

Type	xs:string
Properties	content: `simple`
Used by	Element: attribute Complex Type: ownElementType

Source	`<xs:element name="value" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">value of custom defined (own) element</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

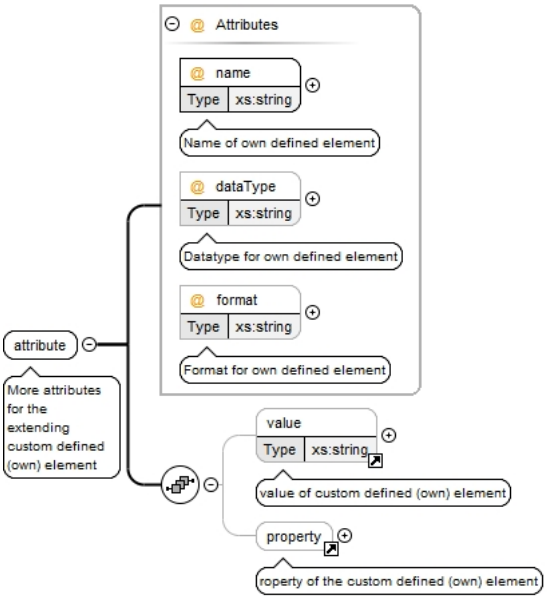
Element `property`

Namespace	employment
Annotations	`roperty of the custom defined (own) element`
Diagram	
Properties	content: `complex`
Used by	Element: attribute Complex Type: ownElementType
Model	attribute+
Children	attribute

Instance	`<property xmlns="employment">`
	`  <attribute dataType="" format="" name="">{1,unbounded}</attribute>`
	`</property>`

Source	`<xs:element name="property">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">roperty of the custom defined (own) element</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:complexType>`
	`  <xs:sequence>`
	`  <xs:element ref="attribute" maxOccurs="unbounded"/>`
	`  </xs:sequence>`
	`  </xs:complexType>`
	`</xs:element>`

Element `attribute`

Namespace	employment
Annotations	`More attributes for the extending custom defined (own) element`
Diagram	

Properties	content: complex
Used by	Element property
Model	value{0,1} , property{0,1}
Children	property, value

Instance	`<attribute dataType="" format="" name="" xmlns="employment">`
	`  <value>{0,1}</value>`
	`  <property>{0,1}</property>`
	`</attribute>`

Attributes	

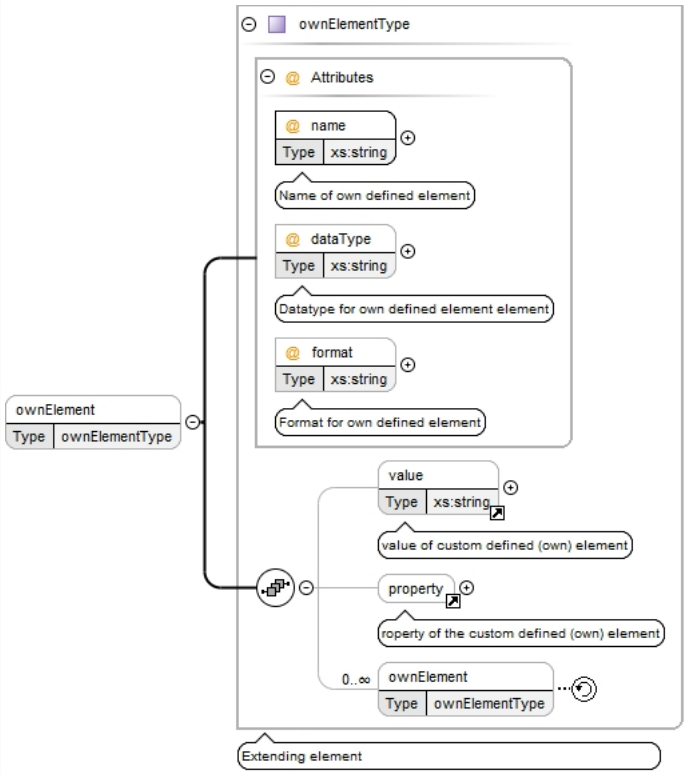
QName	Type	Use	
dataType	xs:string	optional	
	Datatype for own defined element		
format	xs:string	optional	
	Format for own defined element		
name	xs:string	required	
	Name of own defined element		

Source

```
<xs:element name="attribute">
  <xs:annotation>
    <xs:documentation xml:lang="sv">More attributes for the extending custom defined (own) element</
xs:documentation>
  </xs:annotation>
  <xs:complexType>
    <xs:sequence>
      <xs:element ref="value" minOccurs="0"/>
      <xs:element ref="property" minOccurs="0"/>
    </xs:sequence>
    <xs:attribute name="name" type="xs:string" use="required">
      <xs:annotation>
        <xs:documentation xml:lang="en">Name of own defined element</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="dataType" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="en">Datatype for own defined element</xs:documentation>
      </xs:annotation>
    </xs:attribute>
    <xs:attribute name="format" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="en">Format for own defined element</xs:documentation>
      </xs:annotation>
    </xs:attribute>
  </xs:complexType>
</xs:element>
```

Element ownElementType / ownElement

Namespace	employment

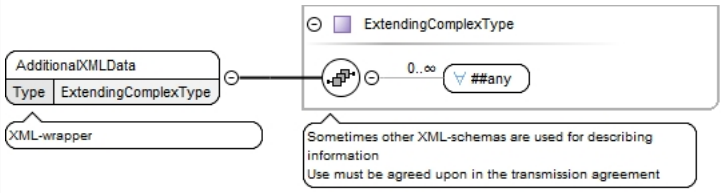
Diagram	
Type	ownElementType
Properties	content: `complex` minOccurs: `0` maxOccurs: `unbounded`
Model	value{0,1} , property{0,1} , ownElement*
Children	ownElement, property, value

Instance	`<ownElement dataType="" format="" name="" xmlns="employment">`
	`  <value>{0,1}</value>`
	`  <property>{0,1}</property>`
	`  <ownElement dataType="" format="" name="">{0,unbounded}</ownElement>`
	`</ownElement>`

Attributes	**QName** / **Type** / **Use**
	dataType / xs:string / optional
	`Datatype for own defined element element`
	format / xs:string / optional
	`Format for own defined element`
	name / xs:string / required
	`Name of own defined element`

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

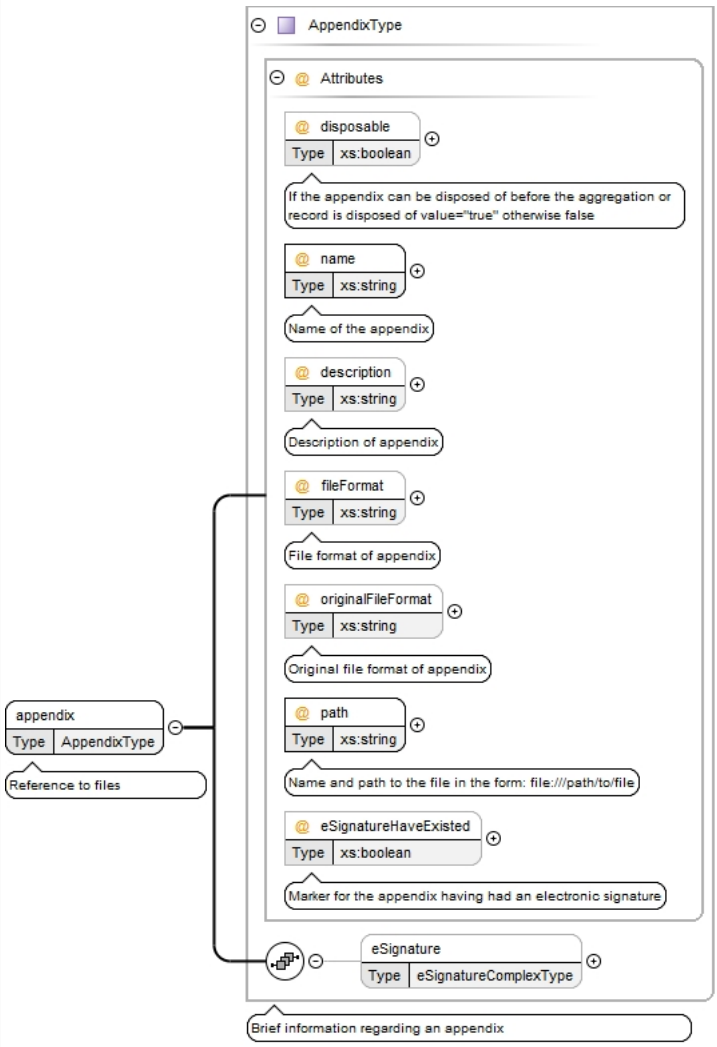
Element `AdditionalXMLData`

Namespace	employment
Annotations	`XML-wrapper`
Diagram	
Type	ExtendingComplexType

Properties	content:	complex
Used by	Complex Type	additionalElementsType
Model	ANY element from ANY namespace	

Source	`<xs:element name="AdditionalXMLData" type="ExtendingComplexType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">XML-wrapper</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`	

Element appendix

Namespace	employment
Annotations	Reference to files
Diagram	
Type	AppendixType
Properties	content: complex
Used by	Complex Type additionalElementsType
Model	eSignature{0,1}
Children	eSignature

Instance	`<appendix description="" disposable="" eSignatureHaveExisted="" fileFormat="" name="" originalFileFormat="" path=""`
	`  <eSignature dateESignatureIsVerified="" present="">{0,1}</eSignature>`
	`</appendix>`

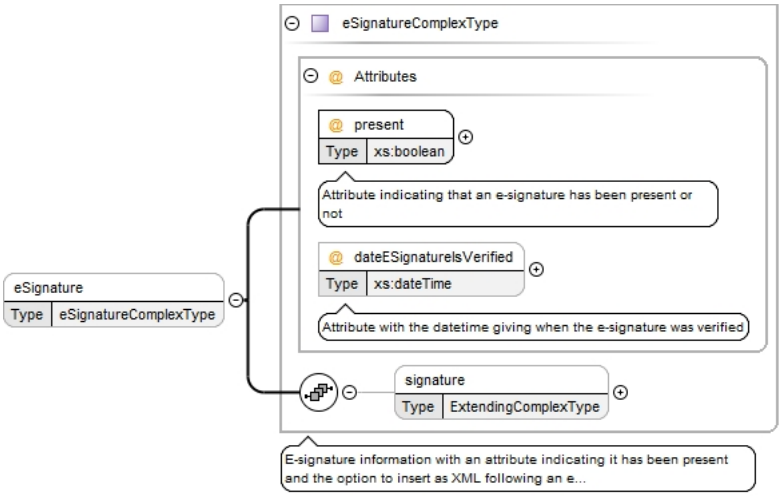
Attributes		

QName	Type	Use	
description	xs:string	optional	
	Description of appendix		

	QName	**Type**	**Use**	
	disposable	xs:boolean	optional	
		`If the appendix can be disposed of before the aggregation or record is disposed of value="true" otherwise false`		
	eSignatureHaveExisted	xs:boolean	optional	
		`Marker for the appendix having had an electronic signature`		
	fileFormat	xs:string	optional	
		`File format of appendix`		
	name	xs:string	required	
		`Name of the appendix`		
	originalFileFormat	xs:string	optional	
		`Original file format of appendix`		
	path	xs:string	required	
		`Name and path to the file in the form: file:///path/to/file`		

Source	`<xs:element name="appendix" type="AppendixType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Reference to files</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`			

Element `AppendixType / eSignature`

Namespace	employment
Diagram	
Type	eSignatureComplexType
Properties	content: `complex` minOccurs: `0`
Model	signature{0,1}
Children	signature

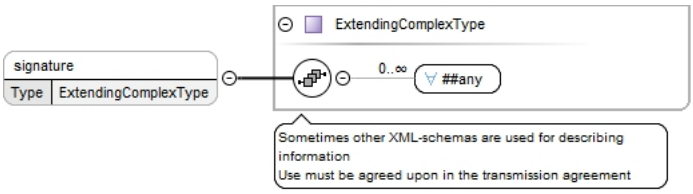
Instance	`<eSignature dateESignatureIsVerified="" present="" xmlns="employment">`
	`  <signature>{0,1}</signature>`
	`</eSignature>`

Attributes	**QName** / **Type** / **Use**
	dateESignatureIsVerified — xs:dateTime — optional
	`Attribute with the datetime giving when the e-signature was verified`
	present — xs:boolean — required
	`Attribute indicating that an e-signature has been present or not`

Source	`<xs:element name="eSignature" type="eSignatureComplexType" minOccurs="0"/>`

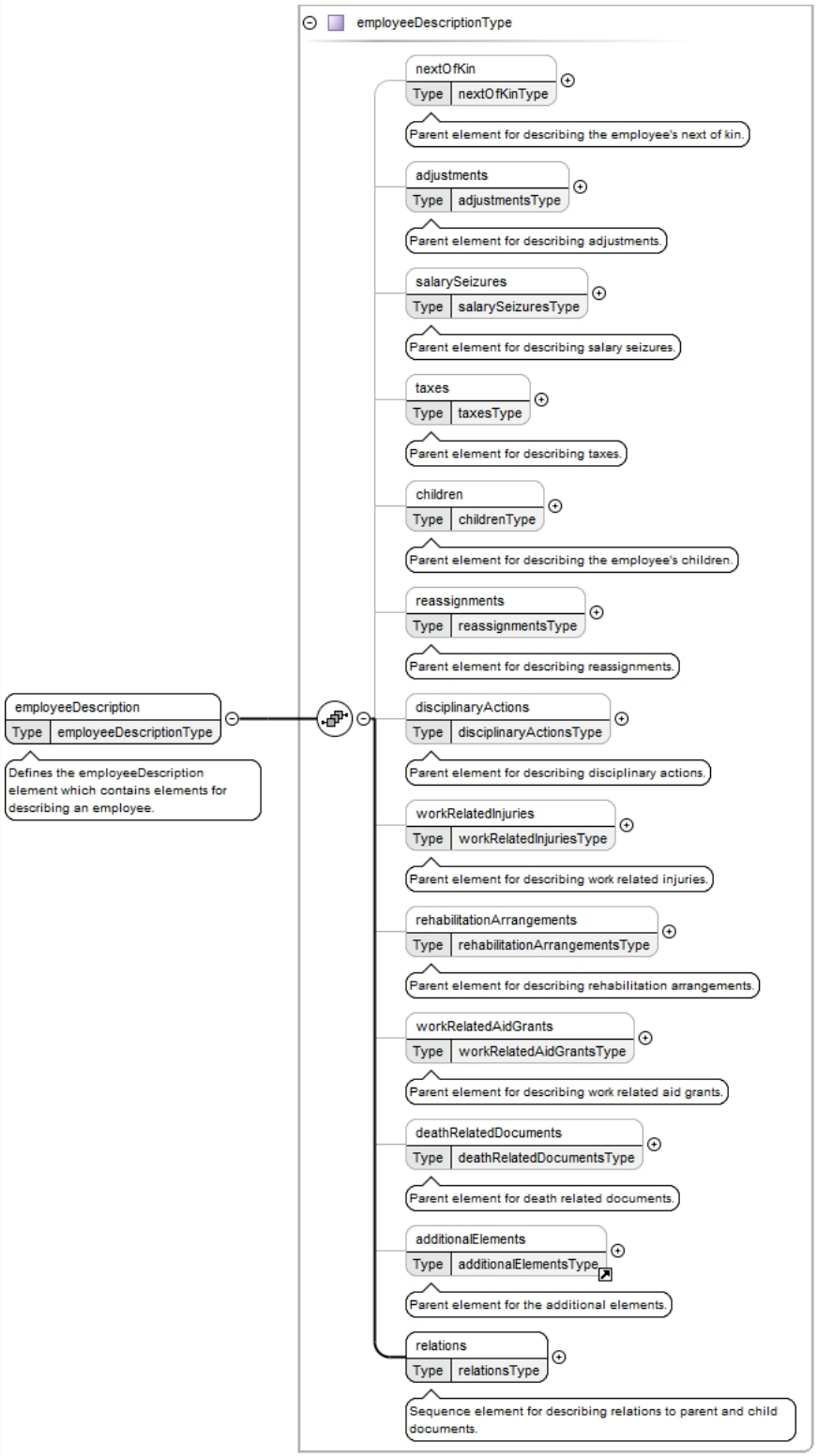
Element `eSignatureComplexType / signature`

Namespace	employment

Diagram	
Type	ExtendingComplexType
Properties	content: `complex` minOccurs: `0`
Model	ANY element from ANY namespace
Source	`<xs:element name="signature" type="ExtendingComplexType" minOccurs="0"/>`

Element `employeeType / employeeDescription`

Namespace	employment
Annotations	`Defines the employeeDescription element which contains elements for describing an employee.`

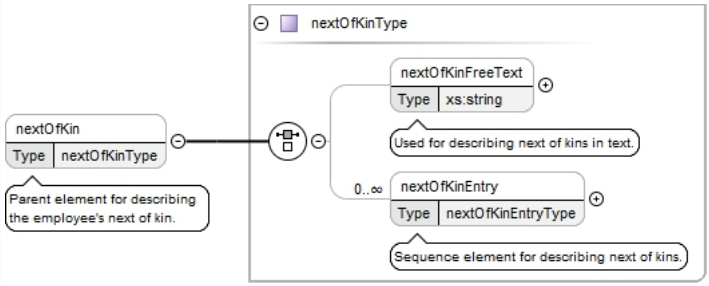
Diagram	
Type	employeeDescriptionType
Properties	content: `complex`
Model	nextOfKin{0,1} , adjustments{0,1} , salarySeizures{0,1} , taxes{0,1} , children{0,1} , reassignments{0,1} , disciplinaryActions{0,1} , workRelatedInjuries{0,1} , rehabilitationArrangements{0,1} , workRelatedAidGrants{0,1} , deathRelatedDocuments{0,1} , additionalElements{0,1} , relations
Children	additionalElements, adjustments, children, deathRelatedDocuments, disciplinaryActions, nextOfKin, reassignments, rehabilitationArrangements, relations, salarySeizures, taxes, workRelatedAidGrants, workRelatedInjuries
Instance	`<employeeDescription xmlns="employment">` `  <nextOfKin>{0,1}</nextOfKin>`

```
  <adjustments>{0,1}</adjustments>
  <salarySeizures>{0,1}</salarySeizures>
  <taxes>{0,1}</taxes>
  <children>{0,1}</children>
  <reassignments>{0,1}</reassignments>
  <disciplinaryActions>{0,1}</disciplinaryActions>
  <workRelatedInjuries>{0,1}</workRelatedInjuries>
  <rehabilitationArrangements>{0,1}</rehabilitationArrangements>
  <workRelatedAidGrants>{0,1}</workRelatedAidGrants>
  <deathRelatedDocuments>{0,1}</deathRelatedDocuments>
  <additionalElements>{0,1}</additionalElements>
  <relations>{1,1}</relations>
</employeeDescription>
```

Source

```
<xs:element name="employeeDescription" type="employeeDescriptionType">
  <xs:annotation>
    <xs:documentation xml:lang="en">Defines the employeeDescription element which contains elements
 for describing an employee.</xs:documentation>
  </xs:annotation>
</xs:element>
```

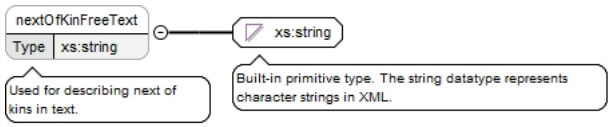
Element employeeDescriptionType / nextOfKin

Namespace	employment
Annotations	Parent element for describing the employee's next of kin.
Diagram	
Type	nextOfKinType
Properties	content: complex minOccurs: 0
Model	nextOfKinFreeText{0,1} | nextOfKinEntry*
Children	nextOfKinEntry, nextOfKinFreeText

Instance	`<nextOfKin xmlns="employment">`
	`  <nextOfKinFreeText>{0,1}</nextOfKinFreeText>`
	`  <nextOfKinEntry otherTypeOfRelation="" typeOfRelation="">{0,unbounded}</nextOfKinEntry>`
	`</nextOfKin>`

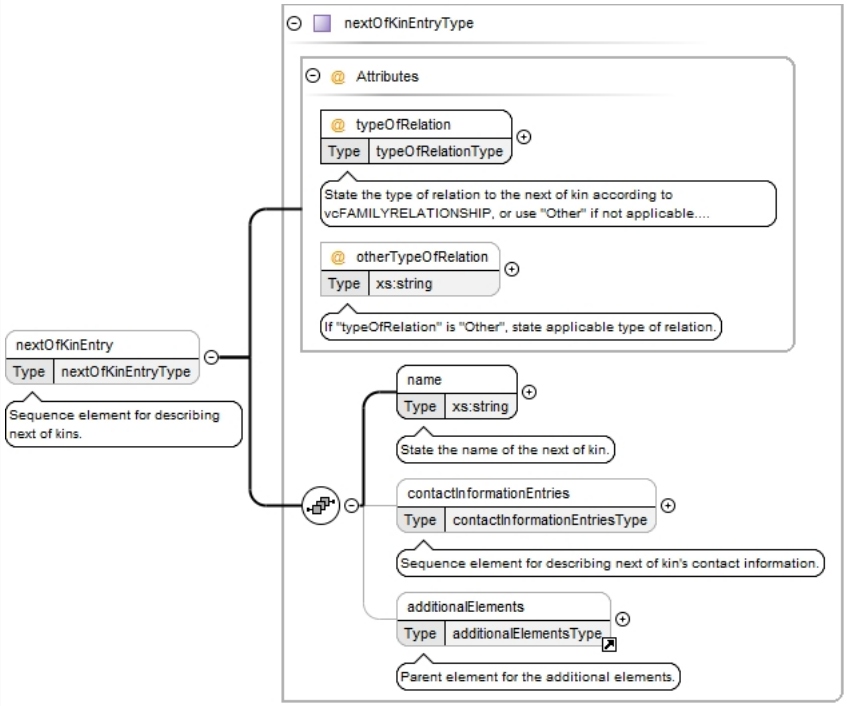
Source	`<xs:element name="nextOfKin" type="nextOfKinType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for describing the employee's next of kin.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element nextOfKinType / nextOfKinFreeText

Namespace	employment
Annotations	Used for describing next of kins in text.
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="nextOfKinFreeText" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for describing next of kins in text.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element nextOfKinType / nextOfKinEntry

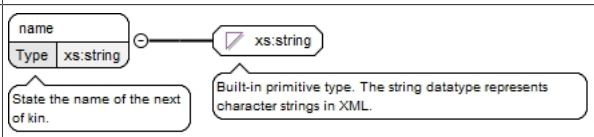
Namespace	employment
Annotations	Sequence element for describing next of kins.
Diagram	
Type	nextOfKinEntryType
Properties	content: complex minOccurs: 0 maxOccurs: unbounded
Model	name , contactInformationEntries{0,1} , additionalElements{0,1}
Children	additionalElements, contactInformationEntries, name

Instance	`<nextOfKinEntry otherTypeOfRelation="" typeOfRelation="" xmlns="employment">`
	`  <name>{1,1}</name>`
	`  <contactInformationEntries>{0,1}</contactInformationEntries>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</nextOfKinEntry>`

Attributes	**QName** / **Type** / **Use** **otherTypeOfRelation** / xs:string / optional If "typeOfRelation" is "Other", state applicable type of relation. **typeOfRelation** / typeOfRelationType / required State the type of relation to the next of kin according to vcFAMILYRELATIONSHIP, or use "Other" if not applicable. Example "child"

