

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

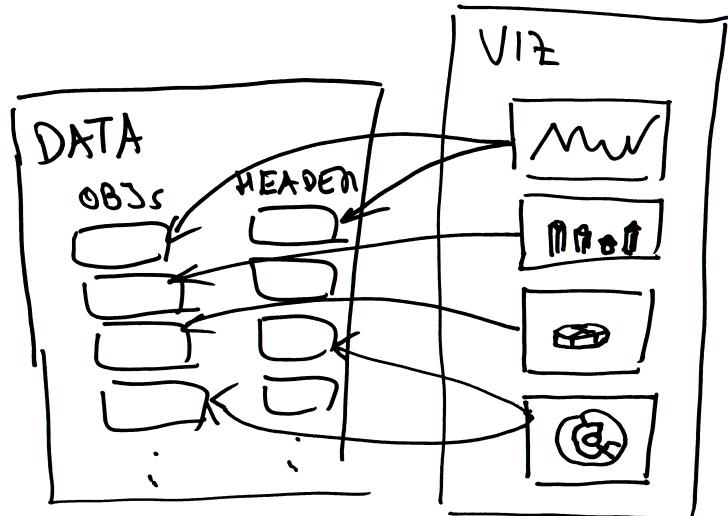


Fig. 1. Concept

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]

		BASIC TYPES OF COMPARISONS				
		Component comparison	Item comparison	Time series comparison	Frequency distribution comparison	Correlation comparison
BASIC CHART TYPES	Pie					
	Bar					
	Column					
	Line					
	Dot					

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

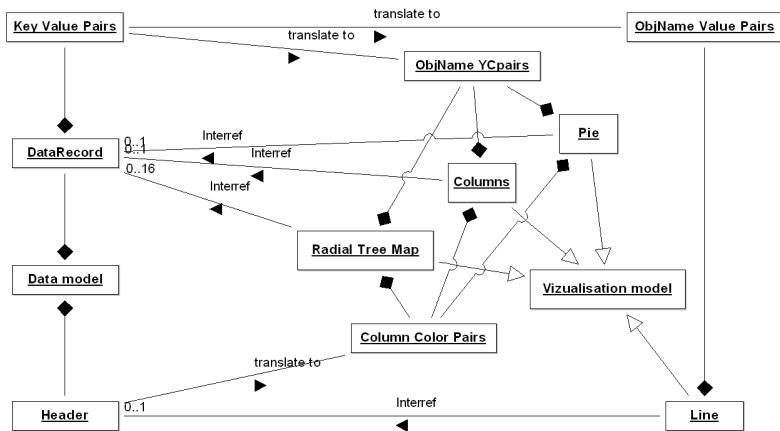


Fig. 3. Meta model

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 or several objects with all columns. Therefore 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 is 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.

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

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

We were using several statistical data from our "Methoden der Datenanalyse" 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 would 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

