

# Schema documentation for archimate3\_Diagram.xsd

## Table of Contents

- Namespace: "http://www.opengroup.org/xsd/archimate/3.0/" ..... 1
- Schema(s) ..... 1
  - Main schema archimate3\_Diagram.xsd ..... 1
- Complex Type(s) ..... 1
  - Redefines Complex Type ViewsType ..... 1
  - Complex Type DiagramsType ..... 2
  - Complex Type Diagram ..... 3
  - Complex Type ViewNodeType ..... 3
  - Complex Type ViewConceptType ..... 4
  - Complex Type StyleType ..... 4
  - Complex Type RGBColorType ..... 5
  - Complex Type FontType ..... 5
  - Complex Type ConnectionType ..... 5
  - Complex Type LocationType ..... 6
  - Complex Type Label ..... 6
  - Complex Type Container ..... 7
  - Complex Type Element ..... 8
  - Complex Type Line ..... 9
  - Complex Type SourcedConnectionType ..... 9
  - Complex Type Relationship ..... 10
  - Complex Type NestingRelationship ..... 11
- Simple Type(s) ..... 11
  - Simple Type RGBValueType ..... 11
  - Simple Type nonNegativeHalfGranularityDecimal ..... 11
  - Simple Type nonNegativeDecimal ..... 12
  - Simple Type FontStyleType ..... 12
  - Simple Type XPATH\_2.0\_Expression ..... 12
  - Simple Type FontStyleEnum ..... 12
- Element Group(s) ..... 13
  - Element Group NodeRefGroup ..... 13
  - Element Group ConnectionRefGroup ..... 13
  - Element Group ViewConceptRefGroup ..... 13
- Attribute Group(s) ..... 13
  - Attribute Group LocationGroup ..... 13
  - Attribute Group SizeGroup ..... 14
  - Attribute Group NodeRefAttributeGroup ..... 14
  - Attribute Group ConnectionRefAttributeGroup ..... 14
  - Attribute Group ViewConceptRefAttributeGroup ..... 14

## Namespace: "http://www.opengroup.org/xsd/archimate/3.0/"

### Schema(s)

#### Main schema archimate3\_Diagram.xsd

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
Properties	attribute form default: unqualified
	element form default: qualified
	version: 3.1

### Complex Type(s)

#### Redefines Complex Type ViewsType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
-----------	---

Type	extension of ViewsType
Type hierarchy	<ul style="list-style-type: none"> <li>ViewsType</li> <li>ViewsType</li> </ul>
Model	viewpoints{0,1} , diagrams{0,1}
Children	diagrams, viewpoints
Source	<pre>&lt;xs:complexType name="ViewsType"&gt;   &lt;xs:complexContent&gt;     &lt;xs:extension base="ViewsType"&gt;       &lt;xs:sequence&gt;         &lt;xs:element name="diagrams" type="DiagramsType" minOccurs="0" maxOccurs="1"/&gt;       &lt;/xs:sequence&gt;     &lt;/xs:extension&gt;   &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type DiagramsType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
Used by	Element ViewsType/diagrams
Model	view+
Children	view
Source	<pre>&lt;xs:complexType name="DiagramsType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is a container for all of the Diagrams in the model.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="view" type="Diagram" minOccurs="1" maxOccurs="unbounded"&gt;       &lt;xs:key name="NodeKey"&gt;         &lt;xs:selector xpath="//archimate:node"/&gt;         &lt;xs:field xpath="@identifier"/&gt;       &lt;/xs:key&gt;       &lt;xs:key name="ConnectionKey"&gt;         &lt;xs:selector xpath="//archimate:connection"/&gt;         &lt;xs:field xpath="@identifier"/&gt;       &lt;/xs:key&gt;       &lt;xs:keyref name="NodeRef" refer="NodeKey"&gt;         &lt;xs:selector xpath="//archimate:nodeRef"/&gt;         &lt;xs:field xpath="@ref"/&gt;       &lt;/xs:keyref&gt;       &lt;xs:keyref name="NodeRefAttribute" refer="NodeKey"&gt;         &lt;xs:selector xpath="//archimate:*/&gt;         &lt;xs:field xpath="@nodeRef"/&gt;       &lt;/xs:keyref&gt;       &lt;xs:keyref name="ConnectionRef" refer="ConnectionKey"&gt;         &lt;xs:selector xpath="//archimate:connectionRef"/&gt;         &lt;xs:field xpath="@ref"/&gt;       &lt;/xs:keyref&gt;       &lt;xs:keyref name="ConnectionRefAttribute" refer="ConnectionKey"&gt;         &lt;xs:selector xpath="//archimate:*/&gt;         &lt;xs:field xpath="@connectionRef"/&gt;       &lt;/xs:keyref&gt;       &lt;!-- ViewConcept --&gt;       &lt;xs:key name="ViewConceptKey"&gt;         &lt;xs:selector xpath="//archimate:node   //archimate:connection"/&gt;         &lt;xs:field xpath="@identifier"/&gt;       &lt;/xs:key&gt;       &lt;xs:keyref name="ViewConceptRef" refer="ViewConceptKey"&gt;         &lt;xs:selector xpath="//archimate:viewConceptRef"/&gt;         &lt;xs:field xpath="@ref"/&gt;       &lt;/xs:keyref&gt;       &lt;xs:keyref name="ViewConceptRefAttribute" refer="ViewConceptKey"&gt;         &lt;xs:selector xpath="//archimate:*/&gt;         &lt;xs:field xpath="@viewConceptRef"/&gt;       &lt;/xs:keyref&gt;       &lt;xs:keyref name="ConnectionSourceRef" refer="ViewConceptKey"&gt;         &lt;xs:selector xpath="//archimate:connection"/&gt;         &lt;xs:field xpath="@source"/&gt;       &lt;/xs:keyref&gt;       &lt;xs:keyref name="ConnectionTargetRef" refer="ViewConceptKey"&gt;         &lt;xs:selector xpath="//archimate:connection"/&gt;         &lt;xs:field xpath="@target"/&gt;       &lt;/xs:keyref&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt; &lt;/xs:complexType&gt;</pre>

