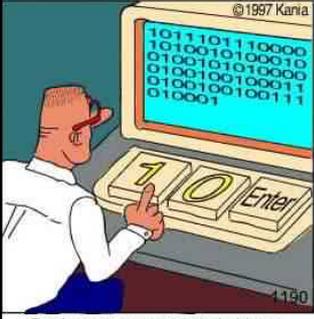


Software Visualization



Lecture
WS 02/03
Algorithm Animation

Real programmers code in binary.

Lecture: Software Visualization, WS02/03 © Dr. Stephan Diehl, Universität des Saarlandes

Algorithm Animation

- What is it about?
- History
- Examples
- Some Design Issues
- Architectures

Lecture: Software Visualization, WS02/03 © Dr. Stephan Diehl, Universität des Saarlandes

Algorithm Animation

- A short definition
 - Algorithm Animation = Visualization of the behavior of an algorithm
 - to animate = „to bring to live“ 
- Motivation
 - Understanding, Teaching
 - Design
 - Optimization
 - Debugging

Lecture: Software Visualization, WS02/03 © Dr. Stephan Diehl, Universität des Saarlandes

Algorithm

- “Algorithm”
named after **Abu Ja'far Muhammad ibn Musa Al-Khwarizmi**, who wrote ~ 840 a treatise on algebra and a treatise on arithmetical calculation with Hindu-Arabic numerals. The Arabic text is lost but a Latin translation, *Algorithmi de numero Indorum (Al-Khwarizmi on the Hindu numerals)*, gave rise to the word algorithm deriving from his name in the title.

Lecture: Software Visualization, WS02/03 © Dr. Stephan Diehl, Universität des Saarlandes

Algorithm

- Definition of the term „Algorithm“ ???
 - There are too many ! Formal and informal.
 - See „What is an Algorithm?“ by Yiannis N. Moschovakis, in Bjorn Engquist and Wilfried Schmid (Eds.), Mathematics Unlimited -- 2001 and Beyond, Springer, 2001.

Lecture: Software Visualization, WS02/03 © Dr. Stephan Diehl, Universität des Saarlandes

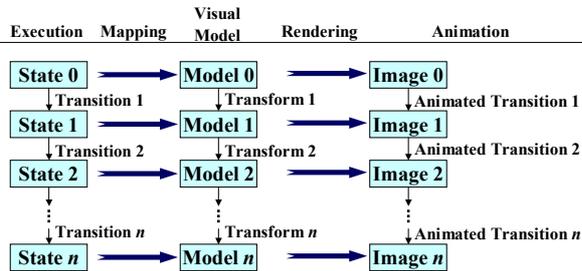
Algorithm Animation

Execution	Mapping	Animation
State 0	→	Image 0
↓ Transition 1		↓ Animated Transition 1
State 1	→	Image 1
↓ Transition 2		↓ Animated Transition 2
State 2	→	Image 2
⋮		⋮
↓ Transition n		↓ Animated Transition n
State n	→	Image n

Lecture: Software Visualization, WS02/03 © Dr. Stephan Diehl, Universität des Saarlandes

Algorithm Animation

- Visual Model = graphical object, geometric data



Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Algorithm Animation

- **Challenge:** Finding the right model
 - Appropriate graphical abstractions for states and transitions between states.

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

A Short History of Algorithm Animation

- First algorithm animations:
 - List processing with language L6 [Knowlton:66]
 - Video „Sorting Out Sorting“ at SIGGRAPH'81 [Baecker:81]

Algorithm Animation Systems (Tools):

- **Georgia Institute of Technology: Stasko**
 - XTango/Tango (1990)
 - Smooth Animations (Path-Transition Paradigm)
 - Polka + Samba (Frontend)
 - Polka3D (1992)
- **MIT later DEC-SRC: Brown (and Najork)**
 - Balsa (1985)
 - Interesting Events, Multiple Views
 - Balsa II (1988)
 - Zeus (1992)
 - Anim3D, Zeus3D (1993)
 - CAT (WWW, 1996), JCAT (1997)

• **And many more by other researchers:** Animal, CATAI, Daphne, Ganimal, Gasp, GeoWin, Jawaaw, Jeliot, Leonardo, Mocha, ...

