InfoVis Systems & Toolkits

CS 7450 - Information Visualization
September 14, 2015
John Stasko

Background

- In previous classes, we have examined different techniques for presenting multivariate data
  - We’ll continue to show more later too

- Today we look at systems that implement these ideas and provide some of their own new visualization techniques
Agenda

- Toolkits that can be used to build systems
  - D3, Processing, ...
- Systems providing a view or views
  - Improvise, Many Eyes, Polaris, ...
- Commercial systems (to come our next class)
  - Spotfire, InfoZoom, Tableau, QlikView...

Toolkits & Infrastructures

- Set of components or capabilities that allow others to put together visualization systems
- Growing trend
Toolkit Design

• What would you include in a toolkit like this if you designed it?

History Lesson

InfoVis Toolkit

http://ivtk.sourceforge.net

Fekete
InfoVis '04
Heer++ Series

Series of toolkits from Jeff Heer and his research group

Prefuse

Java2D

Heer et al, CHI '05

What happened???

Web!!!
**Heer++ Series**

Series of toolkits from Jeff Heer and his research group

- **Prefuse**
  - Java2D
  - Heer et al., CHI ’05

- **Flare**
  - ActionScript & Flash
  - Bostock & Heer, TVCG (InfoVis) ’09

- **Protovis**
  - Declarative spec.
  - Heer & Bostock, TVCG (InfoVis) ’10

---

**D3: Data-Driven Documents**

- Newest entry in the Heer-Bostock line of toolkits
- “Not just an infovis toolkit”
- Javascript-based
- Very similar to Protovis...
  - Except makes use explicitly of web standards such as Scalable Vector Graphics (SVG) rather than a proprietary “marks” graphics set
D3 Design Pattern

- Declarative Syntax like Protovis
- Creating/Modifying selections of the HTML DOM
- “An elegant for-loop with a bunch of useful helper functions”
- Excellent support for changing data
  - Taking advantage of CSS3 Transformations and Transitions
- Integrates seamlessly into any webpage

D3 Website

D3 is a JavaScript library for manipulating documents based on data. D3 helps you bring data to life using HTML, SVG, and CSS. D3’s emphasis on web standards gives you the full capabilities of modern browsers without tying yourself to a proprietary framework, combining powerful visualization components with a data-driven approach to DOM manipulation.
D3 Support

- Active community online
  - Including Mike Bostock often answering questions

- Tutorial from Chad in our last class

Vega

https://trifacta.github.io/vega/
Characteristics

- Declarative visualization grammar like D3
- No CSS, DOM, etc., though
- Describe visualizations in JSON format
- Generate interactive views via HTML5 Canvas or SVG

