Prefuse: a toolkit for interactive information visualization

Jeffery Heer, UC Berkeley
Stuart K. Card, Palo Alto Research Center
James A. Landay, University of Washington
- What is information visualization (infovis)?

  ◦ Visual representations of abstract information
  ◦ Demystify data and reveal hidden patterns
Existing visualizations
Existing visualizations

- **Constrained to one application domain**
  - E.g.: Polaris, table-based
  - DOI Trees, tree-based
What’s new in Prefuse?
What’s new in Prefuse?

composable & reusable
Prefuse:

- **What?**
  - An extensible java user interface toolkit for constructing interactive information visualization applications

- **Why?**
  - Support customized visualization, animation, and interaction

- **How?**
  - Application building by stringing together fine-grained, reusable components
An example using Prefuse

Space Distortion
142 Lines of Code
An example using Prefuse

Animated Radial Layout
190 Lines of Code
A tutorial: Implementing radial graph with Prefuse
A code example

```java
// create graph and registry
Graph g = new XMLGraphReader().loadGraph(datafile);
ItemRegistry registry = new ItemRegistry(g);

// initialize renderers
Renderer nodeR = new TextItemRenderer();
Renderer edgeR = new DefaultEdgeRenderer();
registry.setRendererFactory(
    new DefaultRendererFactory(nodeR, edgeR));

// initialize action lists
ActionList layout = new ActionList(registry);
layout.add(new TreeFilter(true));
layout.add(new RadialTreeLayout());
layout.add(new ColorFunction());
ActionList animate = new ActionList(registry,1500);
animate.setPacingFunction(new SlowInSlowOutPacer());
animate.add(new PolarLocationAnimator());
animate.add(new ColorAnimator());
animate.add(new RepaintAction());
animate.alwaysRunAfter(layout);

// initialize display
Display disp = new Display(registry);
disp.setSize(500,500);
disp.addControlListener(new DragControl());
disp.addControlListener(new FocusControl(layout));

// initialize enclosing window frame
JFrame frame = new JFrame("prefuse example");
frame.getContentPane().add(disp);
frame.pack(); frame.setVisible(true);
layout.runNow();
```

**Code Sample 1: Radial Graph Explorer**
A typical pipeline of infovis
Design of Prefuse

- Prefuse provides interfaces and default implementations of data structures for unstructured, graph and tree data
- Abstract Data:
  - Data element: entity with attributes
Design of Prefuse

- Filtering:
  - Process of mapping abstract data to a representation suitable for visualization then generate corresponding visual analogues.

Diagram:
- **DATA**
  - Abstract Data: Nodes, Edges
  - Filtering

- **VISUAL FORM**
  - Visual Analogues: VisualItems in ItemRegistry
  - Rendering

- **VIEW**
  - Display: Interactive Display
  - User

- **ActionList**
  - Filter
  - Layout
  - Color
  - Size
  - Renderers
Design of Prefuse

- Visual Analogues:
  - To arrange data and stored in a centralized structure called ItemRegistry to house a specific visualization
  - Prefuse provides several VisualItems to visualize different types of entities
Writing apps – a code sample

```java
// create graph and registry
Graph g = new XMLGraphReader().loadGraph(datafile);
ItemRegistry registry = new ItemRegistry(g);
```

- Load data
- Initialize ItemRegistry to house visualization
Design of Prefuse

• Actions:
  ◦ Composable modules that update the VisualItems
  ◦ Mechanism for selecting visualized data and setting visual properties, performing tasks such as filtering, layout, color assignment and sizing
Design of Prefuse

- **ActionsList:**
  - Configurable runnable Class that sequentially execute Actions
Writing apps – a code sample

```java
// initialize action lists
ActionList layout = new ActionList(registry);
layout.add(new TreeFilter(true));
layout.add(new RadialTreeLayout());
layout.add(new ColorFunction());

ActionList animate = new ActionList(registry, 1500);
animate.setPacingFunction(new SlowInSlowOutPacer());
animate.add(new PolarLocation Animator());
animate.add(new ColorAnimator());
animate.add(new RepaintAction());
animate.alwaysRunAfter(layout);
```

- Specify two ActionList:
  - Filter data to tree structure, apply radial tree layout and assign colors to nodes.
  - Add an animation transition for when the focus of the visualization changes.
Design of Prefuse

- **Renderer:**
  - Manage mappings between VisualItems and appearances
Writing apps – a code sample

```java
// initialize renderers
Renderer nodeR = new TextItemRenderer();
Renderer edgeR = new DefaultEdgeRenderer();
registry.setRendererFactory(
    new DefaultRendererFactory(nodeR, edgeR));
```

- Initiate renderers
  - Assign renderers to appropriate items
Design of Prefuse

- **Display:**
  - Perform presentation of visualized data
  - Apply view transformations
  - Support interaction with visualized items
Writing apps – a code sample

```java
// initialize display
Display disp = new Display(registry);
disp.setSize(500, 500);
disp.addControlListener(new DragControl());
disp.addControlListener(new FocusControl(layout));
```

