Information Visualization: Graphs and Networks

Ashim Garg

Brown University
Center for Geometric Computing

New Developments

"There is a river, no – a flood of information"
B. Shneiderman

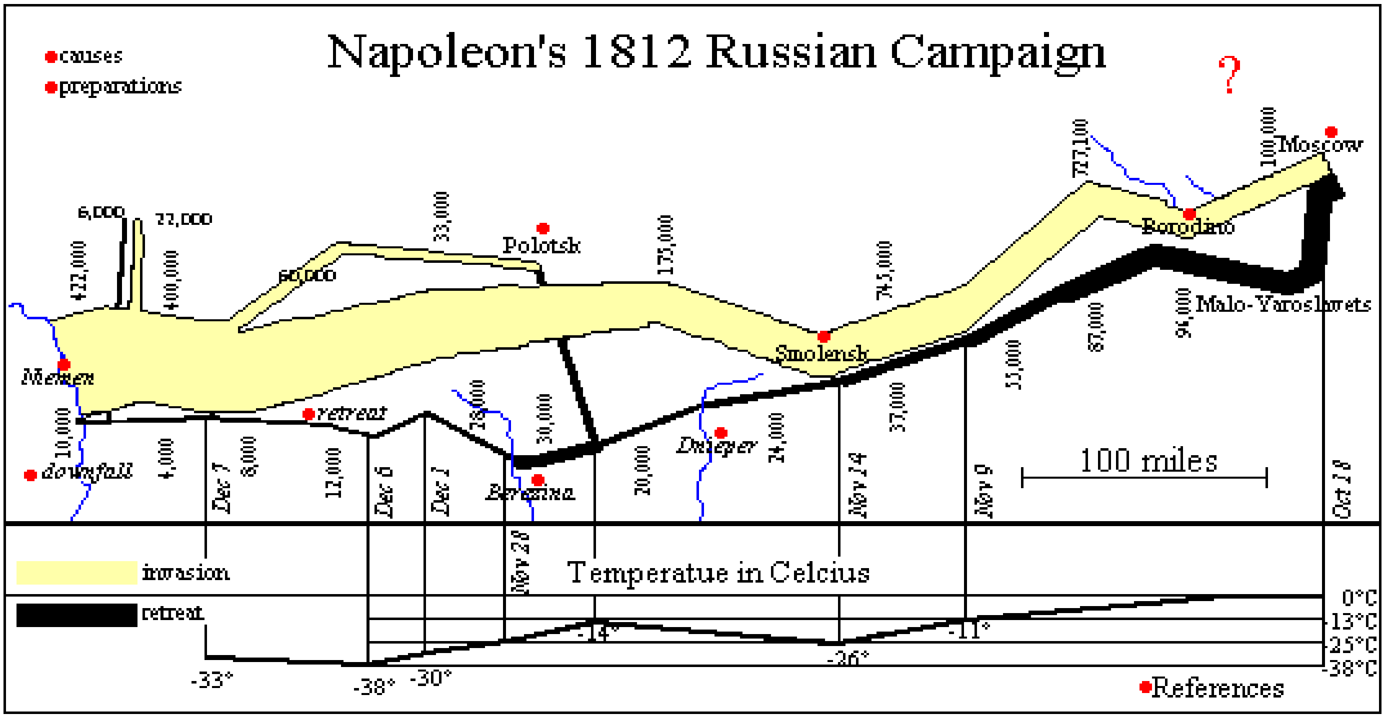
- World-Wide Web
- Hypertext/Hypermedia
- Distributed Information Repositories/Digital Libraries
- Wide Variety of Users
  - Digital Librarian (e.g., creates Indices)
  - Information Visualization Administrator
  - End User
- Executable programs over the WWW
  - Java
  - CGI
  - Mocha [Baker, Cruz, Liotta & Tamassia 96]

Information Access

- users now access increasingly more complex and volumnious information
  - Web-searching
- important to reduce the overall cost of information access in information intensive tasks [Robertson Card Mackinlay 93]
- access costs include:
  - retrieval time
  - cognitive load on user during information interpretation and assimilation

Information Visualization

- focuses on presenting information in a visual form to make it easier for a user to interpret and assimilate information



- Several variables depicted in a limited space for the advance (yellow band) and for the retreat (*black* band) of the French army
  - size of the army depicted by the width of the bands
  - 2D location and geographical landmarks
  - temperature in certain days during the retreat (map made famous by Edward Tufte)