Source	`<xs:element name="nextOfKinEntry" type="nextOfKinEntryType" minOccurs="0" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing next of kins.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

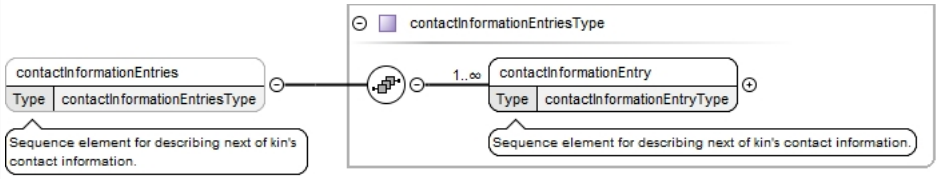
Element nextOfKinEntryType / name

Namespace	employment
Annotations	State the name of the next of kin.
Diagram	

Type	xs:string
Properties	content: simple

Source	`<xs:element name="name" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the name of the next of kin.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

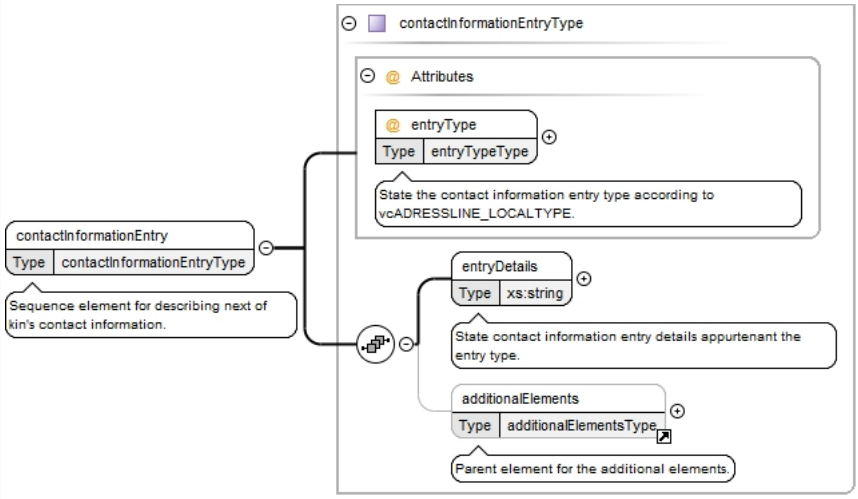
Element nextOfKinEntryType / contactInformationEntries

Namespace	employment
Annotations	Sequence element for describing next of kin's contact information.
Diagram	
Type	contactInformationEntriesType
Properties	content: complex minOccurs: 0
Model	contactInformationEntry+
Children	contactInformationEntry

Instance	`<contactInformationEntries xmlns="employment">`
	`  <contactInformationEntry entryType="">{1,unbounded}</contactInformationEntry>`
	`</contactInformationEntries>`

Source	`<xs:element name="contactInformationEntries" type="contactInformationEntriesType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing next of kin's contact`
	` information.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element contactInformationEntriesType / contactInformationEntry

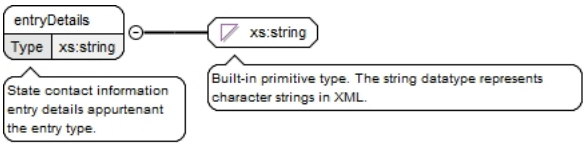
Namespace	employment
Annotations	Sequence element for describing next of kin's contact information.
Diagram	
Type	contactInformationEntryType
Properties	content: complex maxOccurs: unbounded
Model	entryDetails , additionalElements{0,1}
Children	additionalElements, entryDetails

Instance	`<contactInformationEntry entryType="" xmlns="employment">`
	`  <entryDetails>{1,1}</entryDetails>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</contactInformationEntry>`

Attributes	**QName** / **Type** / **Use** **entryType** / entryTypeType / required State the contact information entry type according to vcADRESSLINE_LOCALTYPE.

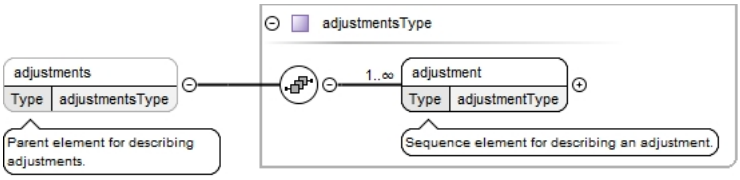
Source	`<xs:element name="contactInformationEntry" type="contactInformationEntryType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing next of kin's contact information.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element contactInformationEntryType / entryDetails

Namespace	employment
Annotations	State contact information entry details appurtenant the entry type.
Diagram	
Type	xs:string
Properties	content: simple

Source	`<xs:element name="entryDetails" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State contact information entry details appurtenant the entry type.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element employeeDescriptionType / adjustments

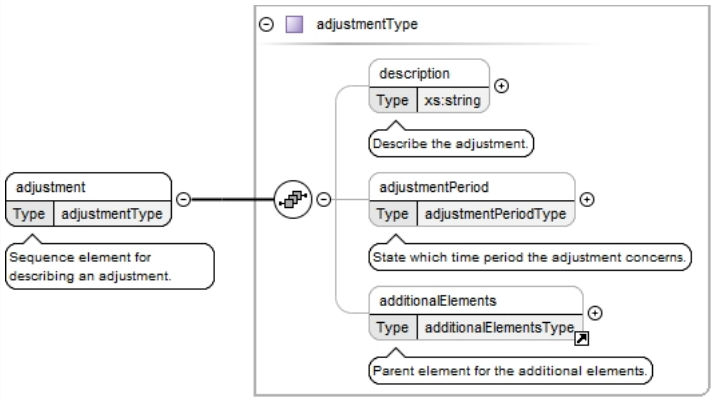
Namespace	employment
Annotations	Parent element for describing adjustments.
Diagram	
Type	adjustmentsType
Properties	content: complex minOccurs: 0
Model	adjustment+
Children	adjustment

Instance	`<adjustments xmlns="employment">`
	`  <adjustment>{1,unbounded}</adjustment>`
	`</adjustments>`

Source	`<xs:element name="adjustments" type="adjustmentsType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for describing adjustments.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element adjustmentsType / adjustment

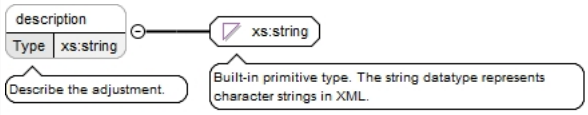
Namespace	employment
Annotations	Sequence element for describing an adjustment.

Diagram	
Type	adjustmentType
Properties	content: complex maxOccurs: unbounded
Model	description{0,1} , adjustmentPeriod{0,1} , additionalElements{0,1}
Children	additionalElements, adjustmentPeriod, description

Instance	`<adjustment xmlns="employment">`
	`  <description>{0,1}</description>`
	`  <adjustmentPeriod>{0,1}</adjustmentPeriod>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</adjustment>`

Source	`<xs:element name="adjustment" type="adjustmentType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing an adjustment.</`
	`xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

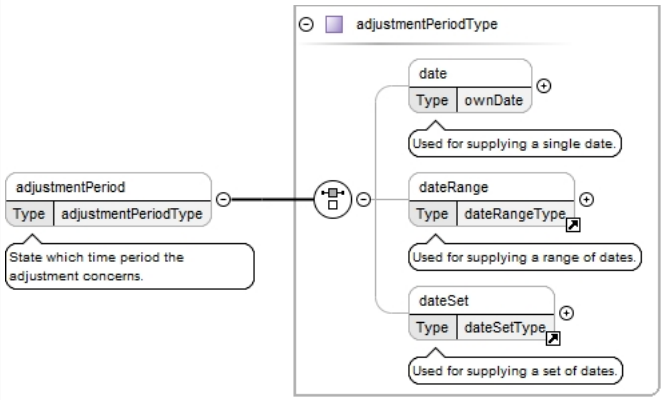
Element adjustmentType / description

Namespace	employment
Annotations	Describe the adjustment.
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the adjustment.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element adjustmentType / adjustmentPeriod

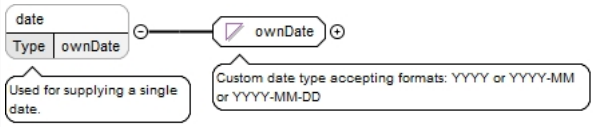
Namespace	employment
Annotations	State which time period the adjustment concerns.

Diagram	
Type	adjustmentPeriodType
Properties	content: `complex` minOccurs: `0`
Model	date{0,1} | dateRange{0,1} | dateSet{0,1}
Children	date, dateRange, dateSet

Instance	`<adjustmentPeriod xmlns="employment">`
	`  <date>{0,1}</date>`
	`  <dateRange>{0,1}</dateRange>`
	`  <dateSet>{0,1}</dateSet>`
	`</adjustmentPeriod>`

Source	`<xs:element name="adjustmentPeriod" type="adjustmentPeriodType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State which time period the adjustment concerns.</`
	`xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

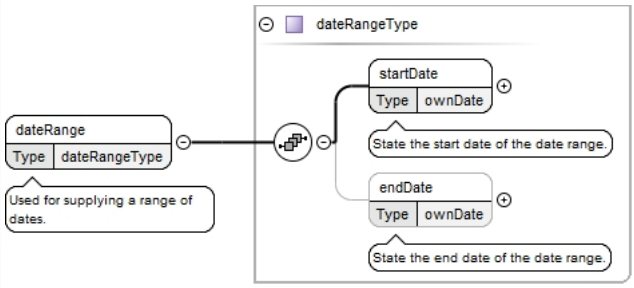
Element `adjustmentPeriodType / date`

Namespace	employment
Annotations	`Used for supplying a single date.`
Diagram	
Type	ownDate
Properties	content: `simple` minOccurs: `0`
Facets	pattern `[\d]{4}\-[\d]{2}\-[\d]{2}|[\d]{4}\-[\d]{2}|[\d]{4}`

Source	`<xs:element name="date" minOccurs="0" type="ownDate">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for supplying a single date.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `dateRange`

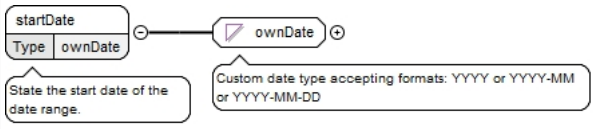
Namespace	employment
Annotations	`Used for supplying a range of dates.`

Diagram	
Type	dateRangeType
Properties	content: `complex`
Used by	Complex Types adjustmentPeriodType, dateSetType, seizurePeriodType
Model	startDate , endDate{0,1}
Children	endDate, startDate

Instance	`<dateRange xmlns="employment">`
	`  <startDate>{1,1}</startDate>`
	`  <endDate>{0,1}</endDate>`
	`</dateRange>`

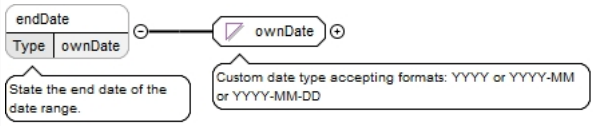
Source	`<xs:element name="dateRange" type="dateRangeType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for supplying a range of dates.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `dateRangeType / startDate`

Namespace	employment
Annotations	`State the start date of the date range.`
Diagram	
Type	ownDate
Properties	content: `simple`
Facets	pattern `[\d]{4}\-[\d]{2}\-[\d]{2}|[\d]{4}\-[\d]{2}|[\d]{4}`

Source	`<xs:element name="startDate" type="ownDate">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the start date of the date range.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

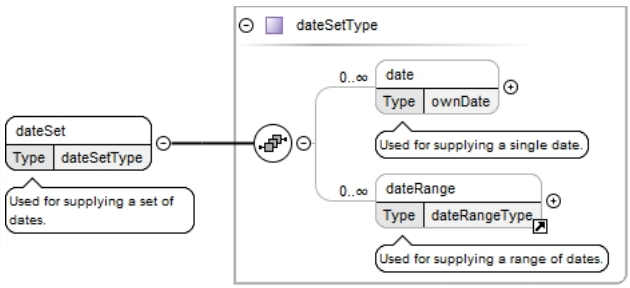
Element `dateRangeType / endDate`

Namespace	employment
Annotations	`State the end date of the date range.`
Diagram	
Type	ownDate
Properties	content: `simple` minOccurs: `0` maxOccurs: `1`

Facets	pattern	[\d]{4}\-[\d]{2}\-[\d]{2}|[\d]{4}\-[\d]{2}|[\d]{4}

Source		`<xs:element name="endDate" minOccurs="0" maxOccurs="1" type="ownDate">`
		`  <xs:annotation>`
		`  <xs:documentation xml:lang="en">State the end date of the date range.</xs:documentation>`
		`  </xs:annotation>`
		`</xs:element>`

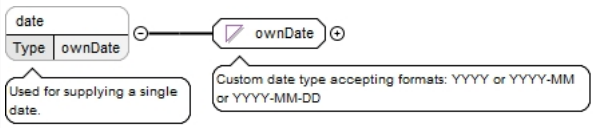
Element dateSet

Namespace	employment	
Annotations	Used for supplying a set of dates.	
Diagram		
Type	dateSetType	
Properties	content:	complex
Used by	Complex Types	adjustmentPeriodType, seizurePeriodType
Model	date* , dateRange*	
Children	date, dateRange	

Instance	`<dateSet xmlns="employment">`
	`  <date>{0,unbounded}</date>`
	`  <dateRange>{0,unbounded}</dateRange>`
	`</dateSet>`	

Source	`<xs:element name="dateSet" type="dateSetType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for supplying a set of dates.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`	

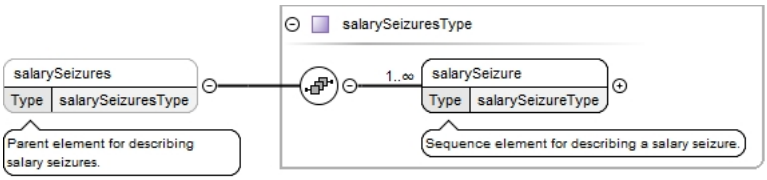
Element dateSetType / date

Namespace	employment	
Annotations	Used for supplying a single date.	
Diagram		
Type	ownDate	
Properties	content:	simple
	minOccurs:	0
	maxOccurs:	unbounded
Facets	pattern	[\d]{4}\-[\d]{2}\-[\d]{2}|[\d]{4}\-[\d]{2}|[\d]{4}

Source		`<xs:element name="date" minOccurs="0" maxOccurs="unbounded" type="ownDate">`
		`  <xs:annotation>`
		`  <xs:documentation xml:lang="en">Used for supplying a single date.</xs:documentation>`
		`  </xs:annotation>`
		`</xs:element>`

Element employeeDescriptionType / salarySeizures

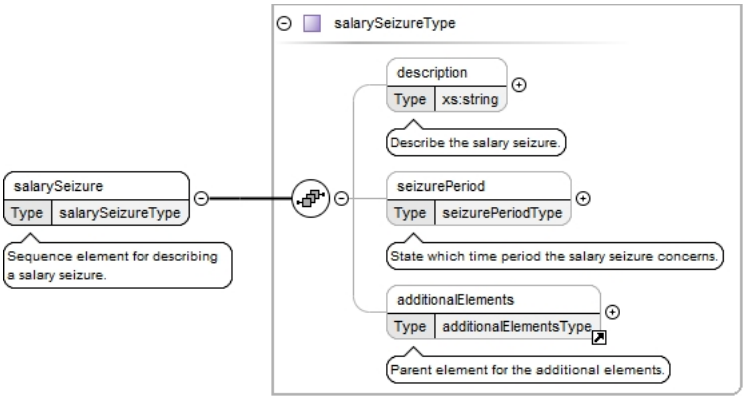
Namespace	employment

Annotations	Parent element for describing salary seizures.
Diagram	
Type	salarySeizuresType
Properties	content: complex minOccurs: 0
Model	salarySeizure+
Children	salarySeizure

Instance	`<salarySeizures xmlns="employment">`
	`  <salarySeizure>{1,unbounded}</salarySeizure>`
	`</salarySeizures>`

Source	`<xs:element name="salarySeizures" type="salarySeizuresType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for describing salary seizures.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element salarySeizuresType / salarySeizure

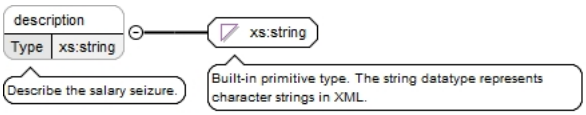
Namespace	employment
Annotations	Sequence element for describing a salary seizure.
Diagram	
Type	salarySeizureType
Properties	content: complex maxOccurs: unbounded
Model	description{0,1} , seizurePeriod{0,1} , additionalElements{0,1}
Children	additionalElements, description, seizurePeriod

Instance	`<salarySeizure xmlns="employment">`
	`  <description>{0,1}</description>`
	`  <seizurePeriod>{0,1}</seizurePeriod>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</salarySeizure>`

Source	`<xs:element name="salarySeizure" type="salarySeizureType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing a salary seizure.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

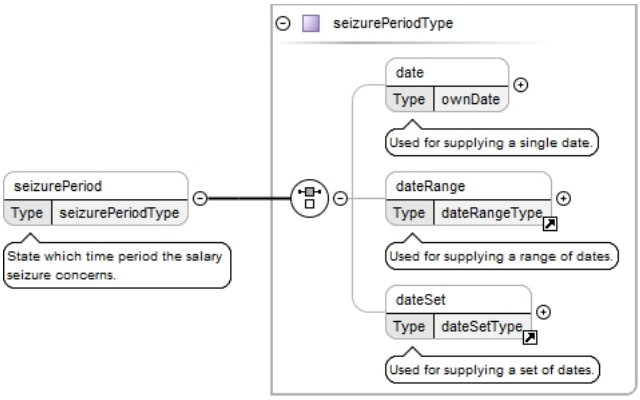
Element salarySeizureType / description

Namespace	employment
Annotations	Describe the salary seizure.

Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the salary seizure.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

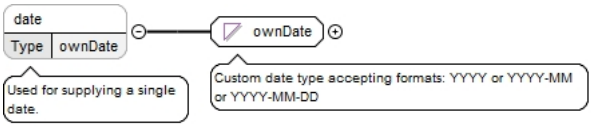
Element salarySeizureType / seizurePeriod

Namespace	employment
Annotations	State which time period the salary seizure concerns.
Diagram	
Type	seizurePeriodType
Properties	content: complex minOccurs: 0
Model	date{0,1} | dateRange{0,1} | dateSet{0,1}
Children	date, dateRange, dateSet

Instance	`<seizurePeriod xmlns="employment">`
	`  <date>{0,1}</date>`
	`  <dateRange>{0,1}</dateRange>`
	`  <dateSet>{0,1}</dateSet>`
	`</seizurePeriod>`

Source	`<xs:element name="seizurePeriod" type="seizurePeriodType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State which time period the salary seizure concerns.</`
	`xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

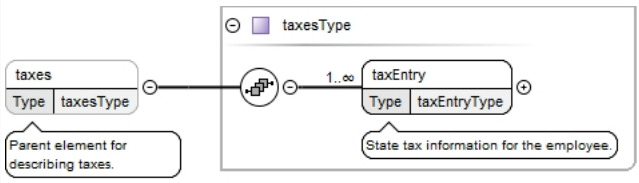
Element seizurePeriodType / date

Namespace	employment
Annotations	Used for supplying a single date.
Diagram	
Type	ownDate
Properties	content: simple minOccurs: 0

Facets	pattern `[\d]{4}\-[\d]{2}\-[\d]{2}|[\d]{4}\-[\d]{2}|[\d]{4}`

Source	`<xs:element name="date" minOccurs="0" type="ownDate">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for supplying a single date.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

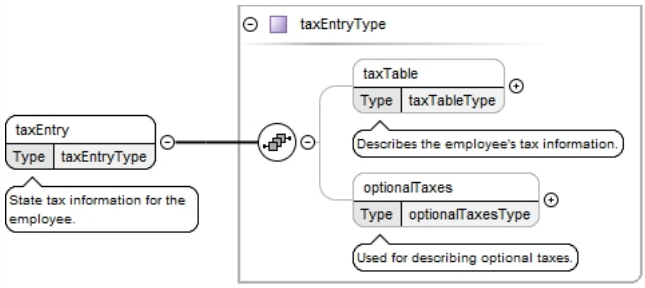
Element `employeeDescriptionType / taxes`

Namespace	employment
Annotations	`Parent element for describing taxes.`
Diagram	
Type	taxesType
Properties	content: `complex` minOccurs: `0`
Model	taxEntry+
Children	taxEntry

Instance	`<taxes xmlns="employment">`
	`  <taxEntry>{1,unbounded}</taxEntry>`
	`</taxes>`

Source	`<xs:element name="taxes" type="taxesType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for describing taxes.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

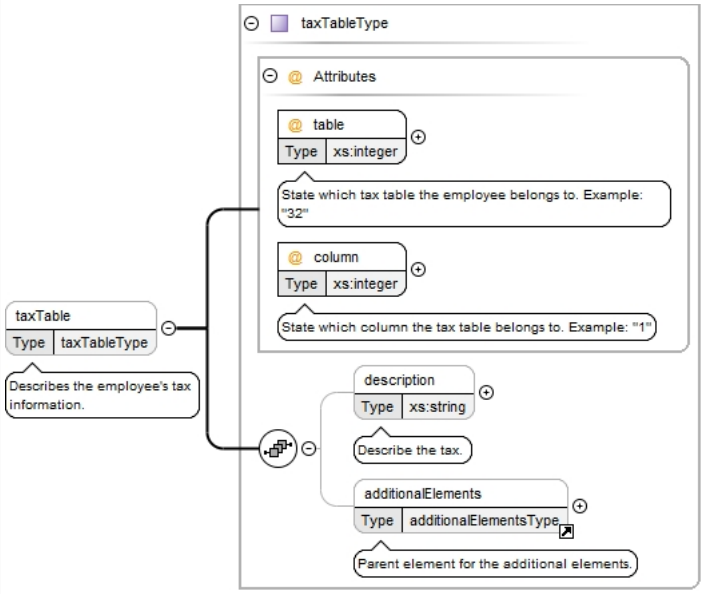
Element `taxesType / taxEntry`

Namespace	employment
Annotations	`State tax information for the employee.`
Diagram	
Type	taxEntryType
Properties	content: `complex` maxOccurs: `unbounded`
Model	taxTable{0,1} , optionalTaxes{0,1}
Children	optionalTaxes, taxTable

Instance	`<taxEntry xmlns="employment">`
	`  <taxTable column="" table="">{0,1}</taxTable>`
	`  <optionalTaxes>{0,1}</optionalTaxes>`
	`</taxEntry>`

Source	`<xs:element name="taxEntry" type="taxEntryType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State tax information for the employee.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element taxEntryType / taxTable

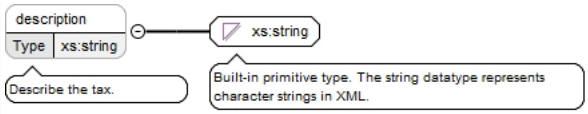
Namespace	employment
Annotations	Describes the employee's tax information.
Diagram	
Type	taxTableType
Properties	content: complex minOccurs: 0
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description

Instance	`<taxTable column="" table="" xmlns="employment">`
	`  <description>{0,1}</description>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</taxTable>`

Attributes	**QName** / **Type** / **Use** **column** / xs:integer / required State which column the tax table belongs to. Example: "1" **table** / xs:integer / required State which tax table the employee belongs to. Example: "32"

