DEFOG
A System for Data-Backed Visual Composition

Lauro Lins, David Koop, Juliana Freire, Claudio Silva

SCI Institute, University of Utah

August 2010
Motivation

Examine Statistics of Workflow Evolution: Provenance: A First Study

Leon Li, Dudy Engel, Erik M. Anderson, Steven P. Goldhagen, Ramez Elmasri, John Hu, and Claudia T. Silber
ACM/IEEE/US National Computing Conference, University of Utah

1 Introduction

Provenance (also referred to as audit trial, lineage, and pedigree) captures information about the steps used to produce a given data product. Such information is a record of the decision-making processes, the contributions of individuals, and the history and provenance of the data. Provenance helps in understanding the causes that led to a given data result. It also supports the verification and validation of the data. Provenance is important for ensuring the reliability and trustworthiness of the data. It is a crucial component of reproducible research, and it plays a key role in scientific workflows. Provenance is also important for legal and regulatory compliance, such as in the healthcare industry. In this paper, we focus on the provenance of scientific workflows. We present a novel approach to capturing provenance information in scientific workflows and demonstrate its effectiveness.

2 Workflow Evolution: Provenance: Background

Scientific workflows involve a sequence of tasks, each performed by a different tool. These tasks are connected by data, and the workflow is often represented as a directed graph. Provenance information is typically captured as a directed graph, where each node represents a task and each edge represents the flow of data between tasks. This graph is then analyzed to understand the evolution of the workflow over time.

Fig. 4. Workflow Structural, Parameter and Layout Activity

Fig. 5. Plot of Branching Factors for the six tasks from two different users. The branching structure for Task 3 is depicted on the right.

Vistrails Screenshots + Adobe Illustrator
Python + R
Python + R
Python + R + GraphViz + Inkscape

SSDBM 2008
The DEFOG System

- Streamlines the process of creating visualizations
- Supports the composition of data-driven visualizations
- Allows users to explore creative data visualizations that are either hard or impossible to express with other tools
  - Users can compose novel representations by combining existing visualizations techniques

- Here is an example...
Analyzing WWW2010 Submissions

Input = XML file with WWW 2010 submissions
Analyzing WWW2010 Submissions

- Data Mining and Machine Learning
- Bridging Structured and Unstructured Data
- Software Architecture and Infrastructure
- User Interfaces and Rich Interaction
- Web Services and Service-Oriented Computing
- Social Networks and Communities
- Semantic Web
- Security and Privacy
- Internet Monetization
- Networking and Mobility
- Performance, Scalability and Availability
- Rich Media
- Search

Excel

DEFOG
Analyzing WWW2010 Submissions

DEFOG used to create visualizations and compose them for presentation
WWW 2010: Area Overlap
DEFOG Features

• Extensible set of data visualization techniques
  – Statistical plots (scatter plots, histograms, etc)
  – Graph drawing
  – Add your own!

• Flexible combination of visualizations and techniques:
  – Side by side, nested, linked
  – E.g., graph inside a plot, graph of plots, plots with elements connected to elements in other plots

• Integration of visual manipulation and programming

• Work in progress:
  – Provenance capture
  – Scalability---support millions of objects
DEFOG combines...

**Python**
- Language to represent data and computations
- Used to configure the “face programs” of the scene elements
- Python console to inspect anything at anytime

**Tableau**
- DEFOG adopts the drag-and-drop approach of Tableau to express data manipulation and visualization
- Tableau can be seen as a “face program” of DEFOG

**2D Vector Drawing**
- DEFOG combines free drawing with data analysis and visualization
- Operations supported by drawing tools are available in DEFOG (e.g. copy-and-paste, transform, free drawing, fine tuning)
DEFOG has been used to help scientists make sense of their data
Familial Association Between Cancer Sites

- Collaborators: Lisa Canon-Albright and Craig Teerlink
  Genetic Epidemiology, University of Utah

Excel Spreadsheet

DEFOG

JNCI Submission

Cancer Sites Network
Biochemical Pathways

- Collaborators: Elizabeth Skovran and Mary Lidstrom
  Lidstrom Laboratory, University of Washington

Excel Spreadsheet -> DEFOG -> Data Analysis
Our laboratory studies the molecular basis for type 2 diabetes as well as the role iron plays in development of the diabetic phenotype... We have run ~136 mice producing over $2.8 \times 10^{-7}$ data points.

We have started using DEFOG to visualize these data. This has led to one novel observation. That mice shift from fatty acid oxidation during the day (inactive period) to glucose oxidation at night (active period) as measured by the RER fluctuation. Highlighting the circadian shift in nutrient utilization during a 24 hour cycle.”

Robert Cooksey
DEFOG vs. Other Visualization Tools
### Visualization Programming Languages

**Protovis**

```javascript
// Create the root panel and set the visualization's size to 150x150
var viz = new pv.Panel();
viz.width(150);
viz.height(150);

// Add the horizontal rules (grid lines), we add them first so they go in the back.
viz.add(pv.Rule());
.vdata(pv.Range(0, 2, .5));
.bottom(function(d, i, M) { return (.4 * 80) + 1; });
.vadd(pv.Label());

// Add the bars with the height corresponding to the values in the data property
viz.add(pv.Bar());
.vdata([1, .2, 1.1, 1.3, .7]);
.vinit(5); // size
.bottom(0); // bottom
.left((function() { this.index < 3 ? 0 : 30; })); // this.index is the position of the datum in the array
.add(pv.Label()); // Add a label to the bottom of each bar

// Render everything.
viz.render();
```

**Microsoft Research Veda**

```javascript
myData = DataSet("mydata.csv");
currentYear = slider.Value + 1900;
bubbles := from row in myData
where row.Year == currentYear
select new Circle()
| X = row.Latitude,
Y = row.Longitude,
Radius = row.Population * scalingFactor,
Fill = BlackBodyPalette(1., 1., row.DeltaCarbon)
|
Scene["USMap"].Add(bubbles);
```
Visualization Programming Languages

- These languages provide constructs that make it easier to generate visualizations than general programming languages.
- But visualizations are still created programmatically.
- They can be integrated with DEFOG.

```javascript
X = row.Latitude,
Y = row.Longitude,
Radius = row.Population * scalingFactor,
Fill = BlackBodyPalette(1., 1., row.DeltaCarbon)
};
Scene["USMap"].Add(bubbles);
```
Tableau

• Database visualization system
• Drag-and-drop interface to explore data visualizations
  – Hides the underlying query language from the users
• Supports efficiently data exploration through “basic” visualizations
• Doesn’t support rich visualizations which are needed to understand more complicated datasets (the norm in science)
• The Tableau way of expressing “basic” visualizations can be integrated into DEFOG as a “face program”
MSR Live Labs Pivot

• Visual interaction with a collection of records (all of the same type?)
• Potential to be the core of an interface to explore Web search results (not there yet)
• Not suitable for exploratory tasks which involve heterogeneous data (not in tabular format)
MSR NodeXL

• Add network drawing and algorithms to Excel spreadsheets
• Excel is widely used and network datasets (many already) in Excel can benefit from NodeXL
Potential Synergy

DEFOG

SQL Server
Protovis
NodeXL
Excel
PowerPoint
Word
Thank you
DEFOG and Provenance Analytics
DEMO