## Complex Type Diagram

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Type	extension of ViewType		
Type hierarchy	<ul style="list-style-type: none"> <li>• ReferenceableType <ul style="list-style-type: none"> <li>• NamedReferenceableType <ul style="list-style-type: none"> <li>• ViewType <ul style="list-style-type: none"> <li>• Diagram</li> </ul> </li> </ul> </li> </ul> </li> </ul>		
Used by	Element	DiagramsType/view	
Model	name+ , documentation* , ANY element from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/' , properties{0,1} , node* , connection*		
Children	connection, documentation, name, node, properties		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>identifier</b>	xs:ID	required
	<b>viewpoint</b>	ViewpointTypeType	optional
	<b>viewpointRef</b>	xs:IDREF	optional
	<b>Wildcard:</b> ANY attribute from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/'		
Source	<pre>&lt;xs:complexType name="Diagram"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is a container for all of the Nodes and Connections in the Diagrams.&lt;/   &lt;/xs:annotation&gt;   &lt;xs:complexTypeContent&gt;     &lt;xs:extension base="ViewType"&gt;       &lt;xs:sequence&gt;         &lt;xs:element name="node" type="ViewNodeType" minOccurs="0" maxOccurs="unbounded"/&gt;         &lt;xs:element name="connection" type="ConnectionType" minOccurs="0" maxOccurs="unbounded"/&gt;       &lt;/xs:sequence&gt;     &lt;/xs:extension&gt;   &lt;/xs:complexTypeContent&gt; &lt;/xs:complexType&gt;</pre>		

## Complex Type ViewNodeType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Type	extension of ViewConceptType		
Type hierarchy	<ul style="list-style-type: none"> <li>• ViewConceptType <ul style="list-style-type: none"> <li>• ViewNodeType</li> </ul> </li> </ul>		
Properties	abstract:	true	
Used by	Elements	Container/node, Diagram/node	
	Complex Types	Container, Label	
Model	label+ , documentation* , style{0,1} , viewRef*		
Children	documentation, label, style, viewRef		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>h</b>	xs:positiveInteger	required
		The height (associated with y) attribute from the top side to the bottom side of the bounding box of a concept.	
	<b>identifier</b>	xs:ID	required
	<b>w</b>	xs:positiveInteger	required
		The width (associated with x) attribute from the Left side to the right side of the bounding box of a concept.	
	<b>x</b>	xs:nonNegativeInteger	required
		The x (towards the right, associated with width) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.	
<b>y</b>	xs:nonNegativeInteger	required	

	QName	Type	Use
		The y (towards the bottom, associated with height) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.	
	<b>Wildcard:</b> ANY attribute from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/'		
Source	<pre>&lt;xs:complexType name="ViewNodeType" abstract="true"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Graphical node type. It can contain child node types.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexContent&gt;     &lt;xs:extension base="ViewConceptType"&gt;       &lt;xs:attributeGroup ref="LocationGroup"/&gt;       &lt;xs:attributeGroup ref="SizeGroup"/&gt;       &lt;xs:anyAttribute namespace="##other" processContents="strict"/&gt;     &lt;/xs:extension&gt;   &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt;</pre>		

### Complex Type ViewConceptType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Properties	abstract:	true	
Used by	Complex Types	ConnectionType, ViewNodeType	
Model	label+ , documentation* , style{0,1} , viewRef*		
Children	documentation, label, style, viewRef		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>identifier</b>	xs:ID	required
	<b>Wildcard:</b> ANY attribute from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/'		
Source	<pre>&lt;xs:complexType name="ViewConceptType" abstract="true"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;This is an abstract super-type of Node and Connection.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:group ref="LabelGroup" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;xs:group ref="DocumentationGroup" minOccurs="0" maxOccurs="unbounded"/&gt;     &lt;xs:element name="style" type="StyleType" minOccurs="0" maxOccurs="1"/&gt;     &lt;xs:element name="viewRef" type="ReferenceType" minOccurs="0" maxOccurs="unbounded"&gt;       &lt;xs:annotation&gt;         &lt;xs:documentation&gt;the "viewRef" of an "Concept" is to a view that allows drill-down diagrams.&lt;/xs:documentation&gt;       &lt;/xs:annotation&gt;     &lt;/xs:element&gt;   &lt;/xs:sequence&gt;   &lt;xs:attributeGroup ref="IdentifierGroup"/&gt;   &lt;xs:anyAttribute namespace="##other" processContents="strict"/&gt; &lt;/xs:complexType&gt;</pre>		

