

Übung 6 – Mensch-Maschine-Interaktion 2

Exercise 5: Information Visualization: Tree Map Visualization

Tasks

- Find and describe possible solutions to visualize a hierarchical graph by a compact representation on a given area. Present a sketch for each solution.
- Write a Java application which loads GraphML files and visualizes the graph as a tree map. The size of the area should be as large as the number of nodes in the subtree in relation to the overall number of nodes.

A tree map visualizes a tree by representing each node by an area. If a node contains subnodes, its area is divided into subareas. You can find an example at:

http://www.peets.com/selector_coffee/coffee_selector.asp

You can find more information about tree maps at:

<ftp://ftp.cs.umd.edu/pub/hcil/Reports-Abstracts-Bibliography/2001-06html/2001-06.pdf>

For the representation of the graphs GraphML (Graph Markup Language,

<http://graphml.graphdrawing.org/>) is used, an XML based format. It describes a graph by nodes and edges. Each edge connects two nodes. Example:

```
<node id="n0"/>
<node id="n1"/>
<edge id="e0" source="n0" target="n1"/>
```

For processing the GraphML graphs use the *gravisto Graph Visualization Toolkit*

(<http://www.gravisto.org>). It supports loading of GraphML files and provides Java classes representing the nodes and edges. For loading a GraphML file create a new GraphMLReader and a new graph (adjacent) and load the file as follows:

```
Graph graph = new AdjListGraph();
try {
    xmlReader.read(fileName, graph);
}
catch (IOException e) {
}
```

You get the first node of the graph (which is the root node) with:

```
root = (Node)graph.getNodes().get(0);
```

Important methods of class Node:

```
// returns neighbours connected by outgoing edges:
public Collection getOutNeighbors()
// returns the number of outgoing (or undirected) edges:
public int getOutDegree()
```

On the exercises website

http://www.medien.ifi.lmu.de/fileadmin/mimuc/mmi2_ss05/uebung/uebung6-tree.zip

you can find a template for your Java class (TreeMapTemplate.java), four example tree files (tree0x.xml) and the gravisto.jar library file.

Submission

- Submission is by email to mmi2.2005@hcilab.org.
- Please use a zip file attachment named `uebung6-gruppeN.zip` containing a PDF file for task a) and java and class files for task b)
- The deadline for submission is **Friday, June 17th 2005, 8 a.m.**