Visualisation in ADOxx

Extending the SeMFIS Toolkit with Model-based Information Visualization approaches

Amine Saoudi, Clemens Böhm

Institute for Business Informatics at the University of Vienna
PR Large Scale Information Systems SS 2015
Univ.-Prof. PD Mag. Dr. Hans-Georg Fill

Abstract. ADOxx[1] is a universal tool for developing modelling methods[2] and OMILAB[3] ist the home for a range of projects realized with ADOxx. For developers the tools and libraries are free to use under the single condition that developed code has to be made available for the developing community. ADOxx has a built in Visualization engine[4] to draw rectangles, lines, arcs, etc., to fill geometric objects with a desired color and to combine these objects to complex visuals. Most of these visuals are static and are used as icons. Models can be created and simulations can be run to evaluate processes. The generated data can make use of the visualization engine and be presented by dynamic graphics[5]. SeMFIS[6] is one of the projects hosted on the OMILAB-site "omilab.org". The Large Scale Information Systems 2015 course offered the opportunity to contribute to the developers community by extending the SeMFIS Toolkit with model-based information visualization approaches.

Keywords: ADOxx, OMILAB, SeMFIS, meta modelling, dynamic, visualization, line, chart, column, pie, radial, treemap

Table of Contents

1 The Concept . . . 3
2 The Visualization Matrix . . . 4
3 The Meta Model . . . 5
4 The Data Model . . . 6
5 The Visualization Model . . . 7
5.1 Line Chart . . . 7
5.2 Pie Chart . . . 8
5.3 Columns Chart . . . 9
5.4 Radial Tree Map . . . 10
6 Interref & Data Mapping . . . 11
6.1 Interref . . . 11
6.2 Mapping the data . . . 11
7 Conclusion & Outlook . . . 12
A ADOscripts . . . 13
A.1 csv_import.asc . . . 13
A.2 map_column.asc . . . 16
A.3 map_rows.asc . . . 18
B GRAPHREPs . . . 22
B.1 GraphRep of class "LineChart" . . . 22
B.2 GraphRep of class "PieChart" . . . 24
B.3 GraphRep of class "ColumnChart" . . . 26
B.4 GraphRep of class "RadialTreeMap" . . . 29

1 The Concept

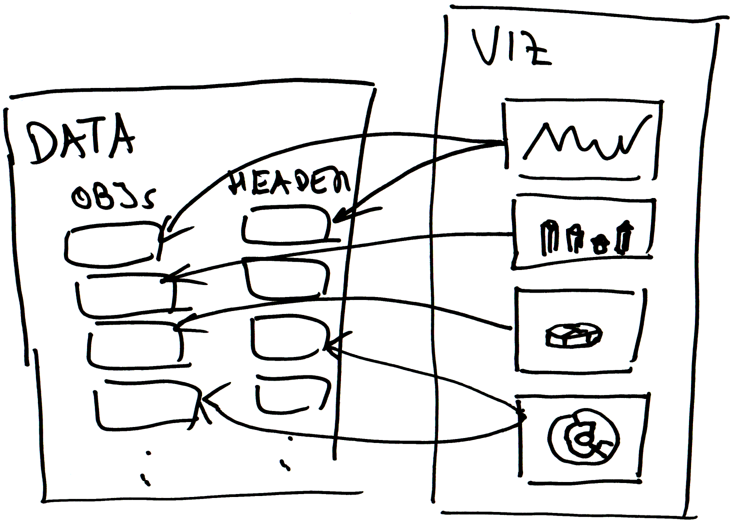

Fig. 1. Concept

The objects of the Visualization models are referencing objects of the Data model. Either the "Headers" are referenced that are representing the column names of the data file, or the data objects are referenced directly. In the case that the Data model is accessed by a "Header" the data objects are accessed via this "Header".

2 The Visualization Matrix

Gene Zelazny describes the process from analysing the type of data to visualize until the selection of the appropriate chart type.[7]

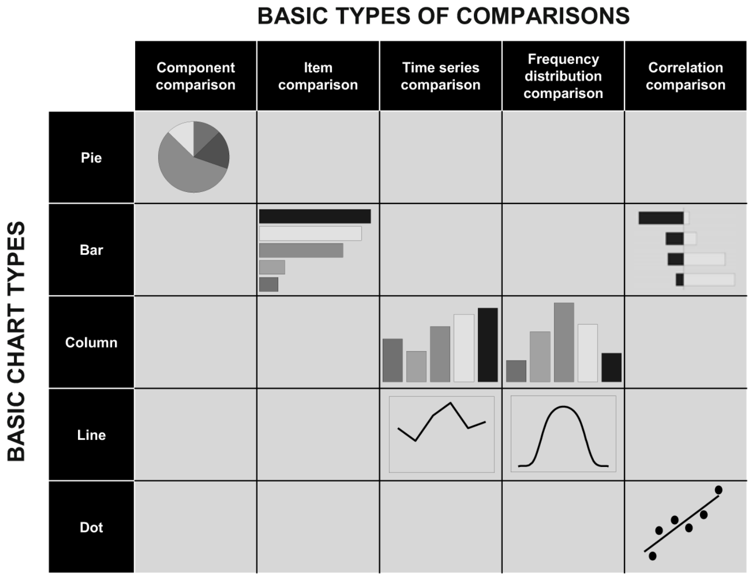

Fig. 2. Visualization matrix[8]

In our work we focus on these type of charts:

