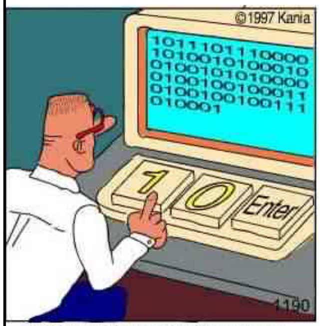
Software Visualization



Real programmers code in binary.

Lecture
WS 02/03

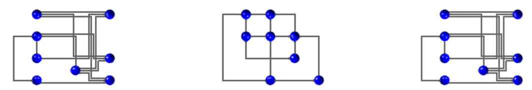
Graph Drawing

Graph

- Widely used mathematical structure to describe relationships between objects
- Objects are called **nodes**
- Relationships are called **edges**
- Mathematical notation: G=(V,E)

Graph Drawing

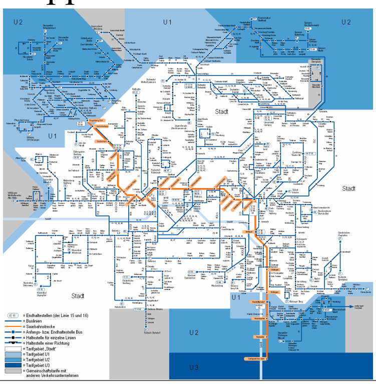
- Art to draw a graph such that the relationships between the objects are easily understood by looking at the picture
- Nodes are drawn as circles or rectangles
- Edges are drawn as curves



Hand made graph drawing **Optimized drawing** **Animation of transformation**

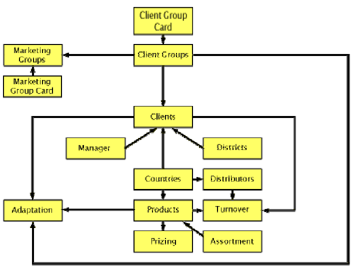
Applications

- Bus map



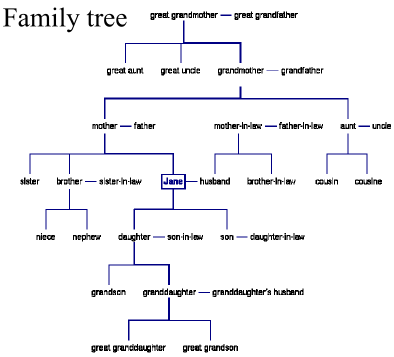
Applications

- Entity-Relationship Diagram



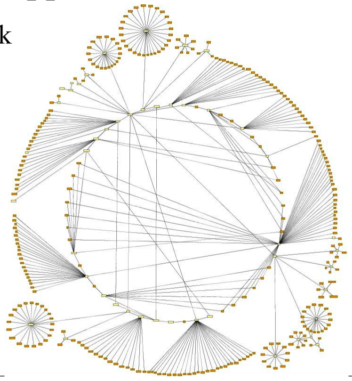
Applications

- Family tree



Applications

- Social network

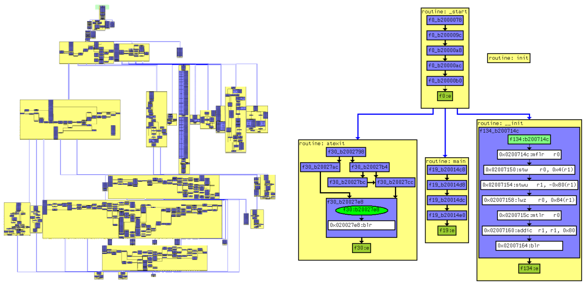


Applications

- Word graphs
- VLSI design
- Work flow diagrams
- Petri nets
- Finite Automata
- many more ...