Information Structures

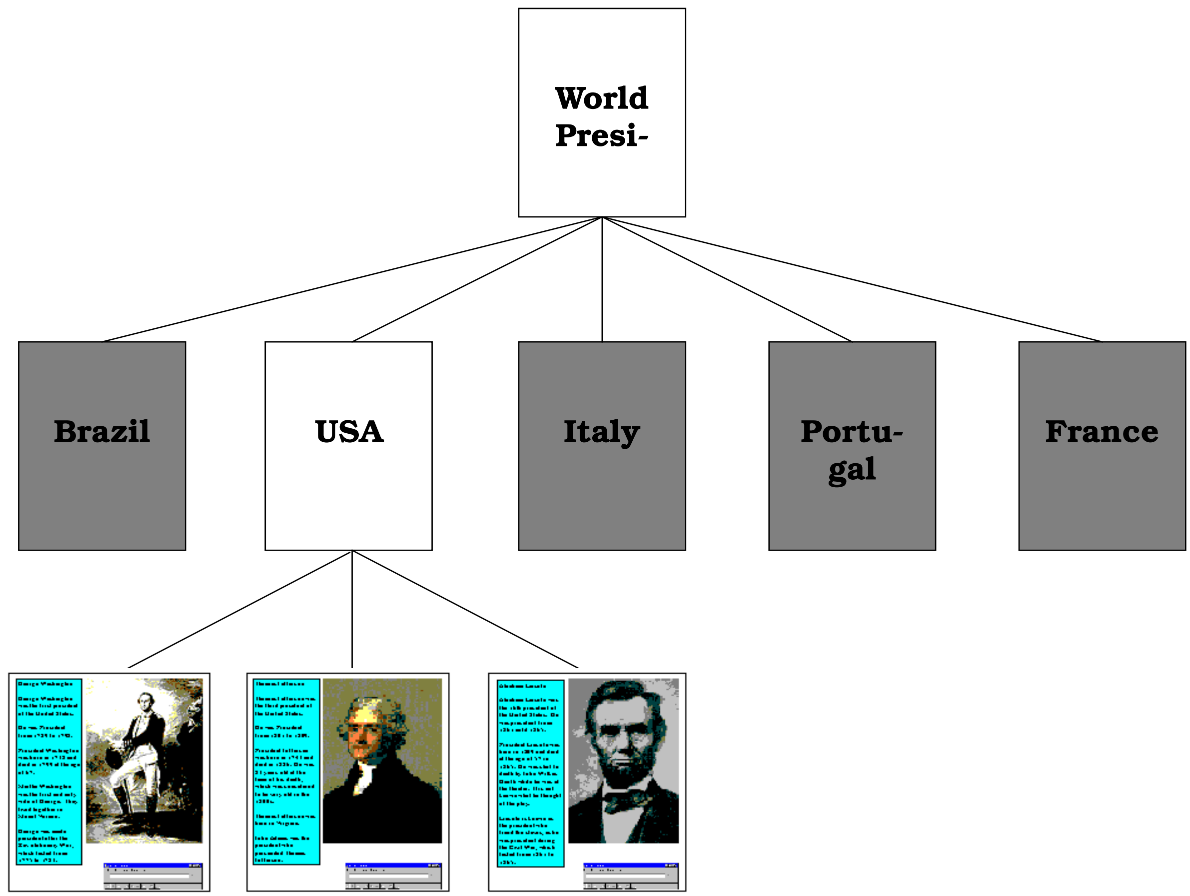
- sequences : research advances in databases

1969 Hypertext Editing System
1970 Relational Model (RM)
1976 Semantic Data Models (SDMs)
1977 First Visual Query Language: QBE
1982 First Graph-based QL (Guide) for a SDM
1984 Graph-based QL (SKI) for a SDM
1985 Graph-based QL (ISIS) for a SDM
1987 Graph-based QL (G^+) for the RM
1987 First Object-oriented DBs (OODBs)
1990 First Work on the WWW starts (CERN)
1992 First tailorable QL (DOODLE) for an OODB
1993 ARPA/NASA/NSF Digital Libraries Initiative
1994 Standardization of the OODB Model
1995 Tailorable QL (OPOSSUM)
1996 SDCR on Human - Computer Interaction

- visualization techniques: Perspective Walls [Mackinlay Card Robertson 91], Life Lines [Plaisant Milash Rose Widoff Schneiderman 96]