- Line
- Columns
- Pie
- Radial Tree Map

3 The Meta Model

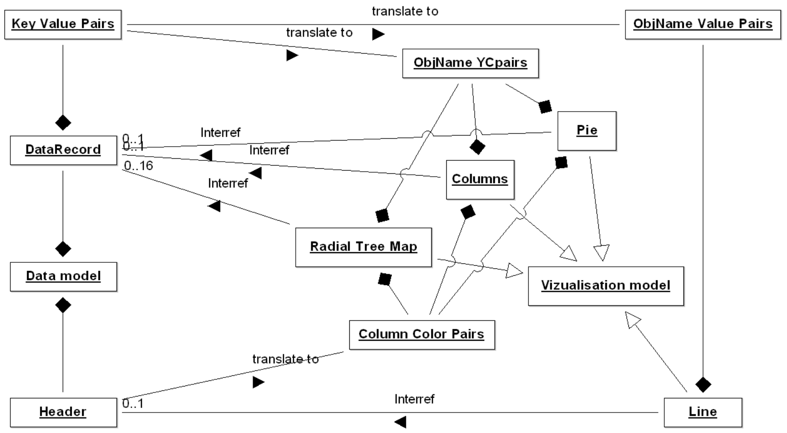

Fig. 3. Meta model

The metamodel shows the Data model on the left side and the Visualization models on the right. The Objects of the Visualization models reference objects in the Data model. In our work are two different types of visualization. The Line chart visualizes one column of multiple rows of the data table. The Pie, Column and RadialTree charts visualize one ore several objects with all columns. Therefor two different algorithms satisfy the different needs by mapping the values of the Data model to the Visualization model. Both algorithms preserve the original values. All the graphical representation is done by the GraphRep attributes of the respective class.

4 The Data Model

Design

The Data Model is realized in ADOxx as a class with a record attribute containing 0..* classes of type "Key Value Pairs". "Key Value Pairs" is an already defined class in the SeMFIS toolkit and has the STRING-type attributes "Key" and "Value". To have the attributes of type string is the most generic approach since the data is not limited to numbers, values can be "Yes", "No", "True", "False", "green", etc. as well.

Data import

Our file "csv_import.asc" manages to import data from a csv-file to our generic data model.

The importer needs a Unix-style file ending (New line / "\n") within the csv-file and commas (",") as separators. Although it could be easily modified to accept a Windows or Mac file ending as well as a semi-colon ";" as a separator.

The file ist stored in the library's database. A new Pulldown menu "CSV" is created in the Import/Export Menu section with a menu item "Import" by this piece of code in the Library Attribute's Add-ons section.

```
ITEM "Import" importexport:"~CSV"
  CC "AdoScript" FREAD file:("db:\\csv_import.asc")
  EXECUTE (text)
```

Listing 1.1. Library attributes / Add-ons / External coupling

We were using several statistical data from our "Methoden der Datananalyse" course by Marcus Hudec as well as the AAPL historical prices from finance.yahoo.com and the seating in the parliament of Austria from 1945 until 2013. The seating data was very versatile as we could use that for a line chart to show the performance of a single party over the years, to show the distribution of seats after an election with a Pie chart or to visualize the seating over several elections with a Radial Tree Map.

5 The Visualization Model

5.1 Line Chart

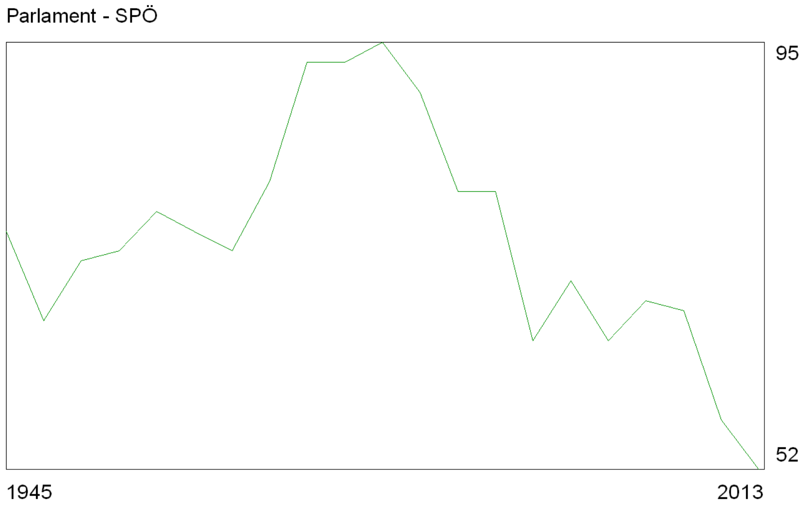

Fig. 4. Line Chart

The line chart visualization offers the possibility to see the progress of a value over time. The standard for the time axis is left to right. Therefor the chart is expecting an ascending order in the data model. Usually a log file would just add a new entry at the end of the file so the downmost entry world be the last. The progress would be displayed correctly. Our AAPL stock historical prices downloaded from finance.yahoo.com have an descending order. For such cases the Notebook Option "Reverse" has to be checked.

5.2 Pie Chart

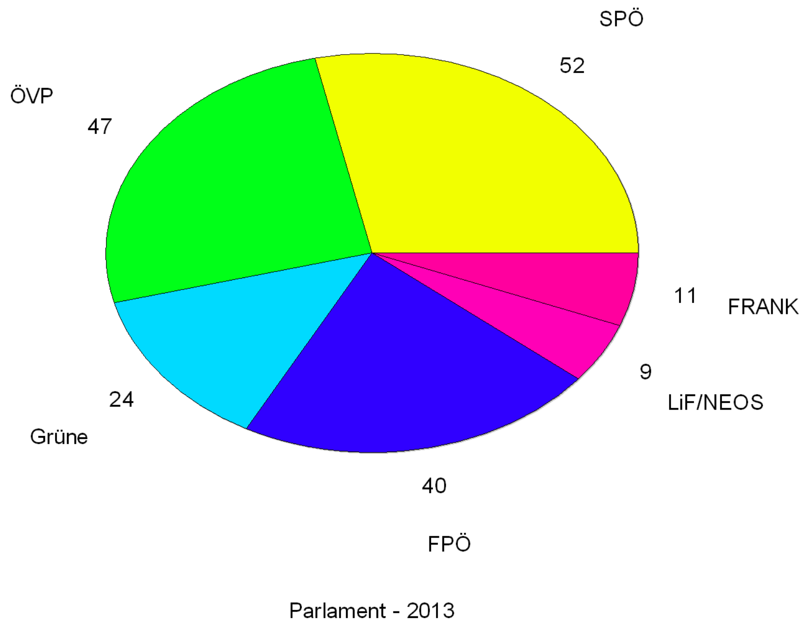

Fig. 5. Pie Chart

The values of all eintries in the Record attribute are added to a Total. Only entries with a value>0 are graphically presented. The value and key of each segment is shown next to the segment.