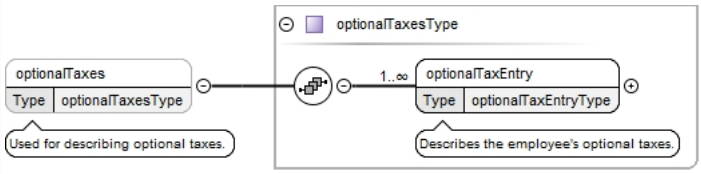
Source	`<xs:element name="taxTable" type="taxTableType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describes the employee's tax information.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element taxTableType / description

Namespace	employment
Annotations	Describe the tax.
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the tax.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

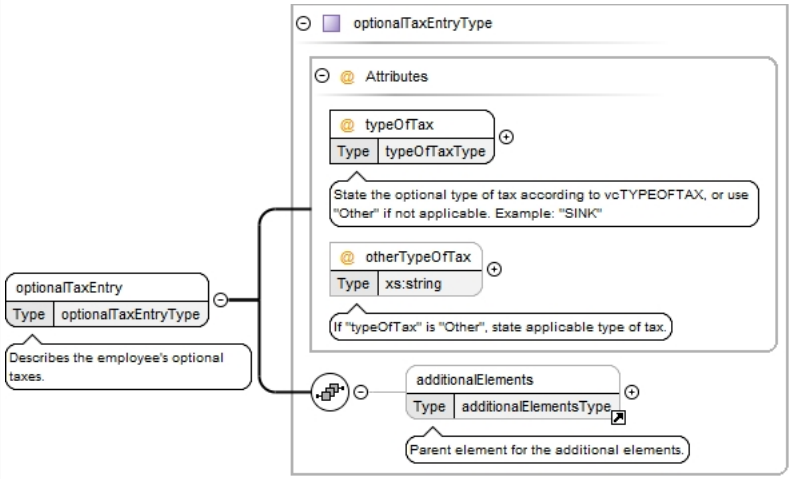
Element `taxEntryType / optionalTaxes`

Namespace	employment
Annotations	`Used for describing optional taxes.`
Diagram	
Type	optionalTaxesType
Properties	content: `complex` minOccurs: `0`
Model	optionalTaxEntry+
Children	optionalTaxEntry

Instance	`<optionalTaxes xmlns="employment">`
	`  <optionalTaxEntry otherTypeOfTax="" typeOfTax="">{1,unbounded}</optionalTaxEntry>`
	`</optionalTaxes>`

Source	`<xs:element name="optionalTaxes" type="optionalTaxesType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for describing optional taxes.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `optionalTaxesType / optionalTaxEntry`

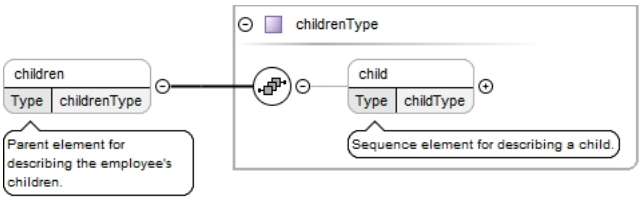
Namespace	employment
Annotations	`Describes the employee's optional taxes.`
Diagram	
Type	optionalTaxEntryType
Properties	content: `complex` maxOccurs: `unbounded`
Model	additionalElements{0,1}
Children	additionalElements

Instance	`<optionalTaxEntry otherTypeOfTax="" typeOfTax="" xmlns="employment">`
	`  <additionalElements>{0,1}</additionalElements>`
	`</optionalTaxEntry>`

Attributes	**QName** / **Type** / **Use**
	otherTypeOfTax / xs:string / optional
	`If "typeOfTax" is "Other", state applicable type of tax.`
	typeOfTax / typeOfTaxType / required
	`State the optional type of tax according to vcTYPEOFTAX, or use "Other" if not applicable. Example: "SINK"`

Source	`<xs:element name="optionalTaxEntry" type="optionalTaxEntryType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describes the employee's optional taxes.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

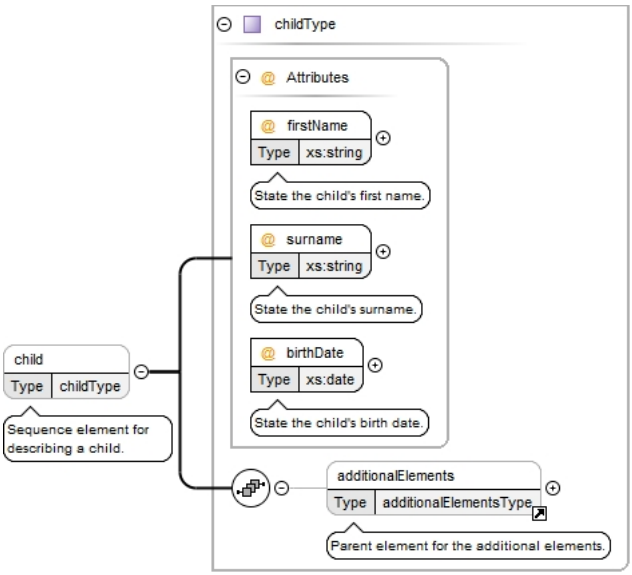
Element `employeeDescriptionType / children`

Namespace	employment
Annotations	`Parent element for describing the employee's children.`
Diagram	
Type	childrenType
Properties	content: `complex` minOccurs: `0`
Model	child{0,1}
Children	child

Instance	`<children xmlns="employment">`
	`  <child birthDate="" firstName="" surname="">{0,1}</child>`
	`</children>`

Source	`<xs:element name="children" type="childrenType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for describing the employee's children.</`
	`xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `childrenType / child`

Namespace	employment
Annotations	`Sequence element for describing a child.`
Diagram	
Type	childType
Properties	content: `complex` minOccurs: `0`
Model	additionalElements{0,1}
Children	additionalElements

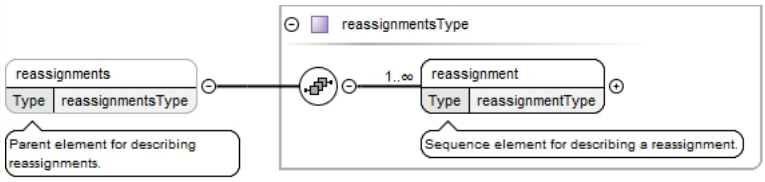
Instance	`<child birthDate="" firstName="" surname="" xmlns="employment">`
	`  <additionalElements>{0,1}</additionalElements>`
	`</child>`

Attributes	

QName	Type	Use	
birthDate	xs:date	required	
	State the child's birth date.		
firstName	xs:string	required	
	State the child's first name.		
surname	xs:string	required	
	State the child's surname.		

Source	`<xs:element name="child" type="childType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing a child.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

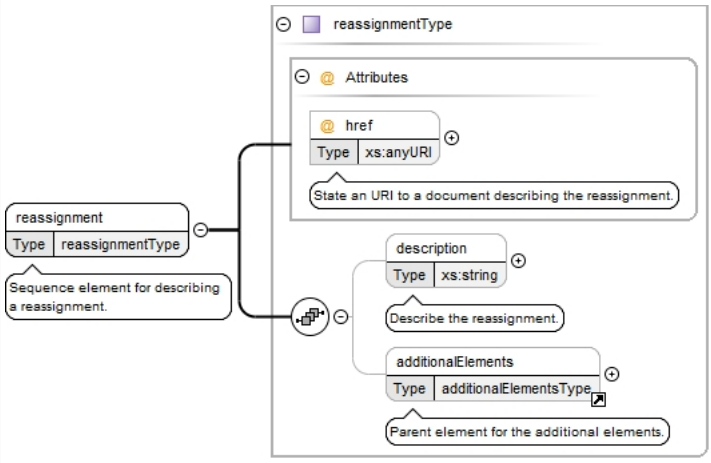
Element employeeDescriptionType / reassignments

Namespace	employment
Annotations	Parent element for describing reassignments.
Diagram	
Type	reassignmentsType
Properties	content: complex minOccurs: 0
Model	reassignment+
Children	reassignment

Instance	`<reassignments xmlns="employment">`
	`  <reassignment href="">{1,unbounded}</reassignment>`
	`</reassignments>`

Source	`<xs:element name="reassignments" type="reassignmentsType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for describing reassignments.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element reassignmentsType / reassignment

Namespace	employment
Annotations	Sequence element for describing a reassignment.
Diagram	

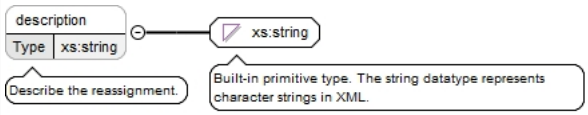
Type	reassignmentType
Properties	content: complex maxOccurs: unbounded
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description

Instance	`<reassignment href="" xmlns="employment">`
	`  <description>{0,1}</description>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</reassignment>`

Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional State an URI to a document describing the reassignment.

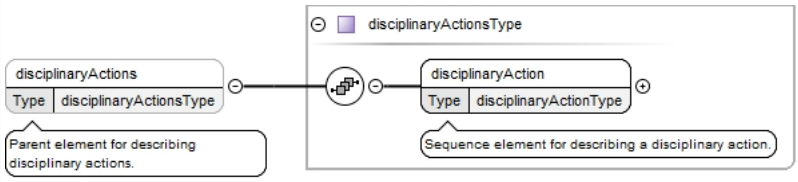
Source	`<xs:element name="reassignment" type="reassignmentType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing a reassignment.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element reassignmentType / description

Namespace	employment
Annotations	Describe the reassignment.
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the reassignment.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

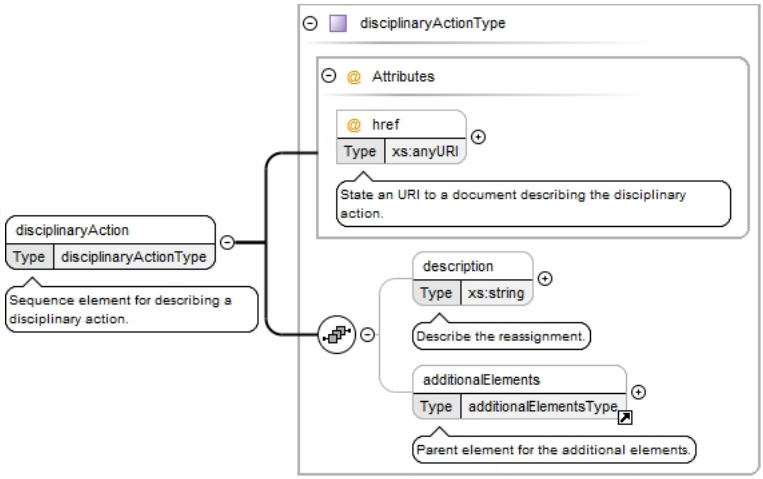
Element employeeDescriptionType / disciplinaryActions

Namespace	employment
Annotations	Parent element for describing disciplinary actions.
Diagram	
Type	disciplinaryActionsType
Properties	content: complex minOccurs: 0
Model	disciplinaryAction
Children	disciplinaryAction

Instance	`<disciplinaryActions xmlns="employment">`
	`  <disciplinaryAction href="">{1,1}</disciplinaryAction>`
	`</disciplinaryActions>`

Source	`<xs:element name="disciplinaryActions" type="disciplinaryActionsType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for describing disciplinary actions.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element disciplinaryActionsType / disciplinaryAction

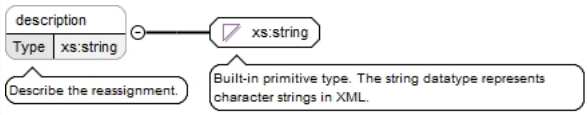
Namespace	employment
Annotations	Sequence element for describing a disciplinary action.
Diagram	
Type	disciplinaryActionType
Properties	content: complex
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description

Instance	`<disciplinaryAction href="" xmlns="employment">`
	`  <description>{0,1}</description>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</disciplinaryAction>`

Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional State an URI to a document describing the disciplinary action.

Source	`<xs:element name="disciplinaryAction" type="disciplinaryActionType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing a disciplinary action.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

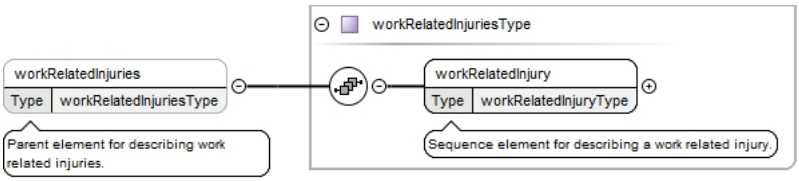
Element disciplinaryActionType / description

Namespace	employment
Annotations	Describe the reassignment.
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the reassignment.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element employeeDescriptionType / workRelatedInjuries

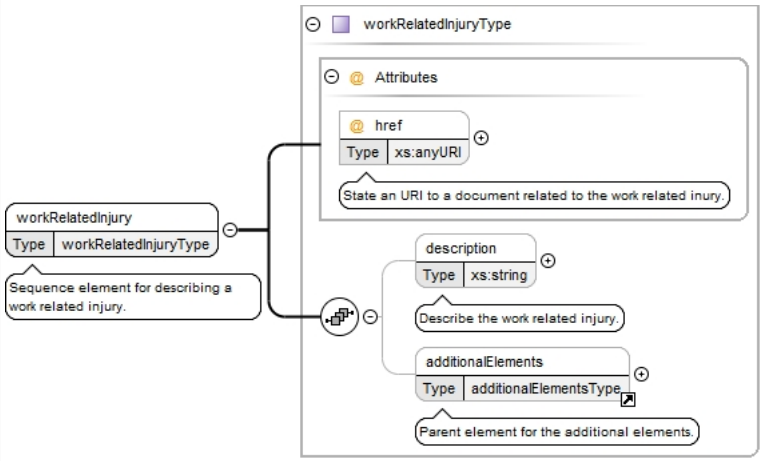
Namespace	employment
Annotations	Parent element for describing work related injuries.

Diagram	
Type	workRelatedInjuriesType
Properties	content: complex minOccurs: 0
Model	workRelatedInjury
Children	workRelatedInjury

Instance	`<workRelatedInjuries xmlns="employment">`
	`  <workRelatedInjury href="">{1,1}</workRelatedInjury>`
	`</workRelatedInjuries>`

Source	`<xs:element name="workRelatedInjuries" type="workRelatedInjuriesType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for describing work related injuries.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element workRelatedInjuriesType / workRelatedInjury

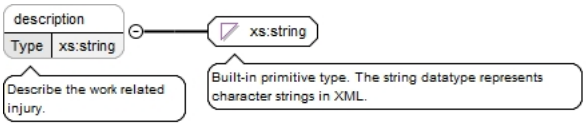
Namespace	employment
Annotations	Sequence element for describing a work related injury.
Diagram	
Type	workRelatedInjuryType
Properties	content: complex
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description

Instance	`<workRelatedInjury href="" xmlns="employment">`
	`  <description>{0,1}</description>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</workRelatedInjury>`

Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional State an URI to a document related to the work related inury.

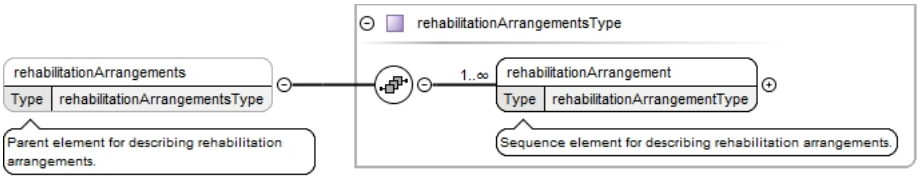
Source	`<xs:element name="workRelatedInjury" type="workRelatedInjuryType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing a work related injury.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element workRelatedInjuryType / description

Namespace	employment
Annotations	Describe the work related injury.
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the work related injury.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element employeeDescriptionType / rehabilitationArrangements

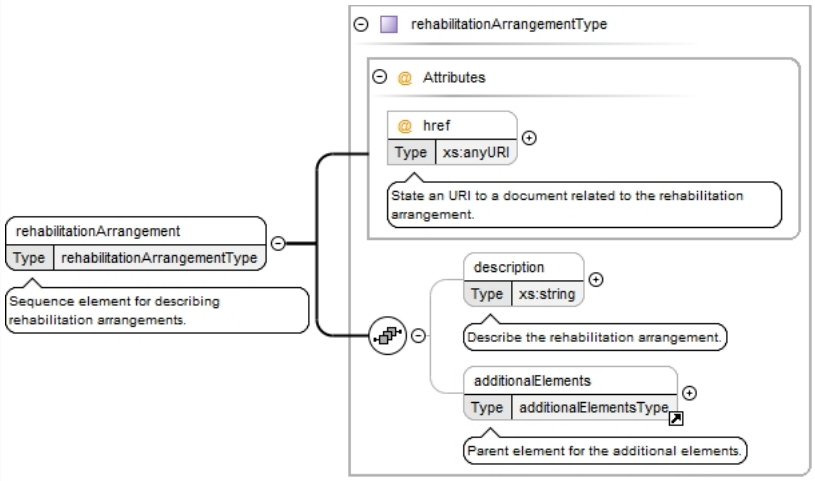
Namespace	employment
Annotations	Parent element for describing rehabilitation arrangements.
Diagram	
Type	rehabilitationArrangementsType
Properties	content: complex minOccurs: 0
Model	rehabilitationArrangement+
Children	rehabilitationArrangement

Instance	`<rehabilitationArrangements xmlns="employment">`
	`  <rehabilitationArrangement href="">{1,unbounded}</rehabilitationArrangement>`
	`</rehabilitationArrangements>`

Source	`<xs:element name="rehabilitationArrangements" type="rehabilitationArrangementsType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for describing rehabilitation arrangements.</`
	`xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element rehabilitationArrangementsType / rehabilitationArrangement

Namespace	employment
Annotations	Sequence element for describing rehabilitation arrangements.

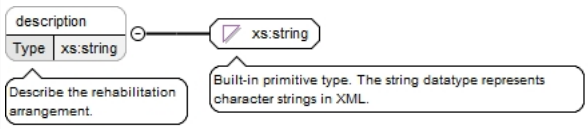
Diagram	
Type	rehabilitationArrangementType
Properties	content: `complex` maxOccurs: `unbounded`
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description

Instance	`<rehabilitationArrangement href="" xmlns="employment">`
	`  <description>{0,1}</description>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</rehabilitationArrangement>`

Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional `State an URI to a document related to the rehabilitation arrangement.`

Source	`<xs:element name="rehabilitationArrangement" type="rehabilitationArrangementType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing rehabilitation arrangements.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

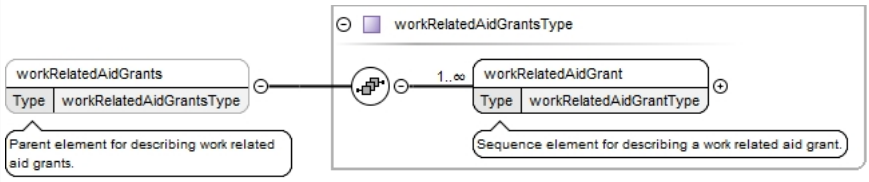
Element `rehabilitationArrangementType / description`

Namespace	employment
Annotations	`Describe the rehabilitation arrangement.`
Diagram	
Type	xs:string
Properties	content: `simple` minOccurs: `0`

Source	`<xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the rehabilitation arrangement.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `employeeDescriptionType / workRelatedAidGrants`

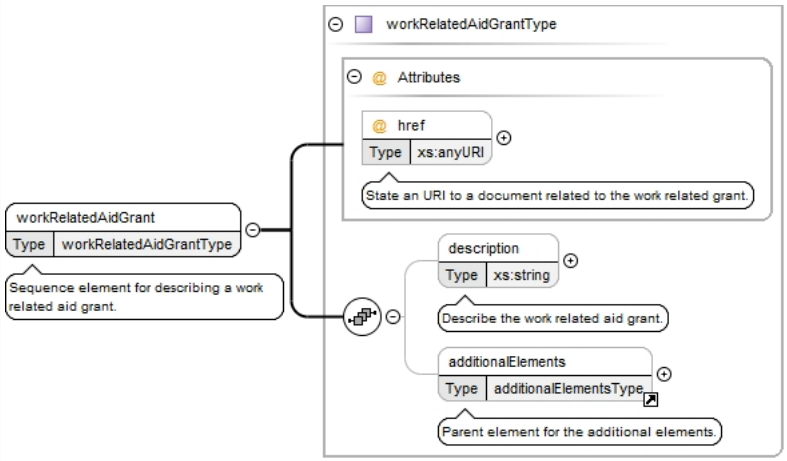
Namespace	employment
Annotations	`Parent element for describing work related aid grants.`

Diagram	
Type	workRelatedAidGrantsType
Properties	content: complex minOccurs: 0
Model	workRelatedAidGrant+
Children	workRelatedAidGrant

Instance	`<workRelatedAidGrants xmlns="employment">`
	`  <workRelatedAidGrant href="">{1,unbounded}</workRelatedAidGrant>`
	`</workRelatedAidGrants>`

Source	`<xs:element name="workRelatedAidGrants" type="workRelatedAidGrantsType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for describing work related aid grants.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element workRelatedAidGrantsType / workRelatedAidGrant

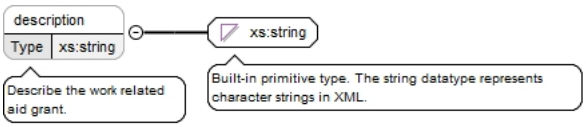
Namespace	employment
Annotations	Sequence element for describing a work related aid grant.
Diagram	
Type	workRelatedAidGrantType
Properties	content: complex maxOccurs: unbounded
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description

Instance	`<workRelatedAidGrant href="" xmlns="employment">`
	`  <description>{0,1}</description>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</workRelatedAidGrant>`

Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional State an URI to a document related to the work related grant.

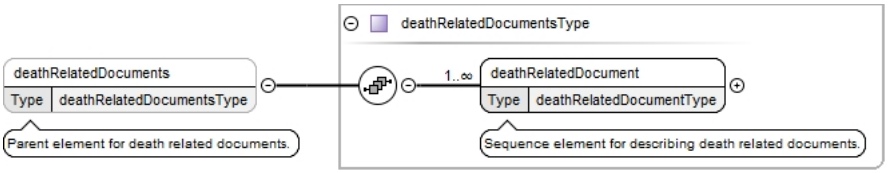
Source	`<xs:element name="workRelatedAidGrant" type="workRelatedAidGrantType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing a work related aid grant.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element workRelatedAidGrantType / description

Namespace	employment
Annotations	Describe the work related aid grant.
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the work related aid grant.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

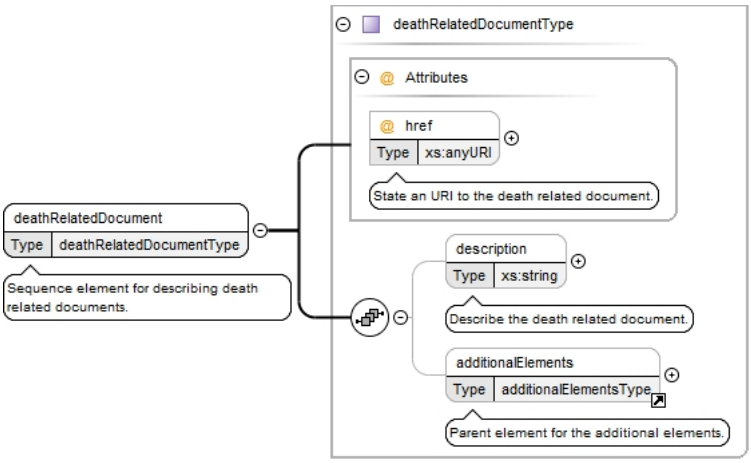
Element employeeDescriptionType / deathRelatedDocuments

Namespace	employment
Annotations	Parent element for death related documents.
Diagram	
Type	deathRelatedDocumentsType
Properties	content: complex minOccurs: 0
Model	deathRelatedDocument+
Children	deathRelatedDocument

Instance	`<deathRelatedDocuments xmlns="employment">`
	`  <deathRelatedDocument href="">{1,unbounded}</deathRelatedDocument>`
	`</deathRelatedDocuments>`

Source	`<xs:element name="deathRelatedDocuments" type="deathRelatedDocumentsType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for death related documents.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element deathRelatedDocumentsType / deathRelatedDocument

Namespace	employment
Annotations	Sequence element for describing death related documents.
Diagram	

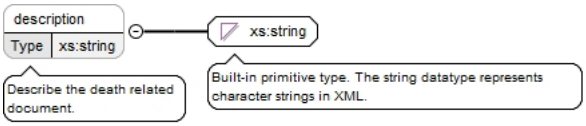
Type	deathRelatedDocumentType
Properties	content: complex maxOccurs: unbounded
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description

Instance	`<deathRelatedDocument href="" xmlns="employment">`
	`  <description>{0,1}</description>`
	`  <additionalElements>{0,1}</additionalElements>`
	`</deathRelatedDocument>`

Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional State an URI to the death related document.