### Complex Type styleType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Used by	Elements	SourcedConnectionType/style, ViewConceptType/style	
Model	ALL(lineColor{0,1} fillColor{0,1} font{0,1})		
Children	fillColor, font, lineColor		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>lineWidth</b>	xs:positiveInteger	optional
		An integer value representing the width (thickness) of the lines of each node in pixels.	
Source	<pre>&lt;xs:complexType name="StyleType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A Style type for a node and a connection. It is a container for style properties such as color, font.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:all&gt;     &lt;xs:element name="lineColor" type="RGBColorType" minOccurs="0" maxOccurs="1"/&gt;     &lt;xs:element name="fillColor" type="RGBColorType" minOccurs="0" maxOccurs="1"/&gt;   &lt;/xs:all&gt; &lt;/xs:complexType&gt;</pre>		

	<pre> &lt;xs:element name="font" type="FontType" minOccurs="0" maxOccurs="1"/&gt; &lt;/xs:all&gt; &lt;xs:attribute name="lineWidth" type="xs:positiveInteger" use="optional"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;An integer value representing the width (thickness) of the lines of each node in pixels.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt; &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>
--	--

### Complex Type RGBColorType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/			
Used by	Elements	FontType/color, StyleType/fillColor, StyleType/lineColor		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>	
	<b>a</b>	restriction of xs:unsignedByte	optional	
	<b>b</b>	RGBValueType	required	
	<b>g</b>	RGBValueType	required	
	<b>r</b>	RGBValueType	required	
Source	<pre> &lt;xs:complexType name="RGBColorType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;RGB Color type. The r, g, b attributes range from 0 - 255. The a (alpha) transparency attribute is optional. 0 = full transparency, 100 = opaque.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:attribute name="r" type="RGBValueType" use="required"/&gt;   &lt;xs:attribute name="g" type="RGBValueType" use="required"/&gt;   &lt;xs:attribute name="b" type="RGBValueType" use="required"/&gt;   &lt;xs:attribute name="a" use="optional"&gt;     &lt;xs:simpleType&gt;       &lt;xs:restriction base="xs:unsignedByte"&gt;         &lt;xs:minInclusive value="0"/&gt;         &lt;xs:maxInclusive value="100"/&gt;       &lt;/xs:restriction&gt;     &lt;/xs:simpleType&gt;   &lt;/xs:attribute&gt; &lt;/xs:complexType&gt; </pre>			

### Complex Type FontType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/			
Used by	Element	StyleType/font		
Model	color{0,1}			
Children	color			
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>	
	<b>name</b>	xs:string	optional	
	<b>size</b>	nonNegativeHalfGranularityDecimal	optional	
		Size is given in points.		
	<b>style</b>	FontStyleType	optional	
Source	<pre> &lt;xs:complexType name="FontType"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Font type.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="color" type="RGBColorType" minOccurs="0" maxOccurs="1"/&gt;   &lt;/xs:sequence&gt;   &lt;xs:attribute name="name" type="xs:string" use="optional"/&gt;   &lt;xs:attribute name="size" type="nonNegativeHalfGranularityDecimal" use="optional"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;Size is given in points.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt;   &lt;xs:attribute name="style" type="FontStyleType" use="optional"/&gt; &lt;/xs:complexType&gt; </pre>			

### Complex Type ConnectionType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/			
-----------	---	--	--	--

Type	extension of ViewConceptType		
Type hierarchy	<ul style="list-style-type: none"> <li>ViewConceptType <ul style="list-style-type: none"> <li>ConnectionType</li> </ul> </li> </ul>		
Properties	abstract:	true	
Used by	Element	Diagram/connection	
	Complex Types	Line, SourcedConnectionType	
Model	label+ , documentation* , style{0,1} , viewRef* , sourceAttachment{0,1} , bendpoint* , targetAttachment{0,1}		
Children	bendpoint, documentation, label, sourceAttachment, style, targetAttachment, viewRef		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>identifier</b>	xs:ID	required
	<b>source</b>	xs:IDREF	optional
	<b>target</b>	xs:IDREF	optional
	<b>Wildcard:</b> ANY attribute from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/'		
Source	<pre>&lt;xs:complexType name="ConnectionType" abstract="true"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Graphical connection type. If the 'relationshipRef' attribute is present, the connection should reference an existing ArchiMate relationship. If the connection is an ArchiMate relationship type, the connection's label, documentation and properties may be determined (i.e inherited) from those in the referenced ArchiMate relationship. Otherwise the connection's label, documentation and properties can be provided and will be additional to (or over-ride) those contained in the referenced ArchiMate relationship.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexContent&gt;     &lt;xs:extension base="ViewConceptType"&gt;       &lt;xs:sequence&gt;         &lt;xs:element name="sourceAttachment" type="LocationType" minOccurs="0" maxOccurs="1"/&gt;         &lt;xs:element name="bendpoint" type="LocationType" minOccurs="0" maxOccurs="unbounded"/&gt;         &lt;xs:element name="targetAttachment" type="LocationType" minOccurs="0" maxOccurs="1"/&gt;       &lt;/xs:sequence&gt;       &lt;xs:attribute name="source" type="xs:IDREF" use="optional"/&gt;       &lt;xs:attribute name="target" type="xs:IDREF" use="optional"/&gt;     &lt;/xs:extension&gt;   &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt;</pre>		