5.3 Columns Chart

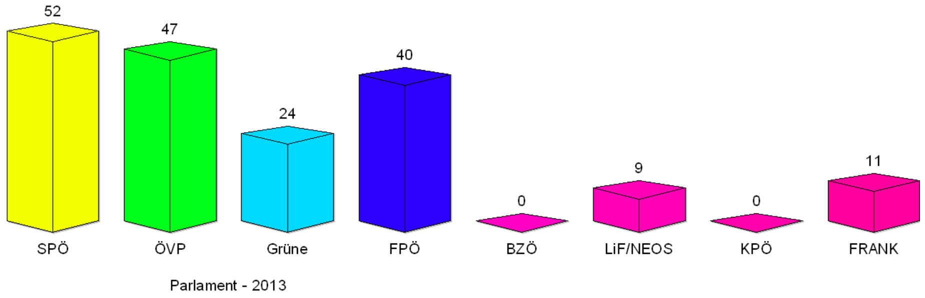

Fig. 6. Columns Chart

All entries are graphically presented. The key of each segment is shown below the respective column, while the value is displayed above.

5.4 Radial Tree Map

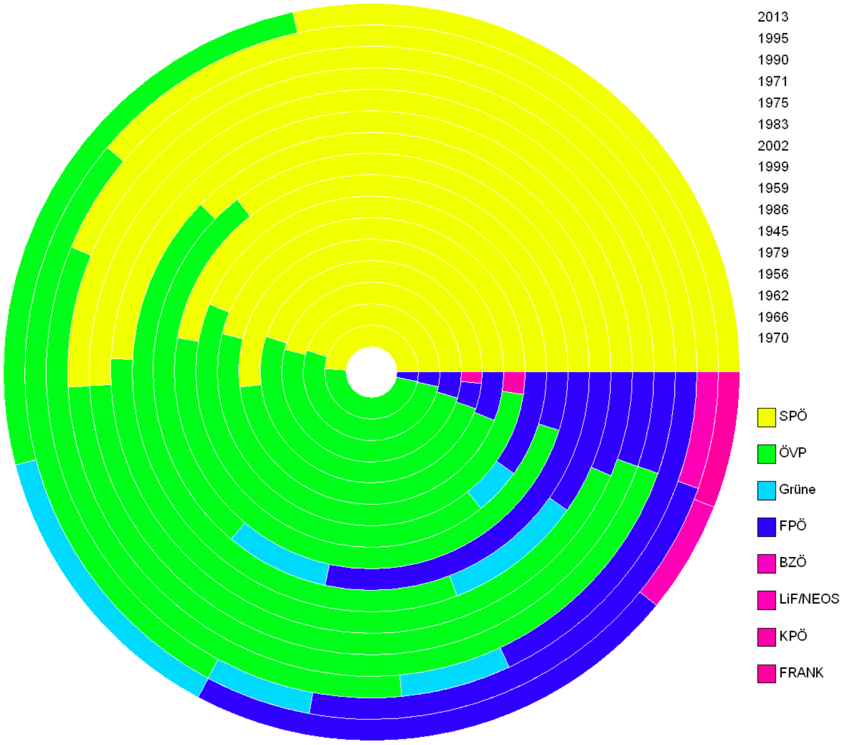

Fig. 7. Radial Tree Map

The radial Tree Map visualizes several objects with all column values (Values>0). For performance reasons the number of data objects that can be referenced is limited to 16 by the "Facets" entry of the "Interref" attribute "Data model".

```
REFDOMAIN
OBJREF
       mt:"Data model"
       c:"DataRecord"
       max: 16
```

Listing 1.2. Facet of the Interref attribute

6 Interref & Data Mapping

6.1 Interref

In ADOxx an attribute of a class of type "Interref" allows the referencing of other models or classes within these models.

In our work the procedure is as follows: The user selects in the Notebook of the visualization object an object of a data model as data source.

6.2 Mapping the data

After the selection of the referenced object an ON_EVENT "AfterEditAttributeValue" in the library's Add-ons section is triggered.

```
ON_EVENT "AfterEditAttributeValue" {
  IF( attrtypeid=9 ) {
    CC "Core" GET_ATTR_VAL objid: (instid) attrname:"External tool coupling"
    CC "AdoScript" FREAD file:(val)
    EXECUTE(text)
  }
}
```

Listing 1.3. Library attributes / Add-ons / External coupling

This code piece selects the stored value of the active class's variable “External tool coupling” where the name and path of a mapping script is stored (like a callback).
The mapping script is executed and does all the mapping from the data model to the visualization model. Colors for the graphical representation are preset by an algorithm that divides the 360 degree Hue value range (of the HSV-color model) by the amount of dimensions that will be displayed (Value>0).

The mapping scripts that do the translation depending on the type of visualization:

- "map_column.asc" takes the values from one column out of multiple objects resulting in a series of values, limited to a maximum of 256 objects for performance reasons. This script is expecting more than one data objects to work properly (to avoid a division by 0).
- "map_rows.asc" takes one (Pie and Columns chart) or several (RadialTreeMap) rows with the values of all the columns for that row(s). This script is expecting more than one Value>0 per data object to work properly (to avoid a division by 0).

In general the scripts are created for data values >=0.
See the appendix for the code.

7 Conclusion & Outlook

We have managed to create a generic Data model and some dynamic visualization classes and hope that our work will be used a lot of times to visualize data on the ADOxx-platform. Either as is or just as inspiration for related work. Despite our work more chart types are to be implemented in the future to let the users of ADOxx select from a full range.

