Challenges in Evaluating Information Visualizations

Kasper Hornbæk
University of Copenhagen
Concerns in evaluation

• Increase in InfoViz studies that include evaluation
• But concerns over those studies remain, see  
  – The BELIV workshops  
  – Reviews of InfoViz papers (e.g., Munzner, 2008)
• Concerns are shared by research in  
  – Usability evaluation (e.g., Gray & Salzman, 1998)  
  – CHI (e.g., Greenberg & Buxton, 2008)
Conceptual challenges

• Observations about (some) InfoViz studies:
  – Win-lose evaluations
  – Point designs
  – Lack of clarity in key constructs
  – Lack of theory-driven experiments
Example: The notion of overview

• An overview
  – “is constructed from, and represents, a collection of objects of interest” (Greene et al., 2000, p. 381)
  – “implies a qualitative awareness of one aspect of some data, preferably acquired rapidly and, even better, pre-attentively: that is, without cognitive effort” (Spence, 2007, p.19)
The notion of overview, cont’

- 1391 occurrences of overview in 60 studies analyzed
- Used in three senses:
  - 81% technical (overview+detail visualization)
  - 14% user-centered (user forming an overview)
  - 5% other ( “give an overview of the literature”)

The notion of overview, cont’

• This review shows
  – Confusion about what an overview is
  – Few studies of the psychology of overview
  – A difference between overviews and overviewing

• Does an overview give an overview?

Example: Fisheye interfaces

- DOI function composed of a priori interest and distance components

\[
DOI(x|\cdot) = F(API(x), D(\cdot, x))
\]
Fisheye interfaces, cont’

• Mixed empirical results (e.g., Cockburn et al. 2008)
  – some optimistic (e.g., papers at CHI2010),
  – some pessimistic (e.g., Lam & Munzner 2008, “we do not know when, how or even if they are useful.”)

• Is a priori interest useful?
public class SimpleDateFormat extends Format {

    protected List parsePattern() {
        String[] ERA = symbols.getERAs();
        String[] months = symbols.getMonthNames();
        String[] shortMonths = symbols.getShortMonths();
        for (int i = 0; i < length; i++) {
            int tokenLen = token.length();
            Rule rule;
            rule = new TextField(Calendar.ERA, ERA);
            break;
            rule = selectNumberRule(Calendar.YEAR, tokenLen);
            if (tokenLen > 4) {
                rule = selectNumberRule(Calendar.YEAR, tokenLen);
            } else {
                rule = TwoDigitYearField.INSTANCE;
            }
            break;
            rule = new TextField(Calendar.MONTH, months);
            if (tokenLen > 4) {
                rule = new TextField(Calendar.MONTH, months);
            } else if (tokenLen == 3) {
                rule = new TextField(Calendar.MONTH, shortMonths);
                } else if (tokenLen == 2) {
            case 'd': // day in month (number)
            case 'h': // hour in am/pm (number, 1..12)
            case 'H': // hour in day (number, 0..23)
            case 'm': // minute in hour (number)
            case 's': // second in minute (number)
            case 'z': // millisecond (number)
            case 'Z': // day in week (text)
            case 'W': // week in year (text)
            case 'U': // week in year (number)
        }
    }

    protected NumberRule selectNumberRule(int field, int padding) {
Fisheye interfaces, cont’

- Our data suggests maybe
  - Hard to understand a priori components of the DOI function
  - Appears less useful in real-life programming than lines linked directly to the users’ focus
- But no one (?!?) has generated direct evidence about this question

Jakobsen & Hornbæk, LNCS (2011)
package org.eclipse.nebula.widgets.gallery;

public class DefaultGalleryItemRenderer extends AbstractColor

    Color foregroundColor;
    boolean showLabels = true;

    public boolean isShowLabels() {
        public void setShowLabels(boolean showLabels) {

        public DefaultGalleryItemRenderer() {

            // Set defaults

            foregroundColor = Display.getDefault().getSystemColor(Darken);

            backgroundColor = Display.getDefault().getSystemColor(Darken);

            selectionForeground = Display.getDefault().getSystemColor(Darken);

            selectionBackgroundColor = Display.getDefault().getSystemColor(Darken);