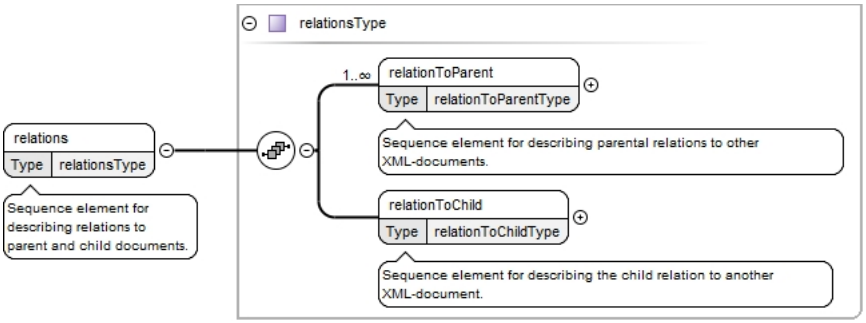
Source	`<xs:element name="deathRelatedDocument" type="deathRelatedDocumentType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing death related documents.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element deathRelatedDocumentType / description

Namespace	employment
Annotations	Describe the death related document.
Diagram	
Type	xs:string
Properties	content: simple minOccurs: 0

Source	`<xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the death related document.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

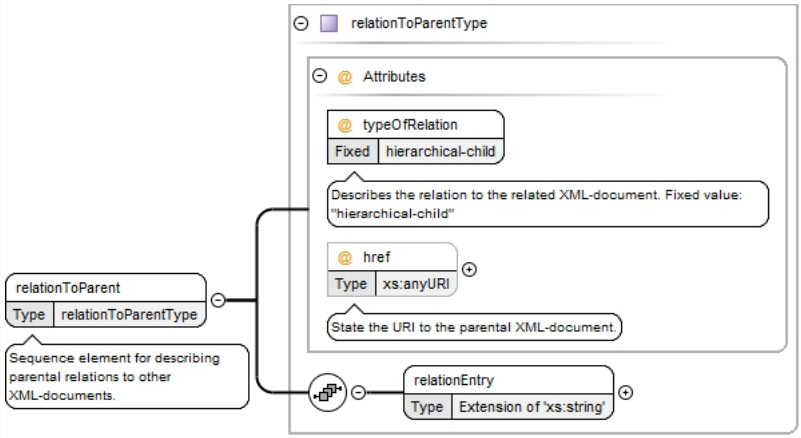
Element employeeDescriptionType / relations

Namespace	employment
Annotations	Sequence element for describing relations to parent and child documents.
Diagram	
Type	relationsType
Properties	content: complex
Model	relationToParent+ , relationToChild
Children	relationToChild, relationToParent

Instance	`<relations xmlns="employment">`
	`  <relationToParent href="" typeOfRelation="hierarchical-child">{1,unbounded}</relationToParent>`
	`  <relationToChild href="" typeOfRelation="hierarchical-parent">{1,1}</relationToChild>`
	`</relations>`

Source	`<xs:element name="relations" type="relationsType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing relations to parent and child documents.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

Element `relationsType / relationToParent`

Namespace	employment
Annotations	Sequence element for describing parental relations to other XML-documents.
Diagram	
Type	relationToParentType
Properties	content: complex maxOccurs: unbounded
Model	relationEntry
Children	relationEntry

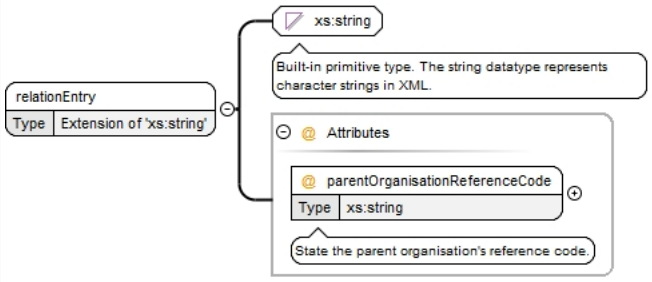
Instance	`<relationToParent href="" typeOfRelation="hierarchical-child" xmlns="employment">`
	`  <relationEntry parentOrganisationReferenceCode="">{1,1}</relationEntry>`
	`</relationToParent>`

Attributes

QName	Type	Fixed	Use	
href	xs:anyURI		optional	
	State the URI to the parental XML-document.			
typeOfRelation		hierarchical-child	required	
	Describes the relation to the related XML-document. Fixed value: "hierarchical-child"			

Source	`<xs:element name="relationToParent" type="relationToParentType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing parental relations to other XML-documents.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`

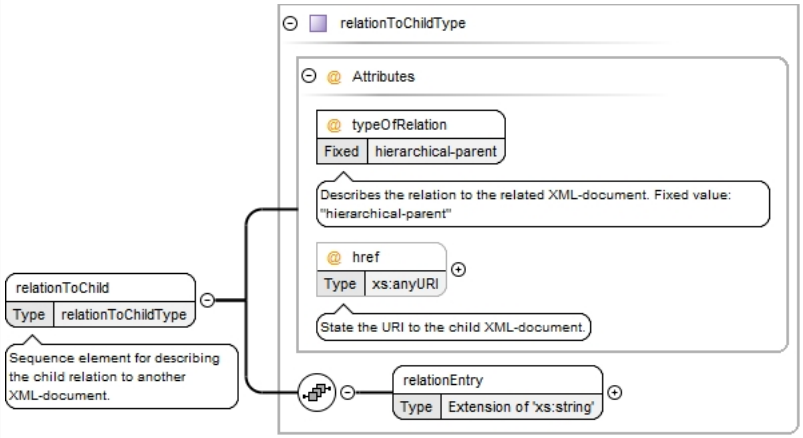
Element `relationToParentType / relationEntry`

Namespace	employment
Diagram	
Type	extension of xs:string

Properties	content: complex			
Attributes	**QName**	**Type**	**Use**	
	parentOrganisationReferenceCode	xs:string	required	
		State the parent organisation's reference code.		

Source	`<xs:element name="relationEntry">`
	`  <xs:complexType>`
	`  <xs:simpleContent>`
	`  <xs:extension base="xs:string">`
	`  <xs:attribute name="parentOrganisationReferenceCode" type="xs:string" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the parent organisation's reference code.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:attribute>`
	`  </xs:extension>`
	`  </xs:simpleContent>`
	`  </xs:complexType>`
	`</xs:element>`			

Element relationsType / relationToChild

Namespace	employment				
Annotations	Sequence element for describing the child relation to another XML-document.				
Diagram					
Type	relationToChildType				
Properties	content: complex				
Model	relationEntry				
Children	relationEntry				

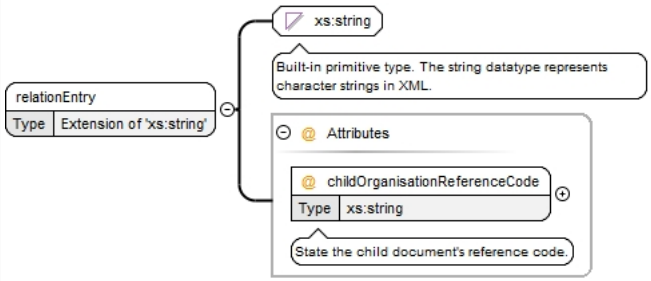
Instance	`<relationToChild href="" typeOfRelation="hierarchical-parent" xmlns="employment">`
	`  <relationEntry childOrganisationReferenceCode="">{1,1}</relationEntry>`
	`</relationToChild>`				

Attributes	**QName**	**Type**	**Fixed**	**Use**	
	href	xs:anyURI		optional	
		State the URI to the child XML-document.			
	typeOfRelation		hierarchical-parent	required	
		Describes the relation to the related XML-document. Fixed value: "hierarchical-parent"			

Source	`<xs:element name="relationToChild" type="relationToChildType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing the child relation to another XML-document.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:element>`				

Element relationToChildType / relationEntry

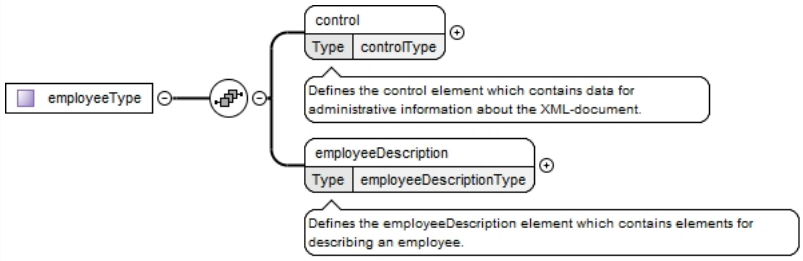
Namespace	employment

Diagram	
Type	extension of xs:string
Properties	content: complex
Attributes	**QName** / **Type** / **Use** **childOrganisationReference-Code** / xs:string / required State the child document's reference code.

Source	`<xs:element name="relationEntry">`
	`  <xs:complexType>`
	`  <xs:simpleContent>`
	`  <xs:extension base="xs:string">`
	`  <xs:attribute name="childOrganisationReferenceCode" type="xs:string" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the child document's reference code.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:attribute>`
	`  </xs:extension>`
	`  </xs:simpleContent>`
	`  </xs:complexType>`
	`</xs:element>`

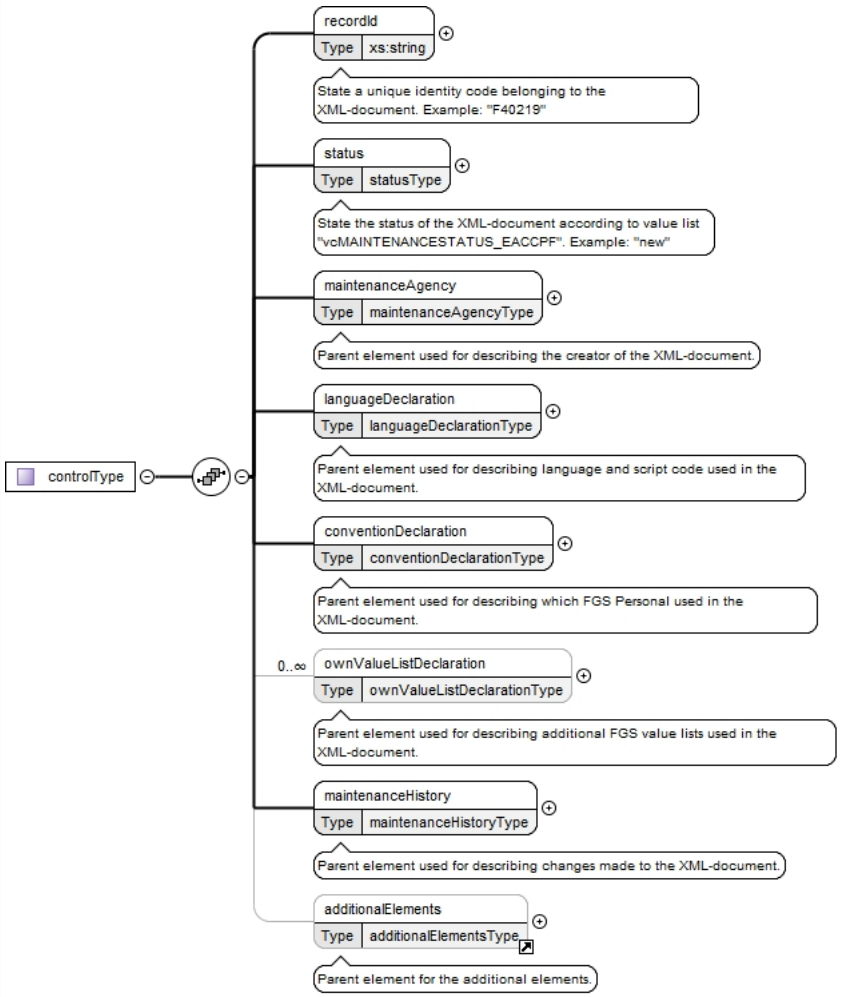
Complex Type(s)

Complex Type employeeType

Namespace	employment
Diagram	
Used by	Element employee
Model	control , employeeDescription
Children	control, employeeDescription

Source	`<xs:complexType name="employeeType">`
	`  <xs:sequence>`
	`  <xs:element name="control" type="controlType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Defines the control element which contains data for administrative information about the XML-document.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element name="employeeDescription" type="employeeDescriptionType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Defines the employeeDescription element which contains elements for describing an employee.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

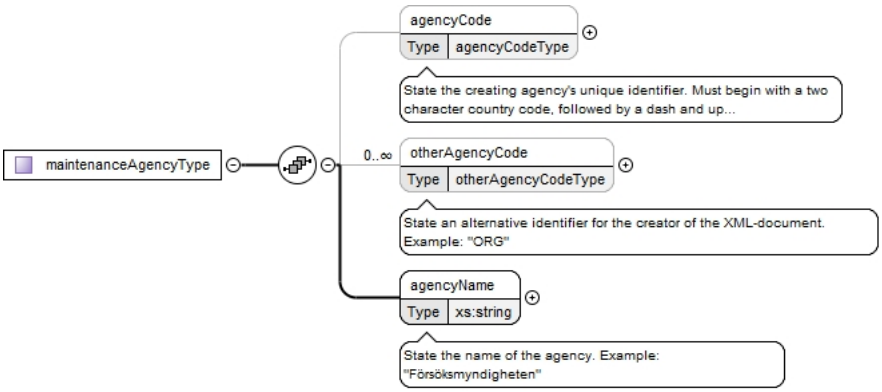
Complex Type `controlType`

Namespace	employment
Diagram	
Used by	Element employeeType/control
Model	recordId , status , maintenanceAgency , languageDeclaration , conventionDeclaration , ownValueListDeclaration* , maintenanceHistory , additionalElements{0,1}
Children	additionalElements, conventionDeclaration, languageDeclaration, maintenanceAgency, maintenanceHistory, ownValueListDeclaration, recordId, status
Source	

```
<xs:complexType name="controlType">
  <xs:sequence>
    <xs:element name="recordId" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="en">State a unique identity code belonging to the XML-document.
 Example: "F40219"</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="status" type="statusType">
      <xs:annotation>
        <xs:documentation xml:lang="en">State the status of the XML-document according to value list
 "vcMAINTENANCESTATUS_EACCPF". Example: "new"</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="maintenanceAgency" type="maintenanceAgencyType">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element used for describing the creator of the XML-
document.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="languageDeclaration" type="languageDeclarationType">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element used for describing language and script code
 used in the XML-document.</xs:documentation>
      </xs:annotation>
```

```
    </xs:element>
    <xs:element name="conventionDeclaration" type="conventionDeclarationType">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element used for describing which FGS Personal used
 in the XML-document.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="ownValueListDeclaration" type="ownValueListDeclarationType" minOccurs="0"
 maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element used for describing additional FGS value
 lists used in the XML-document.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="maintenanceHistory" type="maintenanceHistoryType">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element used for describing changes made to the XML-
document.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="additionalElements" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

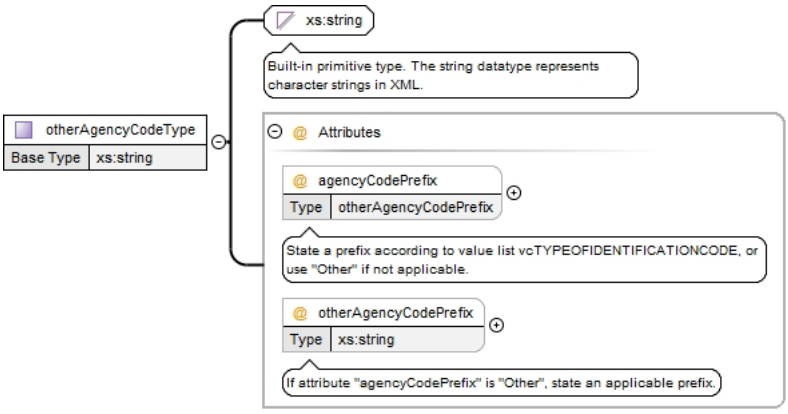
Complex Type `maintenanceAgencyType`

Namespace	employment
Diagram	
Used by	Element controlType/maintenanceAgency
Model	agencyCode{0,1} , otherAgencyCode* , agencyName
Children	agencyCode, agencyName, otherAgencyCode
Source	

```
<xs:complexType name="maintenanceAgencyType">
  <xs:sequence>
    <xs:element name="agencyCode" type="agencyCodeType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">State the creating agency's unique identifier. Must begin
 with a two character country code, followed by a dash and up to 10 characters. Example: "SE-
A:ARA/123</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="otherAgencyCode" minOccurs="0" maxOccurs="unbounded"
 type="otherAgencyCodeType">
      <xs:annotation>
        <xs:documentation xml:lang="en">State an alternative identifier for the creator of the XML-
document. Example: "ORG"</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="agencyName" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="en">State the name of the agency. Example:
 "Försöksmyndigheten"</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Complex Type `otherAgencyCodeType`

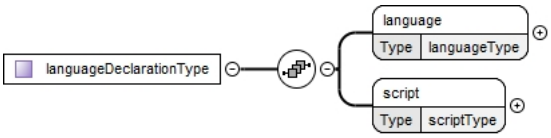
Namespace	employment

Diagram	
Type	extension of xs:string
Used by	Element maintenanceAgencyType/otherAgencyCode
Attributes	**QName** / **Type** / **Use** **agencyCodePrefix** / otherAgencyCodePrefix / optional State a prefix according to value list vcTYPEOFIDENTIFICATIONCODE, or use "Other" if not applicable. **otherAgencyCodePrefix** / xs:string / optional If attribute "agencyCodePrefix" is "Other", state an applicable prefix.

Source

```
<xs:complexType name="otherAgencyCodeType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="agencyCodePrefix" type="otherAgencyCodePrefix" use="optional">
        <xs:annotation>
          <xs:documentation xml:lang="en">State a prefix according to value list
 vcTYPEOFIDENTIFICATIONCODE, or use "Other" if not applicable.</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="otherAgencyCodePrefix" type="xs:string" use="optional">
        <xs:annotation>
          <xs:documentation xml:lang="en">If attribute "agencyCodePrefix" is "Other", state an
 applicable prefix.</xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Complex Type `languageDeclarationType`

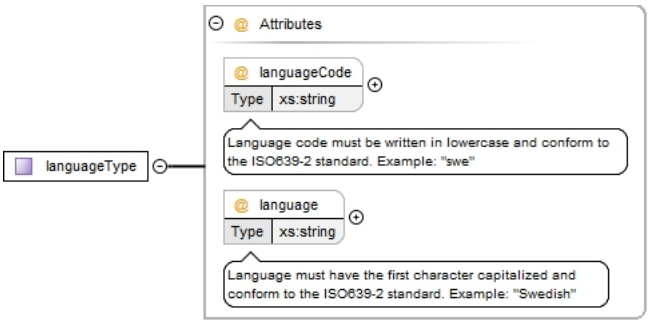
Namespace	employment
Diagram	
Used by	Element controlType/languageDeclaration
Model	language , script
Children	language, script

Source

```
<xs:complexType name="languageDeclarationType">
  <xs:sequence>
    <xs:element name="language" type="languageType"/>
    <xs:element name="script" type="scriptType"/>
  </xs:sequence>
</xs:complexType>
```

Complex Type `languageType`

Namespace	employment

Diagram	
Used by	Element languageDeclarationType/language

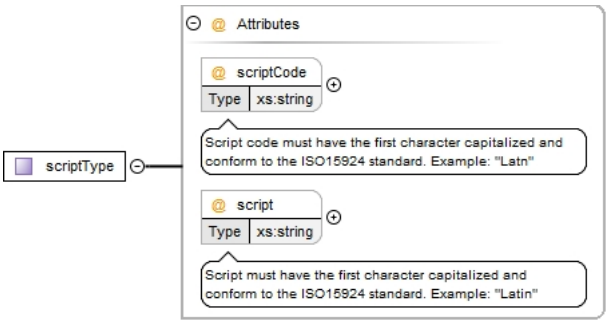
Attributes

QName	Type	Use	
language	xs:string	optional	
	Language must have the first character capitalized and conform to the ISO639-2 standard. Example: "Swedish"		
languageCode	xs:string	optional	
	Language code must be written in lowercase and conform to the ISO639-2 standard. Example: "swe"		

Source

```
<xs:complexType name="languageType">
  <xs:attribute name="languageCode" type="xs:string">
    <xs:annotation>
      <xs:documentation xml:lang="en">Language code must be written in lowercase and conform to the
 ISO639-2 standard. Example: "swe"</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="language" type="xs:string">
    <xs:annotation>
      <xs:documentation xml:lang="en">Language must have the first character capitalized and conform
 to the ISO639-2 standard. Example: "Swedish"</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Complex Type `scriptType`

Namespace	employment
Diagram	
Used by	Element languageDeclarationType/script

Attributes

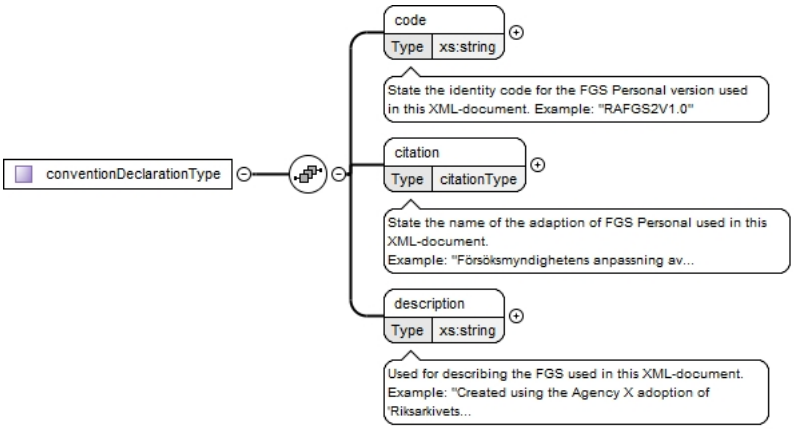
QName	Type	Use	
script	xs:string	optional	
	Script must have the first character capitalized and conform to the ISO15924 standard. Example: "Latin"		
scriptCode	xs:string	optional	
	Script code must have the first character capitalized and conform to the ISO15924 standard. Example: "Latn"		

