Paper Presentation: A Model and Framework for Visualization Exploration

Written By T.J. Jankun-Kelly et al.
Presented by Michael Shah
September 26, 2012
Duration ~10-15 minutes
What is Visualization Exploration?

"Visualization exploration is the process of extracting insight from data via interaction with visual depictions of that data" (357).
What do we want from a Visualization Exploration Framework

- A concrete formal model to describe the exploration of a visualization
  - Can assist users
    - Where have you been, where are you going, etc.
  - Can give developers insight
    - We really want that perfect metadata model—although it is not the goal here.
  - Avoid redundancy
  - A tool better than interaction logs
  - A generalizeable tool that can be applied to many software frameworks
Modeling User Interaction within Visualization

- Cyclic Process - Users fine tune
  - Interactive Parameter Control
    - Parameters updated by user, and viewed as user requests them
  - Dynamic Manipulation
    - New parameters generated from old parameters
Fundamental Operation of Visualization Exploration

"The fundamental operation of the visualization exploration process is the application of a set of parameter values to the visualization transform to generate a visualization result" (359).
P-Set Model

- **Goal:** Encapsulate interactions a user has with system, and capture a greater exploration session.
  - **Elements**
    - Visualization Transforms
    - Parameter Values
    - Results
    - Derivation
P-Set Model (continued)

- Visualization transform
  - Takes in arguments, gives a result
    - (label+args+result=signature)

- Visualization parameter
  - It's type (color map, heat map, etc.)
  - Corresponding values (color and data values)

- Visualization results
  - Identified with P-Set, stores the actual value

- Derivations
  - Describes relation of past & previous results
  - Contains: timestamp, parameter derivations, p-set derivations, and results
What is not captured?

- Does not specify the components of the visualization transform (no small suboperations)
- Does not describe the form of the parameter and result values (no data model for the values)
Representation

- XML format (See Appendix B for full schema)
  - Appears to do its job - it's convenient and flexible
  - Easily Shared
  - Parsed and transformed for analysis later
(a) Some new value is derived  
(b) Parameter sets are added together  
(c) A new result is obtained by applying parameter set to visualization transform
Derivation Calculus

Parameters: \((s_1[n_1], \ldots, s_i[n_j]) \xrightarrow{\delta p} (p_1, \ldots, p_k)\)

P-Sets: \(\{p_1, \ldots, p_m\} | s_{in} \xrightarrow{\delta s} s_{out}\)

Results: \(t(s_{out}) \xrightarrow{\delta r} r\).

*s_1*[param type] - input \hspace{1cm} p_1 - output parameter

S_{in} \rightarrow S_{out} - How parameters in existing set changed

t(S_{out}) \rightarrow r : Transform of P-set that generates result.
Example 3a - Focus+Context Graph

\[
(s_1 \ [\text{focus}_1]) \xrightarrow{\delta_p} \ (\text{focus}_2)
\]

\[
\{\text{focus}_2\} | s_1 \xrightarrow{\delta_s} s_2
\]

\[
t_{\text{graph}} (s_2) \xrightarrow{\delta_r} r_2
\]
Example 3b

\[ (s_3 \ [\text{view}_3]) \xrightarrow{\delta_p} (\text{view}_4), \ (s_4 \ [\text{view}_4]) \xrightarrow{\delta_p} (\text{view}_5) \]

\{\text{view}_4\} | s_3 \xrightarrow{\delta_s} s_4, \ \{\text{view}_5\} | s_4 \xrightarrow{\delta_s} s_5

\[ t_{dvr} (s_4) \xrightarrow{\delta_r} r_4, \ t_{dvr} (s_5) \xrightarrow{\delta_r} r_5 \]
Software Framework

- Core layer: Contains P-Set Model
- Representation layer: XML Layer
- Interaction layer: Couple with interface/vis

"To adapt the framework to a new visualization method, only classes describing the visualization transform, parameters, and results need to be created. Result and parameter types may be common to several transforms and could thus be reused."

- Do I believe in it?
Example 1: Image Graph-based Volume

Fig. 5. Representation of a brain vessel visualization performed using an Image Graph. The feature of interest is the bulge in the lowest vessel in image e (top left image, captions added for clarity). During the visualization, the user dragged the zoom edge going to image e over the edge to b to zoom the other images in the Image Graph (top right). These derivations, including the propagation of the zoom factor from e to results f and g, are recorded using the derivation calculus (middle). The session can also be represented visually (bottom).
Example 2: Network Traffic (before)
Example 2: Network Traffic (after)
Example 3: Web Sheet
Conclusions and Future Work

- Good for training (transforms can be reproduced and replayed)
- Captures order of derivation and parameter results
- Clearly it is being used in the 3 examples provided
- Appears to be efficient in capturing only necessary transforms as specified in software framework
Thoughts While Reading

● Where can this framework be applied outside the examples given?
● Is the granularity enough?
  ○ How would a "visualization transform derivation" model strengthen this?
  ○ How about meta-data?
● Can we learn things beyond salient details?
● If you can build a formal model, that's usually a good sign of progress.
● Was there too much fluff in this paper?
Paper Presentation: A Model and Framework for Visualization Exploration

Written By T.J. Jankun-Kelly et al.
Presented by Michael Shah
September 26, 2012
Duration ~10-15 minutes
Suggested Paper to Read Later