## Complex Type LocationType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Used by	Elements	ConnectionType/bendpoint, ConnectionType/sourceAttachment, ConnectionType/targetAttachment, SourcedConnectionType/bendpoint, SourcedConnectionType/sourceAttachment, SourcedConnectionType/targetAttachment	
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>x</b>	xs:nonNegativeInteger	required
		The x (towards the right, associated with width) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.	
	<b>y</b>	xs:nonNegativeInteger	required
	The y (towards the bottom, associated with height) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.		
Source	<pre>&lt;xs:complexType name="LocationType"&gt;   &lt;xs:attributeGroup ref="LocationGroup"/&gt; &lt;/xs:complexType&gt;</pre>		

## Complex Type Label

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Type	extension of ViewNodeType		
Type hierarchy	<ul style="list-style-type: none"> <li>ViewConceptType <ul style="list-style-type: none"> <li>ViewNodeType</li> </ul> </li> </ul>		

	<ul style="list-style-type: none"> <li>• Label</li> </ul>																																																								
Model	label+ , documentation* , style{0,1} , viewRef*																																																								
Children	documentation, label, style, viewRef																																																								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td>conceptRef</td> <td>xs:IDREF</td> <td>optional</td> <td></td> </tr> <tr> <td></td> <td colspan="3">conceptRef is a reference to an concept for this particular label, along with the attributeRef which references the particular concept's part which this label represents.</td> </tr> <tr> <td>h</td> <td>xs:positiveInteger</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">The height (associated with y) attribute from the top side to the bottom side of the bounding box of a concept.</td> </tr> <tr> <td>identifier</td> <td>xs:ID</td> <td>required</td> <td></td> </tr> <tr> <td>w</td> <td>xs:positiveInteger</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">The width (associated with x) attribute from the Left side to the right side of the bounding box of a concept.</td> </tr> <tr> <td>x</td> <td>xs:nonNegativeInteger</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">The x (towards the right, associated with width) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.</td> </tr> <tr> <td>xpathPart</td> <td>XPATH_2.0_Expression</td> <td>optional</td> <td></td> </tr> <tr> <td></td> <td colspan="3">conceptRef is a reference to an concept for this particular label, along with the partRef which references the particular concept's part which this label represents. If this attribute is set, then there is no need to add a label tag in the Label parent (since it is contained in the model). the XPATH statement is meant to be interpreted in the context of what the conceptRef points to.</td> </tr> <tr> <td>y</td> <td>xs:nonNegativeInteger</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">The y (towards the bottom, associated with height) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.</td> </tr> </tbody> </table>	QName	Type	Use		conceptRef	xs:IDREF	optional			conceptRef is a reference to an concept for this particular label, along with the attributeRef which references the particular concept's part which this label represents.			h	xs:positiveInteger	required			The height (associated with y) attribute from the top side to the bottom side of the bounding box of a concept.			identifier	xs:ID	required		w	xs:positiveInteger	required			The width (associated with x) attribute from the Left side to the right side of the bounding box of a concept.			x	xs:nonNegativeInteger	required			The x (towards the right, associated with width) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.			xpathPart	XPATH_2.0_Expression	optional			conceptRef is a reference to an concept for this particular label, along with the partRef which references the particular concept's part which this label represents. If this attribute is set, then there is no need to add a label tag in the Label parent (since it is contained in the model). the XPATH statement is meant to be interpreted in the context of what the conceptRef points to.			y	xs:nonNegativeInteger	required			The y (towards the bottom, associated with height) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.		
	QName	Type	Use																																																						
	conceptRef	xs:IDREF	optional																																																						
		conceptRef is a reference to an concept for this particular label, along with the attributeRef which references the particular concept's part which this label represents.																																																							
	h	xs:positiveInteger	required																																																						
		The height (associated with y) attribute from the top side to the bottom side of the bounding box of a concept.																																																							
	identifier	xs:ID	required																																																						
	w	xs:positiveInteger	required																																																						
		The width (associated with x) attribute from the Left side to the right side of the bounding box of a concept.																																																							
	x	xs:nonNegativeInteger	required																																																						
		The x (towards the right, associated with width) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.																																																							
	xpathPart	XPATH_2.0_Expression	optional																																																						
		conceptRef is a reference to an concept for this particular label, along with the partRef which references the particular concept's part which this label represents. If this attribute is set, then there is no need to add a label tag in the Label parent (since it is contained in the model). the XPATH statement is meant to be interpreted in the context of what the conceptRef points to.																																																							
y	xs:nonNegativeInteger	required																																																							
	The y (towards the bottom, associated with height) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.																																																								
	<b>Wildcard:</b> ANY attribute from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/'																																																								
Source	<pre> &lt;xs:complexType name="Label"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Node type to allow a Label in a Artifact. the "label" element holds the info for the Note.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexContent&gt;     &lt;xs:extension base="ViewNodeType"&gt;       &lt;xs:attribute name="conceptRef" type="xs:IDREF" use="optional"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;conceptRef is a reference to an concept for this particular label, along with the attributeRef which references the particular concept's part which this label represents.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:attribute&gt;       &lt;xs:attribute name="xpathPart" type="XPATH_2.0_Expression" use="optional"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;conceptRef is a reference to an concept for this particular label, along with the partRef which references the particular concept's part which this label represents. If this attribute is set, then there is no need to add a label tag in the Label parent (since it is contained in the model). the XPATH statement is meant to be interpreted in the context of what the conceptRef points to.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:attribute&gt;     &lt;/xs:extension&gt;   &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt; </pre>																																																								