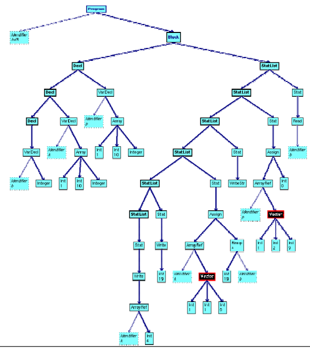
Applications in SV

- Control flow graph (aiCall)



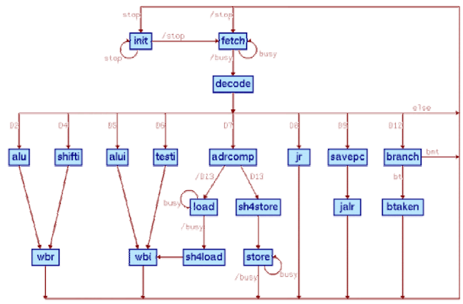
Applications in SV

- Syntax tree



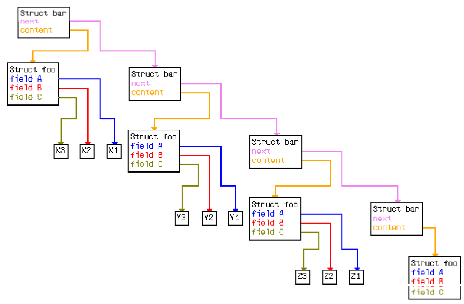
Applications in SV

- Finite state diagram



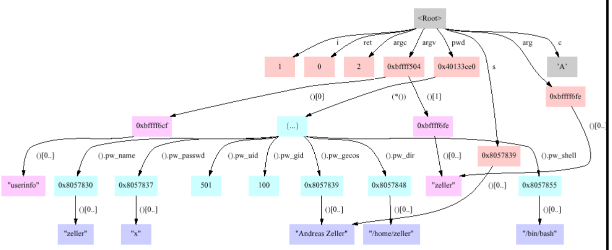
Applications in SV

- Data structure



Applications in SV

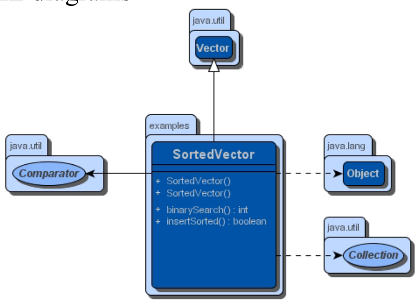
- Memory graphs



Lecture: Software Visualization, WS02/03 © Carsten Görg, Universität des Saarlandes

Applications in SV

- UML diagrams

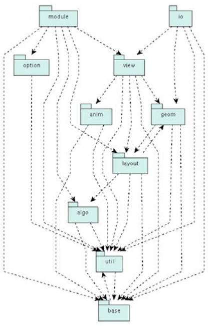


yDoc: Javadoc extension with automatic diagramming support

Lecture: Software Visualization, WS02/03 © Carsten Görg, Universität des Saarlandes

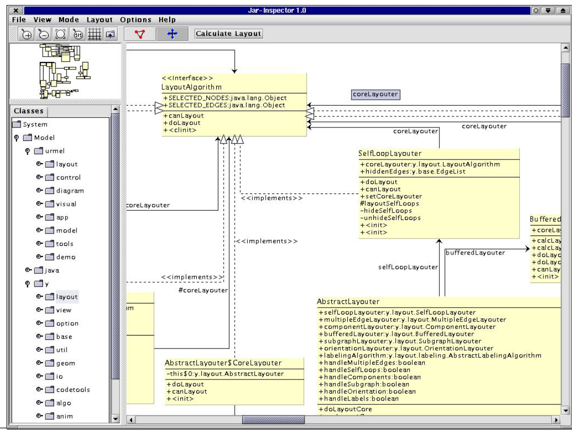
Applications in SV

- UML diagrams



A package hierarchy

Lecture: Software Visualization, WS02/03 © Carsten Görg, Universität des Saarlandes



Lecture: Software Visualization, WS02/03 © Carsten Görg, Universität des Saarlandes