References

1. http://www.adoxx.org/live/adoxx-documentation, last access 15.06.2015
2. Fill, Hans-Georg, Karagiannis, Dimitris (2013): On the Conceptualisation of Modelling Methods Using the ADOxx Meta Modelling Platform, Enterprise Modelling and Information Systems Architectures - An International Journal, Vol. 8, Issue 1, 4-25.
3. http://www.omilab.org/web/guest/about, last access 15.06.2015
4. Fill, Hans-Georg, Höfferer, Peter (2006): Visual Enhancements of Enterprise Models, in: Lehner, F., Nösekabel, H., Kleinschmidt, P. (2006): Multikonferenz Wirtschaftsinformatik 2006, GITO Verlag, 541-550.
5. Fill, Hans-Georg, Eberhart, Andreas, Laslop, Andrea, Reischl, Ilona, Lang, Thomas, Karagiannis, Dimitris (2011): An Approach to Support the Performance Management of Public Health Authorities using an IT based Modeling Method, in: Bernstein, A., Schwabe, G. (2011): Proceedings of the 10th International Conference on Wirtschaftsinformatik WI 2.011, Volume 1, 38-47.
6. Fill, Hans-Georg (2012): SeMFIS: A Tool for Managing Semantic Conceptual Models, in: Störrle, H. et al.: ECMFA 2012 - Joint Proceedings Co-located Events at the 8th European Conference on Modelling Foundations and Applications, Technical University of Denmark, 229-240, Lyngby, Denmark.
7. Zelazny G.(1999). Wie aus Zahlen Bilder werden: Der Weg zur visuellen Kommunikation, Dr. Th. Gabler Verlag; Auflage: 5. Aufl. 1999. 4., korr. Nachdruck 2003
8. H.-G. Fill, Visualisation for Semantic Information Systems, Gabler, Wiesbaden, 2009

A ADOscripts

A.1 csv_import.asc

```
#user selects csv-file
CC "AdoScript" FILE_DIALOG open filter1:"CSV-files" type1:"*.csv" default-ext:"csv"
IF (path = "") {
  CC "AdoScript" ERRORBOX "No file selected!"
    EXIT
}
SET csvpath:(path)

CC "AdoScript" QUERYBOX "First column is an index?" yes-no
SET index_endbutton:(endbutton)

#user request for new model name
#SEND "EXEC_NEW_DLG modeltype:\"Data model\" show-models" to:"Modeling" answer:modelid

FOR sub in: (csvpath) sep: "\\" {
  SET res: (search(lower(sub),".csv",0) )

  IF( res > -1 ) {
    SEND ("EXEC_NEW_DLG modelname:\"" + copy(sub,0,res)  + "\" modeltype:\"Data model\" show-models") to:"Modeling" answer:modelid
  }
}

IF (modelid = "") {
  CC "AdoScript" ERRORBOX "No model created!"
    EXIT
}
SET new_modelid: (VAL modelid)

#csv-file read
CC "AdoScript" FREAD file: (csvpath)
SET sCsvData: ( text )

#activate the new model
CC "Modeling" ACTIVATE_MODEL modelid:(new_modelid)

CC "Core" GET_CLASS_ID classname: ("DataRecord")
SET datarecord_classid: (classid)

CC "Core" GET_ATTR_ID classid:(datarecord_classid) attrname:"KeyValuePair"
SET keyvaluepair_attrid:(attrid)

SET nRow:0
SET tHeader:""

FOR sRow in: (sCsvData) sep: "\n" {
  IF( sRow = "" ) {
    BREAK
  }

  IF( nRow > 0 ) {
    #create data new object
    CC "Core" CREATE_OBJ modelid:(new_modelid) classid: (datarecord_classid)
    SET datarecord_objid: (objid)

    SET nColumn:0
    FOR sCell in: (sRow) sep: "," {
    IF ( nColumn < 1 AND index_endbutton="no" ) {
      CC "Core" SET_ATTR_VAL objid: (datarecord_objid) attrname:"Name" val:(nRow)
    }
    IF ( nColumn < 1 AND index_endbutton="yes" ) {
      CC "Core" SET_ATTR_VAL objid: (datarecord_objid) attrname:"Name" val:(sCell)
    }
    ELSE {
      CC "Core" ADD_REC_ROW objid: (datarecord_objid) attrid: (keyvaluepair_attrid)
      SETL nNewRowID: (rowid)
      CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:"Key" val:(token(tHeader, nColumn,","))

            CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:"Value" val:(sCell)
    }
      SET nColumn: (nColumn + 1)
    }
  }
  ELSE {
    SET tHeader:(sRow)

  CC "Core" GET_CLASS_ID classname: ("Header")
  SET header_classid: (classid)

    SET nColumn:0
    FOR sCell in: (sRow) sep: "," {

        #create an index if there is no index column
    IF ( index_endbutton="no" ) {
      CC "Core" CREATE_OBJ modelid:(new_modelid) classid: (header_classid)
      CC "Core" SET_ATTR_VAL objid: (objid) attrname:"Name" val:(sCell)
      CC "Core" SET_ATTR_VAL objid: (objid) attrname:"Index" val:(nColumn+1)
    }

        #skip the first column(0) if it is an index
    IF ( index_endbutton="yes" AND nColumn > 0 ) {
      CC "Core" CREATE_OBJ modelid:(new_modelid) classid: (header_classid)
      CC "Core" SET_ATTR_VAL objid: (objid) attrname:"Name" val:(sCell)
      CC "Core" SET_ATTR_VAL objid: (objid) attrname:"Index" val:(nColumn)
    }
    SET nColumn: (nColumn + 1)
  }
  }
  SET nRow: (nRow + 1)
}
CC "Modeling" CLOSE modelid:(new_modelid) quiet save
```

