

Lecture 12: Toolkits

Information Visualization
CPSC 533C, Fall 2011

Tamara Munzner

UBC Computer Science

Wed, 19 October 2011

Required Readings

Prefuse: A Toolkit for Interactive Information Visualization. Jeffrey Heer, Stuart K. Card, James Landay. Proc ACM CHI, 421-430, 2005.

ProtoVis: A Graphical Toolkit for Visualization. Michael Bostock and Jeffrey Heer. IEEE Trans. Visualization & Computer Graphics (Proc. InfoVis), 2009.

D3: Data-Driven Documents. Michael Bostock, Vadim Ogievetsky, Jeffrey Heer. IEEE Trans. Visualization & Computer Graphics (Proc. InfoVis), 2011.

Further Reading

Readings in Information Visualization: Using Vision To Think, Chapter 1. Stuart K. Card, Jock Mackinlay, and Ben Shneiderman. Morgan Kaufmann, 1999.

A Taxonomy of Visualization Techniques using the Data State Reference Model. Ed H. Chi. Proc. InfoVis 2000.

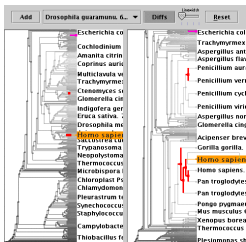
Wrangler: Interactive Visual Specification of Data Transformation Scripts. Sean Kandel, Andreas Paepcke, Joseph Hellerstein, Jeffrey Heer. Proc. CHI 2011.

Toolkits

- imperative: how
 - low-level rendering: Processing, OpenGL
 - parameterized visual objects: prefuse
 - also flare: prefuse for Flash
- declarative: what
 - Protovis, D3, ggplot2
 - separation of specification from execution
- considerations
 - expressiveness
 - can I build it?
 - efficiency
 - how long will it take?
 - accessibility
 - do I know how?

OpenGL

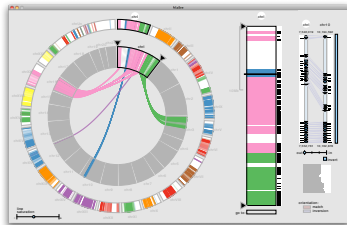
- graphics library
 - pros
 - power and flexibility, complete control for graphics
 - hardware acceleration
 - many language bindings: C, C++, Java (w/ JOGL)
 - cons
 - big learning curve if you don't know already
 - no vis support, must roll your own everything
 - example app: TreeJuxtaposer



[Fig 5. Munzner et al. TreeJuxtaposer: Scalable Tree Comparison using Focus+Context with Guaranteed Visibility. Proc SIGGRAPH 2003, pp 453-462.]

Processing

- layer on top of Java/OpenGL
- visualization esp. for artists/designers
- pros
 - great sandbox for rapid prototyping
 - huge user community, great documentation
- cons
 - poor widget library support
- example app: MizBee



[Fig 1. Meyer et al. MizBee: A Multiscale Synteny Browser. Proc. InfoVis 2009.]

prefuse

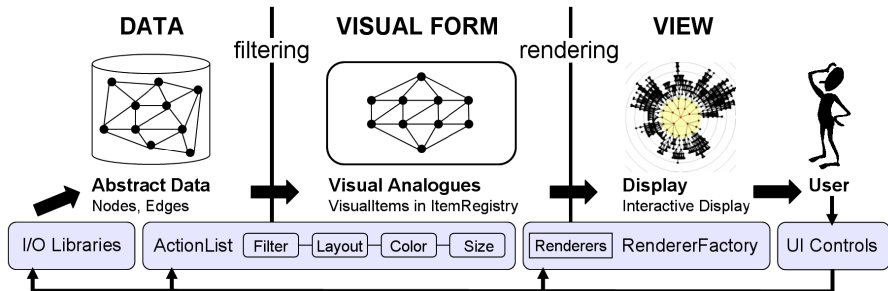
- infovis toolkit, in Java
- fine-grained building blocks for tailored visualizations
- pros
 - heavily used
 - very powerful abstractions
 - quickly implement most techniques covered so far!
- cons
 - no longer under active development
 - nontrivial learning curve
- example app: DOITrees Revisited



[Fig 3. Heer, Card, and Landay. Prefuse: A Toolkit for Interactive Information Visualization. Proc. CHI 2005, 421-430]

Prefuse

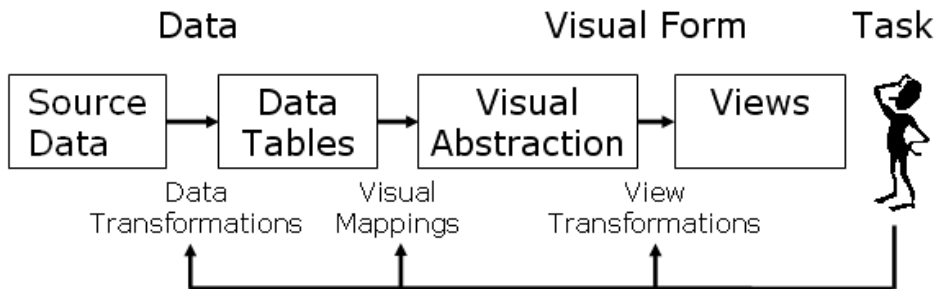
- separation: abstract data, visual form, view
 - data: tables, networks (nodes, edges)
 - visual form: layout, color, size, ...
 - view: multiple renderers



[Fig 2. Heer, Card, and Landay. Prefuse: A Toolkit for Interactive Information Visualization. Proc. CHI 2005, 421-430]