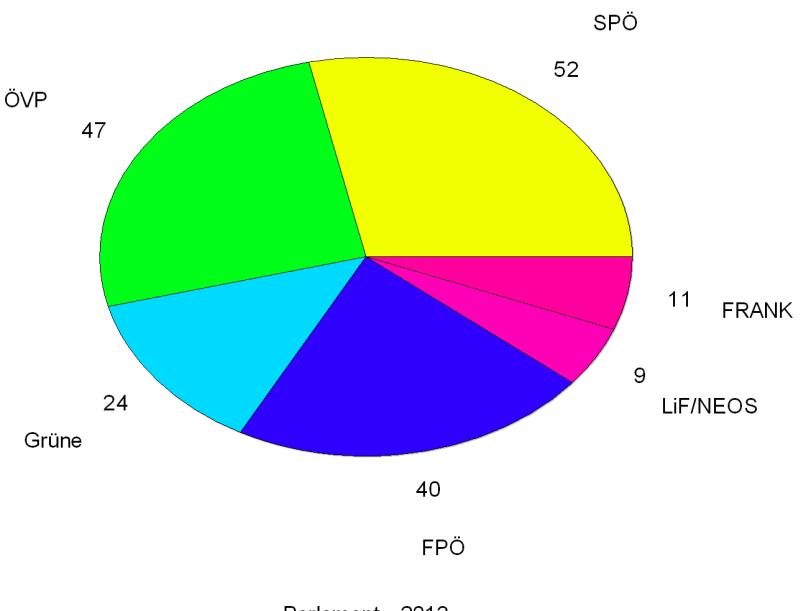


Fig. 5. Pie Chart

The values of all entries 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

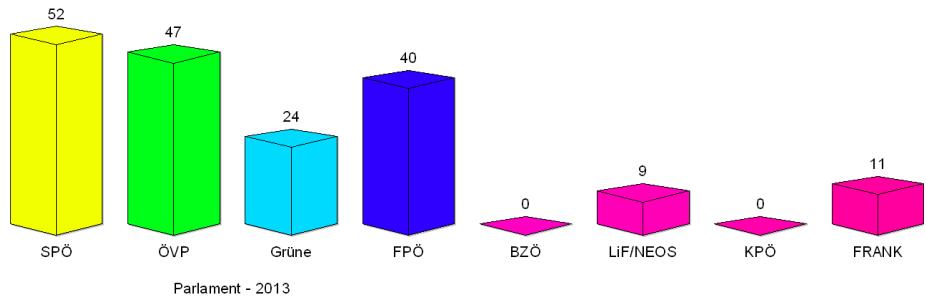


Fig. 6. Columns Chart

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

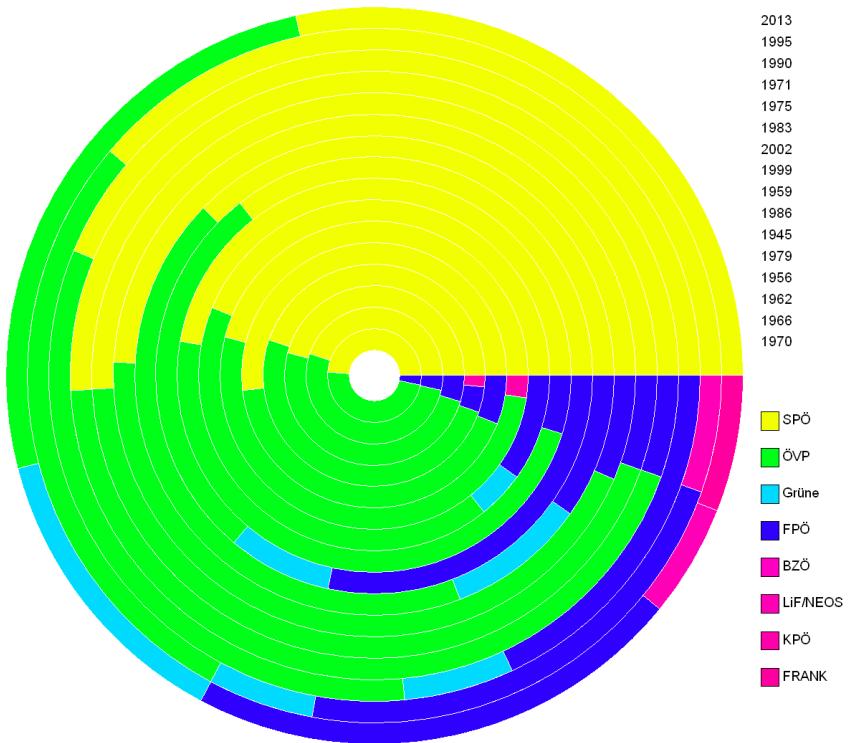


Fig. 7. Radial Tree Map

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".

```
1 REFDOMAIN
2 OBJREF
3   mt:"Data model"
4   c:"DataRecord"
5   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.

```
1 ON_EVENT "AfterEditAttributeValue" {
2   IF( attrtypeid=9 ) {
3     CC "Core" GET_ATTR_VAL objid: (instid) attrname:"External
4       tool coupling"
5     CC "AdoScript" FREAD file:(val)
6     EXECUTE( text )
7 }
```

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 (Radial-TreeMap) 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, GIT 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., korrig. Nachdruck 2003
8. H.-G. Fill, *Visualisation for Semantic Information Systems*, Gabler, Wiesbaden, 2009

A ADOscripts

A.1 csv_import.asc

```
1 #user selects csv-file
2 CC "AdoScript" FILE_DIALOG open filter1:"CSV-files" type1:"*.
  csv" default-ext:"csv"
3 IF (path = "") {
4   CC "AdoScript" ERRORBOX "No file selected!"
5   EXIT
6 }
7 SET csvpath:(path)
8
9 CC "AdoScript" QUERYBOX "First column is an index?" yes-no
10 SET index_endbutton:(endbutton)
11
12 #user request for new model name
13 #SEND "EXEC_NEW_DLG modeltype:\\"Data model\\" show-models" to:\"Modeling" answer:modelid
14
15 FOR sub in: (csvpath) sep: "\\" {
16   SET res: (search(lower(sub), ".csv", 0))
17
18   IF( res > -1 ) {
19     SEND ("EXEC_NEW_DLG modelname:\\" + copy(sub, 0, res) + "\"
      modeltype:\\"Data model\\" show-models") to:\"Modeling"
      answer:modelid
20   }
21 }
22
23 IF (modelid = "") {
24   CC "AdoScript" ERRORBOX "No model created!"
25   EXIT
26 }
27 SET new_modelid: (VAL modelid)
28
29 #csv-file read
30 CC "AdoScript" FREAD file: (csvpath)
31 SET sCsvData: (text)
32
33 #activate the new model
34 CC "Modeling" ACTIVATE_MODEL modelid:(new_modelid)
35
36 CC "Core" GET_CLASS_ID classname: ("DataRecord")
37 SET datarecord_classid: (classid)
38
39 CC "Core" GET_ATTR_ID classid:(datarecord_classid) attrname:
  KeyValuePair"
40 SET keyvaluepair_attrid:(attrid)
41
```

