Visualization
Diva and Ptolemy II

John Reekie
UC Berkeley

In collaboration with:
Michael Shilman
Heloise Hse
UC Berkeley

Why visualization?

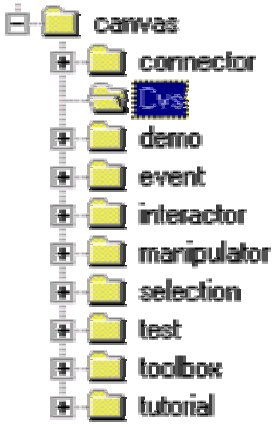
- Design is *exploration* of the design space
  - Much of early design is understanding the *problem*
  - From any given point, one can only see a limited distance
- Complex systems require insight and understanding
  - Artifacts promote reflection
  - Alternate projections of information promote insight
  - Emergent properties are part of the process

What do we need?

- Graphics infrastructure
  - Flexible and uncommitted
  - Persistent-mode objects
  - Scalable floating-point coordinate system
- Visualization infrastructure
  - Data structuring
  - Visualization protocols
- Applications!
  - Ptolemy II
  - Software design
  - CAD/EDA
  - Group awareness

The Diva Canvas

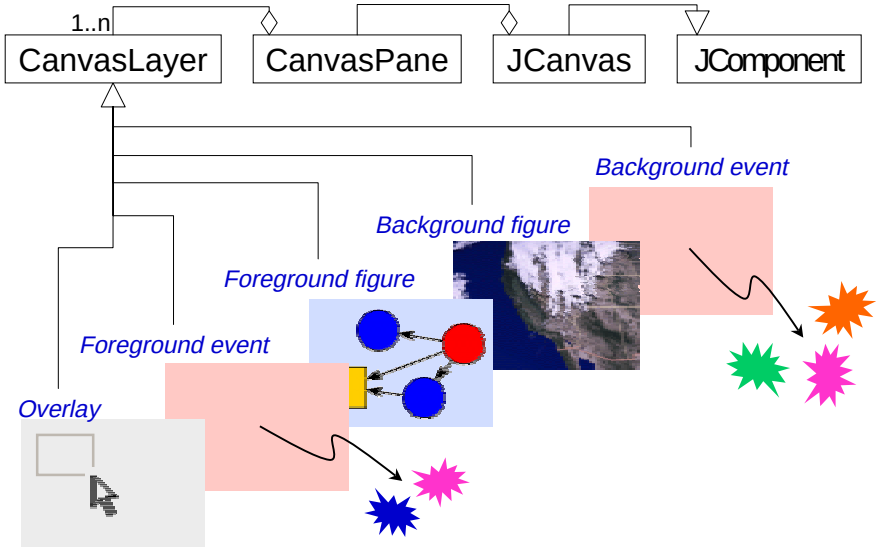
- Persistent-mode graphics infrastructure for:
  - 2D visualization surfaces
  - 2D graphical editors
- Why build our own? We couldn't find a package that is:
  - Open-source
  - Flexible and powerful
  - Based on Java2D
- Non-trivial
  - 20 klocs



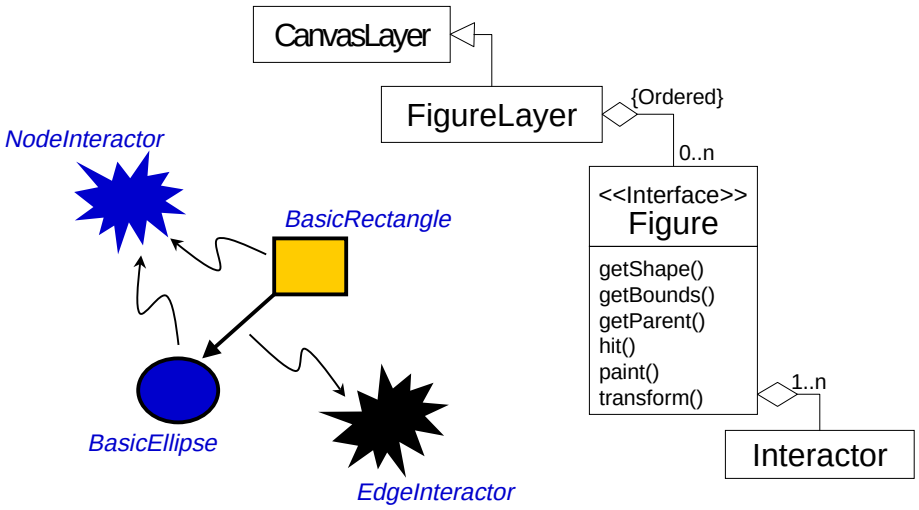
Java2D

- The "new graphics" from Sun
  - Arbitrary shapes and clip regions
  - Transforms
  - Alpha-channel
  - Anti-aliasing
  - Image support
  - Silly demos
- Down-sides
  - Complex
  - Performance performance performance