Listing 1.4. csv_import.asc

A.2 map_column.asc

```
CC "Modeling" GET_SELECTED
IF (objids = "") {
    CC "AdoScript" ERRORBOX "No object has been selected!"
    EXIT
}

# from the list of selected objects, extract the first objectid
SET selected:(VAL token(objids,0," "))

# get the class of the selected object
CC "Core" GET_CLASS_ID objid:(selected)
SET linechart_classid:(classid)

CC "Core" GET_ATTR_ID classid:(linechart_classid) attrname:"Yvalues"
SET yvalues_attrid:(attrid)

#delete all rows of the record attribute
CC "Core" GET_REC_ATTR_ROW_COUNT objid:(selected) attrid:(yvalues_attrid)
FOR d from:1 to:(count) {
  CC "Core" GET_REC_ATTR_ROW_ID objid:(selected) attrid:(yvalues_attrid) index:1
  CC "Core" REMOVE_REC_ROW objid:(selected) attrid:(yvalues_attrid) rowid:(rowid)
}

CC "Core" GET_ATTR_ID classid:(linechart_classid) attrname:"Data model"
#SET datamodel_ir_attrid:(attrid)

CC "Core" GET_INTERREF objid:(selected) attrid:(attrid) index:(0)
SET data_modelid:(tmodelid)

CC "Core" LOAD_MODEL modelid:(data_modelid)

#CC "Core" GET_ATTR_ID classid:(linechart_classid) attrname:"Data model"
#SET datamodel_ir_attrid:(attrid)

CC "Core" GET_INTERREF objid:(selected) attrid:(attrid) index:(0)
SET header_objid:(tobjid)

CC "Core" GET_ATTR_VAL objid:(header_objid) attrname:"Index"
SET index_val:(val)
```

```

CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(data_modelid) classname:"DataRecord"

SET x: 0
FOR datarecord_objid in:(objids) {
  CC "Core" GET_CLASS_ID classname: ("DataRecord")
  CC "Core" GET_ATTR_ID classid: (classid) attrname:"KeyValuePair"

  CC "Core" GET_ATTR_VAL objid:(VAL datarecord_objid) attrname:"Name"
  SET objval:(val)

  CC "Core" GET_REC_ATTR_ROW_ID objid:(VAL datarecord_objid) attrid:(attrid) index:(index_val)

  CC "Core" GET_ATTR_VAL objid:(rowid) attrname:"Value"

  CC "Core" ADD_REC_ROW objid: (selected) attrid: (yvalues_attrid)
  SETL nNewRowID: (rowid)
  CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:"Key" val:(objval)
  CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:"Value" val:(val)

  SET x:(x+1)

  IF( x>255 ) {
    BREAK
  }
}
CC "Core" DISCARD_MODEL modelid:(data_modelid)
```

Listing 1.5. map_column.asc

A.3 map_rows.asc

```
CC "Modeling" GET_SELECTED
IF (objids = "") {
   CC "AdoScript" ERRORBOX "No object has been selected!"
   EXIT
}

# from the list of selected objects, extract the first objectid
SET selected:(VAL token(objids,0," "))

# get the class of the selected object
CC "Core" GET_CLASS_ID objid:(selected)
SET column_classid:(classid)

CC "Core" GET_ATTR_ID classid:(column_classid) attrname:"Header"
SET header_attrid:(attrid)

#delete all rows of the Header record attribute
CC "Core" GET_REC_ATTR_ROW_COUNT objid:(selected) attrid:(header_attrid)
FOR d from:1 to:(count) {
  CC "Core" GET_REC_ATTR_ROW_ID objid:(selected) attrid:(header_attrid) index:1
  CC "Core" REMOVE_REC_ROW objid:(selected) attrid:(header_attrid) rowid:(rowid)
}

CC "Core" GET_ATTR_ID classid:(column_classid) attrname:"YCvalues"
SET ycvalues_attrid:(attrid)

#delete all rows of YCvalues record attribute
CC "Core" GET_REC_ATTR_ROW_COUNT objid:(selected) attrid:(ycvalues_attrid)
FOR d from:1 to:(count) {
  CC "Core" GET_REC_ATTR_ROW_ID objid:(selected) attrid:(ycvalues_attrid) index:1
  CC "Core" REMOVE_REC_ROW objid:(selected) attrid:(ycvalues_attrid) rowid:(rowid)
}

CC "Core" GET_ATTR_ID classid:(column_classid) attrname:"Data model"
SET datamodel_ir_attrid:(attrid)

CC "Core" GET_INTERREF objid:(selected) attrid:(attrid) index:(0)
```

```
SET data_modelid:(tmodelid)

CC "Core" LOAD_MODEL modelid:(data_modelid)

CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:(data_modelid) classname:"Header"
SET count:0
FOR header_objid in:(objids) {
  SET count:(count+1)
}

SET i:0
FOR header_objid in:(objids) {
    CC "Core" GET_ATTR_VAL objid:(VAL header_objid) attrname:"Name"
    SET header:(val)

    CC "Core" ADD_REC_ROW objid: (selected) attrid: (header_attrid)
    CC "Core" SET_ATTR_VAL objid: (rowid) attrname:"Header" val:(header)

    SET h: (i*6/count)

    SET s:1
    SET v:255

    SET region: (INT h)
    SET ff: (h - region)
    SET p: (v * (1.0 - s))
    SET q: (v * (1.0 - (s * ff)))
    SET t: (v * (1.0 - (s * (1.0 - ff))))

    IF( region = 0 ) {
      SET color:(rgbval(v,t,p))
    }
    ELSE {
      IF( region = 1 ) {
        SET color:(rgbval(q,v,p))
      }
      ELSE {
        IF( region = 2 ) {
          SET color:(rgbval(p,v,t))
        }
        ELSE {
          IF( region = 3 ) {
            SET color:(rgbval(p,q,v))
          }
          ELSE {
            IF( region = 4 ) {
              SET color:(rgbval(t,p,v))
            }
            ELSE {
              SET color:(rgbval(v,p,q))
            }
          }
        }
      }
    }
    CC "Core" SET_ATTR_VAL objid: (rowid) attrname:"Color" val:(color)
    SET i:(i+1)
}

CC "Core" GET_INTERREF_COUNT objid:(selected) attrid:(datamodel_ir_attrid)
SET datamodelir_count:(count)

FOR ir from:0 to:(datamodelir_count-1) {
  CC "Core" GET_INTERREF objid:(selected) attrid:(datamodel_ir_attrid) index:(ir)
  SET data_modelid:(tmodelid)
  SET datarecord_objid:(tobjid)

  CC "Core" GET_CLASS_ID classname: ("DataRecord")
  CC "Core" GET_ATTR_ID classid: (classid) attrname:"KeyValuePair"

  CC "Core" GET_ATTR_VAL objid:(datarecord_objid) attrname:"Name"
  SET objval:(val)

  CC "Core" GET_REC_ATTR_ROW_COUNT objid:(datarecord_objid) attrid:(attrid)

  FOR i from:1 to:(count) {
    CC "Core" GET_REC_ATTR_ROW_ID objid:(datarecord_objid) attrid:(attrid) index:(i)

    CC "Core" GET_ATTR_VAL objid:(rowid) attrname:"Key"
    SET keyval:(val)
    CC "Core" GET_ATTR_VAL objid:(rowid) attrname:"Value"
    SET valueval:(val)

    CC "Core" ADD_REC_ROW objid: (selected) attrid: (ycvalues_attrid)
    SETL nNewRowID: (rowid)
    CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:"Key" val:(objval)
    CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:"Value" val:(valueval)
  }
}

CC "Core" DISCARD_MODEL modelid:(data_modelid)
```