```

42 SET nRow:0
43 SET tHeader: ""
44
45 FOR sRow in: (sCsvData) sep: "\n" {
46   IF( sRow = "" ) {
47     BREAK
48   }
49
50   IF( nRow > 0 ) {
51     #create data new object
52     CC "Core" CREATE_OBJ modelid:(new_modelid) classid: (
53       datarecord_classid)
54     SET datarecord_objid: (objid)
55
56     SET nColumn:0
57     FOR sCell in: (sRow) sep: "," {
58       IF ( nColumn < 1 AND index_endbutton="no" ) {
59         CC "Core" SET_ATTR_VAL objid: (datarecord_objid)
60         attrname:"Name" val:(nRow)
61       }
62       IF ( nColumn < 1 AND index_endbutton="yes" ) {
63         CC "Core" SET_ATTR_VAL objid: (datarecord_objid)
64         attrname:"Name" val:(sCell)
65       }
66       ELSE {
67         CC "Core" ADD_REC_ROW objid: (datarecord_objid) attrid:
68         (keyvaluepair_attrid)
69         SETL nNewRowID: (rowid)
70         CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:"Key"
71         val:(token(tHeader, nColumn, ","))
72
73         CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:
74         :"Value" val:(sCell)
75       }
76       SET nColumn: (nColumn + 1)
77     }
78   }
79   ELSE {
80     SET tHeader:(sRow)
81
82     CC "Core" GET_CLASS_ID classname: ("Header")
83     SET header_classid: (classid)
84
85     SET nColumn:0
86     FOR sCell in: (sRow) sep: "," {
87
88       #create an index if there is no index column
89       IF ( index_endbutton="no" ) {
90         CC "Core" CREATE_OBJ modelid:(new_modelid) classid: (
91           header_classid)

```

```

85      CC "Core" SET_ATTR_VAL objid: (objid) attrname:"Name"
86      val:(sCell)
87      CC "Core" SET_ATTR_VAL objid: (objid) attrname:"Index"
88      val:(nColumn+1)
89      }
90
91      #skip the first column(0) if it is an index
92      IF ( index_endbutton="yes" AND nColumn > 0 ) {
93          CC "Core" CREATE_OBJ modelid:(new_modelid) classid:(
94              header_classid)
95          CC "Core" SET_ATTR_VAL objid: (objid) attrname:"Name"
96          val:(sCell)
97          CC "Core" SET_ATTR_VAL objid: (objid) attrname:"Index"
98          val:(nColumn)
99          }
100         SET nColumn: (nColumn + 1)
101     }
102     SET nRow: (nRow + 1)
103   }
104 CC "Modeling" CLOSE modelid:(new_modelid) quiet save

```

Listing 1.4. csv_import.asc

A.2 map_column.asc

```
1 CC "Modeling" GET_SELECTED
2 IF (objids = "") {
3     CC "AdoScript" ERRORBOX "No object has been selected!"
4     EXIT
5 }
6
7 # from the list of selected objects , extract the first
8 #      objectid
9 SET selected:(VAL token(objids,0," "))
10
11 # get the class of the selected object
12 CC "Core" GET_CLASS_ID objid:(selected)
13 SET linechart_classid:(classid)
14
15 CC "Core" GET_ATTR_ID classid:(linechart_classid) attrname:"Yvalues"
16 SET yvalues_attrid:(attrid)
17
18 #delete all rows of the record attribute
19 CC "Core" GET_REC_ATTR_ROW_COUNT objid:(selected) attrid:(yvalues_attrid)
20 FOR d from:1 to:(count) {
21     CC "Core" GET_REC_ATTR_ROW_ID objid:(selected) attrid:(yvalues_attrid) index:1
22     CC "Core" REMOVE_REC_ROW objid:(selected) attrid:(yvalues_attrid) rowid:(rowid)
23 }
24
25 CC "Core" GET_ATTR_ID classid:(linechart_classid) attrname:"Data model"
26 #SET datamodel_ir_attrid:(attrid)
27
28 CC "Core" GET_INTERREF objid:(selected) attrid:(attrid) index:(0)
29 SET data_modelid:(tmodelid)
30
31 CC "Core" LOAD_MODEL modelid:(data_modelid)
32 #CC "Core" GET_ATTR_ID classid:(linechart_classid) attrname:"Data model"
33 #SET datamodel_ir_attrid:(attrid)
34
35 CC "Core" GET_INTERREF objid:(selected) attrid:(attrid) index:(0)
36 SET header_objid:(tobjid)
37
38 CC "Core" GET_ATTR_VAL objid:(header_objid) attrname:"Index"
39 SET index_val:(val)
```