Information Structures, cont'd

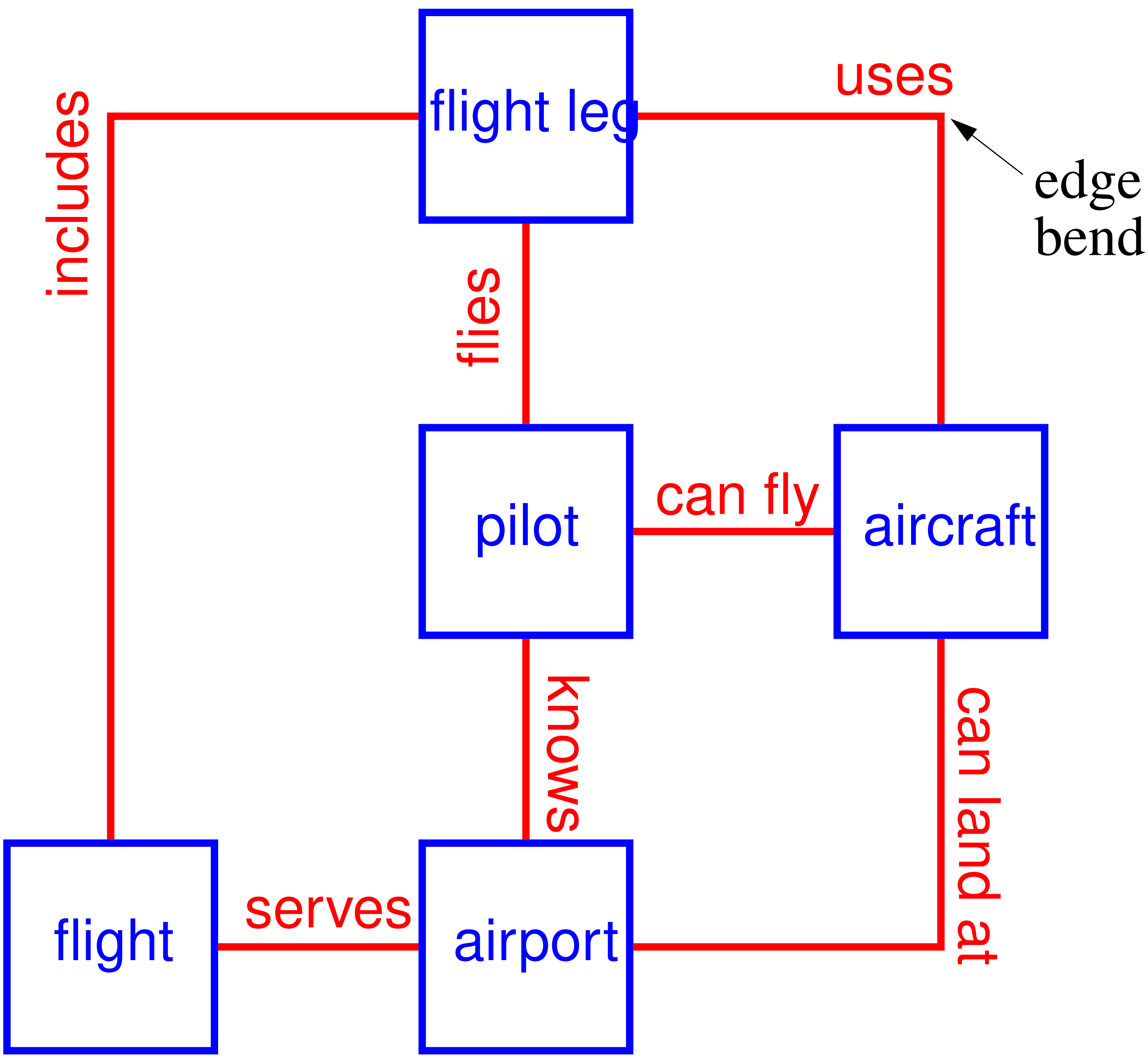
- trees : world presidents



- visualization techniques:
  Cone Trees [Mackinlay Card Robertson 91]
  TreeMaps[Johnson Schneiderman91]

Information Structures, Cont'd

- graphs and networks
  - vertices and edges
  - example: airline database
  - usually drawn as node-link diagrams



Databases "Meet" Information Visualization: Database Visualization

- HCI to DB:
  - Advanced visualization techniques (e.g., to display schema and database objects)
  - User studies
- DB to HCI:
  - Very large database applications (data warehousing, data mining, Web) provide new opportunities for Human Computer Interaction in general, and Information Visualization in particular
  - Powerful query languages to filter visualizations

References:

- ACM Computer Surveys, December 1996: "Strategic Directions in Human-Computer Interaction", Myers, Hollan, Cruz, eds.
- ACM SIGMOD Record, December 1996, "Special Issue on Information Visualization", Catarci and Cruz, eds.