Listing 1.6. map_rows.asc

B GRAPHREPs

B.1 GraphRep of class "LineChart"

```
GRAPHREP sizing: keep-aspect-ratio
SHADOW off

AVAL set-format:"%m" dm:"Data model"
AVAL set-format:"%o" dr:"Data model"

AVAL set-default: black bgd:"Background"
AVAL set-default:$00FF00 fgd:"Line"
AVAL set-count-rows rowcount:"Yvalues"

FILL color:(bgd)
RECTANGLE x:-320pt y:-180pt w:640pt h:360pt

FONT "Arial" h:16pt

IF(dm = "") {
  PEN w:1 color:(fgd)
  LINE x1:(-320pt) y1:(0pt) x2:(-200pt) y2:(-110pt)
  LINE x1:(-200pt) y1:(-110pt) x2:(-100pt) y2:(180pt)
  LINE x1:(-100pt) y1:(180pt) x2:(0pt) y2:(-40pt)
  LINE x1:(-0pt) y1:(-40pt) x2:(100pt) y2:(60pt)
  LINE x1:(100pt) y1:(60pt) x2:(200pt) y2:(-180pt)
  LINE x1:(200pt) y1:(-180pt) x2:(320pt) y2:(-100pt)
}
ELSE {
  TEXT (dm+" - "+dr) x:-320pt y:-190pt w:l h:b
  AVAL set-default:"0" reverse:"Reverse"

  SET high:0
  SET low:0

  FOR i from:1 to:(rowcount) {
    AVAL set-row:(i) set-col:"Value" y:"Yvalues"

    IF( (VAL y)>high )
      SET high:(VAL y)
    ENDIF

    IF( low=0 OR (VAL y)<low )
      SET low:(VAL y)
    ENDIF
  }

  TEXT (high) x:330pt y:-180pt w:l h:t
  TEXT (low) x:330pt y:180pt w:l h:b


  SET y_from:0

  PEN w:1 color:(fgd)

  SET xstep:(640pt/(rowcount-1))

  FOR i from:0 to:(rowcount-1) {
    IF( reverse="1" )
       AVAL set-row:(rowcount-i) set-col:"Value" y:"Yvalues"
    ELSE
       AVAL set-row:(i+1) set-col:"Value" y:"Yvalues"
    ENDIF

    SET y_to:(VAL y)

    IF( i>0 )
      LINE x1:((i-1)*xstep-320pt) y1:(180pt-(y_from-low)/(high-low)*360pt) x2:(i*xstep-320pt) y2:(180pt-(y_to-low)/(high-low)*360pt)
    ENDIF

    SET  y_from:(y_to)
  }
  IF( reverse="1" )
    AVAL set-row:(rowcount) set-col:"Key" l:"Yvalues"
    AVAL set-row:(1) set-col:"Key" r:"Yvalues"
  ELSE
    AVAL set-row:(rowcount) set-col:"Key" r:"Yvalues"
    AVAL set-row:(1) set-col:"Key" l:"Yvalues"
  ENDIF

  TEXT (l) x:-320pt y:190pt w:l h:t
  TEXT (r) x:320pt y:190pt w:r h:t
}
```

Listing 1.7. GraphRep of class "LineChart"

B.2 GraphRep of class "PieChart"

```
GRAPHREP sizing: keep-aspect-ratio

AVAL set-format:"%m" dm:"Data model"
AVAL set-format:"%o" dr:"Data model"

AVAL set-count-rows rowcount:"YCvalues"

FONT "Arial" h:16pt color:black

SET xsize:(120pt)
SET ysize:(90pt)

IF(dm = "") {
    SET a:(3.142*0/100)
    SET b:(3.142*2*40/100)

    FILL color:$FFFF00
    PIE x:0pt y:0pt rx:(xsize) ry:(ysize) x1:(cos(a)*xsize) y1:(-sin(a)*ysize) x2:(cos(b)*xsize) y2:(-sin(b)*ysize)

    SET a:(3.142*2*40/100)
    SET b:(3.142*2*70/100)

    FILL color:$00FFFF
    PIE x:0pt y:0pt rx:(xsize) ry:(ysize) x1:(cos(a)*xsize) y1:(-sin(a)*ysize) x2:(cos(b)*xsize) y2:(-sin(b)*ysize)
    SET a:(3.142*2*70/100)
    SET b:(3.142*2*100/100)

    FILL color:$FF00FF
    PIE x:0pt y:0pt rx:(xsize) ry:(ysize) x1:(cos(a)*xsize) y1:(-sin(a)*ysize) x2:(cos(b)*xsize) y2:(-sin(b)*ysize)
}
ELSE {
  TEXT (dm+" - "+dr) x:0pt y:(ysize*1.8) w:c h:c

  SET total:0
  SET cum:0

  FOR i from:1 to:(rowcount) {
    AVAL set-row:(i) set-col:"Value" h:"YCvalues"
    SET total: (total+(VAL h))
  }

  SET a:(3.142*0/total)

  FOR i from:1 to:(rowcount) {
    AVAL set-row:(i) set-col:"Value" h:"YCvalues"
    AVAL set-row:(i) set-col:"Header" n:"Header"
    AVAL set-row:(i) set-col:"Color" col:"Header"

    IF( (VAL h)>0 ) {
      SET cumtext:(cum+(VAL h)/2)
      SET cum:(cum+(VAL h))

      SET b:(3.142*2*cum/total)
      SET c:(3.142*2*cumtext/total)

      FILL color:(col)
      PIE x:0pt y:0pt rx:(xsize) ry:(ysize) x1:(cos(a)*xsize) y1:(-sin(a)*ysize) x2:(cos(b)*xsize) y2:(-sin(b)*ysize)

      TEXT (copy(h,0,4))  x:(cos(c)*xsize*1.2) y:(-sin(c)*ysize*1.2) w:c h:c
      TEXT (n) x:(cos(c)*xsize*1.5) y:(-sin(c)*ysize*1.5) w:c h:c
    }
    SET a:(b)
  }
}
```