```

40
41 CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:( data_modelid )
      classname:"DataRecord"
42
43 SET x: 0
44 FOR datarecord_objid in:(objids) {
45   CC "Core" GET_CLASS_ID classname: ("DataRecord")
46   CC "Core" GET_ATTR_ID classid: (classid) attrname:""
        KeyValuePair"
47
48   CC "Core" GET_ATTR_VAL objid:(VAL datarecord_objid) attrname
        :"Name"
49   SET objval:( val)
50
51   CC "Core" GET_REC_ATTR_ROW_ID objid:(VAL datarecord_objid)
        attrid:( attrid) index:( index_val)
52
53   CC "Core" GET_ATTR_VAL objid:( rowid) attrname:" Value"
54
55   CC "Core" ADD_REC_ROW objid: (selected) attrid: (
        yvalues_attrid)
56   SETL nNewRowID: (rowid)
57   CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:"Key" val
        :( objval)
58   CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:" Value"
        val:( val)
59
60   SET x:(x+1)
61
62   IF( x>255 ) {
63     BREAK
64   }
65 }
66 CC "Core" DISCARD_MODEL modelid:( data_modelid )

```

Listing 1.5. map_column.asc

A.3 map_rows.asc

```
1 CC "Modeling" GET_SELECTED
2 IF (objids = "") {
3     CC "AdoScript" ERRORBOX "No object has been selected!"
4     EXIT
5 }
6
7 # from the list of selected objects , extract the first
8 #      objectid
9 SET selected:(VAL token(objids,0," "))
10
11 # get the class of the selected object
12 CC "Core" GET_CLASS_ID objid:(selected)
13 SET column_classid:(classid)
14
15 CC "Core" GET_ATTR_ID classid:(column_classid) attrname:"Header"
16 SET header_attrid:(attrid)
17
18 #delete all rows of the Header record attribute
19 CC "Core" GET_REC_ATTR_ROW_COUNT objid:(selected) attrid:(header_attrid)
20 FOR d from:1 to:(count) {
21     CC "Core" GET_REC_ATTR_ROW_ID objid:(selected) attrid:(header_attrid) index:1
22     CC "Core" REMOVE_REC_ROW objid:(selected) attrid:(header_attrid) rowid:(rowid)
23 }
24
25 CC "Core" GET_ATTR_ID classid:(column_classid) attrname:"YCvalues"
26 SET ycvalues_attrid:(attrid)
27
28 #delete all rows of YCvalues record attribute
29 CC "Core" GET_REC_ATTR_ROW_COUNT objid:(selected) attrid:(ycvalues_attrid)
30 FOR d from:1 to:(count) {
31     CC "Core" GET_REC_ATTR_ROW_ID objid:(selected) attrid:(ycvalues_attrid) index:1
32     CC "Core" REMOVE_REC_ROW objid:(selected) attrid:(ycvalues_attrid) rowid:(rowid)
33 }
34
35 CC "Core" GET_ATTR_ID classid:(column_classid) attrname:"Data model"
36 SET datamodel_ir_attrid:(attrid)
37 CC "Core" GET_INTERREF objid:(selected) attrid:(attrid) index :(0)
```

```

38 SET data_modelid:( tmodelid )
39
40 CC "Core" LOAD_MODEL modelid:( data_modelid )
41
42 CC "Core" GET_ALL_OBJS_OF_CLASSNAME modelid:( data_modelid )
43   classname:"Header"
44 SET count:0
45 FOR header_objid in:( objids ) {
46   SET count:( count+1 )
47 }
48 SET i:0
49 FOR header_objid in:( objids ) {
50   CC "Core" GET_ATTR_VAL objid:(VAL header_objid) attrname:"Name"
51   SET header:( val )
52
53   CC "Core" ADD_REC_ROW objid: ( selected ) attrid: (
54     header_attrid )
55   CC "Core" SET_ATTR_VAL objid: ( rowid ) attrname:"Header"
56   val:( header )
57
58   SET h: ( i*6/count )
59
60   SET s:1
61   SET v:255
62
63   SET region: (INT h)
64   SET ff: (h - region)
65   SET p: (v * (1.0 - s))
66   SET q: (v * (1.0 - (s * ff)))
67   SET t: (v * (1.0 - (s * (1.0 - ff))))
68
69   IF( region = 0 ) {
70     SET color:( rgbval(v,t,p))
71   }
72   ELSE {
73     IF( region = 1 ) {
74       SET color:( rgbval(q,v,p))
75     }
76     ELSE {
77       IF( region = 2 ) {
78         SET color:( rgbval(p,v,t))
79       }
80       ELSE {
81         IF( region = 3 ) {
82           SET color:( rgbval(p,q,v))
83         }

```