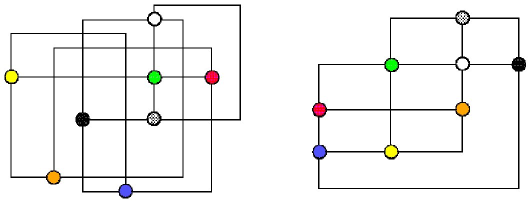
Aesthetics Criteria

- Crossing minimization
- Bend minimization
- Area minimization (homogenous density)
- Angle maximization
- Length minimization
- Symmetries (reflected in layout)
- Clustering (large graphs)

Lecture: Software Visualization, WS02/03 © Carsten Görg, Universität des Saarlandes

Crossing minimization

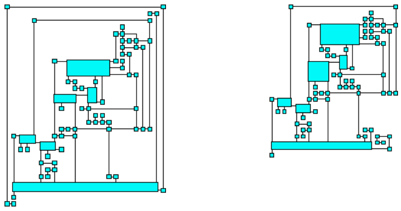
- Graphs that can be drawn without crossings are called planar graphs
- If a graph is not planar, it should be drawn with as few crossings as possible



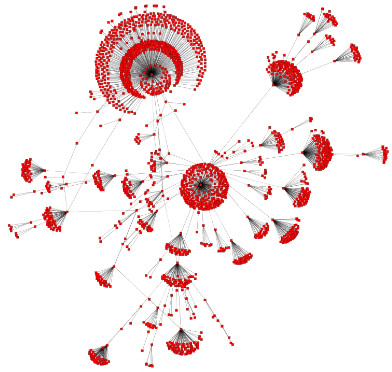
Lecture: Software Visualization, WS02/03 © Carsten Görg, Universität des Saarlandes

Compaction

- Optimize the area of the drawing, the total edge length and the length of the longest edge

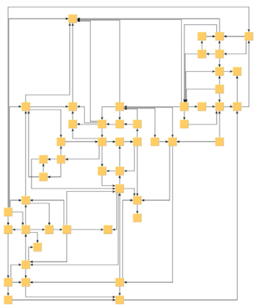


Clustered network



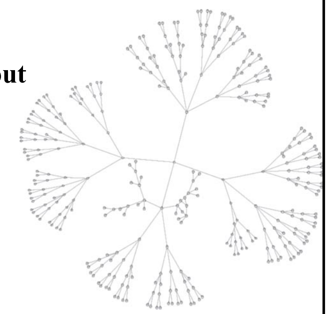
Algorithm Approaches

- **Orthogonal layout**
- Force-directed layout
- Hierarchic layout
- Tree layout
- Circular layout



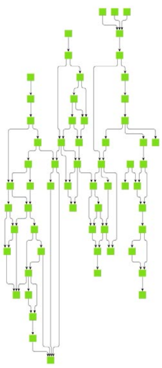
Algorithm Approaches

- Orthogonal layout
- **Force-directed layout**
- Hierarchic layout
- Tree layout
- Circular layout



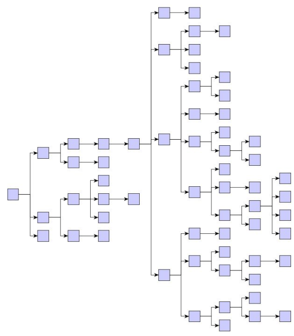
Algorithm Approaches

- Orthogonal layout
- Force-directed layout
- **Hierarchic layout**
- Tree layout
- Circular layout



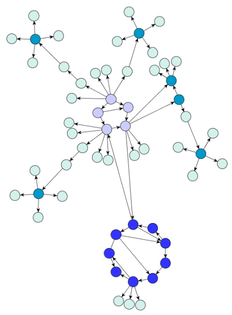
Algorithm Approaches

- Orthogonal layout
- Force-directed layout
- Hierarchic layout
- **Tree layout**
- Circular layout



Algorithm Approaches

- Orthogonal layout
- Force-directed layout
- Hierarchic layout
- Tree layout
- **Circular layout**



Force-directed layout

- Spring embedder
- Magnetic fields
  - parallel
  - polar
  - orthogonal
- Simulated annealing