## Complex Type Container

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
Type	extension of ViewNodeType
Type hierarchy	<ul style="list-style-type: none"> <li>• ViewConceptType</li> </ul>

	<ul style="list-style-type: none"> <li>• ViewNodeType</li> <li>• Container</li> </ul>																																								
Used by	Complex Type      Element																																								
Model	label+ , documentation* , style{0,1} , viewRef* , node*																																								
Children	documentation, label, node, style, viewRef																																								
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td><b>h</b></td> <td>xs:positiveInteger</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">The height (associated with y) attribute from the top side to the bottom side of the bounding box of a concept.</td> </tr> <tr> <td><b>identifier</b></td> <td>xs:ID</td> <td>required</td> <td></td> </tr> <tr> <td><b>w</b></td> <td>xs:positiveInteger</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">The width (associated with x) attribute from the Left side to the right side of the bounding box of a concept.</td> </tr> <tr> <td><b>x</b></td> <td>xs:nonNegativeInteger</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">The x (towards the right, associated with width) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.</td> </tr> <tr> <td><b>y</b></td> <td>xs:nonNegativeInteger</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">The y (towards the bottom, associated with height) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.</td> </tr> </tbody> </table>	QName	Type	Use		<b>h</b>	xs:positiveInteger	required			The height (associated with y) attribute from the top side to the bottom side of the bounding box of a concept.			<b>identifier</b>	xs:ID	required		<b>w</b>	xs:positiveInteger	required			The width (associated with x) attribute from the Left side to the right side of the bounding box of a concept.			<b>x</b>	xs:nonNegativeInteger	required			The x (towards the right, associated with width) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.			<b>y</b>	xs:nonNegativeInteger	required			The y (towards the bottom, associated with height) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.		
	QName	Type	Use																																						
	<b>h</b>	xs:positiveInteger	required																																						
		The height (associated with y) attribute from the top side to the bottom side of the bounding box of a concept.																																							
	<b>identifier</b>	xs:ID	required																																						
	<b>w</b>	xs:positiveInteger	required																																						
		The width (associated with x) attribute from the Left side to the right side of the bounding box of a concept.																																							
	<b>x</b>	xs:nonNegativeInteger	required																																						
		The x (towards the right, associated with width) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.																																							
<b>y</b>	xs:nonNegativeInteger	required																																							
	The y (towards the bottom, associated with height) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.																																								
	<b>Wildcard:</b> ANY attribute from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/'																																								
Source	<pre>&lt;xs:complexType name="Container"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Node type to allow a Container in a Artifact. This is a visual grouping     container.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexContent&gt;     &lt;xs:extension base="ViewNodeType"&gt;       &lt;xs:sequence&gt;         &lt;xs:element name="node" type="ViewNodeType" minOccurs="0" maxOccurs="unbounded"&gt;           &lt;xs:annotation&gt;             &lt;xs:documentation&gt;This is to support Nested Nodes on the Diagram The order of sibling             nodes in their parent View or Node container as declared in the model instance dictates the z-order             of the nodes. Given nodes A, B, and C as declared in that order, node B is considered to be in             front of node A, node C is considered to be in front of node B, and node C is considered to be in             front of nodes A and B.&lt;/xs:documentation&gt;           &lt;/xs:annotation&gt;         &lt;/xs:element&gt;       &lt;/xs:sequence&gt;     &lt;/xs:extension&gt;   &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt;</pre>																																								