```

84         SET color:(rgbval(t,p,v))
85     }
86     ELSE {
87         SET color:(rgbval(v,p,q))
88     }
89 }
90 }
91 }
92 }
93 CC "Core" SET_ATTR_VAL objid: (rowid) attrname:"Color" val
94 :(color)
95 SET i:(i+1)
96 }
97 CC "Core" GET_INTERREF_COUNT objid:(selected) attrid:(
98     datamodel_ir_attrid)
99 SET datamodelir_count:(count)
100 FOR ir from:0 to:(datamodelir_count-1) {
101     CC "Core" GET_INTERREF objid:(selected) attrid:(
102         datamodel_ir_attrid) index:(ir)
103     SET data_modelid:(tmodelid)
104     SET datarecord_objid:(tobjid)
105     CC "Core" GET_CLASS_ID classname: ("DataRecord")
106     CC "Core" GET_ATTR_ID classid: (classid) attrname:""
107         KeyValuePair"
108     CC "Core" GET_ATTR_VAL objid:(datarecord_objid) attrname:""
109         Name"
110     SET objval:(val)
111     CC "Core" GET_REC_ATTR_ROW_COUNT objid:(datarecord_objid)
112         attrid:(attrid)
113     FOR i from:1 to:(count) {
114         CC "Core" GET_REC_ATTR_ROW_ID objid:(datarecord_objid)
115             attrid:(attrid) index:(i)
116         CC "Core" GET_ATTR_VAL objid:(rowid) attrname:"Key"
117         SET keyval:(val)
118         CC "Core" GET_ATTR_VAL objid:(rowid) attrname:"Value"
119         SET valueval:(val)
120     CC "Core" ADD_REC_ROW objid: (selected) attrid: (
121         ycvalues_attrid)
122     SETL nNewRowID: (rowid)
123     CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:"Key"
124         val:(objval)

```

```
124     CC "Core" SET_ATTR_VAL objid: (nNewRowID) attrname:"Value"  
125         val:( valueeval)  
126     }  
127  
128 CC "Core" DISCARD_MODEL modelid:( data_modelid)
```

Listing 1.6. map_rows.asc

B GRAPHREPs

B.1 GraphRep of class "LineChart"

```

1 GRAPHREP sizing: keep-aspect-ratio
2 SHADOW off
3
4 AVAL set-format:"%m" dm:"Data model"
5 AVAL set-format:"%o" dr:"Data model"
6
7 AVAL set-default: black bgd:"Background"
8 AVAL set-default:$00FF00 fgd:"Line"
9 AVAL set-count-rows rowcount:"Yvalues"
10
11 FILL color:(bgd)
12 RECTANGLE x:-320pt y:-180pt w:640pt h:360pt
13
14 FONT "Arial" h:16pt
15
16 IF(dm = "") {
17   PEN w:1 color:(fgd)
18   LINE x1:(-320pt) y1:(0pt) x2:(-200pt) y2:(-110pt)
19   LINE x1:(-200pt) y1:(-110pt) x2:(-100pt) y2:(180pt)
20   LINE x1:(-100pt) y1:(180pt) x2:(0pt) y2:(-40pt)
21   LINE x1:(-0pt) y1:(-40pt) x2:(100pt) y2:(60pt)
22   LINE x1:(100pt) y1:(60pt) x2:(200pt) y2:(-180pt)
23   LINE x1:(200pt) y1:(-180pt) x2:(320pt) y2:(-100pt)
24 }
25 ELSE {
26   TEXT (dm+" - "+dr) x:-320pt y:-190pt w:1 h:b
27   AVAL set-default:"0" reverse:"Reverse"
28
29   SET high:0
30   SET low:0
31
32   FOR i from:1 to:(rowcount) {
33     AVAL set-row:(i) set-col:"Value" y:"Yvalues"
34
35     IF( (VAL y)>high )
36       SET high:(VAL y)
37     ENDIF
38
39     IF( low=0 OR (VAL y)<low )
40       SET low:(VAL y)
41     ENDIF
42   }
43
44   TEXT (high) x:330pt y:-180pt w:1 h:t
45   TEXT (low) x:330pt y:180pt w:1 h:b
46

```

```

47
48   SET y_from:0
49
50   PEN w:1 color:(fgd)
51
52   SET xstep:(640pt/(rowcount-1))
53
54   FOR i from:0 to:(rowcount-1) {
55     IF( reverse=="1" )
56       AVAL set-row:(rowcount-i) set-col:"Value" y:"Yvalues"
57     ELSE
58       AVAL set-row:(i+1) set-col:"Value" y:"Yvalues"
59     ENDIF
60
61     SET y_to:(VAL y)
62
63     IF( i>0 )
64       LINE x1:((i-1)*xstep-320pt) y1:(180pt-(y_from-low)/(high
65       -low)*360pt) x2:(i*xstep-320pt) y2:(180pt-(y_to-low)/(high
66       -low)*360pt)
67     ENDIF
68
69     SET y_from:(y_to)
70   }
71   IF( reverse=="1" )
72     AVAL set-row:(rowcount) set-col:"Key" l:"Yvalues"
73     AVAL set-row:(1) set-col:"Key" r:"Yvalues"
74   ELSE
75     AVAL set-row:(rowcount) set-col:"Key" r:"Yvalues"
76     AVAL set-row:(1) set-col:"Key" l:"Yvalues"
77   ENDIF
78
79   TEXT (l) x:-320pt y:190pt w:l h:t
80   TEXT (r) x:320pt y:190pt w:r h:t
81 }
```

Listing 1.7. GraphRep of class "LineChart"

B.2 GraphRep of class "PieChart"