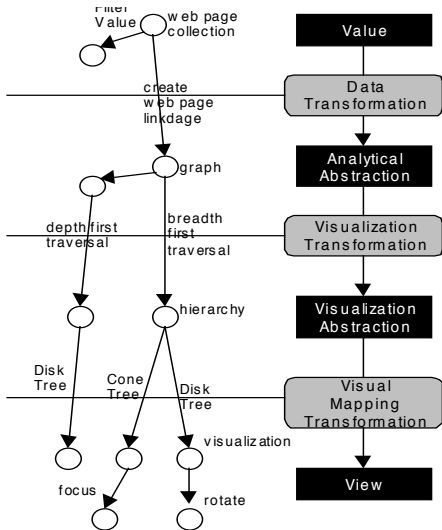
InfoVis Reference Model

- conceptual model underneath prefuse design
- heavily influenced much of infovis (incl nested model)
 - aka infovis pipeline, data state model [Chi99]



[Redrawn Fig 1.23. Card, Mackinlay, and Shneiderman. Readings in Information Visualization: Using Vision To Think, Chapter 1. Morgan Kaufmann, 1999.]

Data State Model



[Fig 2. Chi. A Taxonomy of Visualization Techniques using the Data State Reference Model. Proc. InfoVis 2000.]

Prefuse Design Implications

- separating abstraction, visual form, view
 - supports linked multiple views
 - supports novel visual encoding design
- actions: operator composition
 - supports distortion: layout modification
 - supports animated transitions
- multiple renderers
 - supports semantic zooming
- many/most common methods well supported
 - abstractions map well to infovis concerns
 - nevertheless takes time to wrap head around it
 - good choice for local app

Prefuse Validation

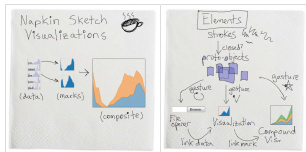
- wide set of old/new app examples
 - expressiveness, effectiveness, scalability
- qualitative usability **for system API**
 - nice methodology!
 - vs for specific application

Declarative Toolkits

- imperative: toolkits/libraries discussed so far
 - say exactly **how** to do it
 - familiar programming model
- declarative: other possibility
 - just say **what** to do
 - Protovis, D3

ProtoVis

- declarative infovis toolkit, in Javascript
 - (also later Java version)
- marks with inherited properties
- pros
 - runs in browser
 - matches mark/channel mental model
 - also much more: interaction, geospatial, trees, ...
- cons
 - not all kinds of operations supported
- example app: NapkinVis (2009 course project)



[Fig 1, 3. Chao. NapkinVis.

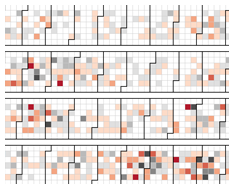
<http://www.cs.ubc.ca/~tmm/courses/533-09/projects.html#will>]

Protovis Validation

- wide set of old/new app examples
 - expressiveness, effectiveness, scalability
 - **accessibility**
- analysis with cognitive dimensions of notation
 - closeness of mapping, hidden dependencies,
 - role-expressiveness, visibility, consistency,
 - viscosity, diffuseness, abstraction,
 - hard mental operations

D3

- declarative infovis toolkit, in Javascript
- Protovis meets Document Object Model / CSS
- pros
 - seamless interoperability with Web
 - explicit transforms of scene with dependency info
- cons
 - even more different from traditional programming model
- example app: calendar (gallery of many)



[Fig 1a. Bostock, Ogievetsky, and Heer. D3: Data-Driven Documents. Proc InfoVis 2011.]

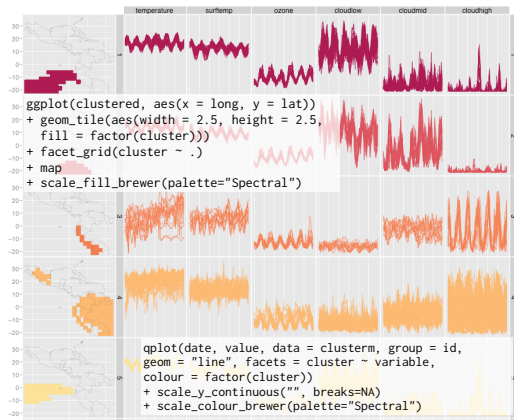
D3 Validation

- wide set of old/new app examples
- software performance
 - initialization, frame rates

Paper Types

- design studies
- technique/algorithm
- evaluation
- model/taxonomy
- **system**
 - today's emphasis

- declarative statistical graphics in R
- implementation of Wilkinson's Grammar of Graphics



[Slide 17. Wickham. ggplot2: past, present, and future.

<http://had.co.nz/ggplot2/resources/2007-past-present-future.pdf>]

Wrangler

- interactive data cleaning
- inference engine: system suggests applicable transforms
- programming by demonstration (vs complex regexps)
- declarative transformation language underlying

- <http://vis.stanford.edu/wrangler/>

Systems

- Tableau: general/powerful database vis
- Mondrian: statistical graphics
- ggobi: high-dimensional analysis

- load data directly, as opposed to build with toolkit

Resource Page: Software

<http://www.cs.ubc.ca/tmm/courses/533-11/resources.html>

Reading For Next Time

- **Mon Oct 31:** no class next week!

Graph Visualisation in Information Visualisation: a Survey. Ivan Herman, Guy Melancon, M. Scott Marshall. IEEE Transactions on Visualization and Computer Graphics, 6(1), pp. 24-44, 2000

Topological Fisheye Views for Visualizing Large Graphs. Emden Gansner, Yehuda Koren and Stephen North, IEEE TVCG 11(4):457-468 (Proc. InfoVis 2005), 2005.

Online Dynamic Graph Drawing. Yaniv Frishman and Ayellet Tal. Proc EuroVis 2007, 75-82.

Reminders

- this Friday: presentation topics due
- next Friday: written project proposals due
- next week: no class