## Complex Type Element

Namespace	http://www.opengroup.org/xsd/archimate/3.0/																
Type	extension of Container																
Type hierarchy	<ul style="list-style-type: none"> <li>• ViewConceptType</li> <li>• ViewNodeType</li> <li>• Container</li> <li>• Element</li> </ul>																
Model	label+ , documentation* , style{0,1} , viewRef* , node*																
Children	documentation, label, node, style, viewRef																
Attributes	<table border="1"> <thead> <tr> <th>QName</th> <th>Type</th> <th>Use</th> <th></th> </tr> </thead> <tbody> <tr> <td><b>elementRef</b></td> <td>xs:IDREF</td> <td>required</td> <td></td> </tr> <tr> <td></td> <td colspan="3">If the 'elementRef' restrictions will only allow references to an existing ArchiMate element.</td> </tr> <tr> <td><b>h</b></td> <td>xs:positiveInteger</td> <td>required</td> <td></td> </tr> </tbody> </table>	QName	Type	Use		<b>elementRef</b>	xs:IDREF	required			If the 'elementRef' restrictions will only allow references to an existing ArchiMate element.			<b>h</b>	xs:positiveInteger	required	
	QName	Type	Use														
	<b>elementRef</b>	xs:IDREF	required														
	If the 'elementRef' restrictions will only allow references to an existing ArchiMate element.																
<b>h</b>	xs:positiveInteger	required															



QName	Type	Use
	The height (associated with y) attribute from the top side to the bottom side of the bounding box of a concept.	
<b>identifier</b>	xs:ID	required
<b>w</b>	xs:positiveInteger	required
	The width (associated with x) attribute from the Left side to the right side of the bounding box of a concept.	
<b>x</b>	xs:nonNegativeInteger	required
	The x (towards the right, associated with width) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.	
<b>y</b>	xs:nonNegativeInteger	required
	The y (towards the bottom, associated with height) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.	
<b>Wildcard:</b> ANY attribute from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/'		
Source	<pre>&lt;xs:complexType name="Element"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Node type to allow an Element in a Artifact. The node's label, documentation and properties may be determined (i.e inherited) from those in the referenced ArchiMate element. Otherwise the node's label, documentation and properties can be provided and will be additional to (or over-ride) those contained in the referenced ArchiMate element.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexContent&gt;     &lt;xs:extension base="Container"&gt;       &lt;xs:attribute name="elementRef" type="xs:IDREF" use="required"&gt;         &lt;xs:annotation&gt;           &lt;xs:documentation&gt;If the 'elementRef' restrictions will only allow references to an existing ArchiMate element.&lt;/xs:documentation&gt;         &lt;/xs:annotation&gt;       &lt;/xs:attribute&gt;     &lt;/xs:extension&gt;   &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt;</pre>	

## Complex Type Line

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Type	extension of ConnectionType		
Type hierarchy	<ul style="list-style-type: none"> <li>• ViewConceptType           <ul style="list-style-type: none"> <li>• ConnectionType               <ul style="list-style-type: none"> <li>• Line</li> </ul> </li> </ul> </li> </ul>		
Model	label+ , documentation* , style{0,1} , viewRef* , sourceAttachment{0,1} , bendpoint* , targetAttachment{0,1}		
Children	bendpoint, documentation, label, sourceAttachment, style, targetAttachment, viewRef		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>identifier</b>	xs:ID	required
	<b>source</b>	xs:IDREF	optional
	<b>target</b>	xs:IDREF	optional
<b>Wildcard:</b> ANY attribute from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/'			
Source	<pre>&lt;xs:complexType name="Line"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Line on the diagram.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexContent&gt;     &lt;xs:extension base="ConnectionType" /&gt;   &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt;</pre>		

## Complex Type SourcedConnectionType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
Type	restriction of ConnectionType

Type hierarchy	<ul style="list-style-type: none"> <li>ViewConceptType</li> <li>ConnectionType <ul style="list-style-type: none"> <li>SourcedConnectionType</li> </ul> </li> </ul>		
Properties	abstract:	true	
Used by	Complex Type	Relationship	
Model	label+ , documentation* , style{0,1} , viewRef* , sourceAttachment{0,1} , bendpoint* , targetAttachment{0,1}		
Children	bendpoint, documentation, label, sourceAttachment, style, targetAttachment, viewRef		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	identifier	xs:ID	required
	source	xs:IDREF	required
	target	xs:IDREF	required
<b>Wildcard:</b> ANY attribute from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/'			
Source	<pre> &lt;xs:complexType name="SourcedConnectionType" abstract="true"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Abstract Type requiring a source and target.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexContent&gt;     &lt;xs:restriction base="ConnectionType"&gt;       &lt;xs:sequence&gt;         &lt;xs:group ref="LabelGroup" minOccurs="0" maxOccurs="unbounded"/&gt;         &lt;xs:group ref="DocumentationGroup" minOccurs="0" maxOccurs="unbounded"/&gt;         &lt;xs:element name="style" type="StyleType" minOccurs="0" maxOccurs="1"/&gt;         &lt;xs:element name="viewRef" type="ReferenceType" minOccurs="0" maxOccurs="unbounded"/&gt;         &lt;xs:element name="sourceAttachment" type="LocationType" minOccurs="0" maxOccurs="1"/&gt;         &lt;xs:element name="bendpoint" type="LocationType" minOccurs="0" maxOccurs="unbounded"/&gt;         &lt;xs:element name="targetAttachment" type="LocationType" minOccurs="0" maxOccurs="1"/&gt;       &lt;/xs:sequence&gt;       &lt;xs:attributeGroup ref="IdentifierGroup"/&gt;       &lt;xs:attribute name="source" type="xs:IDREF" use="required"/&gt;       &lt;xs:attribute name="target" type="xs:IDREF" use="required"/&gt;       &lt;xs:anyAttribute namespace="##other" processContents="strict"/&gt;     &lt;/xs:restriction&gt;   &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt; </pre>		