Force-directed layout

- Algorithm

Initialization

Iteration:

compute forces

move nodes to new positions

anneal temperature

Hierarchic layout

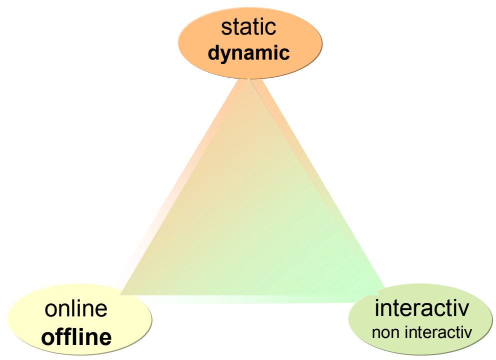
1. Layer assignment (simpel hierarchy)
   - Topological sorting
   - Cycle removal
   - Dummy nodes for „long“ edges
2. Crossing reduction (swapping nodes)
3. Horizontal coordinates
4. Positioning of edges

Graph Drawing Software

- aiSee
- yFiles
- AGD library
- Graph Drawing Server (Internet service)

  http://loki.cs.brown.edu:8081/graphserver/
- jGraph
- a lot more ...

Graph Drawing



Dynamic Graph Drawing for a Sequence of Graphs

Offline Dynamic Graph Drawing

- Computing layouts of graphs which evolve over time
- Goal: preserving the mental map
- Idea: computing global layout and making local arrangements upto some boundary

[S. Diehl, C. Görg, „Dynamic Graph Drawing for a Sequence of Graphs", in Proceedings of Graph Drawing –10th International Symposium GD, Irvine, CA, USA, August 2002, Springer]

Dynamic Graph Drawing for a Sequence of Graphs

Mental Map and Mental Distance

- **Mental Map**: abstract structural information of a graph layout
- Some Mental Map models:
  - Orthogonality
  - Proximity
  - Topology

Dynamic Graph Drawing for a Sequence of Graphs

Mental Map and Mental Distance

- **Mental Distance**: metric indicating how good a layout preserves the mental map of another layout

$MD(l_1,l_2)\geq 0$

$MD(l,l)=0$

- Examples of Mental Distances:
  - Euclidean Mental Distance
  - Orthogonal Mental Distance

Dynamic Graph Drawing for a Sequence of Graphs

Offline Dynamic Graph Drawing Problem

- **Layout Quality**: metric for the quality of a single layout
- **Offline Dynamic Graph Drawing Problem**:
  1. Minimize the sum of the mental distances
  2. Maximize the quality of the single layouts

In general these two optimization goals can not be achieved at the same time !

Dynamic Graph Drawing for a Sequence of Graphs

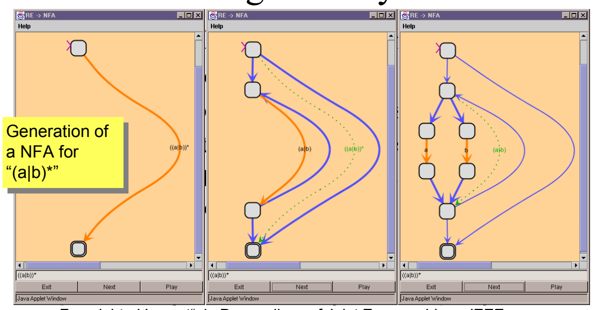
Foresighted Layout

- Dynamic graph drawing algorithm:
  - Given a sequence of graphs
  - Build the supergraph of all graphs
  - Different methods to compact the supergraph
  - Compute global layout of the supergraph, which induces a layout for each graph

[S. Diehl, C. Görg, A. Kerren, „Preserving the Mental Map using Foresighted Layout", in Proceedings of Joint Eurographics – IEEE TCVG Symposium on Visualization VisSym '01, Springer,2001]