Example Code

```json
{
  "width": 400,
  "height": 200,
  "padding": {"top": 10, "left": 30, "bottom": 20, "right": 10},
  "data": [
    {
      "name": "table",
      "values": [
        {"category": "A", "amount": 28},
        {"category": "B", "amount": 55},
        {"category": "C", "amount": 43},
        {"category": "D", "amount": 91},
        {"category": "E", "amount": 81},
        {"category": "F", "amount": 53},
        {"category": "G", "amount": 19},
        {"category": "H", "amount": 87},
        {"category": "I", "amount": 52}
      ]
    },
    {
      "name": "tooltip",
      "init": {},
      "streams": [
        {"type": "rect:mouseover", "expr": "datum"},
        {"type": "rect:mouseout", "expr": "{}"}
      ]
    }
  ],
  "signals": [
    {"name": "tooltip"},
    "init": {},
    "streams": [
      {"type": "rect:mouseover", "expr": "datum"},
      {"type": "rect:mouseout", "expr": "{}"}
    ]
  ]
}```
**Premise**

- Not really for humans to write
- Intended for tools to generate & exchange it

---

**Processing**

- Java based
- Unlike protovis & D3, not specifically designed for InfoVis
  - Data Reader? Layout algorithm?
  - But can definitely be used to build visualizations!
- Well documented, lots of tutorials with contributions from many people and even books

Ben Fry
Processing: the idea

- Programming as scripting
  - PDE: processing development environment
  - A program is called a *sketch*
  - written as a list of statements

![Processing Example]

Processing: Useful Functions

void setup() {
    //your own code here
}

void draw() {
    //your own code here
}

- These are built-in functions that are called automatically.
  - The setup() block runs once.
  - The draw() block runs repeatedly: good for animation
P5.js

• Interpretation of Processing in Javascript

---

Piccolo

---

http://p5js.org

http://code.google.com/p/piccolo2d
Characteristics

- Graphics toolkit with very nice built-in zooming and panning support
- Useful for implementing infovis too
- Will discuss more later in course...

WebGL


When you need 3D or really sophisticated graphics
More toolkits

Aperture.js

http://aperturejs.com/
**Bokeh**


**HighCharts**

http://www.highcharts.com/
dimple

On top of D3

JavaScript InfoVis Toolkit

http://philogb.github.io/jit/
Google Chart Tools

http://code.google.com/apis/visualization/documentation/

Characteristics

- Javascript-based
- Gallery of contributed code segments
- Visualizations are interactive
- Evolving API
Reflection

- What would you seek in a good infovis system or toolkit?

Toolkit Tradeoffs

- InfoVis-focused
  - Many fundamental techniques built-in
  - Can be faster to get something going
  - Often more difficult to implement something “different”
  - Documentation?

- Generic graphics
  - More flexible
  - Can customize better
  - Big learning curve
  - Doc is often better
  - Can take a long time to (re)implement basic techniques
Design Patterns

- Heer used his experience with prefuse etc to describe common design patterns for infovis
  - Reference model
  - Data column
  - Cascaded table
  - Relational graph
  - Proxy tuple
  - Expression

- Scheduler
- Operator
- Renderer
- Production rule
- Camera
- Dynamic query binding

Writing Code is Hard

- Why not just show what you want the visualization to look like?
  - What’s the challenge?
Lyra

- Interactive vis builder tool without needing to program
- Graphical “marks” are bound to data fields
- User shows what vis is to look like, the mapping from data
- Generates code (Vega) that can be run on the web

Satyanarayan & Heer
Computer Graphics Forum (EuroVis) ’14

iVisDesigner

- Interactively create mappings from data elements to graphical elements
- Has scatterplot, timeline, graph templates

Video

Ren, Hollerer, & Yuan
TVCG (InfoVis) ’14

Journalism-driven systems
Datawrapper

https://datawrapper.de/

Chartbuilder

http://quartz.github.io/Chartbuilder/
**Systems/Tools**

- Primarily examining academic/research systems that provide preexisting views
  - Commercial systems next time

**Polaris**

http://www.graphics.stanford.edu/projects/polaris/

http://www.tableausoftware.com

Stolte et al
TVCG '02

Video
Basis

- Relational databases
- Pivot tables from spreadsheets
- N-dimensional data cubes
- Analytic approach is fundamental
- Provides visual representations of these concepts

Visualization

- Table of data (rows, columns)
- Each axis may have nested dimensions
- Table entry is a pane, and has visual marks to represent data
- Analyst drags and drops fields from database schema onto shelves of display
- Much interaction supported
Graphics

- Formal table algebra provided to describe data
- Visual mappings established from data types to appropriate (good) markings and encodings

**Impact:** Became Tableau

Many Eyes

- InfoVis on the web
- Website developed from IBM’s infovis group
- Motivating infovis challenges:
  - Difficulty of creating new visualizations
  - How do you discuss the visualizations?
- Project goals:
  - Enable end-user creation of visualizations
  - Foster large-scale collaborative usage

Viégas et al *TVCG (InfoVis) ’07*
Features

- Provides data upload capabilities and choice from library of visualizations
- Includes
  - Gallery of recently uploaded visualizations for browsing
  - Chosen highlighted visualizations
  - Attached discussion forums for each vis

Recent gallery

Featured visualizations
Data

- Users upload their own data sets
  - All become public
- Format: table or unstructured text
  - Metadata allowed
- Immutable once uploaded
Visualizations

- Preloaded visualization types
  - Has grown over time
- User chooses one to combine with their data
- Provides named, typed slots that the user maps particular pieces of data to
  - System makes some reasonable guesses too
WordTree

Allows the user to control the mapping from data to image
Social Aspects

- Users identified by login ID
- Can leave comments about different visualizations
- Can take snapshot of visualization state
  - Unique URL
- “Blog this” button

Evaluation

- Quantitative, objective
- 1895 posts of March ‘07
- Wide variety of topics of visualizations and motivations for creating visualizations
- Does seem to be fostering discussion

Viégas et al
HICSS ’08
Evaluation

- Qualitative, subjective
- In-depth interviews with some ME users
- Visualizations used largely for communication and collaboration (not necessarily analysis)
  - Privacy and audience management a concern
- Highlights a number of interesting, non-expected uses of the technology

Status

- Shut down by IBM on June 12, 2015

Thoughts?

- What do you think of the design choices they made?

Sense.us

- Related, follow-up to Many Eyes
- Discussion and visualization of US Census data
- Go beyond Many Eyes in terms of annotation, collaboration, and discussion
Components

- **Doubly-linked discussion**
  - Can go from visualization to threaded discussion items or vice-versa

- **Graphical annotation**
  - Simple graphics editor and comments (like transparent layer)

- **Bookmark trail**
  - Small strip of views

- **Comment listings and social navigation**
  - Searchable, sortable indices and links

---

**Figure 1. The sense.us collaborative visualization system.**
(a) An interactive visualization applet, with a graphical annotation for the currently selected comment. The visualization is a stacked time-series visualization of the U.S. labor force, broken down by gender. Here the percentage of the work force in military jobs is shown. (b) A set of graphical annotation tools. (c) A bookmark trail of saved views. (d) Text-entry field for adding comments. Bookmarks can be dragged onto the text field to add a link to that view in the comment. (e) Threaded comments attached to the current view. (f) URL for the current state of the application. The URL is updated automatically as the visualization state changes.
Video

Sample annotations

Statwing

- Data analysis tool
- Upload spreadsheet or dataset, pick relationships you want to explore
- Basic graph types generated

https://www.statwing.com/
Keshif

- Import data from spreadsheets and csv/text files
- Provides multiple coordinated bar charts and histograms
- Large set of examples on website

http://keshif.me/
Some web collections
http://www.visualisingdata.com/resources/

http://www.creativebloq.com/design-tools/data-visualization-712402

https://github.com/showcases/data-visualization
More Next Time...

- Demos of commercial infovis systems
  - Spotfire
  - Table Lens/Eureka
  - SeeIt
  - Qlikview
  - InfoZoom
  - Tableau

HW 2 Feedback

- Things we noticed
- Example solutions
Project Topics Feedback

- Will try to return your proposals soon
- Get to work – Poster session is a few weeks away

Upcoming

- Commercial systems
  - Reading
    Spenke & Beilken '00

- Interaction
  - Reading
    Yi et al '07