## Complex Type Relationship

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Type	extension of SourcedConnectionType		
Type hierarchy	<ul style="list-style-type: none"> <li>ViewConceptType</li> <li>ConnectionType <ul style="list-style-type: none"> <li>SourcedConnectionType <ul style="list-style-type: none"> <li>Relationship</li> </ul> </li> </ul> </li> </ul>		
Used by	Complex Type	NestingRelationship	
Model	label+ , documentation* , style{0,1} , viewRef* , sourceAttachment{0,1} , bendpoint* , targetAttachment{0,1}		
Children	bendpoint, documentation, label, sourceAttachment, style, targetAttachment, viewRef		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	identifier	xs:ID	required
	relationshipRef	xs:IDREF	required
	source	xs:IDREF	required
	target	xs:IDREF	required
<b>Wildcard:</b> ANY attribute from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/'			
Source	<pre> &lt;xs:complexType name="Relationship"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Connector for a Relationship.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexContent&gt; </pre>		

```
<xs:extension base="SourcedConnectionType">
  <xs:attribute name="relationshipRef" type="xs:IDREF" use="required" />
</xs:extension>
</xs:complexContent>
</xs:complexType>
```

## Complex Type NestingRelationship

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Type	extension of Relationship		
Type hierarchy	<ul style="list-style-type: none"> <li>• ViewConceptType</li> <li>• ConnectionType             <ul style="list-style-type: none"> <li>• SourcedConnectionType                 <ul style="list-style-type: none"> <li>• Relationship                     <ul style="list-style-type: none"> <li>• NestingRelationship</li> </ul> </li> </ul> </li> </ul> </li> </ul>		
Model	label+ , documentation* , style{0,1} , viewRef* , sourceAttachment{0,1} , endpoint* , targetAttachment{0,1}		
Children	endpoint, documentation, label, sourceAttachment, style, targetAttachment, viewRef		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>identifier</b>	xs:ID	required
	<b>relationshipRef</b>	xs:IDREF	required
	<b>source</b>	xs:IDREF	required
	<b>target</b>	xs:IDREF	required
	<b>Wildcard:</b> ANY attribute from ANY namespace OTHER than 'http://www.opengroup.org/xsd/archimate/3.0/'		
Source	<pre>&lt;xs:complexType name="NestingRelationship"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;Connector for a Nested Relationship on an element.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:complexContent&gt;     &lt;xs:extension base="Relationship" /&gt;   &lt;/xs:complexContent&gt; &lt;/xs:complexType&gt;</pre>		

## Simple Type(s)

### Simple Type RGBValueType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Type	restriction of xs:unsignedByte		
Facets	maxInclusive	255	
	minInclusive	0	
Used by	Attributes	RGBColorType/@b, RGBColorType/@g, RGBColorType/@r	
Source	<pre>&lt;xs:simpleType name="RGBValueType"&gt;   &lt;xs:restriction base="xs:unsignedByte"&gt;     &lt;xs:minInclusive value="0" /&gt;     &lt;xs:maxInclusive value="255" /&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>		

### Simple Type nonNegativeHalfGranularityDecimal

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Type	restriction of nonNegativeDecimal		
Type hierarchy	<ul style="list-style-type: none"> <li>• xs:decimal             <ul style="list-style-type: none"> <li>• nonNegativeDecimal                 <ul style="list-style-type: none"> <li>• nonNegativeHalfGranularityDecimal</li> </ul> </li> </ul> </li> </ul>		
Facets	minInclusive	0	
	pattern	[1-9][0-9]*(\.\d \.5)?	

Used by	Attribute FontType/@size
Source	<pre>&lt;xs:simpleType name="nonNegativeHalfGranularityDecimal"&gt;   &lt;xs:restriction base="nonNegativeDecimal"&gt;     &lt;xs:pattern value="[1-9][0-9]*(\.0 \.5)?"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>

### Simple Type nonNegativeDecimal

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
Type	restriction of xs:decimal
Facets	minInclusive 0
Used by	Simple Type nonNegativeHalfGranularityDecimal
Source	<pre>&lt;xs:simpleType name="nonNegativeDecimal"&gt;   &lt;xs:restriction base="xs:decimal"&gt;     &lt;xs:minInclusive value="0"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>

### Simple Type FontStyleType

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
Type	list of FontStyleEnum
Used by	Attribute FontType/@style
Source	<pre>&lt;xs:simpleType name="FontStyleType"&gt;   &lt;xs:list itemType="FontStyleEnum"/&gt; &lt;/xs:simpleType&gt;</pre>

### Simple Type XPATH\_2.0\_Expression

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
Type	restriction of xs:token
Facets	pattern .+
Used by	Attribute Label/@xpathPart
Source	<pre>&lt;xs:simpleType name="XPATH_2.0_Expression"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;An XPath 2.0 expression.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:token"&gt;     &lt;xs:pattern value=".+"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>