Issues in Information and Graph Visualization

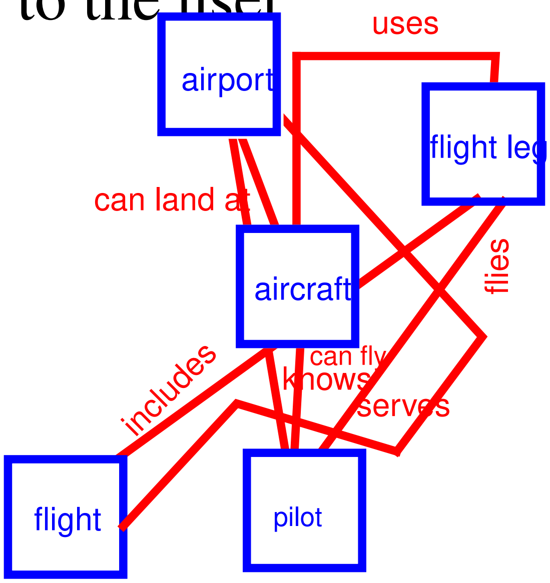
Support for Interaction

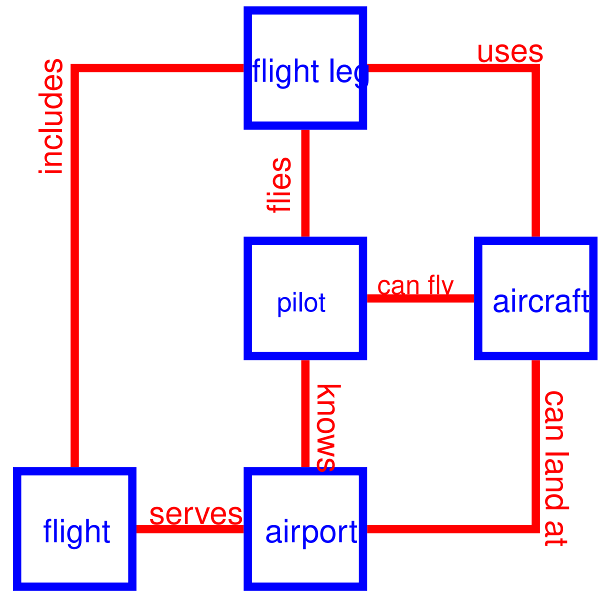
- information filtering
  - too much information can be overwhelming
- focus+context display
  - highlight the objects in focus but also show their context
- performing task-specific operations
  - clicking on a node in a map of a Web-subspace should take us to that site

Issues in Information and Graph Visualization

Automatic Layout

- some drawings are more **readable**, i.e., are better than others in conveying information to the user



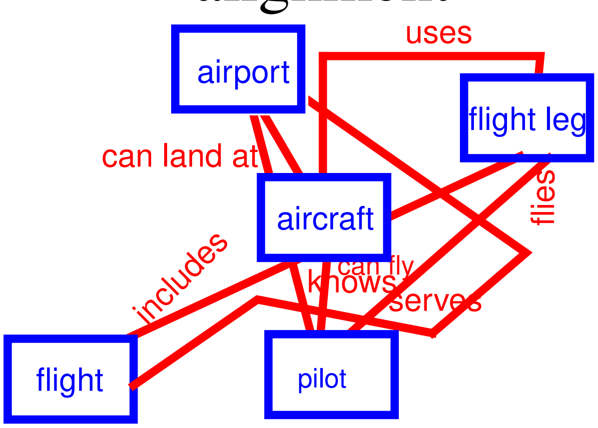


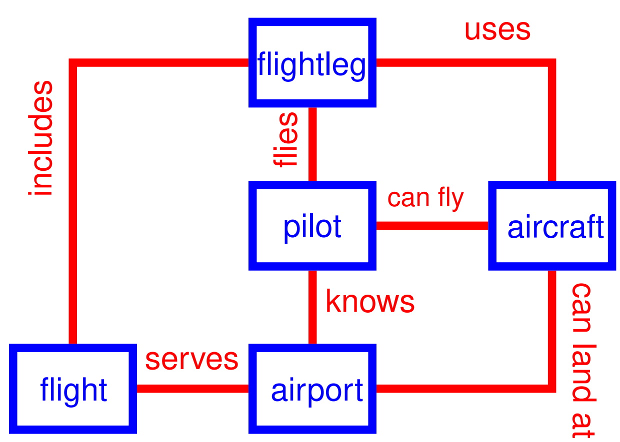
- how can we construct readable drawings?

Readability Criteria

- what makes a drawing more readable?