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

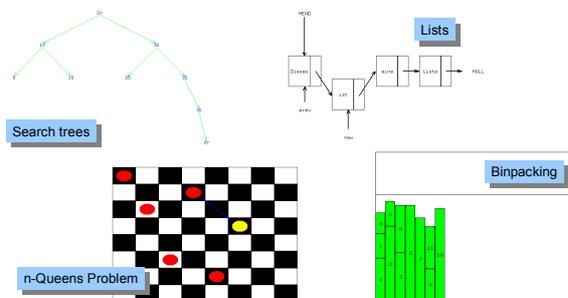
Sorting Out Sorting

- Video

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

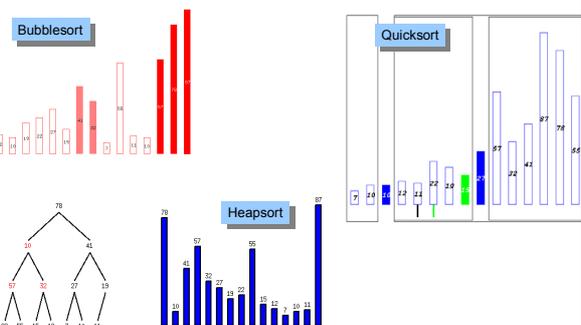
X-Tango



Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

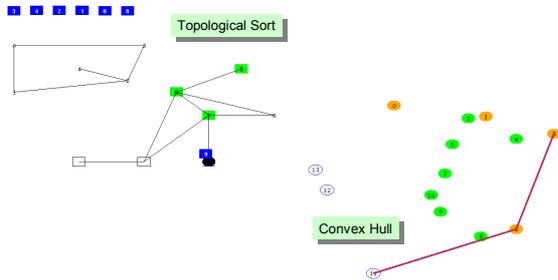
X-Tango



Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

X-Tango



Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Primitives of X-Tango

- Graphical Primitives: lines, nodes, boxes, text
- More complex graphical operations are provided by predefined views.
- Implements the path-transition paradigm
- Algorithms are annotated with IEs

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

3D for Algorithm Animation

- Aesthetics
 - Humans are used to three dimensions
- Data structures or algorithms for 3D geometry
 - E.g. triangulation
- 3D adds additional information to a 2D representation
- Multiple views of an object
- History

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

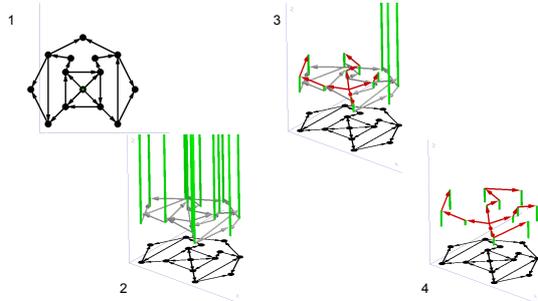
3D Animation of the Shortest Path Algorithm

- Benefit: additional information
- SSSP: Single Source Shortest Path
- Graph drawn in the XY plane
- Z axis indicates costs for every node
- Source has costs = 0
- Shortest Path = ascending path with lowest height
- Result: shortest-paths tree

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

3D Animation of the Shortest Path Algorithm



Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Some Design Issues

- **How are invariants visualized?**
 - In the 3D-Heapsort the heap-property is shown as follows: along each path the columns have increasing height.
- **How does focussing work?**
- **How is recursion displayed?**
 - E.g. frames, colors, sound, ...
- **Goal of the animation system?**
 - Easy-to-use, comprehensible, powerful?
- **Coupling/separation of algorithm and animation?**

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Architectures of Algorithm Animation Tools

- Don't use a tool at all. Implement everything **from scratch**.
- Use at least **libraries** with graphical abstractions, control-elements, etc.
- **Special datatypes**
 - Program the algorithm with datatypes which have built-in visualizations
- **Post-Mortem** Visualization
 - Algorithm and visualization tool are two separate applications
 - Trace
 - Animation plan (typically not a full-fledged programming language)

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Architectures of Algorithm Animation Tools