Source

```
<xs:complexType name="scriptType">
  <xs:attribute name="scriptCode" type="xs:string">
    <xs:annotation>
      <xs:documentation xml:lang="en">Script code must have the first character capitalized and
 conform to the ISO15924 standard. Example: "Latn"</xs:documentation>
    </xs:annotation>
  </xs:attribute>
```

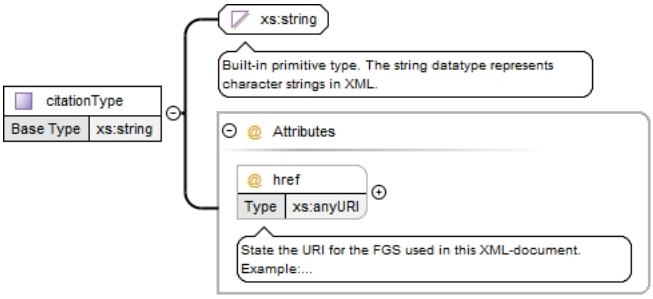
```
  <xs:attribute name="script" type="xs:string">
    <xs:annotation>
      <xs:documentation xml:lang="en">Script must have the first character capitalized and conform
 to the ISO15924 standard. Example: "Latin"</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Complex Type `conventionDeclarationType`

Namespace	employment
Diagram	
Used by	Element controlType/conventionDeclaration
Model	code , citation , description
Children	citation, code, description
Source	

```
<xs:complexType name="conventionDeclarationType">
  <xs:sequence>
    <xs:element name="code" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="en">State the identity code for the FGS Personal version used in
 this XML-document. Example: "RAFGS2V1.0"</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="citation" type="citationType">
      <xs:annotation>
        <xs:documentation xml:lang="en">State the name of the adaption of FGS Personal used in this
 XML-document.</xs:documentation>
        <xs:documentation xml:lang="en">Example: "Försöksmyndighetens anpassning av FGS Personal för
 överföring av personalinformation till det egna E-arktivet"</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="description" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="en">Used for describing the FGS used in this XML-document.</
xs:documentation>
        <xs:documentation xml:lang="en">Example: "Created using the Agency X adoption of
 'Riksarkivets anpassning av FGS Personal'"</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Complex Type `citationType`

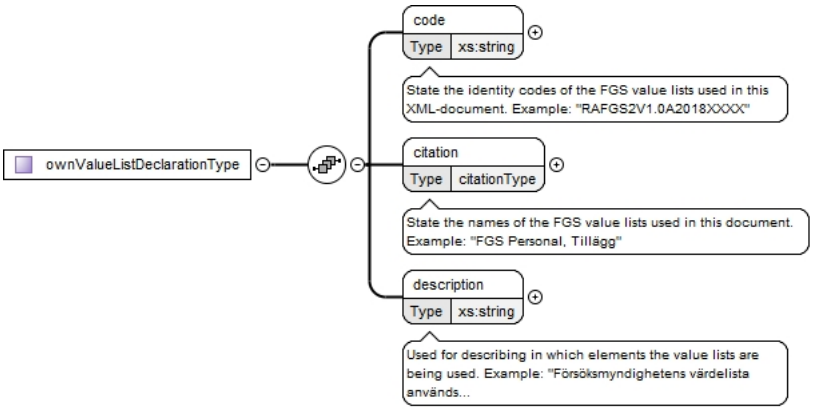
Namespace	employment
Diagram	

Type	extension of xs:string
Used by	Elements conventionDeclarationType/citation, ownValueListDeclarationType/citation
Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional State the URI for the FGS used in this XML-document. Example: "https://myndighet.se/Försöksmyndighetens_FGS_Personal_V_20181224.pdf"

Source

```
<xs:complexType name="citationType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="href" type="xs:anyURI">
        <xs:annotation>
          <xs:documentation xml:lang="en">State the URI for the FGS used in this XML-document.
 Example: "https://myndighet.se/Försöksmyndighetens_FGS_Personal_V_20181224.pdf"</xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Complex Type `ownValueListDeclarationType`

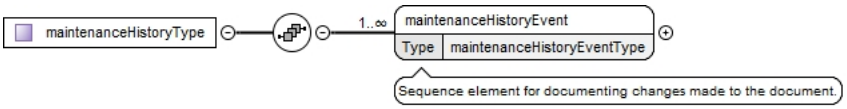
Namespace	employment
Diagram	
Used by	Element controlType/ownValueListDeclaration
Model	code , citation , description
Children	citation, code, description

Source

```
<xs:complexType name="ownValueListDeclarationType">
  <xs:sequence>
    <xs:element name="code" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="en">State the identity codes of the FGS value lists used in this
 XML-document. Example: "RAFGS2V1.0A2018XXXX"</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="citation" type="citationType">
      <xs:annotation>
        <xs:documentation xml:lang="en">State the names of the FGS value lists used in this
 document. Example: "FGS Personal, Tillägg"</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="description" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="en">Used for describing in which elements the value lists
 are being used. Example: "Försöksmyndighetens värdelista används i elementet AnhörigFält"</
xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

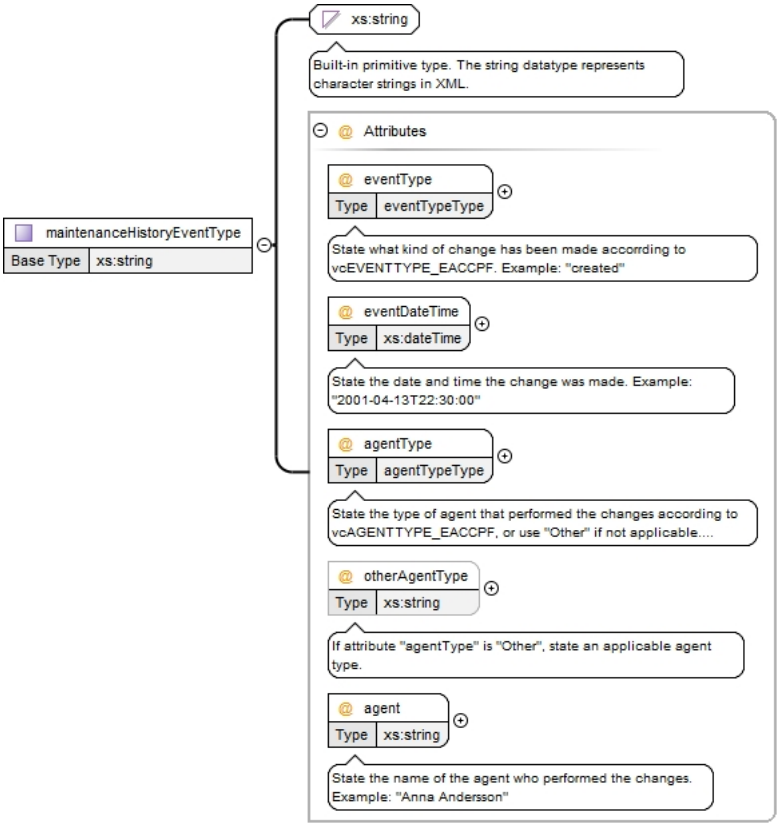
Complex Type `maintenanceHistoryType`

Namespace	employment

Diagram	
Used by	Element controlType/maintenanceHistory
Model	maintenanceHistoryEvent+
Children	maintenanceHistoryEvent

Source	`<xs:complexType name="maintenanceHistoryType">`
	`  <xs:sequence>`
	`  <xs:element name="maintenanceHistoryEvent" type="maintenanceHistoryEventType" minOccurs="1" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for documenting changes made to the document.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `maintenanceHistoryEventType`

Namespace	employment
Diagram	
Type	extension of xs:string
Used by	Element maintenanceHistoryType/maintenanceHistoryEvent
Attributes	

QName	Type	Use	
agent	xs:string	required	
	State the name of the agent who performed the changes. Example: "Anna Andersson"		
agentType	agentTypeType	required	
	State the type of agent that performed the changes according to vcAGENTTYPE_EACCPF, or use "Other" if not applicable. Example: "human"		
eventDateTime	xs:dateTime	required	
	State the date and time the change was made. Example: "2001-04-13T22:30:00"		

QName	Type	Use	
eventType	eventTypeType	required	
	State what kind of change has been made accorrding to vcEVENTTYPE_EACCPF. Example: "created"		
otherAgentType	xs:string	optional	
	If attribute "agentType" is "Other", state an applicable agent type.		

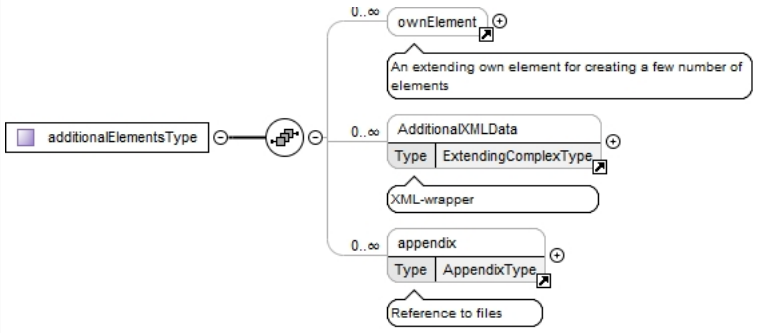
Source

```
<xs:complexType name="maintenanceHistoryEventType">
  <xs:simpleContent>
    <xs:extension base="xs:string">
      <xs:attribute name="eventType" type="eventTypeType" use="required">
        <xs:annotation>
          <xs:documentation xml:lang="en">State what kind of change has been made accorrding to
 vcEVENTTYPE_EACCPF. Example: "created"</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="eventDateTime" type="xs:dateTime" use="required">
        <xs:annotation>
          <xs:documentation xml:lang="en">State the date and time the change was made. Example:
 "2001-04-13T22:30:00"</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="agentType" type="agentTypeType" use="required">
        <xs:annotation>
          <xs:documentation xml:lang="en">State the type of agent that performed the changes
 according to vcAGENTTYPE_EACCPF, or use "Other" if not applicable. Example: "human"</
xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="otherAgentType" type="xs:string" use="optional">
        <xs:annotation>
          <xs:documentation xml:lang="en">If attribute "agentType" is "Other", state an applicable
 agent type.</xs:documentation>
        </xs:annotation>
      </xs:attribute>
      <xs:attribute name="agent" type="xs:string" use="required">
        <xs:annotation>
          <xs:documentation xml:lang="en">State the name of the agent who performed the changes.
 Example: "Anna Andersson"</xs:documentation>
        </xs:annotation>
      </xs:attribute>
    </xs:extension>
  </xs:simpleContent>
</xs:complexType>
```

Complex Type `additionalElementsType`

Namespace: employment

Diagram



Used by: Element additionalElements

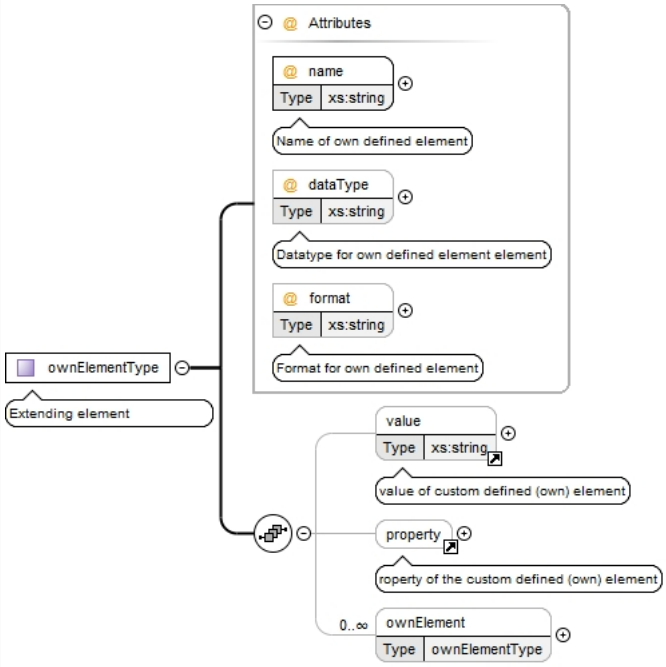
Model: ownElement* , AdditionalXMLData* , appendix*

Children: AdditionalXMLData, appendix, ownElement

Source

```
<xs:complexType name="additionalElementsType">
  <xs:sequence>
    <xs:element ref="ownElement" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="AdditionalXMLData" minOccurs="0" maxOccurs="unbounded"/>
    <xs:element ref="appendix" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
</xs:complexType>
```

Complex Type `ownElementType`

Namespace	employment
Annotations	`Extending element`
Diagram	
Used by	Elements ownElement/ownElement, ownElementType/ownElement
Model	value{0,1} , property{0,1} , ownElement*
Children	ownElement, property, value

Attributes

QName	Type	Use	
dataType	xs:string	optional	
	`Datatype for own defined element element`		
format	xs:string	optional	
	`Format for own defined element`		
name	xs:string	required	
	`Name of own defined element`		

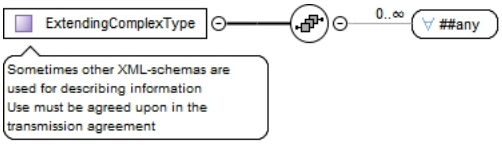
Source

```
<xs:complexType name="ownElementType">
  <xs:annotation>
    <xs:documentation xml:lang="en">Extending element</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element ref="value" minOccurs="0"/>
    <xs:element ref="property" minOccurs="0"/>
    <xs:element name="ownElement" type="ownElementType" minOccurs="0" maxOccurs="unbounded"/>
  </xs:sequence>
  <xs:attribute name="name" use="required" type="xs:string">
    <xs:annotation>
      <xs:documentation xml:lang="en">Name of own defined element</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="dataType" type="xs:string">
    <xs:annotation>
      <xs:documentation xml:lang="en">Datatype for own defined element element</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="format" type="xs:string">
    <xs:annotation>
      <xs:documentation xml:lang="en">Format for own defined element</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Complex Type `ExtendingComplexType`

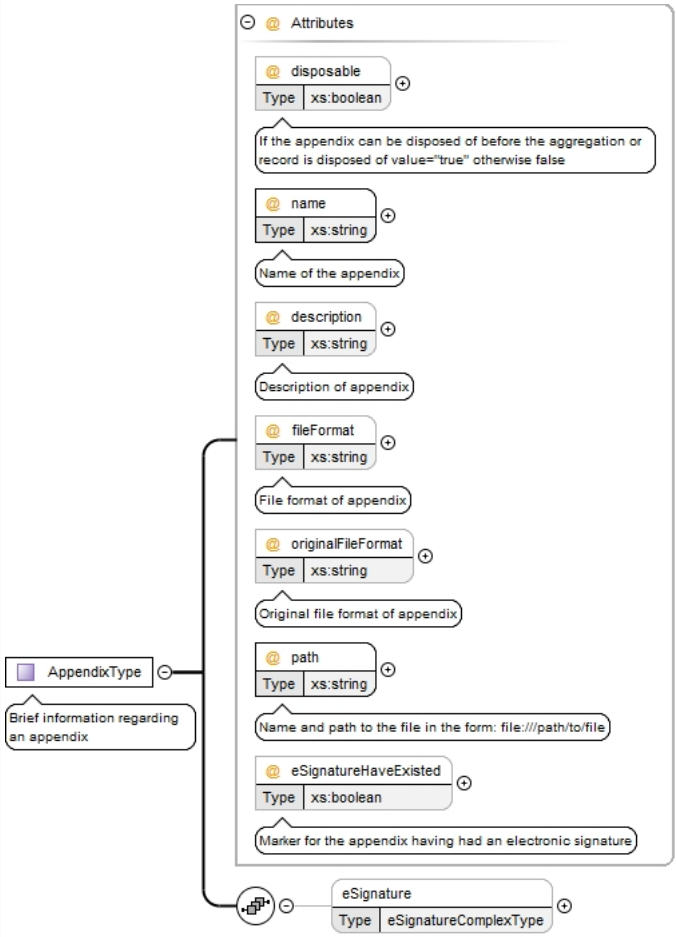
Namespace	employment

Annotations	`Sometimes other XML-schemas are used for describing information`
	
	`Use must be agreed upon in the transmission agreement`

Diagram	
Used by	Elements AdditionalXMLData, eSignatureComplexType/signature
Model	ANY element from ANY namespace

Source	`<xs:complexType name="ExtendingComplexType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sometimes other XML-schemas are used for describing information</xs:documentation>`
	`  <xs:documentation xml:lang="en">Use must be agreed upon in the transmission agreement</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:sequence>`
	`  <xs:any namespace="##any" processContents="lax" minOccurs="0" maxOccurs="unbounded"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `AppendixType`

Namespace	employment
Annotations	`Brief information regarding an appendix`
Diagram	
Used by	Element appendix
Model	eSignature{0,1}

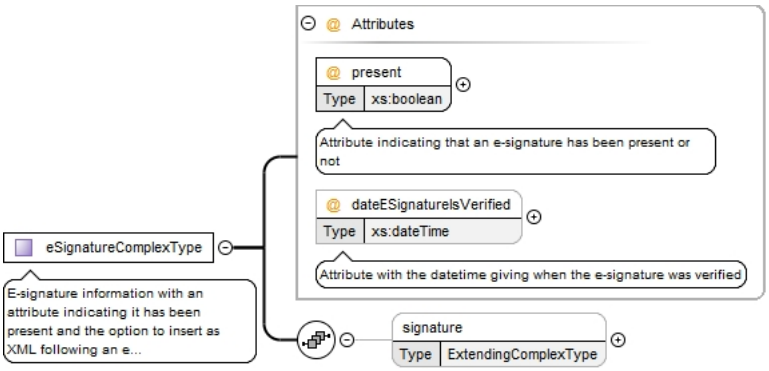
Children	eSignature			
Attributes	**QName**	**Type**	**Use**	
	description	xs:string	optional	
		Description of appendix		
	disposable	xs:boolean	optional	
		If the appendix can be disposed of before the aggregation or record is disposed of value="true" otherwise false		
	eSignatureHaveExisted	xs:boolean	optional	
		Marker for the appendix having had an electronic signature		
	fileFormat	xs:string	optional	
		File format of appendix		
	name	xs:string	required	
		Name of the appendix		
	originalFileFormat	xs:string	optional	
		Original file format of appendix		
	path	xs:string	required	
		Name and path to the file in the form: file:///path/to/file		

Source

```
<xs:complexType name="AppendixType">
  <xs:annotation>
    <xs:documentation xml:lang="en">Brief information regarding an appendix</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="eSignature" type="eSignatureComplexType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="disposable" type="xs:boolean" use="optional">
    <xs:annotation>
      <xs:documentation xml:lang="en">If the appendix can be disposed of before the aggregation or
 record is disposed of value="true" otherwise false</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="name" type="xs:string" use="required">
    <xs:annotation>
      <xs:documentation xml:lang="en">Name of the appendix</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="description" type="xs:string" use="optional">
    <xs:annotation>
      <xs:documentation xml:lang="en">Description of appendix</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="fileFormat" type="xs:string" use="optional">
    <xs:annotation>
      <xs:documentation xml:lang="en">File format of appendix</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="originalFileFormat" type="xs:string" use="optional">
    <xs:annotation>
      <xs:documentation xml:lang="en">Original file format of appendix</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="path" type="xs:string" use="required">
    <xs:annotation>
      <xs:documentation xml:lang="en">Name and path to the file in the form: file:///path/to/file</
xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="eSignatureHaveExisted" type="xs:boolean" use="optional">
    <xs:annotation>
      <xs:documentation xml:lang="en">Marker for the appendix having had an electronic signature</
xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Complex Type `eSignatureComplexType`

Namespace	employment
Annotations	E-signature information with an attribute indicating it has been present and the option to insert as XML following an e Signature XML-schema

Diagram	
Used by	Element AppendixType/eSignature
Model	signature{0,1}
Children	signature

Attributes

QName	Type	Use	
dateESignatureIsVerified	xs:dateTime	optional	
	Attribute with the datetime giving when the e-signature was verified		
present	xs:boolean	required	
	Attribute indicating that an e-signature has been present or not		

Source

```
<xs:complexType name="eSignatureComplexType">
  <xs:annotation>
    <xs:documentation xml:lang="en">E-signature information with an attribute indicating it has been
 present and the option to insert as XML following an e Signature XML-schema</xs:documentation>
  </xs:annotation>
  <xs:sequence>
    <xs:element name="signature" type="ExtendingComplexType" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="present" type="xs:boolean" use="required">
    <xs:annotation>
      <xs:documentation xml:lang="en">Attribute indicating that an e-signature has been present or
 not</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="dateESignatureIsVerified" type="xs:dateTime" use="optional">
    <xs:annotation>
      <xs:documentation xml:lang="en">Attribute with the datetime giving when the e-signature was
 verified</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Complex Type employeeDescriptionType

Namespace	employment

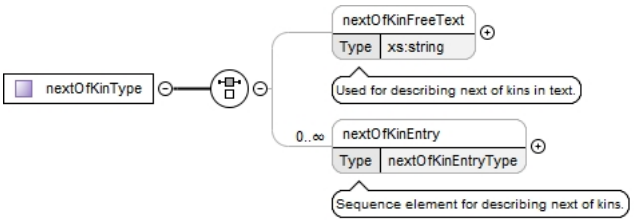
Diagram	
Used by	Element employeeType/employeeDescription
Model	nextOfKin{0,1} , adjustments{0,1} , salarySeizures{0,1} , taxes{0,1} , children{0,1} , reassignments{0,1} , disciplinaryActions{0,1} , workRelatedInjuries{0,1} , rehabilitationArrangements{0,1} , workRelatedAidGrants{0,1} , deathRelatedDocuments{0,1} , additionalElements{0,1} , relations
Children	additionalElements, adjustments, children, deathRelatedDocuments, disciplinaryActions, nextOfKin, reassignments, rehabilitationArrangements, relations, salarySeizures, taxes, workRelatedAidGrants, workRelatedInjuries

Source	`<xs:complexType name="employeeDescriptionType">`
	`  <xs:sequence>`
	`  <xs:element name="nextOfKin" type="nextOfKinType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Parent element for describing the employee's next of kin.</xs:documentation>`

```
      </xs:annotation>
    </xs:element>
    <xs:element name="adjustments" type="adjustmentsType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element for describing adjustments.</
xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="salarySeizures" type="salarySeizuresType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element for describing salary seizures.</
xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="taxes" type="taxesType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element for describing taxes.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="children" type="childrenType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element for describing the employee's children.</
xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="reassignments" type="reassignmentsType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element for describing reassignments.</
xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="disciplinaryActions" type="disciplinaryActionsType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element for describing disciplinary actions.</
xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="workRelatedInjuries" type="workRelatedInjuriesType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element for describing work related injuries.</
xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="rehabilitationArrangements" type="rehabilitationArrangementsType"
 minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element for describing rehabilitation arrangements.</
xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="workRelatedAidGrants" type="workRelatedAidGrantsType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element for describing work related aid grants.</
xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="deathRelatedDocuments" type="deathRelatedDocumentsType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Parent element for death related documents.</
xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="additionalElements" minOccurs="0"/>
    <xs:element name="relations" type="relationsType">
      <xs:annotation>
        <xs:documentation xml:lang="en">Sequence element for describing relations to parent and
 child documents.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