Dynamic Graph Drawing for a Sequence of Graphs

Foresighted Layout



Foresighted Layout", in Proceedings of Joint Eurographics – IEEE TCVG Symposium on Visualization VisSym '01, Springer,2001]

Foresighted Layout with Tolerance

- **Tolerant Offline Dynamic Graph Drawing Problem**:
  1. Each mental distance must be less than δ
  2. Maximize the quality of the single layouts

- ◆ Small δ fortifies dynamic stability
- ◆ Large δ increases local quality

Dynamic Graph Drawing for a Sequence of Graphs

Foresighted Layout with Tolerance

```
foresightedLayoutWithTolerance ( g_1,...,g_n )
{   Compute global layout L for supergraph of g_1,...,g_n
    for i=1 to n do
        L_i=L|g_i
        l_i = adjust (L_i, ...)
        if (i==1) then Draw graph g_1 using l_1
        else Draw graph g_i by morphing from l_i-1 to l_i
}
```


Dynamic Graph Drawing for a Sequence of Graphs

Layout Adjustment Strategies

- ◆ Independent Adjustment
  $l_i = adjust(L_i, \delta)$ — $MD(L_i, l_i) < \delta$
- ◆ Predecessor Dependent Adjustment
  $l_i = adjust(L_i, l_{i-1}, \delta)$ — $MD(l_{i-1}, l_i) < \delta$
- ◆ Context Dependent Adjustment
  $l_i = adjust(L_i, l_{i-1}, L_{i+1}, \delta)$ — $MD(l_{i-1}, l_i) < \delta$, $MD(l_i, L_{i+1}) < \delta$
- ◆ Simultaneous Adjustment
  $(l_1,...,l_n) = adjust(L_1,...,L_n, \delta)$ — $MD(l_i, l_{i+1}) < \delta$ $\forall\, 1 \le i < n$

Dynamic Graph Drawing for a Sequence of Graphs

Layout Adjustment for Force-Directed Layout

```
adjust( L_i, l_i-1, δ)        // Predecessor Dependent
{   l = L_i
    for j=1 to #Iterations do
        Compute forces for each node in l with global T
        Compute new layout l' by applying forces to node in l
        if MentalDistance(l_i-1, l') < δ then l = l'
        anneal T
    return l
}
```


Dynamic Graph Drawing for a Sequence of Graphs

Implementation

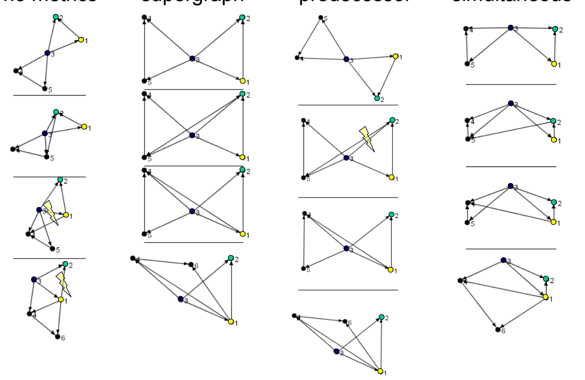
- First prototype implementation: spring embedder with polar and parallel magnetic fields, gravity and simulated annealing
- Implements the presented strategies and mental distances
- Animations available online
  http://www.cs.uni-sb.de/~diehl/ganimation

Dynamic Graph Drawing for a Sequence of Graphs

Example

- Some „butterfly“ graphs
- Different adjustment strategies

Dynamic Graph Drawing for a Sequence of Graphs



Dynamic Graph Drawing for a Sequence of Graphs

Conclusion

- ✓ Generic algorithm
- ✓ Several strategies for layout adjustment
- ✓ Trade stability for local quality

Dynamic Graph Drawing for a Sequence of Graphs

Future work

- Adjustment strategies for hierarchical layout
- Further metrics
- Larger graphs
- Study of effectiveness of animations
- Evaluation of dynamic stabilitiy

Organizational Issues

- Final Exam on February 12th
- Next week due of 2nd projekt (Tree map)
- January 22nd second session of short presentations (SV for Software Engineering)