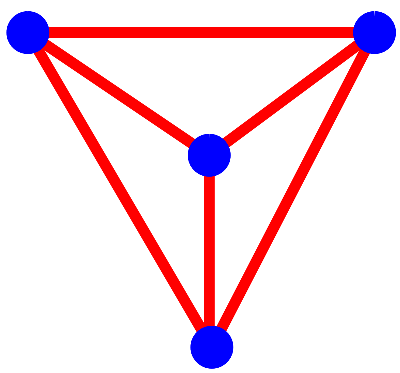
- user studies [Batini Furlani Nardelli 85, Cohen Eades Rusky 94] show that a drawing is more readable if it satisfies the following readability criteria:
  - no edge-bends
  - no edge-crossings
  - small edge-lengths
  - occupy small screen-space
  - display symmetries
  - satisfy application-specific constraints such as clustering and alignment



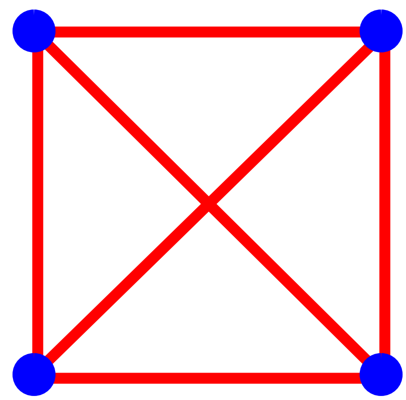


Trade-offs

- in general, not possible to satisfy two readability criteria simultaneously



min # crossings



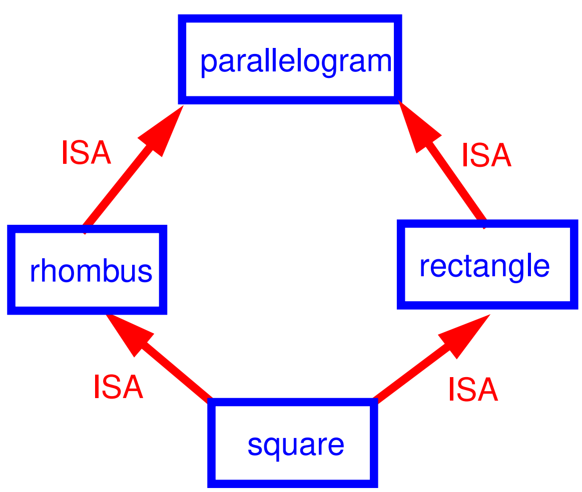
max symmetries

- a wide variety of heuristics for constructing readable drawings are available that focus on satisfying different aesthetic criteria

Visualizing Hierarchical Structures

Hierarchical Structures

- found in a variety of applications
  - example: object oriented databases
  - can be modeled using directed acyclic graphs, i.e., graphs with directed edges but no directed cycles
- are usually drawn in an upward fashion, i.e., such that the edges “flow” in the same direction, e.g., from bottom to top
  - upward drawings are more readable because the placement of vertices reflects the hierarchy



Planar Upward Drawings

- A planar drawing is one with no edge crossings
- planar upward drawings combine the advantages of planarity and upwardness
- is there an efficient technique for constructing planar upward drawings?
- however, such an efficient technique is unlikely to exist....

Upward Planarity Testing

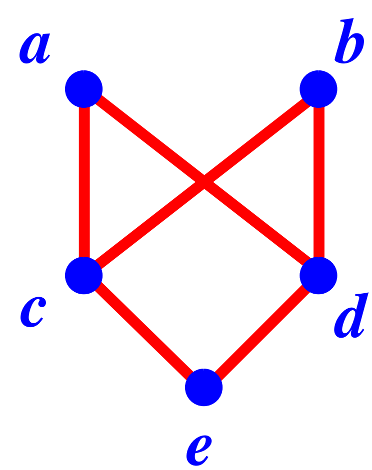
- the problem given below is NP-complete

  [Garg Tamassia 94]

> **Upward Planarity Testing Problem**
> given a directed acyclic graph, test whether it admits an upward planar drawing

- covering digraphs of ordered sets are a subclass directed acyclic graphs

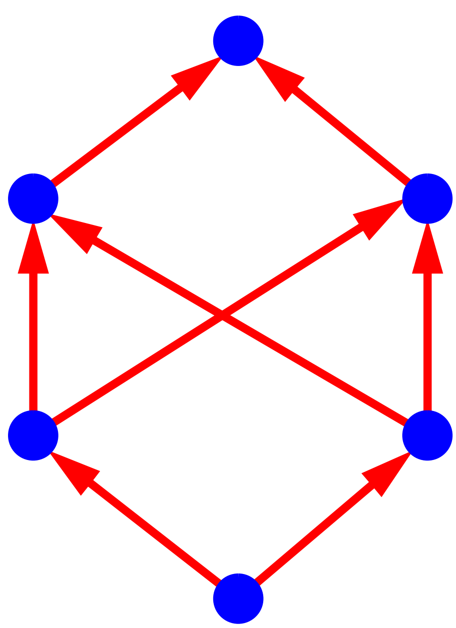
P = {(a,c), (a,d),
(a,e), (b,c),
(b,d), (b,e),
(c,e), (d,e)}