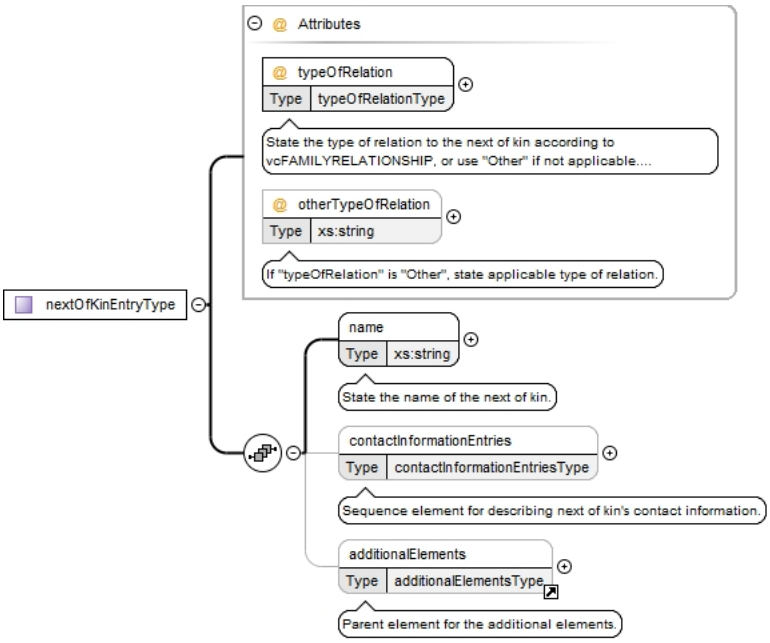
Complex Type `nextOfKinType`

Namespace	employment

Diagram	
Used by	Element employeeDescriptionType/nextOfKin
Model	nextOfKinFreeText{0,1} | nextOfKinEntry*
Children	nextOfKinEntry, nextOfKinFreeText

Source	`<xs:complexType name="nextOfKinType">`
	`  <xs:choice>`
	`  <xs:element name="nextOfKinFreeText" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for describing next of kins in text.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element name="nextOfKinEntry" type="nextOfKinEntryType" minOccurs="0" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing next of kins.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:choice>`
	`</xs:complexType>`

Complex Type `nextOfKinEntryType`

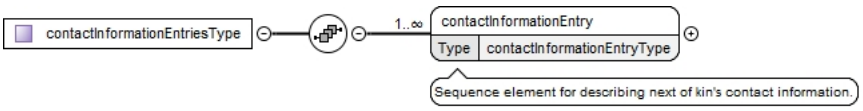
Namespace	employment
Diagram	
Used by	Element nextOfKinType/nextOfKinEntry
Model	name , contactInformationEntries{0,1} , additionalElements{0,1}
Children	additionalElements, contactInformationEntries, name

Attributes	**QName** | **Type** | **Use**
	otherTypeOfRelation | xs:string | optional
	`If "typeOfRelation" is "Other", state applicable type of relation.`
	typeOfRelation | typeOfRelationType | required
	`State the type of relation to the next of kin according to vcFAMILYRELATIONSHIP, or use "Other" if not applicable. Example "child"`

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

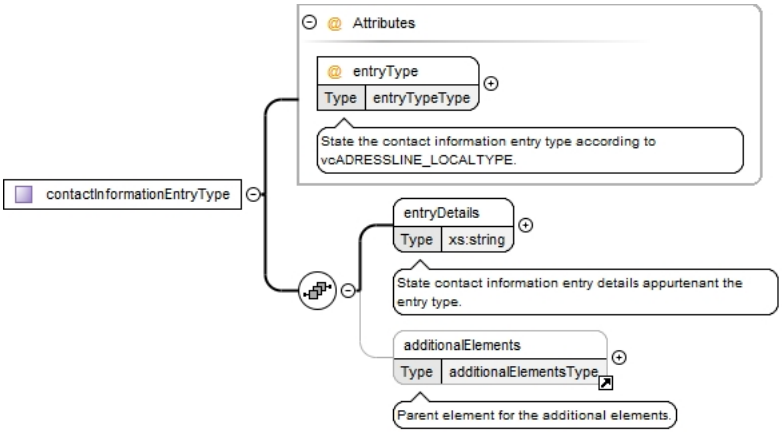
```
  <xs:sequence>
    <xs:element name="name" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="en">State the name of the next of kin.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element name="contactInformationEntries" type="contactInformationEntriesType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Sequence element for describing next of kin's contact
 information.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="additionalElements" minOccurs="0" maxOccurs="1"/>
  </xs:sequence>
  <xs:attribute name="typeOfRelation" type="typeOfRelationType" use="required">
    <xs:annotation>
      <xs:documentation xml:lang="en">State the type of relation to the next of kin according to
 vcFAMILYRELATIONSHIP, or use "Other" if not applicable. Example "child"</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="otherTypeOfRelation" type="xs:string" use="optional">
    <xs:annotation>
      <xs:documentation xml:lang="en">If "typeOfRelation" is "Other", state applicable type of
 relation.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Complex Type `contactInformationEntriesType`

Namespace	employment
Diagram	
Used by	Element nextOfKinEntryType/contactInformationEntries
Model	contactInformationEntry+
Children	contactInformationEntry

Source	`<xs:complexType name="contactInformationEntriesType">`
	`  <xs:sequence>`
	`  <xs:element name="contactInformationEntry" type="contactInformationEntryType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing next of kin's contact information.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `contactInformationEntryType`

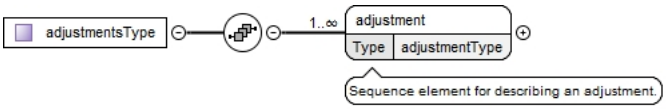
Namespace	employment
Diagram	
Used by	Element contactInformationEntriesType/contactInformationEntry
Model	entryDetails , additionalElements{0,1}

Children	additionalElements, entryDetails
Attributes	**QName** / **Type** / **Use** **entryType** / entryTypeType / required State the contact information entry type according to vcADRESSLINE_LOCALTYPE.

Source

```
<xs:complexType name="contactInformationEntryType">
  <xs:sequence>
    <xs:element name="entryDetails" type="xs:string">
      <xs:annotation>
        <xs:documentation xml:lang="en">State contact information entry details appurtenant the
 entry type.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="additionalElements" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="entryType" type="entryTypeType" use="required">
    <xs:annotation>
      <xs:documentation xml:lang="en">State the contact information entry type according to
 vcADRESSLINE_LOCALTYPE.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

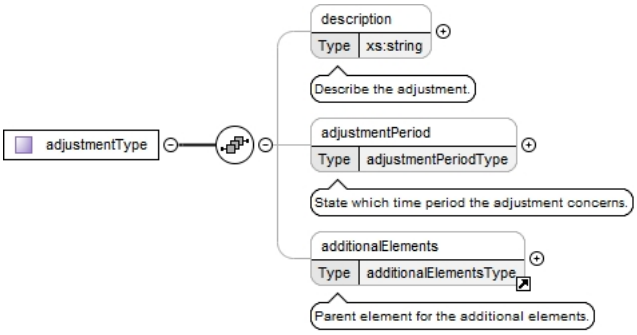
Complex Type `adjustmentsType`

Namespace	employment
Diagram	
Used by	Element employeeDescriptionType/adjustments
Model	adjustment+
Children	adjustment

Source

```
<xs:complexType name="adjustmentsType">
  <xs:sequence>
    <xs:element name="adjustment" type="adjustmentType" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation xml:lang="en">Sequence element for describing an adjustment.</
xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Complex Type `adjustmentType`

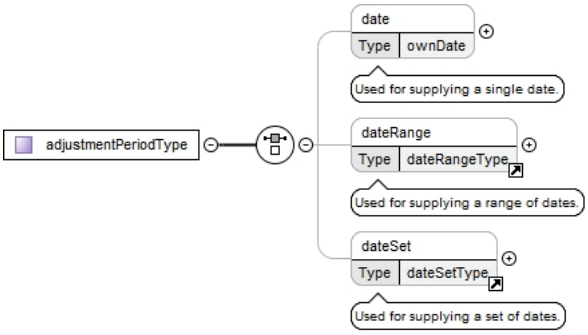
Namespace	employment
Diagram	
Used by	Element adjustmentsType/adjustment
Model	description{0,1} , adjustmentPeriod{0,1} , additionalElements{0,1}
Children	additionalElements, adjustmentPeriod, description

Source

```
<xs:complexType name="adjustmentType">
  <xs:sequence>
    <xs:element name="description" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Describe the adjustment.</xs:documentation>
```

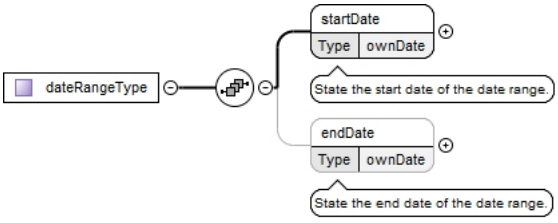
```
      </xs:annotation>
    </xs:element>
    <xs:element name="adjustmentPeriod" type="adjustmentPeriodType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">State which time period the adjustment concerns.</
xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="additionalElements" minOccurs="0"/>
  </xs:sequence>
</xs:complexType>
```

Complex Type `adjustmentPeriodType`

Namespace	employment
Diagram	
Used by	Element adjustmentType/adjustmentPeriod
Model	date{0,1} | dateRange{0,1} | dateSet{0,1}
Children	date, dateRange, dateSet

Source	`<xs:complexType name="adjustmentPeriodType">`
	`  <xs:choice>`
	`  <xs:element name="date" minOccurs="0" type="ownDate">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for supplying a single date.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element ref="dateRange" minOccurs="0"/>`
	`  <xs:element ref="dateSet" minOccurs="0"/>`
	`  </xs:choice>`
	`</xs:complexType>`

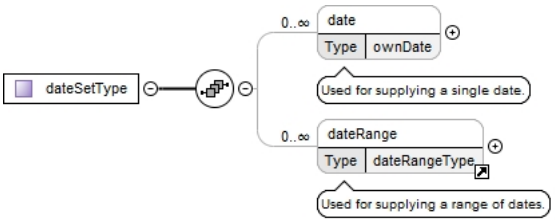
Complex Type `dateRangeType`

Namespace	employment
Diagram	
Used by	Element dateRange
Model	startDate , endDate{0,1}
Children	endDate, startDate

Source	`<xs:complexType name="dateRangeType">`
	`  <xs:sequence>`
	`  <xs:element name="startDate" type="ownDate">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the start date of the date range.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element name="endDate" minOccurs="0" maxOccurs="1" type="ownDate">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the end date of the date range.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`

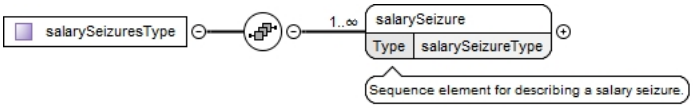
```
</xs:complexType>
```

Complex Type `dateSetType`

Namespace	employment
Diagram	
Used by	Element dateSet
Model	date* , dateRange*
Children	date, dateRange

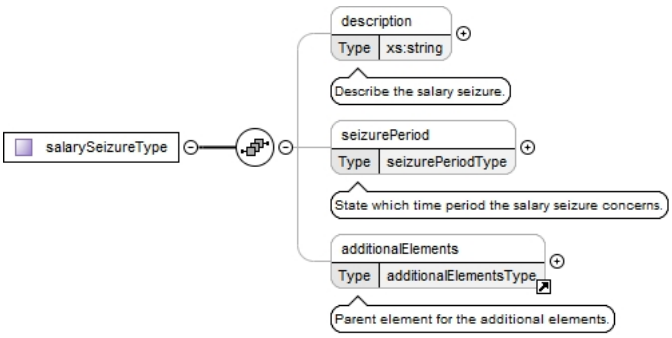
Source	`<xs:complexType name="dateSetType">`
	`  <xs:sequence>`
	`  <xs:element name="date" minOccurs="0" maxOccurs="unbounded" type="ownDate">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for supplying a single date.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element ref="dateRange" minOccurs="0" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for supplying a range of dates.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `salarySeizuresType`

Namespace	employment
Diagram	
Used by	Element employeeDescriptionType/salarySeizures
Model	salarySeizure+
Children	salarySeizure

Source	`<xs:complexType name="salarySeizuresType">`
	`  <xs:sequence>`
	`  <xs:element name="salarySeizure" type="salarySeizureType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing a salary seizure.</`
	`xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

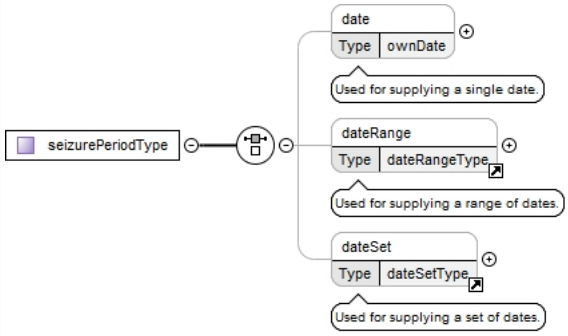
Complex Type `salarySeizureType`

Namespace	employment
Diagram	

Used by	Element salarySeizuresType/salarySeizure
Model	description{0,1} , seizurePeriod{0,1} , additionalElements{0,1}
Children	additionalElements, description, seizurePeriod

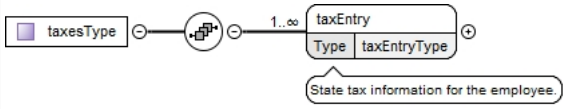
Source	`<xs:complexType name="salarySeizureType">`
	`  <xs:sequence>`
	`  <xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the salary seizure.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element name="seizurePeriod" type="seizurePeriodType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State which time period the salary seizure concerns.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element ref="additionalElements" minOccurs="0"/>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `seizurePeriodType`

Namespace	employment
Diagram	
Used by	Element salarySeizureType/seizurePeriod
Model	date{0,1} | dateRange{0,1} | dateSet{0,1}
Children	date, dateRange, dateSet

Source	`<xs:complexType name="seizurePeriodType">`
	`  <xs:choice>`
	`  <xs:element name="date" minOccurs="0" type="ownDate">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for supplying a single date.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element ref="dateRange" minOccurs="0"/>`
	`  <xs:element ref="dateSet" minOccurs="0"/>`
	`  </xs:choice>`
	`</xs:complexType>`

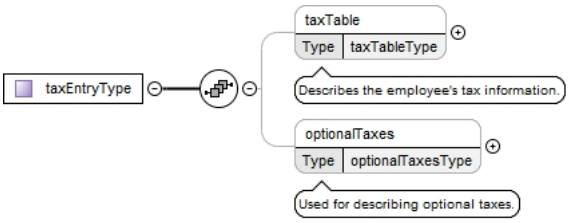
Complex Type `taxesType`

Namespace	employment
Diagram	
Used by	Element employeeDescriptionType/taxes
Model	taxEntry+
Children	taxEntry

Source	`<xs:complexType name="taxesType">`
	`  <xs:sequence>`
	`  <xs:element name="taxEntry" type="taxEntryType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State tax information for the employee.</xs:documentation>`
	`  </xs:annotation>`

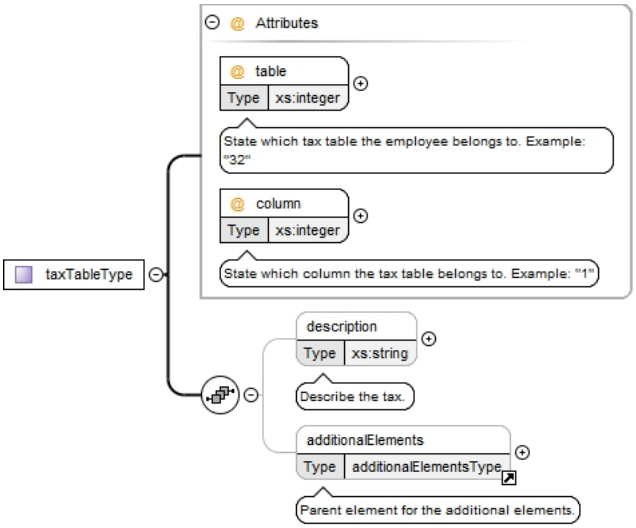
```
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Complex Type `taxEntryType`

Namespace	employment
Diagram	
Used by	Element taxesType/taxEntry
Model	taxTable{0,1} , optionalTaxes{0,1}
Children	optionalTaxes, taxTable

Source	`<xs:complexType name="taxEntryType">`
	`  <xs:sequence>`
	`  <xs:element name="taxTable" type="taxTableType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describes the employee's tax information.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element name="optionalTaxes" type="optionalTaxesType" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Used for describing optional taxes.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

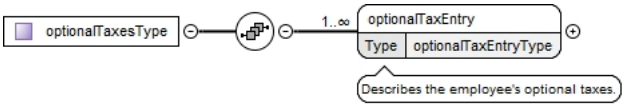
Complex Type `taxTableType`

Namespace	employment
Diagram	
Used by	Element taxEntryType/taxTable
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description
Attributes	

QName	Type	Use	
column	xs:integer	required	
	State which column the tax table belongs to. Example: "1"		
table	xs:integer	required	
	State which tax table the employee belongs to. Example: "32"		

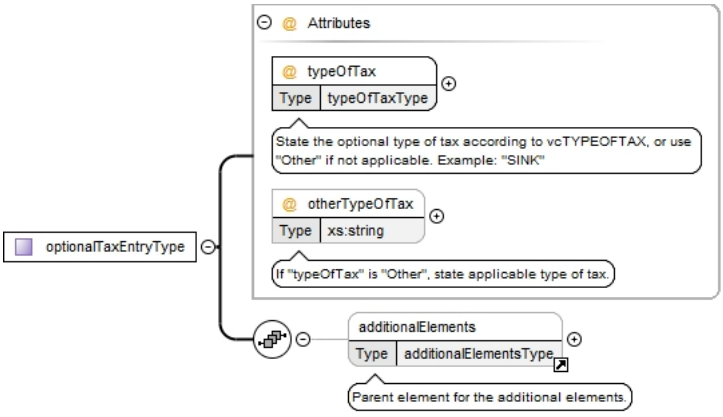
Source	`<xs:complexType name="taxTableType">`
	`  <xs:sequence>`
	`  <xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the tax.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element ref="additionalElements" minOccurs="0"/>`
	`  </xs:sequence>`
	`  <xs:attribute name="table" type="xs:integer" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State which tax table the employee belongs to. Example: "32"</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:attribute>`
	`  <xs:attribute name="column" type="xs:integer" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State which column the tax table belongs to. Example: "1"</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:attribute>`
	`</xs:complexType>`

Complex Type `optionalTaxesType`

Namespace	employment
Diagram	
Used by	Element taxEntryType/optionalTaxes
Model	optionalTaxEntry+
Children	optionalTaxEntry

Source	`<xs:complexType name="optionalTaxesType">`
	`  <xs:sequence>`
	`  <xs:element name="optionalTaxEntry" type="optionalTaxEntryType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describes the employee's optional taxes.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `optionalTaxEntryType`

Namespace	employment
Diagram	
Used by	Element optionalTaxesType/optionalTaxEntry
Model	additionalElements{0,1}
Children	additionalElements
Attributes	

QName	Type	Use	
otherTypeOfTax	xs:string	optional	
	`If "typeOfTax" is "Other", state applicable type of tax.`		

QName	Type	Use	
typeOfTax	typeOfTaxType	required	
	State the optional type of tax according to vcTYPEOFTAX, or use "Other" if not applicable. Example: "SINK"		

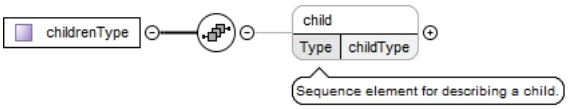
Source

```
<xs:complexType name="optionalTaxEntryType">
  <xs:sequence>
    <xs:element ref="additionalElements" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="typeOfTax" type="typeOfTaxType" use="required">
    <xs:annotation>
      <xs:documentation xml:lang="en">State the optional type of tax according to vcTYPEOFTAX, or
 use "Other" if not applicable. Example: "SINK"</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="otherTypeOfTax" type="xs:string" use="optional">
    <xs:annotation>
      <xs:documentation xml:lang="en">If "typeOfTax" is "Other", state applicable type of tax.</
xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Complex Type `childrenType`

Namespace: employment

Diagram



Used by: Element employeeDescriptionType/children

Model: child{0,1}

Children: child

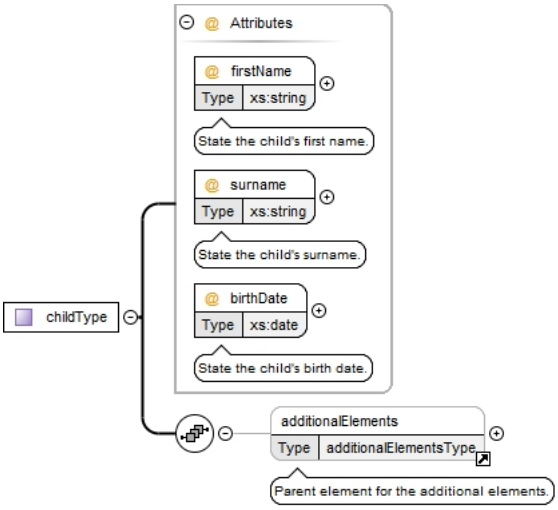
Source

```
<xs:complexType name="childrenType">
  <xs:sequence>
    <xs:element name="child" type="childType" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Sequence element for describing a child.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Complex Type `childType`

Namespace: employment

Diagram



Used by: Element childrenType/child

Model: additionalElements{0,1}

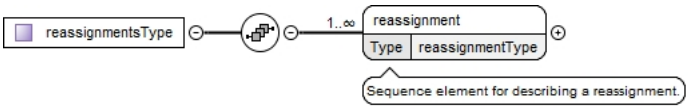
Children: additionalElements

Attributes	**QName**	**Type**	**Use**	
	birthDate	xs:date	required	
		State the child's birth date.		
	firstName	xs:string	required	
		State the child's first name.		
	surname	xs:string	required	
		State the child's surname.		

Source

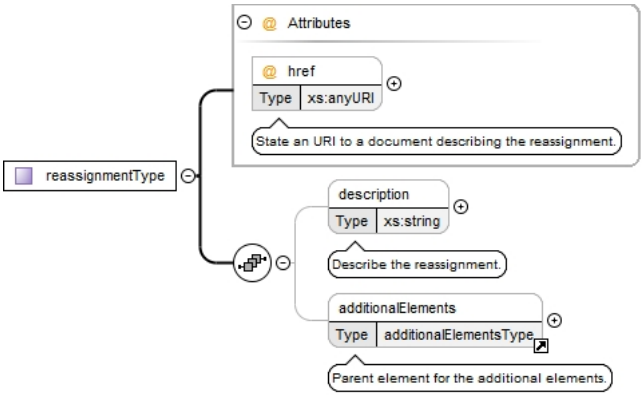
```
<xs:complexType name="childType">
  <xs:sequence>
    <xs:element ref="additionalElements" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="firstName" type="xs:string" use="required">
    <xs:annotation>
      <xs:documentation xml:lang="en">State the child's first name.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="surname" type="xs:string" use="required">
    <xs:annotation>
      <xs:documentation xml:lang="en">State the child's surname.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="birthDate" type="xs:date" use="required">
    <xs:annotation>
      <xs:documentation xml:lang="en">State the child's birth date.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Complex Type `reassignmentsType`

Namespace	employment
Diagram	
Used by	Element employeeDescriptionType/reassignments
Model	reassignment+
Children	reassignment
Source	

```
<xs:complexType name="reassignmentsType">
  <xs:sequence>
    <xs:element name="reassignment" type="reassignmentType" maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation xml:lang="en">Sequence element for describing a reassignment.</
xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

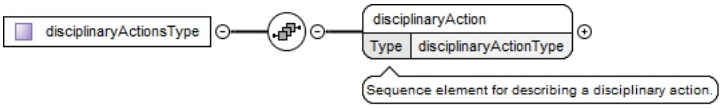
Complex Type `reassignmentType`

Namespace	employment
Diagram	

Used by	Element reassignmentsType/reassignment
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description
Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional State an URI to a document describing the reassignment.