- **Interesting events** and multiple views
 - Annotate interesting program points
 - MVC design pattern
- **Declarative**
 - separates annotations and algorithm
 - State mapping: a demon watches state changes and visualizes the state
 - Constraints-based systems
- **Semantics-Directed** (usually non-intrusive)
 - Visual Interpreter or Debugger

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Example: Interesting Events (Polka)

```
void main() {
  bsort.SendAlgoEvt("Input",n,v);
  for(j=n; j>0; j--)
    for(i=1; i<j; i++)
      if (v[i]>v[i+1])
        { int temp= v[i];
          v[i]=v[i+1];
          v[i+1]=temp;
          bsort.SendAlgoEvt("Exchange",i,i+1);
        }
}
```

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Example: Declarations (Leonardo)

```
void main() {
  for(j=n; j>0; j--)
    for(i=1; i<j; i++)
      if (v[i]>v[i+1])
        { int temp= v[i];
          v[i]=v[i+1];
          v[i+1]=temp;
        }
}

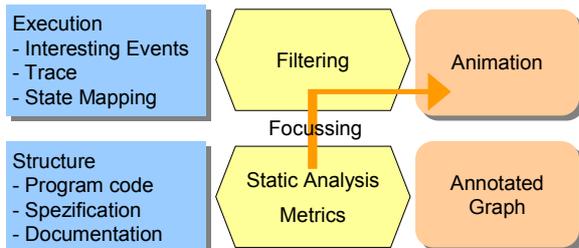
/**
View(Out 1);
Rectangle(Out ID, Out X, Out Y, Out L Out H, 1);
For N: InRange(N,0,n-1)
Assign X=20+20*N Y=20 L=15 H=15*v[N] ID=N;
**/
```

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Visualization Pipeline

Data Acquisition \rightleftarrows Analysis \rightleftarrows Visualization



Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Abstract Algorithm Animation

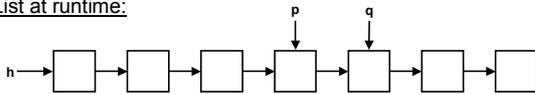
- **Problem:** At each step changes only occur at a small fraction of the data.
- **Approach:** Use Static Program Analysis
 - *Shapeanalyse* [see Sagiv,Reps,Wilhelm]
- Computes an **abstract representation** of linked data structures, which focusses on the active parts of this structures. For each program point it yields a finite set of shape graphs.
- Goal: Animation of **abstract execution** of algorithm called algorithm explanation [Braune&Wilhelm:2000].

Lecture: Software Visualization, WS02/03

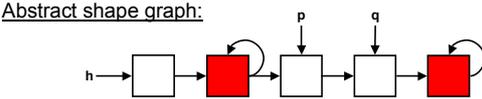
© Dr. Stephan Diehl, Universität des Saarlandes

Abstract Algorithm Animation: An Example

List at runtime:



Abstract shape graph:

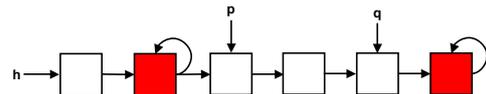
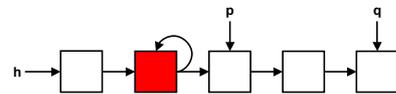


Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Abstract Algorithm Animation: An Example

Possible subsequent graphs for $q := q \rightarrow \text{next}$:



Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Abstract Algorithm Animation

- A transition from an abstract state as_1 to an abstract state as_2 is legal, if a transition from a concrete state cs_1 to a concrete state cs_2 exists where cs_1 is represented by as_1 and cs_2 by as_2 .
- Visual abstract execution must only show legal transitions.

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes

Assignments

- Paper presentations (max. 10 minutes each)
 - We have a choice of 6 papers.
 - Teams of two
 - There will be paper presentations about other software visualization tools later.
 - Presenters have to be here at 2pm (s.t.) next week.
- <http://www.cs.uni-sb.de/~diehl/SoftVisVorles>
 - User:
 - Password:

Lecture: Software Visualization, WS02/03

© Dr. Stephan Diehl, Universität des Saarlandes