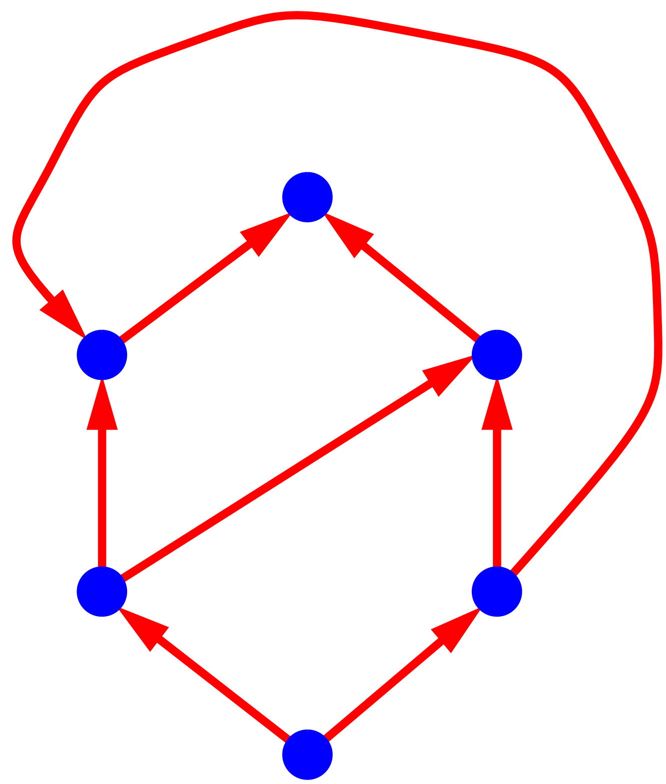
- our NP-completeness result for upward planarity can be used to show the NP-completeness of the planarity testing of ordered sets, a long standing problem in discrete mathematics

Problems with Upward Drawings

- our NP-completeness result is a consequence of a general problem with upward drawings
  - the requirement of upwardness is too restrictive!
  - a graph may not admit an upward drawing that satisfies a readability criterion



upward with edge crossings



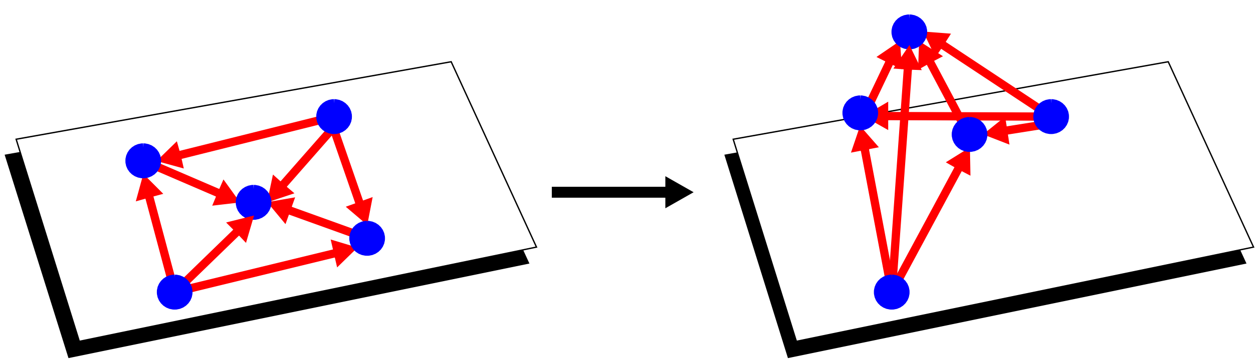
nonupward with no edge crossings

Upward Drawings

- how can we construct readable upward drawings, i.e., upward drawings that satisfy several other readability criteria to a large extent as well?

GET THREE DIMENSIONAL !