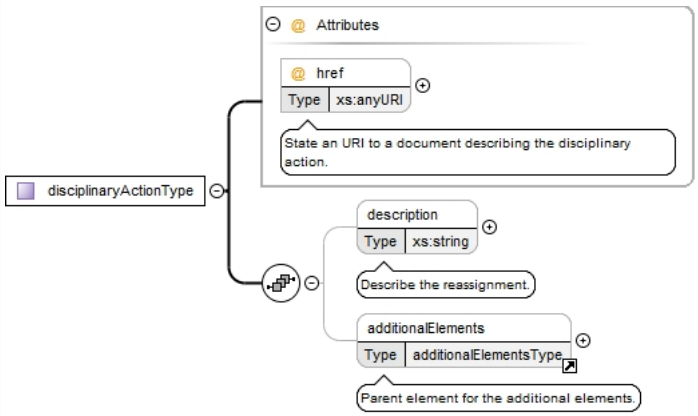
Source	`<xs:complexType name="reassignmentType">`
	`  <xs:sequence>`
	`  <xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the reassignment.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element ref="additionalElements" minOccurs="0"/>`
	`  </xs:sequence>`
	`  <xs:attribute name="href" type="xs:anyURI" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State an URI to a document describing the reassignment.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:attribute>`
	`</xs:complexType>`

Complex Type `disciplinaryActionsType`

Namespace	employment
Diagram	
Used by	Element employeeDescriptionType/disciplinaryActions
Model	disciplinaryAction
Children	disciplinaryAction

Source	`<xs:complexType name="disciplinaryActionsType">`
	`  <xs:sequence>`
	`  <xs:element name="disciplinaryAction" type="disciplinaryActionType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing a disciplinary action.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

Complex Type `disciplinaryActionType`

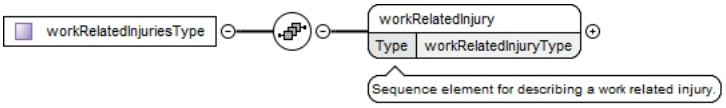
Namespace	employment
Diagram	
Used by	Element disciplinaryActionsType/disciplinaryAction
Model	description{0,1} , additionalElements{0,1}

Children	additionalElements, description			
Attributes	**QName**	**Type**	**Use**	
	href	xs:anyURI	optional	
		State an URI to a document describing the disciplinary action.		

Source

```
<xs:complexType name="disciplinaryActionType">
  <xs:sequence>
    <xs:element name="description" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Describe the reassignment.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="additionalElements" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="href" type="xs:anyURI" use="optional">
    <xs:annotation>
      <xs:documentation xml:lang="en">State an URI to a document describing the disciplinary
 action.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

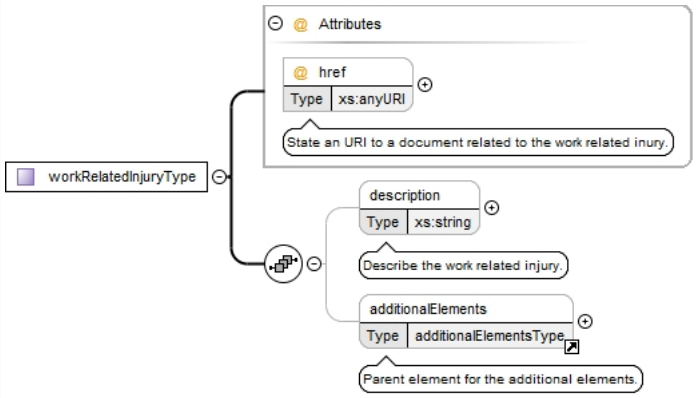
Complex Type workRelatedInjuriesType

Namespace	employment
Diagram	
Used by	Element employeeDescriptionType/workRelatedInjuries
Model	workRelatedInjury
Children	workRelatedInjury

Source

```
<xs:complexType name="workRelatedInjuriesType">
  <xs:sequence>
    <xs:element name="workRelatedInjury" type="workRelatedInjuryType">
      <xs:annotation>
        <xs:documentation xml:lang="en">Sequence element for describing a work related injury.</
xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Complex Type workRelatedInjuryType

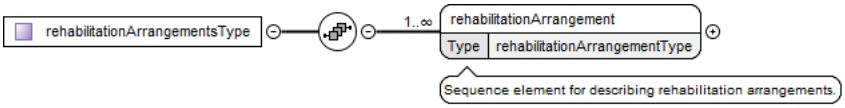
Namespace	employment
Diagram	
Used by	Element workRelatedInjuriesType/workRelatedInjury
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description
Attributes	**QName** / **Type** / **Use**
	href / xs:anyURI / optional

	QName	Type	Use	
		State an URI to a document related to the work related inury.		

Source

```
<xs:complexType name="workRelatedInjuryType">
  <xs:sequence>
    <xs:element name="description" type="xs:string" minOccurs="0">
      <xs:annotation>
        <xs:documentation xml:lang="en">Describe the work related injury.</xs:documentation>
      </xs:annotation>
    </xs:element>
    <xs:element ref="additionalElements" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="href" type="xs:anyURI" use="optional">
    <xs:annotation>
      <xs:documentation xml:lang="en">State an URI to a document related to the work related
 inury.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

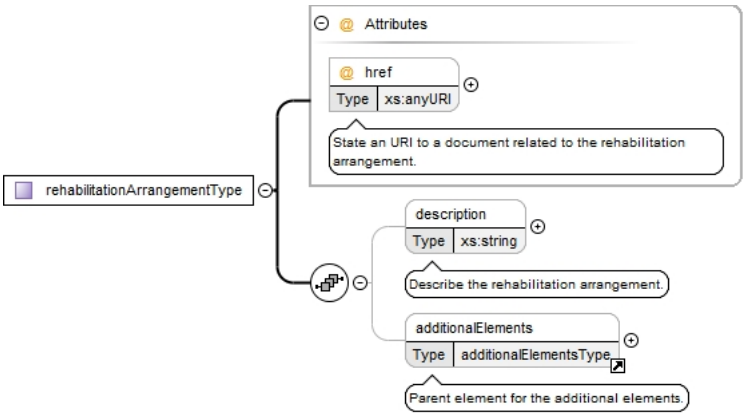
Complex Type `rehabilitationArrangementsType`

Namespace	employment
Diagram	
Used by	Element employeeDescriptionType/rehabilitationArrangements
Model	rehabilitationArrangement+
Children	rehabilitationArrangement

Source

```
<xs:complexType name="rehabilitationArrangementsType">
  <xs:sequence>
    <xs:element name="rehabilitationArrangement" type="rehabilitationArrangementType"
 maxOccurs="unbounded">
      <xs:annotation>
        <xs:documentation xml:lang="en">Sequence element for describing rehabilitation
 arrangements.</xs:documentation>
      </xs:annotation>
    </xs:element>
  </xs:sequence>
</xs:complexType>
```

Complex Type `rehabilitationArrangementType`

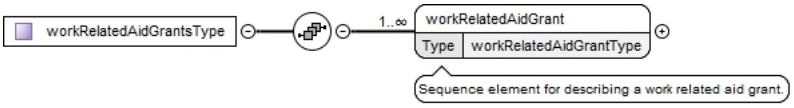
Namespace	employment
Diagram	
Used by	Element rehabilitationArrangementsType/rehabilitationArrangement
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description

Attributes

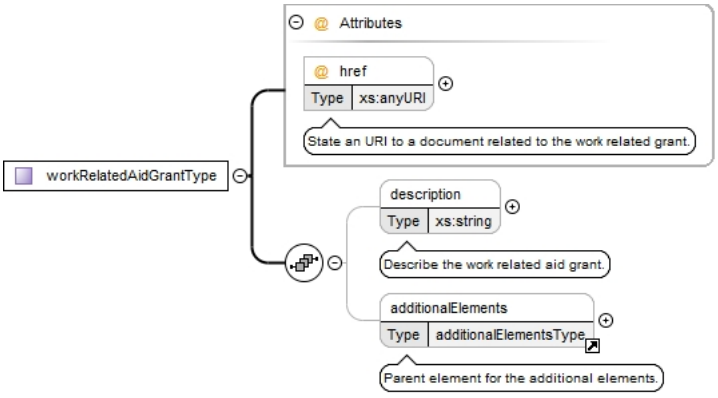
QName	Type	Use	
href	xs:anyURI	optional	
	State an URI to a document related to the rehabilitation arrangement.		

Source	<xs:complexType name="rehabilitationArrangementType"> <xs:sequence> <xs:element name="description" type="xs:string" minOccurs="0"> <xs:annotation> <xs:documentation xml:lang="en">Describe the rehabilitation arrangement.</xs:documentation> </xs:annotation> </xs:element> <xs:element ref="additionalElements" minOccurs="0"/> </xs:sequence> <xs:attribute name="href" type="xs:anyURI" use="optional"> <xs:annotation> <xs:documentation xml:lang="en">State an URI to a document related to the rehabilitation arrangement.</xs:documentation> </xs:annotation> </xs:attribute> </xs:complexType>

Complex Type workRelatedAidGrantsType

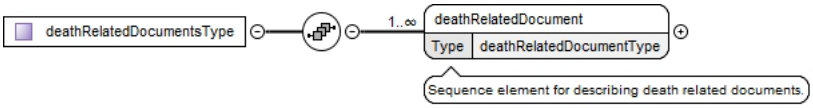
Namespace	employment
Diagram	
Used by	Element employeeDescriptionType/workRelatedAidGrants
Model	workRelatedAidGrant+
Children	workRelatedAidGrant
Source	<xs:complexType name="workRelatedAidGrantsType"> <xs:sequence> <xs:element name="workRelatedAidGrant" type="workRelatedAidGrantType" maxOccurs="unbounded"> <xs:annotation> <xs:documentation xml:lang="en">Sequence element for describing a work related aid grant.</xs:documentation> </xs:annotation> </xs:element> </xs:sequence> </xs:complexType>

Complex Type workRelatedAidGrantType

Namespace	employment
Diagram	
Used by	Element workRelatedAidGrantsType/workRelatedAidGrant
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description
Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional State an URI to a document related to the work related grant.
Source	<xs:complexType name="workRelatedAidGrantType"> <xs:sequence> <xs:element name="description" type="xs:string" minOccurs="0"> <xs:annotation> <xs:documentation xml:lang="en">Describe the work related aid grant.</xs:documentation> </xs:annotation>

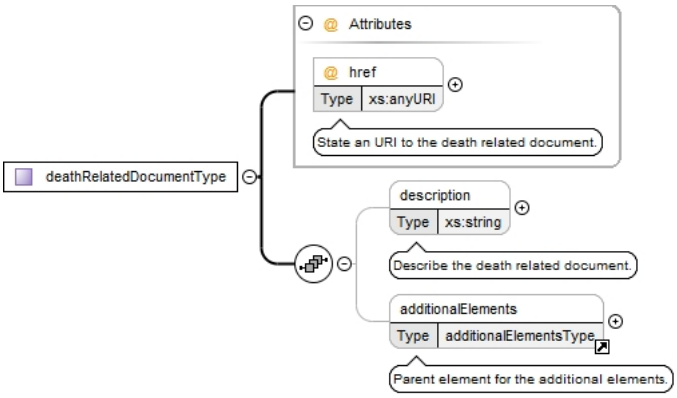
```
    </xs:element>
    <xs:element ref="additionalElements" minOccurs="0"/>
  </xs:sequence>
  <xs:attribute name="href" type="xs:anyURI" use="optional">
    <xs:annotation>
      <xs:documentation xml:lang="en">State an URI to a document related to the work related
 grant.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Complex Type `deathRelatedDocumentsType`

Namespace	employment
Diagram	
Used by	Element employeeDescriptionType/deathRelatedDocuments
Model	deathRelatedDocument+
Children	deathRelatedDocument

Source	`<xs:complexType name="deathRelatedDocumentsType">`
	`  <xs:sequence>`
	`  <xs:element name="deathRelatedDocument" type="deathRelatedDocumentType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing death related documents.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

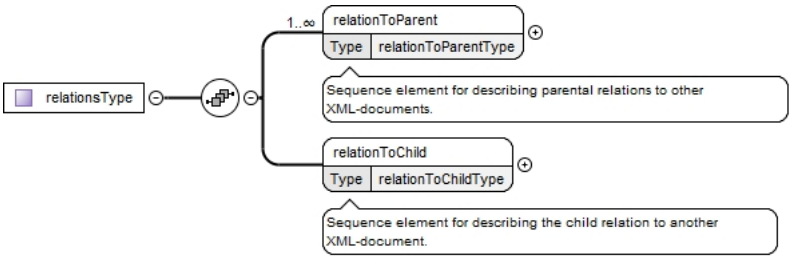
Complex Type `deathRelatedDocumentType`

Namespace	employment
Diagram	
Used by	Element deathRelatedDocumentsType/deathRelatedDocument
Model	description{0,1} , additionalElements{0,1}
Children	additionalElements, description
Attributes	**QName** / **Type** / **Use** **href** / xs:anyURI / optional State an URI to the death related document.

Source	`<xs:complexType name="deathRelatedDocumentType">`
	`  <xs:sequence>`
	`  <xs:element name="description" type="xs:string" minOccurs="0">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describe the death related document.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element ref="additionalElements" minOccurs="0"/>`
	`  </xs:sequence>`
	`  <xs:attribute name="href" type="xs:anyURI" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State an URI to the death related document.</xs:documentation>`

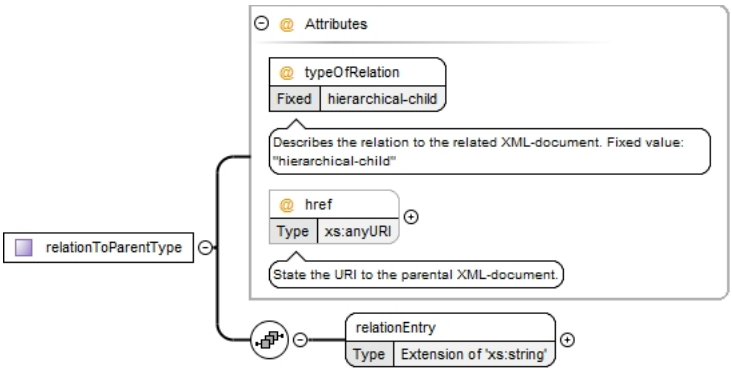
```
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Complex Type `relationsType`

Namespace	employment
Diagram	
Used by	Element employeeDescriptionType/relations
Model	relationToParent+ , relationToChild
Children	relationToChild, relationToParent

Source	`<xs:complexType name="relationsType">`
	`  <xs:sequence>`
	`  <xs:element name="relationToParent" type="relationToParentType" maxOccurs="unbounded">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing parental relations to other XML-documents.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  <xs:element name="relationToChild" type="relationToChildType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Sequence element for describing the child relation to another XML-document.</xs:documentation>`
	`  </xs:annotation>`
	`  </xs:element>`
	`  </xs:sequence>`
	`</xs:complexType>`

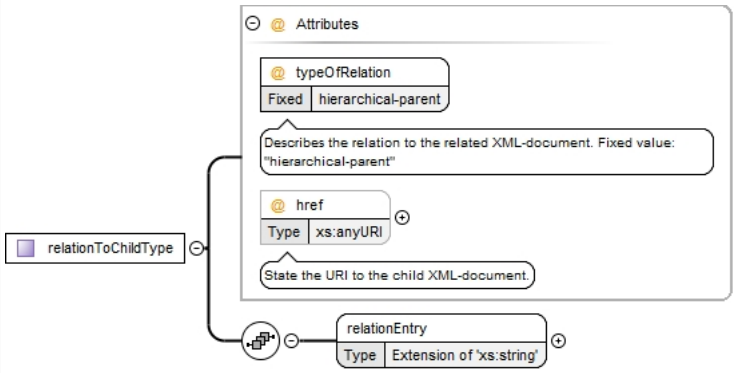
Complex Type `relationToParentType`

Namespace	employment
Diagram	
Used by	Element relationsType/relationToParent
Model	relationEntry
Children	relationEntry
Attributes	**QName** / **Type** / **Fixed** / **Use** **href** / xs:anyURI / / optional State the URI to the parental XML-document. **typeOfRelation** / / hierarchical-child / required Describes the relation to the related XML-document. Fixed value: "hierarchical-child"

Source	`<xs:complexType name="relationToParentType">`
	`  <xs:sequence>`
	`  <xs:element name="relationEntry">`

```
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="parentOrganisationReferenceCode" type="xs:string" use="required">
              <xs:annotation>
                <xs:documentation xml:lang="en">State the parent organisation's reference code.</
xs:documentation>
              </xs:annotation>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="typeOfRelation" use="required" fixed="hierarchical-child">
    <xs:annotation>
      <xs:documentation xml:lang="en">Describes the relation to the related XML-document. Fixed
 value: "hierarchical-child"</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="href" use="optional" type="xs:anyURI">
    <xs:annotation>
      <xs:documentation xml:lang="en">State the URI to the parental XML-document.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

Complex Type `relationToChildType`

Namespace	employment
Diagram	
Used by	Element relationsType/relationToChild
Model	relationEntry
Children	relationEntry

Attributes

QName	Type	Fixed	Use	
href	xs:anyURI		optional	
	State the URI to the child XML-document.			
typeOfRelation		hierarchical-parent	required	
	Describes the relation to the related XML-document. Fixed value: "hierarchical-parent"			

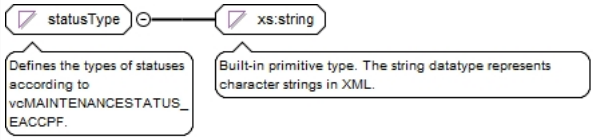
Source

```
<xs:complexType name="relationToChildType">
  <xs:sequence>
    <xs:element name="relationEntry">
      <xs:complexType>
        <xs:simpleContent>
          <xs:extension base="xs:string">
            <xs:attribute name="childOrganisationReferenceCode" type="xs:string" use="required">
              <xs:annotation>
                <xs:documentation xml:lang="en">State the child document's reference code.</
xs:documentation>
              </xs:annotation>
            </xs:attribute>
          </xs:extension>
        </xs:simpleContent>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
  <xs:attribute name="typeOfRelation" use="required" fixed="hierarchical-parent">
    <xs:annotation>
```

```
      <xs:documentation xml:lang="en">Describes the relation to the related XML-document. Fixed
 value: "hierarchical-parent"</xs:documentation>
    </xs:annotation>
  </xs:attribute>
  <xs:attribute name="href" use="optional" type="xs:anyURI">
    <xs:annotation>
      <xs:documentation xml:lang="en">State the URI to the child XML-document.</xs:documentation>
    </xs:annotation>
  </xs:attribute>
</xs:complexType>
```

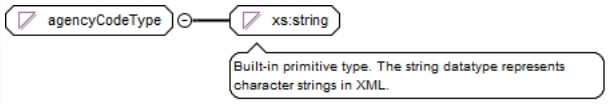
Simple Type(s)

Simple Type `statusType`

Namespace	employment
Annotations	Defines the types of statuses according to vcMAINTENANCESTATUS_EACCPF.
Diagram	
Type	restriction of xs:string
Facets	enumeration `cancelled` enumeration `deleted` enumeration `deletedReplaced` enumeration `derived` enumeration `new` enumeration `revised`
Used by	Element controlType/status
Source	

```
<xs:simpleType name="statusType">
  <xs:annotation>
    <xs:documentation xml:lang="en">Defines the types of statuses according to
 vcMAINTENANCESTATUS_EACCPF.</xs:documentation>
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="cancelled"/>
    <xs:enumeration value="deleted"/>
    <xs:enumeration value="deletedReplaced"/>
    <xs:enumeration value="derived"/>
    <xs:enumeration value="new"/>
    <xs:enumeration value="revised"/>
  </xs:restriction>
</xs:simpleType>
```

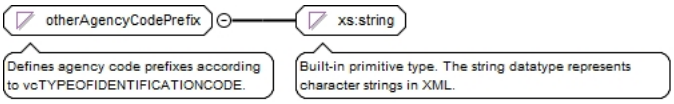
Simple Type `agencyCodeType`

Namespace	employment
Diagram	
Type	restriction of xs:string
Facets	pattern `[A-Z]{2}\-[a-zA-Z0-9/:\-]{1,10}`
Used by	Element maintenanceAgencyType/agencyCode
Source	

```
<xs:simpleType name="agencyCodeType">
  <xs:restriction base="xs:string">
    <xs:pattern value="[A-Z]{2}\-[a-zA-Z0-9/:\-]{1,10}"/>
  </xs:restriction>
</xs:simpleType>
```

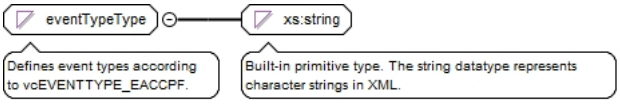
Simple Type `otherAgencyCodePrefix`

Namespace	employment

Annotations	Defines agency code prefixes according to vcTYPEOFIDENTIFICATIONCODE.
Diagram	
Type	restriction of xs:string
Facets	enumeration VAT enumeration DUNS enumeration ORG enumeration HSA enumeration Local enumeration URI enumeration PersonalIdentityNumber enumeration Co-ordinationNumber enumeration Other
Used by	Attribute otherAgencyCodeType/@agencyCodePrefix

Source	`<xs:simpleType name="otherAgencyCodePrefix">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Defines agency code prefixes according to`
	` vcTYPEOFIDENTIFICATIONCODE.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:string">`
	`  <xs:enumeration value="VAT"/>`
	`  <xs:enumeration value="DUNS"/>`
	`  <xs:enumeration value="ORG"/>`
	`  <xs:enumeration value="HSA"/>`
	`  <xs:enumeration value="Local"/>`
	`  <xs:enumeration value="URI"/>`
	`  <xs:enumeration value="PersonalIdentityNumber"/>`
	`  <xs:enumeration value="Co-ordinationNumber"/>`
	`  <xs:enumeration value="Other"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

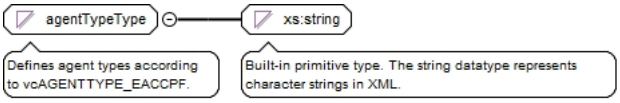
Simple Type eventTypeType

Namespace	employment
Annotations	Defines event types according to vcEVENTTYPE_EACCPF.
Diagram	
Type	restriction of xs:string
Facets	enumeration cancelled enumeration created enumeration deleted enumeration derived enumeration revised1 enumeration updated
Used by	Attribute maintenanceHistoryEventType/@eventType

Source	`<xs:simpleType name="eventTypeType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Defines event types according to vcEVENTTYPE_EACCPF.</`
	`xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:string">`
	`  <xs:enumeration value="cancelled"/>`
	`  <xs:enumeration value="created"/>`
	`  <xs:enumeration value="deleted"/>`
	`  <xs:enumeration value="derived"/>`
	`  <xs:enumeration value="revised1"/>`
	`  <xs:enumeration value="updated"/>`