• Present visualization:
  • Enable user to reposition nodes and select new focus by clicking on a node
Radial Graph using Prefuse

Animated Radial Layout
190 Lines of Code
Summary

- Prefuse:
  - A toolkit consists of composable, reusable units
  - Enables reuse and composition of visualization and interaction techniques

Evaluations

- How?
Summary

- Prefuse:
  - A toolkit consists of composable, reusable units
  - Enables reuse and composition of visualization and interaction techniques

Evaluations

- Application coverage
- Qualitative usability
Application Coverage

- **Goal:**
  - Test expressiveness and efficiency of the toolkit

- **Approach:**
  - Reimplement existing visualizations
    - e.g. Animated radial graphs, animated force-directed layout, the hyperbolic tree browser etc.

- **Results:**
  - **Flexibility:** implementations are greatly simplified
  - **Efficiency:** running time decreased from weeks or days to minutes
Qualitative Usability Study

Goal:
- Understand the learnability and usability of programming for other programmers

Approach:
- Given tutorial, observe 8 programmers of varying background and expertise
  - using the toolkit to build applications
  - interviewing them about experiences

Results:
- Programmers can use the toolkit to quickly build and tailor the visualizations
After Prefuse

Jeffrey Heer
University of Washington
Human-Computer Interaction, HCI, Visualization, Information Visualization, Visual Analytics
Verified email at cs.washington.edu - Homepage

<table>
<thead>
<tr>
<th>Title</th>
<th>Cited by</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td>D3: Data-Driven Documents</td>
<td>1216</td>
<td>2011</td>
</tr>
<tr>
<td>M Bostock, V Ogievetsky, J Heer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEEE Transactions on Visualization and Computer Graphics 17 (12), 2301-2309</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Prefuse: a toolkit for interactive information visualization</td>
<td>758</td>
<td>2005</td>
</tr>
<tr>
<td>J Heer, SK Card, JA Landay</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proceedings of the SIGCHI conference on Human factors in computing systems ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Vizster: Visualizing online social networks</td>
<td>731</td>
<td>2005</td>
</tr>
<tr>
<td>J Heer, D Boyd</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Crowdsourcing graphical perception: using mechanical turk to assess visualization design</td>
<td>394</td>
<td>2010</td>
</tr>
<tr>
<td>J Heer, M Bostock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proceedings of the SIGCHI Conference on Human Factors in Computing Systems</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
After Prefuse

Jeffrey Heer
University of Washington
Human-Computer Interaction, HCI, Visualization, Information Visualization, Visual Analytics
Verified email at cs.washington.edu - Homepage

<table>
<thead>
<tr>
<th>Title</th>
<th>Cited by</th>
<th>Year</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>D3: Data-Driven Documents</strong></td>
<td>1216</td>
<td>2011</td>
</tr>
<tr>
<td>M Bostock, V Ogievetsky, J Heer</td>
<td></td>
<td></td>
</tr>
<tr>
<td>IEEE Transactions on Visualization and Computer Graphics 17 (12), 2301-2309</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Prefuse: a toolkit for interactive information visualization</strong></td>
<td>758</td>
<td>2005</td>
</tr>
<tr>
<td>J Heer, SK Card, JA Landay</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proceedings of the SIGCHI conference on Human factors in computing systems ...</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Vizster: Visualizing online social networks</strong></td>
<td>731</td>
<td>2005</td>
</tr>
<tr>
<td>J Heer, D Boyd</td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Crowdsourcing graphical perception: using mechanical turk to assess visualization design</strong></td>
<td>394</td>
<td>2010</td>
</tr>
<tr>
<td>J Heer, M Bostock</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Proceedings of the SIGCHI Conference on Human Factors in Computing Systems</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>
D3 – Data-Driven Documents

- A JavaScript Library
- Help visualize data on web browser
D3 – not only for data scientists and data analysis

D3 in European MTV Awards
Ending

EXHIBITIONS

26 Mar 2005 False Profit: LIQUIDATE
False Profit, San Francisco, CA
Interactive installation of the Vizster social network browser.

8 Mar 2012 The Art of Networks
Florida Institute of Technology
Curator: Isabel Meirelles
Exhibition of the Stanford Dissertation Browser.

INVITED TALKS

24 Feb 2016 Predictive Interaction
Design@Large Seminar, UC San Diego San Diego, CA

Principles of Data Visualization
27 Jan 2016 Upper Columbia Science Conference Wenatchee, WA
5 Mar 2014 Keynote Address, Visualizing Biological Data (VIZBI) Heidelberg, Germany
Take away

- What is Prefuse and why it?
- How to use it?
- How did we evaluate it?
- What did you learn from author’s experience?