- an approach for constructing a readable upward drawing:
  - use a high quality drawing technique to construct a non upward two dimensional (2D) drawing
  - lift vertices along the third dimension such that the drawing is upward in the third dimension



A High Quality 2D Drawing Technique

- what technique for constructing 2D non upward drawing may be a good choice?
  - generally difficult to evaluate drawing techniques by using analytical tools
- we implemented and experimentally evaluated four well-known techniques that use four different approaches
  [Di Battista Garg Liotta Tamassia Tassinari Vargiu 95]
- Bend-Stretch: uses visibility representation
  [Tamassia Tollis 89]
- Column: uses a numbering of the vertices
  [Biedl Kant 94]
- Giotto: uses a network-flow based approach
  [Tamassia Di Battista Batini 88]
- Pairs: uses a vertex-pairing technique
  [Papakostas Tollis 94]

Evaluation Criteria

- we evaluated the techniques on 9 readability criteria and running time
  - Area : screen space occupied by a layout
  - Cross : number of edge crossings
  - TotalBends : number of edge bends
  - TotalEdgeLen : sum of edge lengths
  - MaxEdgeBends : maximum number of bends in an edge
  - MaxEdgeLen : maximum length of an edge
  - UnifBends : standard deviation of the number of bends
  - UnifLen : standard deviation of edge lengths
  - Screen Ratio : deviation from 4/3 of the height-to-width ratio of a layout
  - RunTime : time taken to create a layout

Test Data for Our Experimental Study

- We started with 112 core graphs collected from:
  - software companies and government organizations (54%)
  - books and articles in software engineering and database systems (33%)
  - Ph.D. and Sc.M. theses in software and database visualization (13%)
  - 11,582 test graphs generated as variations of the core graphs by applying a random sequence of update operations typical in software engineering and database applications
- the test graphs are available on the internet

Related Work

- Himsolt [95] evaluated 3 techniques for drawing general graphs [Himsolt 95]
  - mostly gave subjective evaluations and gave objective evaluations only on edge-lengths using a small test data
- Brandenburg and Ruhrer [95] evaluated five methods for constructing only straight line drawings (edges drawn without any bends)
- Junger and Mutzel [96] evaluated 8 edge-crossing minimizing heuristic for straight line drawings of bipartite graphs
- in another study, we compared four methods for constructing 2D upward drawings of directed acyclic graphs

  [Di Battista Garg Liotta Parise Tamassia Tassinari Vargiu Vismara 96]

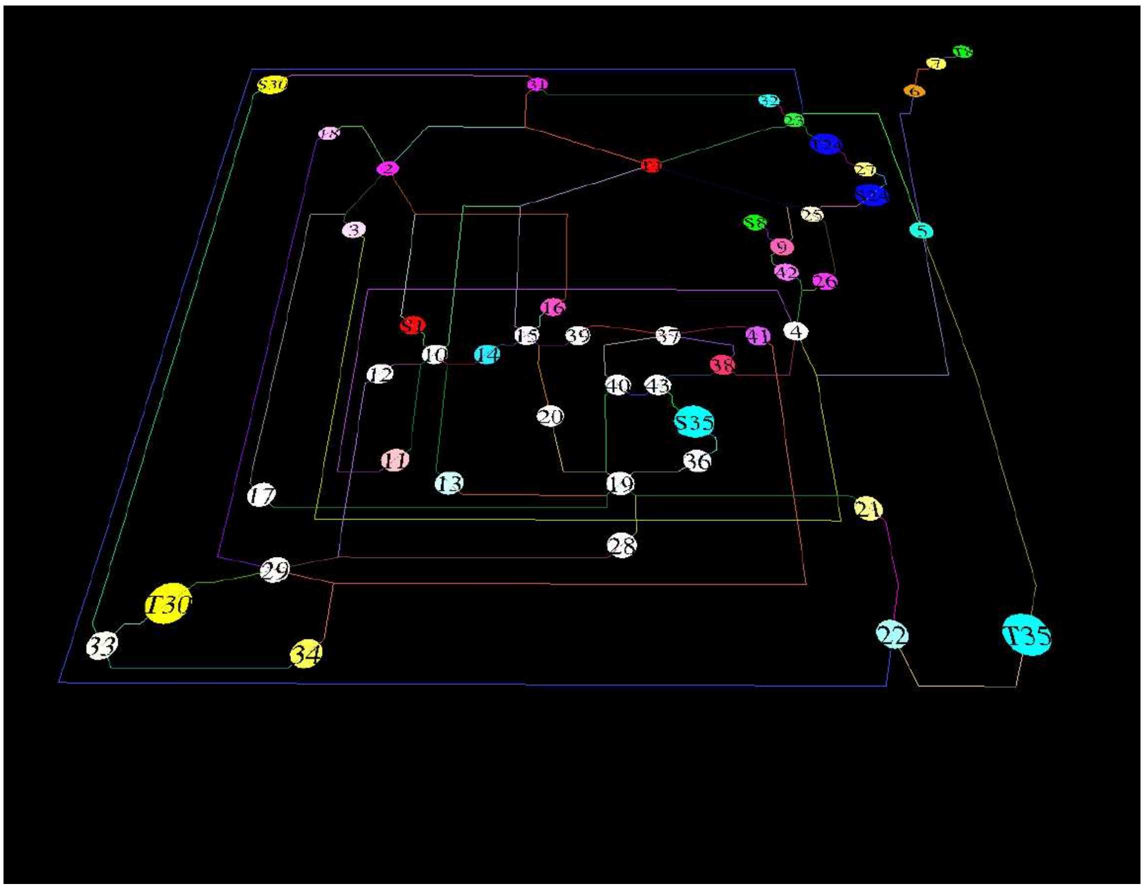
Our Conclusion

- Overall, Giotto is the best technique among the ones evaluated, with reasonably good performance on all the readability criteria considered