            // Create drop shadows

            createColors();

            public void draw(GC gc, GalleryItem item, int index) {

                gc.setForeground(foregroundColor)

                public void setDropShadowSize(int dropShadowSize) {

                    // TODO: force redraw

                    private void createColors() {

                        dropShadowColors.add(c);

                        Jakobsen & Hornbæk, CHI (2009)
Strong inference

• Platt (1964) proposed the notion of strong inference, with the steps of
  – Devising alternative hypotheses
  – Devising a crucial experiment which will rule out some hypotheses
  – Carry out the experiment
Radical solutions

• Newman (1994) reviewed CHI publications and found 25% radical solutions

RS Radical Solution:
A radical solution to the problem of <problem definition> is described, based on <solution strategy>. In comparison with <existing normal solutions> it offers <advantages>, which have been demonstrated in preliminary tests, but it leaves a number of side-effects to be addressed including <list of side-effects>. Strategies are suggested for addressing these side-effects.
Practical challenges
Simple outcome measures

- Binary task completion
- Spatial accuracy
- Error rates
- Comprehension/learning
- Recall
- Completeness
- Expert grading of product

Simple process measures

• 57% of studies measure task completion time
• Task completion time work as a summary measure of the process
• How should we interpret task completion time?

Example: Comparing text viz.

**Executable Object Modeling with Statecharts**

David Harel & Eran Gery, Computer, JULY 1997, 30 no. 7, 31-42

Statecharts, popular for modeling system behavior in the structural analysis paradigm, are part of a fully executable language set for modeling object-oriented systems. The languages form the core of the emerging Unified Modeling Language.

Models for the development of object-oriented systems should be behaviorally expressive and rigorous as well as intuitive and well structured. Thus, any modeling approach must be detailed and precise enough to produce fully executable models and permit the automatic synthesis of efficient code in languages such as C++.

Most OOC modeling methodologies specify a model through graphical notations. Entity-relationship-like diagrams typically specify object classes and their interrelationships, and there is some way to describe which objects do and how they interact. Most methodologies also adopt a state-based formalism to specify behavior, using statecharts[1] or some sublanguage thereof.

However, many methodologies fail to rigorously define the semantics of the languages. Without a rigorous semantic definition, precise model behavior over time is not well defined and full executability and automatic code synthesis is impossible. Adopting a richly expressive behavioral language like statecharts makes modeling easier, but requires great care in defining the way it integrates with the other parts of the model. Statecharts must capture not only the state of the object as a precondition to service requests, but also the dynamics of the objects' internal behavior in responding to those requests and in maintaining relationships with other objects.

These issues are complicated and go beyond recommending a modeling approach or methodology—they are language design concerns, requiring rigorous mathematical underpinnings. Both syntax and semantics must be fully worked out. Any possible combination of constructs must be clearly characterized as semantically legal or illegal, and each legal combination must be given a unique and formal meaning.

To address these needs, we embarked on an effort to develop an integrated set of diagrammatic languages for object modeling, built around statecharts, and to construct a supporting tool that produces a fully executable model and allows automatic code synthesis. The language set includes two constructive modeling languages (languages containing the information needed to execute the model or translate it into executable code):

- Object-model diagrams: specify system structure by identifying object classes and their multiplicities, object relationships and roles, and subclassing relationships.
- Statecharts: describe system behavior. A statechart attached to a class specifies all behavioral aspects of the objects in that class.

Hornbæk & Frøkjær, TOCHI (2003)
Longer and more frequent exploration

Time for further exploration (min.)