Listing 1.8. GraphRep of class "PieChart"

B.3 GraphRep of class "ColumnChart"

```
GRAPHREP sizing: keep-aspect-ratio

AVAL set-format:"%m" dm:"Data model"
AVAL set-format:"%o" dr:"Data model"

SET xofs:(0pt)
SET yofs:(0pt)

FONT "Arial" h:16pt color:black

 IF(dm = "") {
  SET height:(-60pt)
  SET c:$FFFF07
  SET xstep:(0pt)

#top
  FILL color:(c)
  POLYGON 4 x1:(xstep+xofs) y1:(height-10pt-yofs) x2:(xstep-25pt+xofs) y2:(height-6pt-yofs) x3:(xstep+xofs) y3:(height-yofs) x4:(xstep+25pt+xofs) y4:(height-6pt-yofs)

#left
  FILL color:(c)
  POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(height-yofs) x3:(xstep-25pt+xofs) y3:(height-6pt-yofs) x4:(xstep-25pt+xofs) y4:(-6pt-yofs)

#right
  FILL color:(c)
  POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(height-yofs) x3:(xstep+25pt+xofs) y3:(height-6pt-yofs) x4:(xstep+25pt+xofs) y4:(-6pt+-yofs)

  SET height:(-100pt)
  SET c:$07FFFF
  SET xstep:(64pt)

#top
  FILL color:(c)
  POLYGON 4 x1:(xstep+xofs) y1:(height-10pt-yofs) x2:(xstep-25pt+xofs) y2:(height-6pt-yofs) x3:(xstep+xofs) y3:(height-yofs) x4:(xstep+25pt+xofs) y4:(height-6pt-yofs)

#left
  FILL color:(c)
  POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(height-yofs) x3:(xstep-25pt+xofs) y3:(height-6pt-yofs) x4:(xstep-25pt+xofs) y4:(-6pt-yofs)
```

```

#right
  FILL color:(c)
  POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(height-yofs) x3:(xstep+25pt+xofs) y3:(height-6pt-yofs) x4:(xstep+25pt+xofs) y4:(-6pt+-yofs)

  SET height:(-80pt)
  SET c:$FF07FF
  SET xstep:(128pt)

#top
  FILL color:(c)
  POLYGON 4 x1:(xstep+xofs) y1:(height-10pt-yofs) x2:(xstep-25pt+xofs) y2:(height-6pt-yofs) x3:(xstep+xofs) y3:(height-yofs) x4:(xstep+25pt+xofs) y4:(height-6pt-yofs)

#left
  FILL color:(c)
  POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(height-yofs) x3:(xstep-25pt+xofs) y3:(height-6pt-yofs) x4:(xstep-25pt+xofs) y4:(-6pt-yofs)

#right
  FILL color:(c)
  POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(height-yofs) x3:(xstep+25pt+xofs) y3:(height-6pt-yofs) x4:(xstep+25pt+xofs) y4:(-6pt+-yofs)
}
ELSE {
  AVAL set-count-rows rowcount:"YCvalues"

  TEXT (dm + " - " + dr) x:(rowcount*12pt) y:(30pt-yofs) w:c h:c

  SET max:0

  FOR i from:0 to:(rowcount-1) {
    AVAL set-row:(i+1) set-col:"Value" y:"YCvalues"
    IF( VAL y > max )
      SET max:(VAL y)
    ENDIF
  }

  FOR i from:0 to:(rowcount-1) {
    AVAL set-row:(i+1) set-col:"Value" y:"YCvalues"
    AVAL set-row:(i+1) set-col:"Header" n:"Header"
    AVAL set-row:(i+1) set-col:"Color" c:"Header"

    SET height:(-(VAL y)/max*240pt)

    SET xstep:(i*64pt)

    #top
    FILL color:(c)
    POLYGON 4 x1:(xstep+xofs) y1:(height-10pt-yofs) x2:(xstep-25pt+xofs) y2:(height-6pt-yofs) x3:(xstep+xofs) y3:(height-yofs) x4:(xstep+25pt+xofs) y4:(height-6pt-yofs)

    #left
    FILL color:(c)
    POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(height-yofs) x3:(xstep-25pt+xofs) y3:(height-6pt-yofs) x4:(xstep-25pt+xofs) y4:(-6pt-yofs)

    #right
    FILL color:(c)
    POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(height-yofs) x3:(xstep+25pt+xofs) y3:(height-6pt-yofs) x4:(xstep+25pt+xofs) y4:(-6pt+-yofs)

    TEXT (y) x:(xstep+xofs) y:(height-16pt-yofs) w:c h:c
    TEXT (n) x:(xstep+xofs) y:(10pt-yofs) w:c h:c
  }
}
```

Listing 1.9. GraphRep of class "ColumnChart"