Giotto3D: a System for Visualizing Hierarchical Structures in Three Dimensions

Giotto3D

- visualizes hierarchical structures in three dimensions using upward drawings
- uses Giotto to first construct a non upward 2D layout



- lifts the vertices along the z-direction to get an upward drawing

Giotto3D Visualizations

- Vertices are represented by spheres and edges by Bezier tubes
- vertex coloring and size can be used to encode application-specific information
- the footprint (xy-projection) is a high quality drawing with few edge crossings
- implemented with C++ and OpenInventor
- subjective evaluations on graphs derived from applications show good results in practice

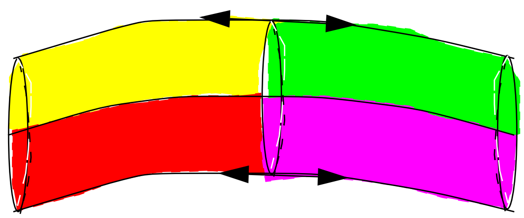
Giotto V isualizations

Vertex Representation

- dra wn as spheres
- size and color can be used to encode application-specific information

Edge Representation

- drawn as Bezier Tubes [Spratt Ambler 94]
  - consists of connected tubular segments whose "top" and "bottom" halves are each Bezier surfaces
  - we achieve smoothness at the joins of two segments by ensuring that that they join tangentially



Support for Interaction in Giotto3D

- 3D-na vigation support
  - rotation, translation, zooming
- information filtering
  - show only selected objects
  - show only the descendents of selected objects
  - (future work) a query language for filtering information
- focus+context displa y
  - gra ying the objects not in focus
  - 3D nature of the dra wing helps

Related W ork

- In cone trees , children of a vertex are placed around a cone with the vertex at the apex of the cone
  [Robertson Mac kinla y Card 91]
- Semnet is a system for displa ying knowledge bases
  [F airc hild P oltroc k Furnas 88]
- Plum is a system for visualizing program information [Reiss 94]

Future W ork

- provide a query language for information filtering
- use Giotto3D for W eb-visualization
- improve Giotto

A Web-based Graph Drawing and Translation Service

Motivation

- make implementations of graph drawing algorithms available
- make implementations easy to use
- vision of distributed geometric computing system, where common servers provide access to graph drawing and computational geometry algorithms
  - draw a graph using a particular algorithm
  - shop around for best algorithm for particular application
  - experimental study and comparison of algorithms
  - demonstrations in educational settings
- first step towards goal is implementation of this prototype graph drawing server

The Service

- easy-to-use , platform-independent interface
  - WWW-based - no downloading or local installation necessary
  - HTML forms or Java-based graph editor as interface
- access to several graph drawing algorithms
  - GIOTTO/GIOTTO with labels
  - PAIRS
  - planarizer
  - more to come
- translation service
  - text formats: parenthetic , MALF, GDS, Tom Sawyer (input only)
  - graphic formats (output only): Postscript, GIF, FrameMaker, FIG, gnuplot
  - more to come

Architecture

- client-server
  - drawing/translation code on server
  - interface code on client
  - users on slow machines can still use service with minimal loss of performance
  - can make implementations available without releasing code
- modular design
  - easy to modify code or add algorithms/ filters

Existing W ork

- do not know of anything quite like this
- number of pac kages a vailable for download
  - M. Himsolt, Graphlet (http://www .uni-passau.de/Graphlet)
  - bibliography maintained by Georg Sanders (http://www .cs.uni-sb.de/RW/users/sander/html/gstools .html)
- U. Erlingsson, M. Krishnamoorthy , graphdra w (http://www .cs.rpi.edu/projects/pb/graphdraindex.html)
  - J a va-based interactive graph dra wing
- S. North' s email server for *dot* (http://www .researc h.att.com/dist/dra wdag/mail_server)
- no known translation service

Using the Server

- http://loki.cs.brown.edu:8081/graphserver/

Extensions

- bug fixes
- more dra wing algorithms
- more graph formats
- improve graph editor

- extend uses
  - programmer' s interface
  - solidify standards for adding implementations to allow additions by others