```

1 GRAPHREP sizing: keep-aspect-ratio
2
3 AVAL set-format:"%m" dm:"Data model"
4 AVAL set-format:"%o" dr:"Data model"
5
6 AVAL set-count-rows rowcount:"YCvalues"
7
8 FONT "Arial" h:16pt color:black
9
10 SET xsize:(120pt)
11 SET ysize:(90pt)
12
13 IF(dm == "") {
14     SET a:(3.142*0/100)
15     SET b:(3.142*2*40/100)
16
17     FILL color:$FFFF00
18     PIE x:0pt y:0pt rx:(xsize) ry:(ysize) x1:(cos(a)*xsize) y1
19         :(-sin(a)*ysize) x2:(cos(b)*xsize) y2:(-sin(b)*ysize)
20
21     SET a:(3.142*2*40/100)
22     SET b:(3.142*2*70/100)
23
24     FILL color:$00FFFF
25     PIE x:0pt y:0pt rx:(xsize) ry:(ysize) x1:(cos(a)*xsize) y1
26         :(-sin(a)*ysize) x2:(cos(b)*xsize) y2:(-sin(b)*ysize)
27     SET a:(3.142*2*70/100)
28     SET b:(3.142*2*100/100)
29
30     FILL color:$FF00FF
31     PIE x:0pt y:0pt rx:(xsize) ry:(ysize) x1:(cos(a)*xsize) y1
32         :(-sin(a)*ysize) x2:(cos(b)*xsize) y2:(-sin(b)*ysize)
33 }
34 ELSE {
35     TEXT (dm+" - "+dr) x:0pt y:(ysize*1.8) w:c h:c
36
37     SET total:0
38     SET cum:0
39
40     FOR i from:1 to:(rowcount) {
41         AVAL set-row:(i) set-col:"Value" h:"YCvalues"
42         SET total: (total+(VAL h))
43     }
44
45     SET a:(3.142*0/total)
46
47     FOR i from:1 to:(rowcount) {
48         AVAL set-row:(i) set-col:"Value" h:"YCvalues"
49     }
50 }
```

```

46   AVAL set-row:( i ) set-col:"Header" n:"Header"
47   AVAL set-row:( i ) set-col:"Color" col:"Header"
48
49   IF( (VAL h)>0 ) {
50     SET cumtext:(cum+(VAL h)/2)
51     SET cum:(cum+(VAL h))
52
53     SET b:(3.142*2*cum/total)
54     SET c:(3.142*2*cumtext/total)
55
56     FILL color:( col )
57     PIE x:0pt y:0pt rx:( xsize ) ry:( ysize ) x1:( cos(a)*xsize )
58     y1:(-sin(a)*ysize) x2:( cos(b)*xsize ) y2:(-sin(b)*ysize)
59     TEXT (copy(h,0,4)) x:(cos(c)*xsize*1.2) y:(-sin(c)*
60     ysize*1.2) w:c h:c
61     TEXT (n) x:(cos(c)*xsize*1.5) y:(-sin(c)*ysize*1.5) w:c
62     h:c
63   }
64 }
```

Listing 1.8. GraphRep of class "PieChart"

B.3 GraphRep of class "ColumnChart"

```

1 GRAPHREP sizing: keep-aspect-ratio
2
3 AVAL set-format:"%m" dm:"Data model"
4 AVAL set-format:"%o" dr:"Data model"
5
6 SET xofs:(0 pt)
7 SET yofs:(0 pt)
8
9 FONT "Arial" h:16 pt color:black
10
11 IF(dm = "") {
12     SET height:(-60 pt)
13     SET c:$FFFF07
14     SET xstep:(0 pt)
15
16 #top
17     FILL color:(c)
18     POLYGON 4 x1:(xstep+xofs) y1:(height-10pt-yofs) x2:(xstep-25
19         pt+xofs) y2:(height-6pt-yofs) x3:(xstep+xofs) y3:(height-
20         yofs) x4:(xstep+25pt+xofs) y4:(height-6pt-yofs)
21
22 #left
23     FILL color:(c)
24     POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(
25         height-yofs) x3:(xstep-25pt+xofs) y3:(height-6pt-yofs) x4
26         :(xstep-25pt+xofs) y4:(-6pt-yofs)
27
28 #right
29     FILL color:(c)
30     POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(
31         height-yofs) x3:(xstep+25pt+xofs) y3:(height-6pt-yofs) x4
32         :(xstep+25pt+xofs) y4:(-6pt-yofs)
33
34     SET height:(-100 pt)
35     SET c:$07FFFF
36     SET xstep:(64 pt)
37
38 #top
39     FILL color:(c)
40     POLYGON 4 x1:(xstep+xofs) y1:(height-10pt-yofs) x2:(xstep-25
41         pt+xofs) y2:(height-6pt-yofs) x3:(xstep+xofs) y3:(height-
42         yofs) x4:(xstep+25pt+xofs) y4:(height-6pt-yofs)
43
44 #left
45     FILL color:(c)
46     POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(
47         height-yofs) x3:(xstep-25pt+xofs) y3:(height-6pt-yofs) x4
48         :(xstep-25pt+xofs) y4:(-6pt-yofs)

```

