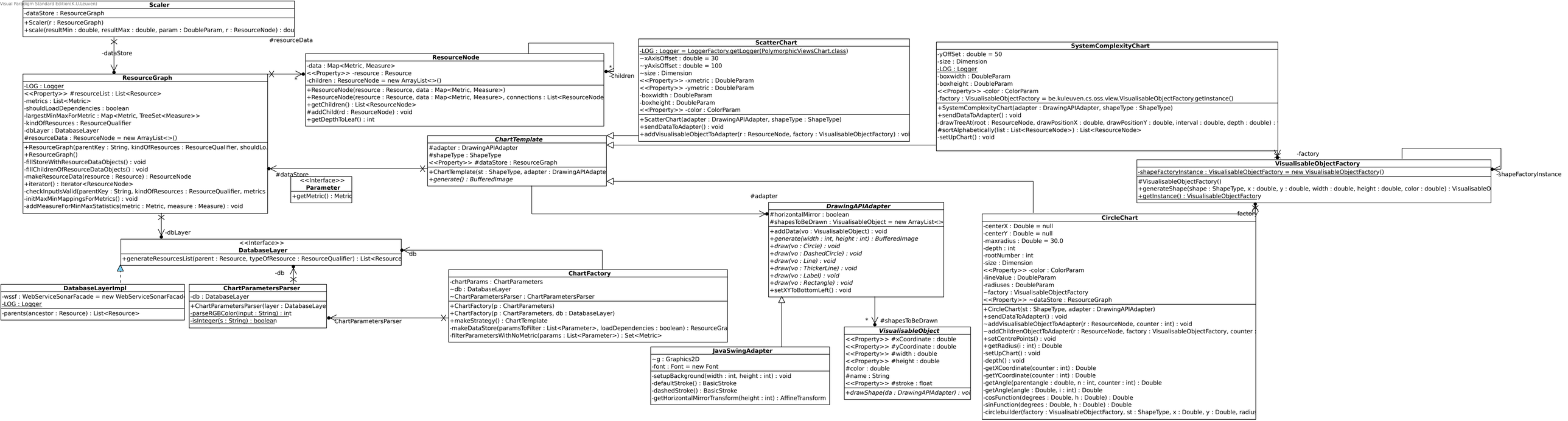
Visual Paradigm Standard Edition(K.U.Leuven)
Scaler
-dataStore : ResourceGraph
+Scaler(r : ResourceGraph)
+scale(resultMin : double, resultMax : double, param : DoubleParam, r : ResourceNode) : dou
-dataStore
#resourceData
ResourceGraph
-LOG : Logger
<<Property>> #resourceList : List<Resource>
-metrics : List<Metric>
-shouldLoadDependencies : boolean
-largestMinMaxForMetric : Map<Metric, TreeSet<Measure>>
-kindOfResources : ResourceQualifier
-dbLayer : DatabaseLayer
#resourceData : ResourceNode = new ArrayList<>()
+ResourceGraph(parentKey : String, kindOfResources : ResourceQualifier, shouldLo
+ResourceGraph()
-fillStoreWithResourceDataObjects() : void
-fillChildrenOfResourceDataObjects() : void
-makeResourceData(resource : Resource) : ResourceNode
+iterator() : Iterator<ResourceNode>
-checkInputIsValid(parentKey : String, kindOfResources : ResourceQualifier, metrics
-initMaxMinMappingsForMetrics() : void
-addMeasureForMinMaxStatistics(metric : Metric, measure : Measure) : void
ResourceNode
-data : Map<Metric, Measure>
<<Property>> -resource : Resource
-children : ResourceNode = new ArrayList<>()
+ResourceNode(resource : Resource, data : Map<Metric, Measure>)
+ResourceNode(resource : Resource, data : Map<Metric, Measure>, connections : List<ResourceNode
+getChildren() : List<ResourceNode>
#addChild(rd : ResourceNode) : void
+getDepthToLeaf() : int
-children
ScatterChart
-LOG : Logger = LoggerFactory.getLogger(PolymorphicViewsChart.class)
~xAxisOffset : double = 30
~yAxisOffset : double = 100
~size : Dimension
<<Property>> -xmetric : DoubleParam
<<Property>> -ymetric : DoubleParam
-boxwidth : DoubleParam
-boxheight : DoubleParam
<<Property>> -color : ColorParam
+ScatterChart(adapter : DrawingAPIAdapter, shapeType : ShapeType)
+sendDataToAdapter() : void
+addVisualisableObjectToAdapter(r : ResourceNode, factory : VisualisableObjectFactory) : voi
SystemComplexityChart
-yOffSet : double = 50
-size : Dimension
-LOG : Logger
-boxwidth : DoubleParam
-boxheight : DoubleParam
<<Property>> -color : ColorParam
-factory : VisualisableObjectFactory = be.kuleuven.cs.oss.view.VisualisableObjectFactory.getInstance()
+SystemComplexityChart(adapter : DrawingAPIAdapter, shapeType : ShapeType)
+sendDataToAdapter() : void
-drawTreeAt(root : ResourceNode, drawPositionX : double, drawPositionY : double, interval : double, depth : double) :
#sortAlphabetically(list : List<ResourceNode>) : List<ResourceNode>
-setUpChart() : void
-factory
ChartTemplate
#adapter : DrawingAPIAdapter
#shapeType : ShapeType
<<Property>> #dataStore : ResourceGraph
+ChartTemplate(st : ShapeType, adapter : DrawingAPIAdapte
+generate() : BufferedImage
#dataStore
<<Interface>>
Parameter
+getMetric() : Metric
VisualisableObjectFactory
-shapeFactoryInstance : VisualisableObjectFactory = new VisualisableObjectFactory()
#VisualisableObjectFactory()