B.4 GraphRep of class "RadialTreeMap"

```
GRAPHREP sizing: keep-aspect-ratio
SHADOW off

AVAL set-format:"%o" dr:"Data model"

AVAL col:"fontcolor"
FONT "Arial" h:16pt color:black

SET size:(200pt)

ELLIPSE x:0pt y:0pt rx:(size) ry:(size)

IF(dr = "") {
    SET size:200pt
    SET a:(3.142*2*0/100)
    SET b:(3.142*2*40/100)

    FILL style:null
    PEN w1
    CLIP_ELLIPSE x:0pt y:0pt rx:(size*2/3) ry:(size*2/3) combine-mode:diff

    FILL color:($FFFF00)
    PIE x:0pt y:0pt rx:(size) ry:(size) x1:(cos(a)*size) y1:(-sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
    CLIP_OFF

    SET a:(3.142*2*40/100)
    SET b:(3.142*2*70/100)

    FILL style:null
    CLIP_ELLIPSE x:0pt y:0pt rx:(size*2/3) ry:(size*2/3) combine-mode:diff

    FILL color:$FF00FF
    PIE x:0pt y:0pt rx:(size) ry:(size) x1:(cos(a)*size) y1:(-sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
    CLIP_OFF

    SET a:(3.142*2*70/100)
    SET b:(3.142*2*100/100)

    FILL style:null
    CLIP_ELLIPSE x:0pt y:0pt rx:(size*2/3) ry:(size*2/3) combine-mode:diff

    FILL color:$00FFFF
    PIE x:0pt y:0pt rx:(size) ry:(size) x1:(cos(a)*size) y1:(-sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
    CLIP_OFF

    SET size:150pt
    SET a:(3.142*2*0/100)
    SET b:(3.142*2*30/100)

    FILL style:null
    CLIP_ELLIPSE x:0pt y:0pt rx:(size-50pt) ry:(size-50pt) combine-mode:diff

    FILL color:$FFFF00
    PIE x:0pt y:0pt rx:(size) ry:(size) x1:(cos(a)*size) y1:(-sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
    CLIP_OFF

    SET a:(3.142*2*30/100)
    SET b:(3.142*2*80/100)

    FILL style:null
    CLIP_ELLIPSE x:0pt y:0pt rx:(size-50pt) ry:(size-50pt) combine-mode:diff

    FILL color:$FF00FF
    PIE x:0pt y:0pt rx:(size) ry:(size) x1:(cos(a)*size) y1:(-sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
    CLIP_OFF

    SET a:(3.142*2*80/100)
    SET b:(3.142*2*100/100)

    FILL style:null
    CLIP_ELLIPSE x:0pt y:0pt rx:(size-50pt) ry:(size-50pt) combine-mode:diff

    FILL color:$00FFFF
    PIE x:0pt y:0pt rx:(size) ry:(size) x1:(cos(a)*size) y1:(-sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
    CLIP_OFF

    SET size:100pt
    SET a:(3.142*2*0/100)
    SET b:(3.142*2*20/100)

    FILL style:null
    CLIP_ELLIPSE x:0pt y:0pt rx:(size-50pt) ry:(size-50pt) combine-mode:diff

    FILL color:($FFFF00)
    PIE x:0pt y:0pt rx:(size) ry:(size) x1:(cos(a)*size) y1:(-sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
    CLIP_OFF

    SET a:(3.142*2*20/100)
    SET b:(3.142*2*50/100)

    FILL style:null
    CLIP_ELLIPSE x:0pt y:0pt rx:(size-50pt) ry:(size-50pt) combine-mode:diff

    FILL color:$FF00FF
    PIE x:0pt y:0pt rx:(size) ry:(size) x1:(cos(a)*size) y1:(-sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
    CLIP_OFF

    SET a:(3.142*2*50/100)
    SET b:(3.142*2*100/100)

    FILL style:null
    CLIP_ELLIPSE x:0pt y:0pt rx:(size-50pt) ry:(size-50pt) combine-mode:diff

    FILL color:$00FFFF
    PIE x:0pt y:0pt rx:(size) ry:(size) x1:(cos(a)*size) y1:(-sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
    CLIP_OFF
}
ELSE {
  AVAL set-count-rows rowcount:"YCvalues"
  AVAL set-count-rows dimension:"Header"

  SET objcount:(rowcount / dimension)
  SET width:(size/(objcount+1))

  AVAL set-default:$FFFFFF background:"Background"
  PEN style:null
  FILL color:(background)
  RECTANGLE x:(-size) y:(-size) w:(size*2.5) h:(size*2)
  PEN w:1

  FOR i from:1 to:(dimension) {
    AVAL set-row:(i) set-col:"Header" n:"Header"
    AVAL set-row:(i) set-col:"Color" c:"Header"
    FILL color:(c)
    RECTANGLE x:(size+10pt) y:(i*20pt) w:10pt h:10pt
    TEXT (n) x:(size+22pt) y:(i*20pt) w:l h:t
  }

  PEN w1 color:$FFFFFF

  FOR r from:0 to:(objcount-1) {
    SET total:0
    SET cum:0

    FOR i from:1 to:(dimension) {
      AVAL set-row:(i+r*dimension) set-col:"Value" h:"YCvalues"
      SET total: (total+(VAL h))
    }

    SET a:(3.142*0/total)

    FILL style:null
    CLIP_ELLIPSE x:0pt y:0pt rx:(size-width) ry:(size-width) combine-mode:diff
    RECTANGLE x:200pt y:-200pt rx:(size) ry:(size)

    FOR i from:1 to:(dimension) {
      AVAL set-row:(i+r*dimension) set-col:"Value" h:"YCvalues"
      AVAL set-row:(i+r*dimension) set-col:"Header" n:"Header"
      AVAL set-row:(i) set-col:"Color" c:"Header"
      IF( (VAL h)>0 ) {
        SET cum:(cum+(VAL h))

        SET b:(3.142*2*cum/total)

        FILL color:(c)
        PIE x:0pt y:0pt rx:(size) ry:(size) x1:(cos(a)*size) y1:(-sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
      }
      IF(i=1)
        AVAL set-row:(i+r*dimension) set-col:"Key" o:"YCvalues"
        TEXT (o) x:(210pt) y:(-size+8pt) w:l h:c
      ENDIF
      SET a:(b)
    }
    CLIP_OFF

    SET size:(size-width)
  }
}
```

Listing 1.10. GraphRep of class "RadialTreeMap"