```

39 #right
40 FILL color:(c)
41 POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(height-yofs)
42 x3:(xstep+25pt+xofs) y3:(height-6pt-yofs) x4
43 :(xstep+25pt+xofs) y4:(-6pt-yofs)
44 SET height:(-80pt)
45 SET c:$FF07FF
46 SET xstep:(128pt)
47
48 #top
49 FILL color:(c)
50 POLYGON 4 x1:(xstep+xofs) y1:(height-10pt-yofs) x2:(xstep-25
51 pt+xofs) y2:(height-6pt-yofs) x3:(xstep+xofs) y3:(height-
52 yofs) x4:(xstep+25pt+xofs) y4:(height-6pt-yofs)
53
54 #left
55 FILL color:(c)
56 POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(height-yofs)
57 x3:(xstep-25pt+xofs) y3:(height-6pt-yofs) x4
58 :(xstep-25pt+xofs) y4:(-6pt-yofs)
59 }
60 ELSE {
61 AVAL set-count-rows rowcount:"YCvalues"
62
63 TEXT (dm + " - " + dr) x:(rowcount*12pt) y:(30pt-yofs) w:c h
64 :c
65 SET max:0
66
67 FOR i from:0 to:(rowcount-1) {
68 AVAL set-row:(i+1) set-col:"Value" y:"YCvalues"
69 IF( VAL y > max )
70 SET max:(VAL y)
71 ENDIF
72 }
73
74 FOR i from:0 to:(rowcount-1) {
75 AVAL set-row:(i+1) set-col:"Value" y:"YCvalues"
76 AVAL set-row:(i+1) set-col:"Header" n:"Header"
77 AVAL set-row:(i+1) set-col:"Color" c:"Header"
78
79 SET height:(-(VAL y)/max*240pt)

```

```

80
81     SET xstep:( i *64pt )
82
83     #top
84     FILL color:(c)
85     POLYGON 4 x1:(xstep+xofs) y1:(height-10pt-yofs) x2:(xstep
86             -25pt+xofs) y2:(height-6pt-yofs) x3:(xstep+xofs) y3:(
87             height-yofs) x4:(xstep+25pt+xofs) y4:(height-6pt-yofs)
88
89     #left
90     FILL color:(c)
91     POLYGON 4 x1:(xstep+xofs) y1:(-yofs) x2:(xstep+xofs) y2:(
92             height-yofs) x3:(xstep-25pt+xofs) y3:(height-6pt-yofs) x4
93             :(xstep-25pt+xofs) y4:(-6pt-yofs)
94
95     TEXT (y) x:(xstep+xofs) y:(height-16pt-yofs) w:c h:c
96     TEXT (n) x:(xstep+xofs) y:(10pt-yofs) w:c h:c
97 }
98 }
```

Listing 1.9. GraphRep of class "ColumnChart"

B.4 GraphRep of class "RadialTreeMap"

```

1 GRAPHREP sizing: keep-aspect-ratio
2 SHADOW off
3
4 AVAL set-format:"%o" dr:"Data model"
5
6 AVAL col:"fontcolor"
7 FONT "Arial" h:16pt color:black
8
9 SET size:(200pt)
10
11 ELLIPSE x:0pt y:0pt rx:(size) ry:(size)
12
13 IF(dr == "") {
14     SET size:200pt
15     SET a:(3.142*2*0/100)
16     SET b:(3.142*2*40/100)
17
18     FILL style:null
19     PEN w1
20     CLIP_ELLIPSE x:0pt y:0pt rx:(size*2/3) ry:(size*2/3)
combine-mode:diff
21
22     FILL color:($FFFF00)
23     PIE x:0pt y:0pt rx:(size) ry:(size) x1:(cos(a)*size) y1:(-
sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
24     CLIP_OFF
25
26     SET a:(3.142*2*40/100)
27     SET b:(3.142*2*70/100)
28
29     FILL style:null
30     CLIP_ELLIPSE x:0pt y:0pt rx:(size*2/3) ry:(size*2/3)
combine-mode:diff
31
32     FILL color:$FF00FF
33     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)
34     CLIP_OFF
35
36     SET a:(3.142*2*70/100)
37     SET b:(3.142*2*100/100)
38
39     FILL style:null
40     CLIP_ELLIPSE x:0pt y:0pt rx:(size*2/3) ry:(size*2/3)
combine-mode:diff
41
42     FILL color:$00FFFF

```