+generateShape(shape : ShapeType, x : double, y : double, width : double, height : double, color : double) : VisualisableO
+getInstance() : VisualisableObjectFactory
-shapeFactoryInstance
#adapter
DrawingAPIAdapter
#horizontalMirror : boolean
#shapesToBeDrawn : VisualisableObject = new ArrayList<>
+addData(vo : VisualisableObject) : void
+generate(width : int, height : int) : BufferedImage
+draw(vo : Circle) : void
+draw(vo : DashedCircle) : void
+draw(vo : Line) : void
+draw(vo : ThickerLine) : void
+draw(vo : Label) : void
+draw(vo : Rectangle) : void
+setXYToBottomLeft() : void
factory
CircleChart
-centerX : Double = null
-centerY : Double = null
-maxradius : Double = 30.0
-depth : int
-rootNumber : int
-size : Dimension
<<Property>> -color : ColorParam
-lineValue : DoubleParam
-radiuses : DoubleParam
~factory : VisualisableObjectFactory
<<Property>> ~dataStore : ResourceGraph
+CircleChart(st : ShapeType, adapter : DrawingAPIAdapter)
+sendDataToAdapter() : void
~addVisualisableObjectToAdapter(r : ResourceNode, counter : int) : void
~addChildrenObjectToAdapter(r : ResourceNode, factory : VisualisableObjectFactory, counter
+setCentrePoints() : void
+getRadius(i : int) : Double
-setUpChart() : void
-depth() : void
-getXCoordinate(counter : int) : Double
-getYCoordinate(counter : int) : Double
-getAngle(parentangle : double, n : int, counter : int) : Double
-getAngle(angle : Double, i : int) : Double
-cosFunction(degrees : Double, h : Double) : Double
-sinFunction(degrees : Double, h : Double) : Double
-circlebuilder(factory : VisualisableObjectFactory, st : ShapeType, x : Double, y : Double, radius
-dbLayer
<<Interface>>
DatabaseLayer
+generateResourcesList(parent : Resource, typeOfResource : ResourceQualifier) : List<Resource
db
-db
DatabaseLayerImpl
-wssf : WebServiceSonarFacade = new WebServiceSonarFacad
-LOG : Logger
-parents(ancestor : Resource) : List<Resource>
ChartParametersParser
-db : DatabaseLayer
+ChartParametersParser(layer : DatabaseLaye
-parseRGBColor(input : String) : int
-isInteger(s : String) : boolean
-ChartParametersParser
ChartFactory
-chartParams : ChartParameters
~db : DatabaseLayer
~ChartParametersParser : ChartParametersParser
+ChartFactory(p : ChartParameters)
+ChartFactory(p : ChartParameters, db : DatabaseLayer)
+makeStrategy() : ChartTemplate
-makeDataStore(paramsToFilter : List<Parameter>, loadDependencies : boolean) : ResourceGra
-filterParametersWithNoMetric(params : List<Parameter>) : Set<Metric>
* #shapesToBeDrawn
VisualisableObject
<<Property>> #xCoordinate : double
<<Property>> #yCoordinate : double
<<Property>> #width : double
<<Property>> #height : double
#color : double
#name : String
<<Property>> #stroke : float
+drawShape(da : DrawingAPIAdapter) : voi
JavaSwingAdapter
~g : Graphics2D
-font : Font = new Font
-setupBackground(width : int, height : int) : void
-defaultStroke() : BasicStroke
-dashedStroke() : BasicStroke
-getHorizontalMirrorTransform(height : int) : AffineTransform