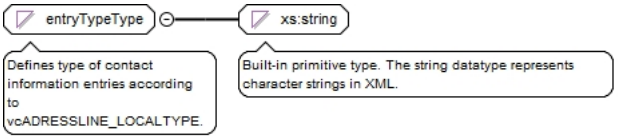
```
  </xs:restriction>
</xs:simpleType>
```

Simple Type `agentTypeType`

Namespace	employment
Annotations	`Defines agent types according to vcAGENTTYPE_EACCPF.`
Diagram	
Type	restriction of xs:string
Facets	enumeration `human` enumeration `machine` enumeration `Other`
Used by	Attribute maintenanceHistoryEventType/@agentType

Source	`<xs:simpleType name="agentTypeType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Defines agent types according to vcAGENTTYPE_EACCPF.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:string">`
	`  <xs:enumeration value="human"/>`
	`  <xs:enumeration value="machine"/>`
	`  <xs:enumeration value="Other"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

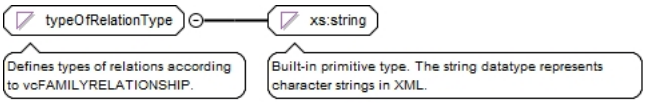
Simple Type `entryTypeType`

Namespace	employment
Annotations	`Defines type of contact information entries according to vcADRESSLINE_LOCALTYPE.`
Diagram	
Type	restriction of xs:string
Facets	enumeration `postalAddress` enumeration `postalCode` enumeration `postalCity` enumeration `postalBox` enumeration `municipalityCode` enumeration `municipality` enumeration `parish` enumeration `parishCode` enumeration `province` enumeration `county` enumeration `country` enumeration `phoneNumber` enumeration `fax` enumeration `email` enumeration `homepage`
Used by	Attribute contactInformationEntryType/@entryType

Source	`<xs:simpleType name="entryTypeType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Defines type of contact information entries according to vcADRESSLINE_LOCALTYPE.</xs:documentation>`

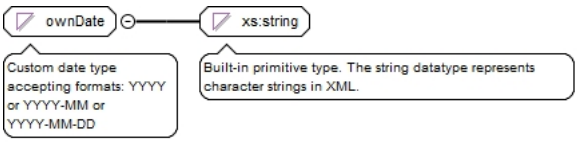
```
  </xs:annotation>
  <xs:restriction base="xs:string">
    <xs:enumeration value="postalAddress"/>
    <xs:enumeration value="postalCode"/>
    <xs:enumeration value="postalCity"/>
    <xs:enumeration value="postalBox"/>
    <xs:enumeration value="municipalityCode"/>
    <xs:enumeration value="municipality"/>
    <xs:enumeration value="parish"/>
    <xs:enumeration value="parishCode"/>
    <xs:enumeration value="province"/>
    <xs:enumeration value="county"/>
    <xs:enumeration value="country"/>
    <xs:enumeration value="phoneNumber"/>
    <xs:enumeration value="fax"/>
    <xs:enumeration value="email"/>
    <xs:enumeration value="homepage"/>
  </xs:restriction>
</xs:simpleType>
```

Simple Type `typeOfRelationType`

Namespace	employment
Annotations	`Defines types of relations according to vcFAMILYRELATIONSHIP.`
Diagram	
Type	restriction of xs:string
Facets	enumeration `father` enumeration `mother` enumeration `child` enumeration `guardian` enumeration `partner` enumeration `Other`
Used by	Attribute nextOfKinEntryType/@typeOfRelation

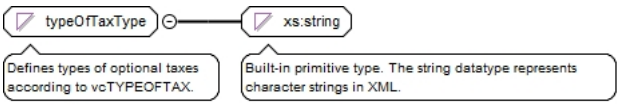
Source	`<xs:simpleType name="typeOfRelationType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Defines types of relations according to vcFAMILYRELATIONSHIP.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:string">`
	`  <xs:enumeration value="father"/>`
	`  <xs:enumeration value="mother"/>`
	`  <xs:enumeration value="child"/>`
	`  <xs:enumeration value="guardian"/>`
	`  <xs:enumeration value="partner"/>`
	`  <xs:enumeration value="Other"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `ownDate`

Namespace	employment
Annotations	`Custom date type accepting formats: YYYY or YYYY-MM or YYYY-MM-DD`
Diagram	
Type	restriction of xs:string
Facets	pattern `[\d]{4}\-[\d]{2}\-[\d]{2}|[\d]{4}\-[\d]{2}|[\d]{4}`
Used by	Elements adjustmentPeriodType/date, dateRangeType/endDate, dateRangeType/startDate, dateSetType/date, seizurePeriodType/date

Source	`<xs:simpleType name="ownDate">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Custom date type accepting formats: YYYY or YYYY-MM or YYYY-MM-DD</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:string">`
	`  <xs:pattern value="[\d]{4}\-[\d]{2}\-[\d]{2}|[\d]{4}\-[\d]{2}|[\d]{4}"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Simple Type `typeOfTaxType`

Namespace	employment
Annotations	Defines types of optional taxes according to vcTYPEOFTAX.
Diagram	
Type	restriction of xs:string
Facets	enumeration: AdditionalTaxPayment enumeration: SINK enumeration: Other
Used by	Attribute: optionalTaxEntryType/@typeOfTax

Source	`<xs:simpleType name="typeOfTaxType">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Defines types of optional taxes according to vcTYPEOFTAX.</xs:documentation>`
	`  </xs:annotation>`
	`  <xs:restriction base="xs:string">`
	`  <xs:enumeration value="AdditionalTaxPayment"/>`
	`  <xs:enumeration value="SINK"/>`
	`  <xs:enumeration value="Other"/>`
	`  </xs:restriction>`
	`</xs:simpleType>`

Namespace: ""

Attribute(s)

Attribute `otherAgencyCodeType / @agencyCodePrefix`

Namespace	No namespace
Annotations	State a prefix according to value list vcTYPEOFIDENTIFICATIONCODE, or use "Other" if not applicable.
Type	otherAgencyCodePrefix
Properties	use: optional
Facets	enumeration: VAT enumeration: DUNS enumeration: ORG enumeration: HSA enumeration: Local enumeration: URI enumeration: PersonalIdentityNumber enumeration: Co-ordinationNumber enumeration: Other
Used by	Complex Type: otherAgencyCodeType

Source	`<xs:attribute name="agencyCodePrefix" type="otherAgencyCodePrefix" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State a prefix according to value list vcTYPEOFIDENTIFICATIONCODE, or use "Other" if not applicable.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute otherAgencyCodeType / @otherAgencyCodePrefix

Namespace	No namespace
Annotations	If attribute "agencyCodePrefix" is "Other", state an applicable prefix.
Type	xs:string
Properties	use: optional
Used by	Complex Type otherAgencyCodeType

Source	`<xs:attribute name="otherAgencyCodePrefix" type="xs:string" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">If attribute "agencyCodePrefix" is "Other", state an applicable prefix.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute languageType / @languageCode

Namespace	No namespace
Annotations	Language code must be written in lowercase and conform to the ISO639-2 standard. Example: "swe"
Type	xs:string
Properties	content: simple
Used by	Complex Type languageType

Source	`<xs:attribute name="languageCode" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Language code must be written in lowercase and conform to the ISO639-2 standard. Example: "swe"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute languageType / @language

Namespace	No namespace
Annotations	Language must have the first character capitalized and conform to the ISO639-2 standard. Example: "Swedish"
Type	xs:string
Properties	content: simple
Used by	Complex Type languageType

Source	`<xs:attribute name="language" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Language must have the first character capitalized and conform to the ISO639-2 standard. Example: "Swedish"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute scriptType / @scriptCode

Namespace	No namespace
Annotations	Script code must have the first character capitalized and conform to the ISO15924 standard. Example: "Latn"
Type	xs:string
Properties	content: simple
Used by	Complex Type scriptType

Source	`<xs:attribute name="scriptCode" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Script code must have the first character capitalized and conform to the ISO15924 standard. Example: "Latn"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute scriptType / @script

Namespace	No namespace

Annotations	Script must have the first character capitalized and conform to the ISO15924 standard. Example: "Latin"
Type	xs:string
Properties	content: simple
Used by	Complex Type scriptType

Source	`<xs:attribute name="script" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Script must have the first character capitalized and conform to the ISO15924 standard. Example: "Latin"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute citationType / @href

Namespace	No namespace
Annotations	State the URI for the FGS used in this XML-document. Example: "https://myndighet.se/Försöksmyndighetens_FGS_Personal_V_20181224.pdf"
Type	xs:anyURI
Properties	content: simple
Used by	Complex Type citationType

Source	`<xs:attribute name="href" type="xs:anyURI">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the URI for the FGS used in this XML-document. Example: "https://myndighet.se/Försöksmyndighetens_FGS_Personal_V_20181224.pdf"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute maintenanceHistoryEventType / @eventType

Namespace	No namespace
Annotations	State what kind of change has been made accorrding to vcEVENTTYPE_EACCPF. Example: "created"
Type	eventTypeType
Properties	use: required
Facets	enumeration cancelled
	enumeration created
	enumeration deleted
	enumeration derived
	enumeration revised1
	enumeration updated
Used by	Complex Type maintenanceHistoryEventType

Source	`<xs:attribute name="eventType" type="eventTypeType" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State what kind of change has been made accorrding to vcEVENTTYPE_EACCPF. Example: "created"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute maintenanceHistoryEventType / @eventDateTime

Namespace	No namespace
Annotations	State the date and time the change was made. Example: "2001-04-13T22:30:00"
Type	xs:dateTime
Properties	use: required
Used by	Complex Type maintenanceHistoryEventType

Source	`<xs:attribute name="eventDateTime" type="xs:dateTime" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the date and time the change was made. Example: "2001-04-13T22:30:00"</xs:documentation>`
	`  </xs:annotation>`

```
</xs:attribute>
```

Attribute `maintenanceHistoryEventType / @agentType`

Namespace	No namespace
Annotations	`State the type of agent that performed the changes according to vcAGENTTYPE_EACCPF, or use "Other" if not applicable. Example: "human"`
Type	agentTypeType
Properties	use: `required`
Facets	enumeration `human` enumeration `machine` enumeration `Other`
Used by	Complex Type maintenanceHistoryEventType

Source	`<xs:attribute name="agentType" type="agentTypeType" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the type of agent that performed the changes according to vcAGENTTYPE_EACCPF, or use "Other" if not applicable. Example: "human"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `maintenanceHistoryEventType / @otherAgentType`

Namespace	No namespace
Annotations	`If attribute "agentType" is "Other", state an applicable agent type.`
Type	xs:string
Properties	use: `optional`
Used by	Complex Type maintenanceHistoryEventType

Source	`<xs:attribute name="otherAgentType" type="xs:string" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">If attribute "agentType" is "Other", state an applicable agent type.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `maintenanceHistoryEventType / @agent`

Namespace	No namespace
Annotations	`State the name of the agent who performed the changes. Example: "Anna Andersson"`
Type	xs:string
Properties	use: `required`
Used by	Complex Type maintenanceHistoryEventType

Source	`<xs:attribute name="agent" type="xs:string" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the name of the agent who performed the changes. Example: "Anna Andersson"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `attribute / @name`

Namespace	No namespace
Annotations	`Name of own defined element`
Type	xs:string
Properties	use: `required`
Used by	Element attribute

Source	`<xs:attribute name="name" type="xs:string" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Name of own defined element</xs:documentation>`
	`  </xs:annotation>`

```
</xs:attribute>
```

Attribute attribute / @dataType

Namespace	No namespace
Annotations	Datatype for own defined element
Type	xs:string
Properties	content: simple
Used by	Element attribute

Source	`<xs:attribute name="dataType" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Datatype for own defined element</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute attribute / @format

Namespace	No namespace
Annotations	Format for own defined element
Type	xs:string
Properties	content: simple
Used by	Element attribute

Source	`<xs:attribute name="format" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Format for own defined element</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute ownElementType / @name

Namespace	No namespace
Annotations	Name of own defined element
Type	xs:string
Properties	use: required
Used by	Complex Type ownElementType

Source	`<xs:attribute name="name" use="required" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Name of own defined element</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute ownElementType / @dataType

Namespace	No namespace
Annotations	Datatype for own defined element element
Type	xs:string
Properties	content: simple
Used by	Complex Type ownElementType

Source	`<xs:attribute name="dataType" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Datatype for own defined element element</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute ownElementType / @format

Namespace	No namespace
Annotations	Format for own defined element
Type	xs:string

Properties	content: simple
Used by	Complex Type ownElementType

Source	`<xs:attribute name="format" type="xs:string">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Format for own defined element</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute eSignatureComplexType / @present

Namespace	No namespace
Annotations	Attribute indicating that an e-signature has been present or not
Type	xs:boolean
Properties	use: required
Used by	Complex Type eSignatureComplexType

Source	`<xs:attribute name="present" type="xs:boolean" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Attribute indicating that an e-signature has been present or not</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute eSignatureComplexType / @dateESignatureIsVerified

Namespace	No namespace
Annotations	Attribute with the datetime giving when the e-signature was verified
Type	xs:dateTime
Properties	use: optional
Used by	Complex Type eSignatureComplexType

Source	`<xs:attribute name="dateESignatureIsVerified" type="xs:dateTime" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Attribute with the datetime giving when the e-signature was verified</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute AppendixType / @disposable

Namespace	No namespace
Annotations	If the appendix can be disposed of before the aggregation or record is disposed of value="true" otherwise false
Type	xs:boolean
Properties	use: optional
Used by	Complex Type AppendixType

Source	`<xs:attribute name="disposable" type="xs:boolean" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">If the appendix can be disposed of before the aggregation or record is disposed of value="true" otherwise false</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute AppendixType / @name

Namespace	No namespace
Annotations	Name of the appendix
Type	xs:string
Properties	use: required
Used by	Complex Type AppendixType
Source	`<xs:attribute name="name" type="xs:string" use="required">`

```
  <xs:annotation>
    <xs:documentation xml:lang="en">Name of the appendix</xs:documentation>
  </xs:annotation>
</xs:attribute>
```

Attribute AppendixType / @description

Namespace	No namespace
Annotations	Description of appendix
Type	xs:string
Properties	use: optional
Used by	Complex Type AppendixType

Source	`<xs:attribute name="description" type="xs:string" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Description of appendix</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute AppendixType / @fileFormat

Namespace	No namespace
Annotations	File format of appendix
Type	xs:string
Properties	use: optional
Used by	Complex Type AppendixType

Source	`<xs:attribute name="fileFormat" type="xs:string" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">File format of appendix</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute AppendixType / @originalFileFormat

Namespace	No namespace
Annotations	Original file format of appendix
Type	xs:string
Properties	use: optional
Used by	Complex Type AppendixType

Source	`<xs:attribute name="originalFileFormat" type="xs:string" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Original file format of appendix</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute AppendixType / @path

Namespace	No namespace
Annotations	Name and path to the file in the form: file:///path/to/file
Type	xs:string
Properties	use: required
Used by	Complex Type AppendixType

Source	`<xs:attribute name="path" type="xs:string" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Name and path to the file in the form: file:///path/to/file</`
	`xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute AppendixType / @eSignatureHaveExisted

Namespace	No namespace

Annotations	Marker for the appendix having had an electronic signature
Type	xs:boolean
Properties	use: optional
Used by	Complex Type AppendixType

Source	`<xs:attribute name="eSignatureHaveExisted" type="xs:boolean" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Marker for the appendix having had an electronic signature</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute contactInformationEntryType / @entryType

Namespace	No namespace
Annotations	State the contact information entry type according to vcADRESSLINE_LOCALTYPE.
Type	entryTypeType
Properties	use: required
Facets	enumeration postalAddress
	enumeration postalCode
	enumeration postalCity
	enumeration postalBox
	enumeration municipalityCode
	enumeration municipality
	enumeration parish
	enumeration parishCode
	enumeration province
	enumeration county
	enumeration country
	enumeration phoneNumber
	enumeration fax
	enumeration email
	enumeration homepage
Used by	Complex Type contactInformationEntryType

Source	`<xs:attribute name="entryType" type="entryTypeType" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the contact information entry type according to vcADRESSLINE_LOCALTYPE.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute nextOfKinEntryType / @typeOfRelation

Namespace	No namespace
Annotations	State the type of relation to the next of kin according to vcFAMILYRELATIONSHIP, or use "Other" if not applicable. Example "child"
Type	typeOfRelationType
Properties	use: required
Facets	enumeration father
	enumeration mother
	enumeration child
	enumeration guardian
	enumeration partner
	enumeration Other
Used by	Complex Type nextOfKinEntryType

Source	`<xs:attribute name="typeOfRelation" type="typeOfRelationType" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the type of relation to the next of kin according to vcFAMILYRELATIONSHIP, or use "Other" if not applicable. Example "child"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `nextOfKinEntryType / @otherTypeOfRelation`

Namespace	No namespace
Annotations	`If "typeOfRelation" is "Other", state applicable type of relation.`
Type	xs:string
Properties	use: `optional`
Used by	Complex Type nextOfKinEntryType

Source	`<xs:attribute name="otherTypeOfRelation" type="xs:string" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">If "typeOfRelation" is "Other", state applicable type of relation.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `taxTableType / @table`

Namespace	No namespace
Annotations	`State which tax table the employee belongs to. Example: "32"`
Type	xs:integer
Properties	use: `required`
Used by	Complex Type taxTableType

Source	`<xs:attribute name="table" type="xs:integer" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State which tax table the employee belongs to. Example: "32"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `taxTableType / @column`

Namespace	No namespace
Annotations	`State which column the tax table belongs to. Example: "1"`
Type	xs:integer
Properties	use: `required`
Used by	Complex Type taxTableType

Source	`<xs:attribute name="column" type="xs:integer" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State which column the tax table belongs to. Example: "1"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `optionalTaxEntryType / @typeOfTax`

Namespace	No namespace
Annotations	`State the optional type of tax according to vcTYPEOFTAX, or use "Other" if not applicable. Example: "SINK"`
Type	typeOfTaxType
Properties	use: `required`
Facets	enumeration `AdditionalTaxPayment` enumeration `SINK` enumeration `Other`

Used by	Complex Type optionalTaxEntryType

Source	`<xs:attribute name="typeOfTax" type="typeOfTaxType" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the optional type of tax according to vcTYPEOFTAX, or use "Other" if not applicable. Example: "SINK"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `optionalTaxEntryType / @otherTypeOfTax`

Namespace	No namespace
Annotations	`If "typeOfTax" is "Other", state applicable type of tax.`
Type	xs:string
Properties	use: `optional`
Used by	Complex Type optionalTaxEntryType

Source	`<xs:attribute name="otherTypeOfTax" type="xs:string" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">If "typeOfTax" is "Other", state applicable type of tax.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `childType / @firstName`

Namespace	No namespace
Annotations	`State the child's first name.`
Type	xs:string
Properties	use: `required`
Used by	Complex Type childType

Source	`<xs:attribute name="firstName" type="xs:string" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the child's first name.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `childType / @surname`

Namespace	No namespace
Annotations	`State the child's surname.`
Type	xs:string
Properties	use: `required`
Used by	Complex Type childType

Source	`<xs:attribute name="surname" type="xs:string" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the child's surname.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `childType / @birthDate`

Namespace	No namespace
Annotations	`State the child's birth date.`
Type	xs:date
Properties	use: `required`
Used by	Complex Type childType

Source	`<xs:attribute name="birthDate" type="xs:date" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the child's birth date.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute reassignmentType / @href

Namespace	No namespace
Annotations	State an URI to a document describing the reassignment.
Type	xs:anyURI
Properties	use: optional
Used by	Complex Type reassignmentType

Source	`<xs:attribute name="href" type="xs:anyURI" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State an URI to a document describing the reassignment.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute disciplinaryActionType / @href

Namespace	No namespace
Annotations	State an URI to a document describing the disciplinary action.
Type	xs:anyURI
Properties	use: optional
Used by	Complex Type disciplinaryActionType

Source	`<xs:attribute name="href" type="xs:anyURI" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State an URI to a document describing the disciplinary action.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute workRelatedInjuryType / @href

Namespace	No namespace
Annotations	State an URI to a document related to the work related inury.
Type	xs:anyURI
Properties	use: optional
Used by	Complex Type workRelatedInjuryType

Source	`<xs:attribute name="href" type="xs:anyURI" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State an URI to a document related to the work related inury.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute rehabilitationArrangementType / @href

Namespace	No namespace
Annotations	State an URI to a document related to the rehabilitation arrangement.
Type	xs:anyURI
Properties	use: optional
Used by	Complex Type rehabilitationArrangementType

Source	`<xs:attribute name="href" type="xs:anyURI" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State an URI to a document related to the rehabilitation arrangement.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute workRelatedAidGrantType / @href

Namespace	No namespace
Annotations	State an URI to a document related to the work related grant.

Type	xs:anyURI
Properties	use: optional
Used by	Complex Type workRelatedAidGrantType

Source	`<xs:attribute name="href" type="xs:anyURI" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State an URI to a document related to the work related grant.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute deathRelatedDocumentType / @href

Namespace	No namespace
Annotations	State an URI to the death related document.
Type	xs:anyURI
Properties	use: optional
Used by	Complex Type deathRelatedDocumentType

Source	`<xs:attribute name="href" type="xs:anyURI" use="optional">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State an URI to the death related document.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute relationToParentType / relationEntry / @parentOrganisationReference-Code

Namespace	No namespace
Annotations	State the parent organisation's reference code.
Type	xs:string
Properties	use: required
Used by	Element relationToParentType/relationEntry

Source	`<xs:attribute name="parentOrganisationReferenceCode" type="xs:string" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the parent organisation's reference code.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute relationToParentType / @typeOfRelation

Namespace	No namespace
Annotations	Describes the relation to the related XML-document. Fixed value: "hierarchical-child"
Properties	use: required fixed: hierarchical-child
Used by	Complex Type relationToParentType

Source	`<xs:attribute name="typeOfRelation" use="required" fixed="hierarchical-child">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describes the relation to the related XML-document. Fixed value: "hierarchical-child"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute relationToParentType / @href

Namespace	No namespace
Annotations	State the URI to the parental XML-document.
Type	xs:anyURI
Properties	use: optional
Used by	Complex Type relationToParentType

Source	`<xs:attribute name="href" use="optional" type="xs:anyURI">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the URI to the parental XML-document.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `relationToChildType / relationEntry / @childOrganisationReferenceCode`

Namespace	No namespace
Annotations	`State the child document's reference code.`
Type	xs:string
Properties	use: `required`
Used by	Element relationToChildType/relationEntry

Source	`<xs:attribute name="childOrganisationReferenceCode" type="xs:string" use="required">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the child document's reference code.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `relationToChildType / @typeOfRelation`

Namespace	No namespace
Annotations	`Describes the relation to the related XML-document. Fixed value: "hierarchical-parent"`
Properties	use: `required` fixed: `hierarchical-parent`
Used by	Complex Type relationToChildType

Source	`<xs:attribute name="typeOfRelation" use="required" fixed="hierarchical-parent">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">Describes the relation to the related XML-document. Fixed value:`
	` "hierarchical-parent"</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`

Attribute `relationToChildType / @href`

Namespace	No namespace
Annotations	`State the URI to the child XML-document.`
Type	xs:anyURI
Properties	use: `optional`
Used by	Complex Type relationToChildType

Source	`<xs:attribute name="href" use="optional" type="xs:anyURI">`
	`  <xs:annotation>`
	`  <xs:documentation xml:lang="en">State the URI to the child XML-document.</xs:documentation>`
	`  </xs:annotation>`
	`</xs:attribute>`