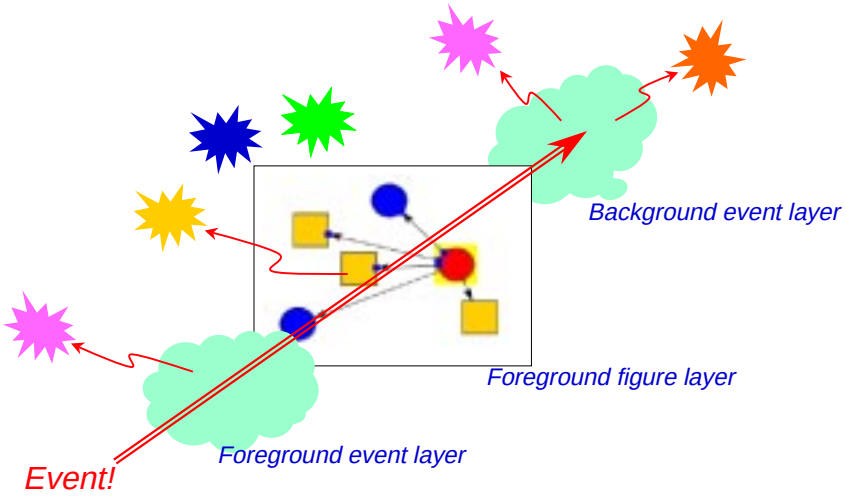
Layered architecture



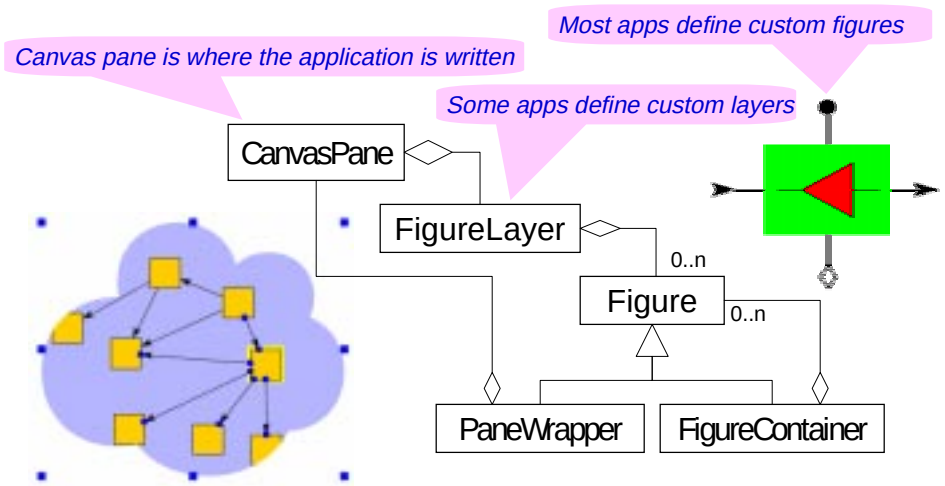
Figures



Event-handling

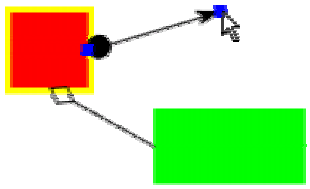


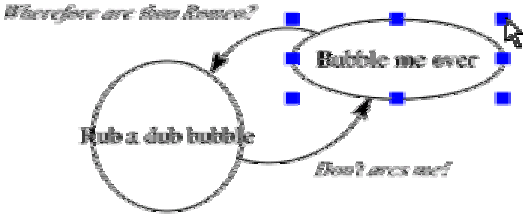
Structural recursion

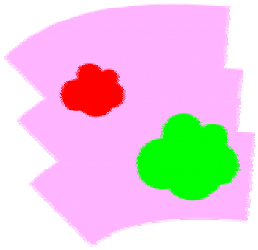


Plus useful stuff

- "Starter" figures
- Interactors
- Manipulators
- Connectors

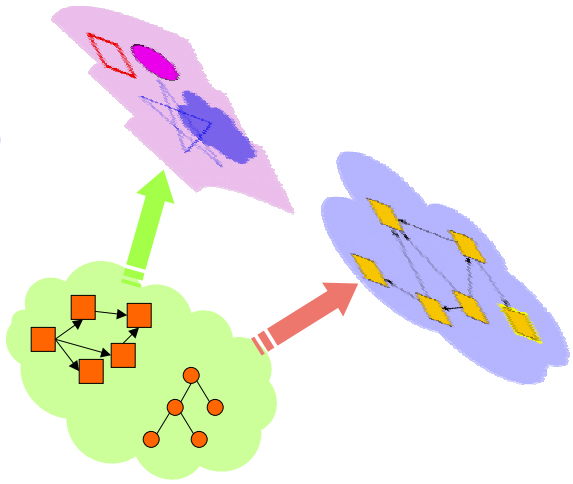






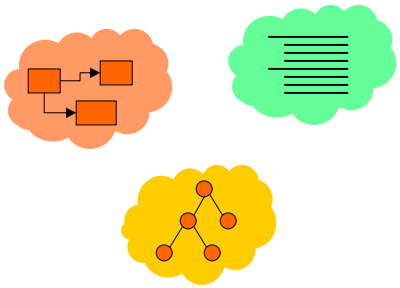
Visualization surfaces

- Surfaces
  - A projection of information to visible (usually) form
  - How do we make them reusable and generic?