### Simple Type FontStyleEnum

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
Type	restriction of xs:NMTOKEN
Facets	enumeration plain enumeration bold enumeration italic enumeration underline
Source	<pre>&lt;xs:simpleType name="FontStyleEnum"&gt;   &lt;xs:annotation&gt;     &lt;xs:documentation&gt;A value describing the style of the font. Style values can be combined and shall be separated by a space. If no style values are declared, a default style is to be assumed. Declaring a style of plain implies that no styling (bold, italic, underline) shall be used by the receiver.&lt;/xs:documentation&gt;   &lt;/xs:annotation&gt;   &lt;xs:restriction base="xs:NMTOKEN"&gt;     &lt;xs:enumeration value="plain"/&gt;     &lt;xs:enumeration value="bold"/&gt;     &lt;xs:enumeration value="italic"/&gt;     &lt;xs:enumeration value="underline"/&gt;   &lt;/xs:restriction&gt; &lt;/xs:simpleType&gt;</pre>

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

## Element Group(s)

### Element Group NodeRefGroup

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
Model	nodeRef
Children	nodeRef
Source	<pre>&lt;xs:group name="NodeRefGroup"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="nodeRef" type="ReferenceType" minOccurs="1" maxOccurs="1"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:group&gt;</pre>

### Element Group ConnectionRefGroup

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
Model	connectionRef
Children	connectionRef
Source	<pre>&lt;xs:group name="ConnectionRefGroup"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="connectionRef" type="ReferenceType" minOccurs="1" maxOccurs="1"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:group&gt;</pre>

### Element Group ViewConceptRefGroup

Namespace	http://www.opengroup.org/xsd/archimate/3.0/
Model	viewConceptRef
Children	viewConceptRef
Source	<pre>&lt;xs:group name="ViewConceptRefGroup"&gt;   &lt;xs:sequence&gt;     &lt;xs:element name="viewConceptRef" type="ReferenceType" minOccurs="1" maxOccurs="1"/&gt;   &lt;/xs:sequence&gt; &lt;/xs:group&gt;</pre>

## Attribute Group(s)

### Attribute Group LocationGroup

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Used by	Complex Types      LocationType, ViewNodeType		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	x	xs:nonNegativeInteger	required
		The x (towards the right, associated with width) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.	
	y	xs:nonNegativeInteger	required
	The y (towards the bottom, associated with height) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.		
Source	<pre>&lt;xs:attributeGroup name="LocationGroup"&gt;   &lt;xs:attribute name="x" type="xs:nonNegativeInteger" use="required"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;The x (towards the right, associated with width) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt;   &lt;xs:attribute name="y" type="xs:nonNegativeInteger" use="required"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;The y (towards the bottom, associated with height) attribute from the Top,Left (i.e. 0,0) corner of the diagram to the Top, Left corner of the bounding box of the concept.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt; &lt;/xs:attributeGroup&gt;</pre>		

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

### Attribute Group SizeGroup

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Used by	Complex Type	ViewNodeType	
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>h</b>	xs:positiveInteger	required
		The height (associated with y) attribute from the top side to the bottom side of the bounding box of a concept.	
	<b>w</b>	xs:positiveInteger	required
	The width (associated with x) attribute from the Left side to the right side of the bounding box of a concept.		
Source	<pre>&lt;xs:attributeGroup name="SizeGroup"&gt;   &lt;xs:attribute name="w" type="xs:positiveInteger" use="required"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;The width (associated with x) attribute from the Left side to the right side of the bounding box of a concept.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt;   &lt;xs:attribute name="h" type="xs:positiveInteger" use="required"&gt;     &lt;xs:annotation&gt;       &lt;xs:documentation&gt;The height (associated with y) attribute from the top side to the bottom side of the bounding box of a concept.&lt;/xs:documentation&gt;     &lt;/xs:annotation&gt;   &lt;/xs:attribute&gt; &lt;/xs:attributeGroup&gt;</pre>		

### Attribute Group NodeRefAttributeGroup

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>nodeRef</b>	xs:IDREF	required
Source	<pre>&lt;xs:attributeGroup name="NodeRefAttributeGroup"&gt;   &lt;xs:attribute name="nodeRef" type="xs:IDREF" use="required" /&gt; &lt;/xs:attributeGroup&gt;</pre>		

### Attribute Group ConnectionRefAttributeGroup

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>connectionRef</b>	xs:IDREF	required
Source	<pre>&lt;xs:attributeGroup name="ConnectionRefAttributeGroup"&gt;   &lt;xs:attribute name="connectionRef" type="xs:IDREF" use="required" /&gt; &lt;/xs:attributeGroup&gt;</pre>		

### Attribute Group ViewConceptRefAttributeGroup

Namespace	http://www.opengroup.org/xsd/archimate/3.0/		
Attributes	<b>QName</b>	<b>Type</b>	<b>Use</b>
	<b>viewConceptRef</b>	xs:IDREF	required
Source	<pre>&lt;xs:attributeGroup name="ViewConceptRefAttributeGroup"&gt;   &lt;xs:attribute name="viewConceptRef" type="xs:IDREF" use="required" /&gt; &lt;/xs:attributeGroup&gt;</pre>		