```

43   PIE x:0 pt y:0 pt rx:( size) ry:( size) x1:(cos(a)*size) y1:(-
44     sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
45   CLIP_OFF
46
47   SET size:150 pt
48   SET a:(3.142*2*0/100)
49   SET b:(3.142*2*30/100)
50
51   FILL style:null
52   CLIP_ELLIPSE x:0 pt y:0 pt rx:( size-50pt) ry:( size-50pt)
53   combine-mode:diff
54
55   FILL color:$FFFF00
56   PIE x:0 pt y:0 pt rx:( size) ry:( size) x1:(cos(a)*size) y1:(-
57     sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
58   CLIP_OFF
59
60   SET a:(3.142*2*30/100)
61   SET b:(3.142*2*80/100)
62
63   FILL style:null
64   CLIP_ELLIPSE x:0 pt y:0 pt rx:( size-50pt) ry:( size-50pt)
65   combine-mode:diff
66
67   FILL color:$FF00FF
68   PIE x:0 pt y:0 pt rx:( size) ry:( size) x1:(cos(a)*size) y1:(-
69     sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
70   CLIP_OFF
71
72   SET a:(3.142*2*80/100)
73   SET b:(3.142*2*100/100)
74
75   FILL style:null
76   CLIP_ELLIPSE x:0 pt y:0 pt rx:( size-50pt) ry:( size-50pt)
77   combine-mode:diff
78
79   FILL color:$00FFFF
80   PIE x:0 pt y:0 pt rx:( size) ry:( size) x1:(cos(a)*size) y1:(-
81     sin(a)*size) x2:(cos(b)*size) y2:(-sin(b)*size)
82   CLIP_OFF
83
84   SET size:100 pt
85   SET a:(3.142*2*0/100)
86   SET b:(3.142*2*20/100)
87
88   FILL style:null
89   CLIP_ELLIPSE x:0 pt y:0 pt rx:( size-50pt) ry:( size-50pt)
90   combine-mode:diff
91
92   FILL color:($FFFF00)

```

```

85   PIE x:0 pt y:0 pt rx:( size) ry:( size) x1:( cos(a)*size) y1:(-
86     sin(a)*size) x2:( cos(b)*size) y2:(-sin(b)*size)
87     CLIP_OFF
88
89   SET a:(3.142*2*20/100)
90   SET b:(3.142*2*50/100)
91
92   FILL style:null
93   CLIP_ELLIPSE x:0 pt y:0 pt rx:( size-50pt) ry:( size-50pt)
94   combine-mode:diff
95
96   FILL color:$FF00FF
97   PIE x:0 pt y:0 pt rx:( size) ry:( size) x1:( cos(a)*size) y1:(-
98     sin(a)*size) x2:( cos(b)*size) y2:(-sin(b)*size)
99     CLIP_OFF
100
101  SET a:(3.142*2*50/100)
102  SET b:(3.142*2*100/100)
103
104  FILL style:null
105  CLIP_ELLIPSE x:0 pt y:0 pt rx:( size-50pt) ry:( size-50pt)
106  combine-mode:diff
107
108 }ELSE {
109   AVAL set-count-rows rowcount:"YCvalues"
110   AVAL set-count-rows dimension:"Header"
111
112   SET objcount:(rowcount / dimension)
113   SET width:(size/(objcount+1))
114
115   AVAL set-default:$FFFFFF background:"Background"
116   PEN style:null
117   FILL color:(background)
118   RECTANGLE x:(-size) y:(-size) w:(size*2.5) h:(size*2)
119   PEN w:1
120
121   FOR i from:1 to:(dimension) {
122     AVAL set-row:(i) set-col:"Header" n:"Header"
123     AVAL set-row:(i) set-col:"Color" c:"Header"
124     FILL color:(c)
125     RECTANGLE x:(size+10pt) y:(i*20pt) w:10pt h:10pt
126     TEXT (n) x:(size+22pt) y:(i*20pt) w:1 h:t
127   }
128
129   PEN w1 color:$FFFFFF

```

```

130
131 FOR r from:0 to:( objcount -1) {
132   SET total:0
133   SET cum:0
134
135   FOR i from:1 to:( dimension ) {
136     AVAL set-row:( i+r*dimension ) set-col:" Value" h:" YCvalues
137   "
138     SET total: ( total+(VAL h) )
139   }
140
141   SET a:(3.142*0/ total)
142
143   FILL style:null
144   CLIP_ELLIPSE x:0 pt y:0 pt rx:( size-width ) ry:( size-width )
145   combine-mode: diff
146   RECTANGLE x:200 pt y:-200 pt rx:( size ) ry:( size )
147
148   FOR i from:1 to:( dimension ) {
149     AVAL set-row:( i+r*dimension ) set-col:" Value" h:" YCvalues
150   "
151     AVAL set-row:( i+r*dimension ) set-col:" Header" n:" Header"
152     AVAL set-row:( i ) set-col:" Color" c:" Header"
153     IF( (VAL h)>0 ) {
154       SET cum:( cum+(VAL h) )
155
156       SET b:(3.142*2*cum/ total)
157
158       FILL color:(c)
159       PIE x:0 pt y:0 pt rx:( size ) ry:( size ) x1:( cos(a)*size )
160       y1:( -sin(a)*size ) x2:( cos(b)*size ) y2:( -sin(b)*size )
161     }
162     IF( i=1 )
163       AVAL set-row:( i+r*dimension ) set-col:" Key" o:" YCvalues
164   "
165
166       TEXT (o) x:(210 pt) y:(- size+8pt) w:l h:c
167     ENDIF
168     SET a:( b)
169   }
170   CLIP_OFF
171
172   SET size:( size-width )
173 }

```

Listing 1.10. GraphRep of class "RadialTreeMap"