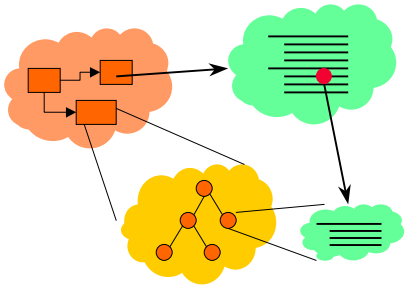
Infospaces (1)

- An *infospace* is a coherent collection of information points
  - Web mining bot
  - Source code statistics
  - System model
  - Circuit simulation
  - Database query
- Infospaces are *dynamic* and *evolving*



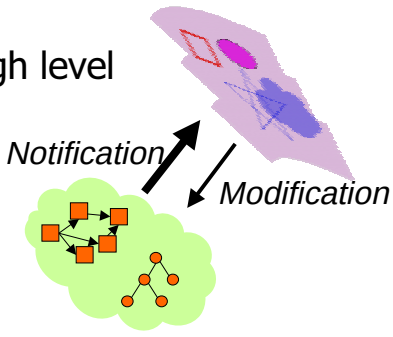
Infospaces (2)

- Infospaces are structured in complex ways
  - Contain other infospaces
  - Reference other infospaces
- Infospaces are application-specific
  - Can we build them from reusable components?
  - How do we connect reusable surfaces?

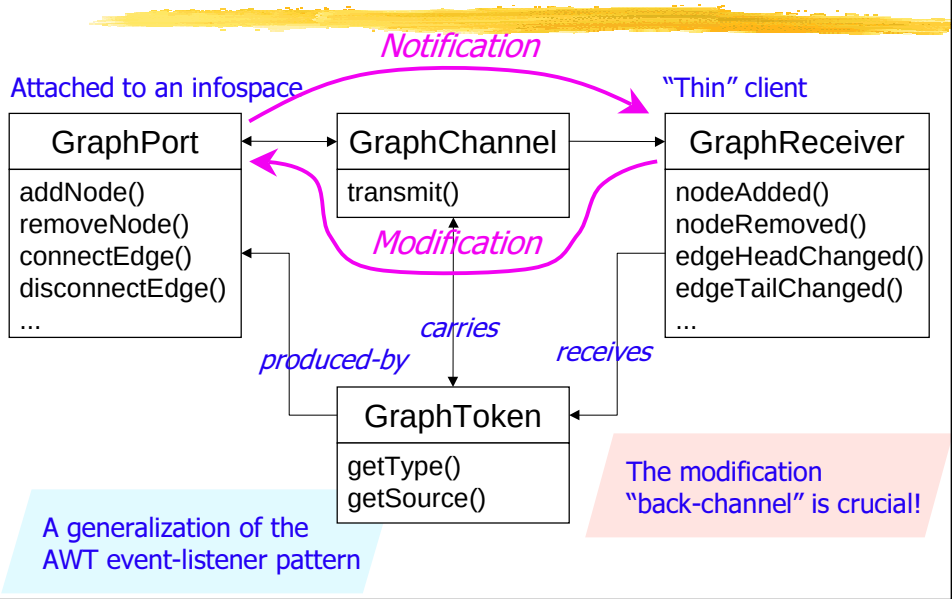


Protocols

- The language for communicating between components
  - Events for notification (up)
  - API for modification (down)
- Protocols are typed at a high level
  - Graph
  - Tuple-set
  - Array
  - 2D layout



An example protocol: graph

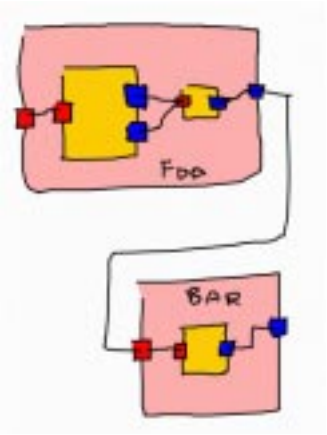


Parameterizing surfaces

- Surface is parameterized by:
  - Node/edge renderer
  - Node/edge interactor
  - Layout engine
  - Controller for more flexible control
- Possible instantiations
  - Bubble diagrams (ellipses, arc edges)
  - Schematic diagrams (rectangles, ports, manhattan edges)
  - Sketch-based graph editing

Sketch-based interaction

- Fluid navigation
  - Scrollbars don't work with arbitrary zooming
  - Reduces interaction modality
- Less formal interfaces
  - Sketching is more "natural"
  - Encourages brainstorming, conceptual sketches
  - Compared to paper, digital sketches can be processed, edited and archived



More information

- Diva 0.2 is released
  - http://www-cad.eecs.berkeley.edu/diva
  - canvas, graph, and sketch "done"
  - protocols and surfaces under construction
- Integration with Ptolemy II
  - First-cut threading and type system visualizations
  - First-cut graphical editors
- Stand-alone applications
  - Digital whiteboard on the 550 Cory LiveBoard