Overview+Detail (N=59)  Linear (N=40)
Standing on the shoulders of giants

• A key challenge is to build on the work of others in selecting:
  – Data
  – Tasks
  – Measures
  – Interfaces

• Crucial for replications, accumulating knowledge, validity/reliability
The system is …

I feel …
accessible, adequate, annoying, anxiety, appealing, boring, clear, cluttered, comfortable, competent, comprehensible, conclusive, confident, conflict, confusing, connected, convenient, desirable, difficult, dislikable, dissatisfied, distracting, easy, effective, efficient, embarrassed, emotional, engaging, enjoyable, entertaining, enthusiasm, excellent, exciting, familiar, favorable, flexible, flustered, friendly, frustrating, fun, good, hate, helpfulness, immediate, important, improving, inefficient, intelligent, interested, intuitive, involved, irritation, learnable, likable, lively, loved, motivating, natural, nice, personal, plain, pleasant, preference, presence, productive, quality, quick, relevant, reliable, respect, responsive, satisfied, sensate, sense of being together, sense of control, sense of success, simple, smooth, sociable, social presence, stimulating, successful, sufficient, surprising, time consuming, timely, tiring, trust, uncomfortable, understand, useful, user-friendly, vexed, vivid, warm, well-organized.
Reliability of questionnaires

- Homegrown questionnaires have low reliability, 6 are below .7

<table>
<thead>
<tr>
<th>Questionnaire</th>
<th>N</th>
<th>Cronbach’s $\kappa$</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Mean</td>
</tr>
<tr>
<td>Standard questionnaires</td>
<td>16</td>
<td>.814</td>
</tr>
<tr>
<td>(e.g., TLX, QUIS, CSUQ)</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Homegrown</td>
<td>20</td>
<td>.736</td>
</tr>
</tbody>
</table>

Table 1: Reliability of satisfaction questionnaires as indicated by Cronbach’s $\alpha$ [6]. Homegrown refers to questionnaires that authors themselves developed to capture ease-of-use.

Hornbæk & Law, CHI (2007)
Example: Selecting tasks

- Tasks are crucial
- But chosen ad hoc, to match evaluation, or habitually
- Which task would you use to evaluate an overview?
Table 3
Tasks in the 24 papers presenting laboratory experiments.

<table>
<thead>
<tr>
<th>Task</th>
<th>Frequency</th>
<th>Sample task</th>
</tr>
</thead>
<tbody>
<tr>
<td>Monitoring</td>
<td>3</td>
<td>Monitor an interface to receive alerts about possible attacks, determine the cause of the alerts, and decide if an attack had occurred (Thompson et al., 2007)</td>
</tr>
<tr>
<td>Navigating</td>
<td>11</td>
<td>Navigate to specific calendar appointments or monthly views (Bederson et al., 2004)</td>
</tr>
<tr>
<td>Exploring</td>
<td>13</td>
<td>Find the most influential paper(s) or author(s) in a citation database (Hornbæk et al., 2002)</td>
</tr>
<tr>
<td>Understanding</td>
<td>8</td>
<td>Write a one-page essay stating the main content of a document just read (Hornbæk and Frøkjær, 2003)</td>
</tr>
<tr>
<td>Planning</td>
<td>3</td>
<td>Plan a weekend-long trip to Stockholm considering all the necessary details of transportation, accommodation, and possible activities (Jhaveri and Räihä, 2005)</td>
</tr>
<tr>
<td>Other</td>
<td>1</td>
<td>Set parameters defining the animation of an overview display to achieve smooth and fluent animation (van Wijk and Nuij, 2003)</td>
</tr>
</tbody>
</table>

Note: 12 experiments involved only a single category of task, 9 experiments involved two categories of task, and 3 experiments involved three categories of task.
What to do?

• More strong, theoretically motivated comparisons (TILCS?)
• More complex measures of outcome and process, coupled with richer data
• Build on existing work, replications
• Task taxonomies, do task-level analysis, do field work
• Studies of adoption and integration