Info Vis

Article

Information Visualization
2015, Vol. 14(3) 250–269
© The Author(s) 2013
Reprints and permissions:
sagepub.co.uk/journalsPermissions.nav
DOI: 10.1177/1473871613513228
ivi.sagepub.com
SAGE

Patterns for visualization evaluation

Niklas Elmqvist and Ji Soo Yi

Abstract

We propose a pattern-based approach to evaluating data visualization: a set of general and reusable solutions to commonly occurring problems in evaluating visualization tools, techniques, and systems. Patterns have had significant impact in a wide array of disciplines, particularly software engineering, and we believe that they provide a powerful lens for characterizing visualization evaluation practices by offering practical, tried-and-tested tips, and tricks that can be adopted immediately. The 20 patterns presented here have also been added to a freely editable Wiki repository. The motivation for creating this evaluation pattern language is to (a) capture and formalize "dark" practices for visualization evaluation not currently recorded in the literature, (b) disseminate these hard-won experiences to researchers and practitioners alike, (c) provide a standardized vocabulary for designing visualization evaluation, and (d) invite the community to add new evaluation patterns to a growing repository of patterns.

Keywords

Visualization, evaluation, validation, patterns, best practices, pattern language

Introduction

Evaluating data visualization systems is generally held to be difficult,[1,2] even to the point where it is seen as a black art consisting of equal parts prior experience and trial and error. Why is this the case? Visualization systems are generally designed to scaffold high-level cognitive activities, such as understanding particular phenomena, finding insight about a problem, and making a decision in the face of complex or massive data.[3] Such high-level tasks are difficult to isolate, characterize, and quantify. Furthermore, it is far from clear that a bottom–up model of assembling higher level tasks from many low-level tasks is a valid approach.[4] As a result, visualization papers tend to have a much lower incidence of evaluation than papers in the broader discipline of human–computer interaction (HCI): Lam et al.[3] show that for the four major visualization venues (EuroVis, 2002–2011; InfoVis, 1995–2010; IVS, 2002–2010; and VAST, 2006–2010), over half of the papers (489 out of 850; 57%) included *no* evaluation of any kind (not even a case study). In comparison, even a cursory read of the proceedings of leading HCI conferences, such as the ACM CHI conference, will show that the vast majority of HCI papers do include at least some form of evaluation.

Of course, validation through empirical evaluation is key to the scientific method and is a cornerstone for most scientific domains. The field of visualization has certainly reached well past the breakthrough stage in its development,[5] but many visualization papers still insist on "validation through awesome example": merely showing pictures of a visualization (stills and video) in the hope of convincing the reader. For the field of visualization to move solidly into replication, empiricism, and theory (see the Breakthrough, Replication, Empiricism, Theory, Automation, Maturity

Purdue University, West Lafayette, IN, USA

Corresponding author:
Niklas Elmqvist, Purdue University, 465 Northwestern Avenue, West Lafayette, IN 47907-2035, USA.
Email: elm@purdue.edu

(BRETAM) model[5]), empirical evaluation with human subjects is vital.

What is needed to promote more and higher quality evaluation in visualization? Despite the difficulty of evaluating visualization systems, it can certainly be done and has in fact been very successfully demonstrated in recent work (e.g., Andrews et al.,[6] Dwyer et al.[7] and Kang et al.[8]). Individual researchers possess vast amounts of tacit knowledge about visualization evaluation that is not formally recorded in the literature. In other words, the problem is perhaps not so much how to effectively evaluate visualization, but rather how to democratize this existing know-how across the entire scientific community. Stated differently, the question is how we can collect best practices from existing work and communicate these to a broader audience. Recent literature has done just that by discussing the different abstraction levels and pitfalls of visualization evaluation,[9,10] as well as by categorizing representative scenarios of evaluation studies of visualizations.[3] However, these efforts are all fairly high level, and there is no natural venue for sharing the nifty tips and tricks of visualization evaluation that individual practitioners and researchers have accumulated over the years.

To remedy this state of affairs, we present a pattern-based approach to visualization evaluation: essentially, a set of general and reusable solutions to commonly occurring problems in evaluating tools, techniques, and systems for visual sensemaking. The patterns' concept was originally introduced for urban planning[11] and has since become a powerful tool for capturing best practices in many domains, particularly in software engineering.[12] The patterns presented in this article are examples of hard-won, tried-and-tested ideas that will be useful while conducting visualization evaluation studies. In fact, some of these patterns capture evaluation practices that experienced visualization researchers are already doing but do not talk about; the aforementioned "dark" knowledge that novices to the field may find difficult to access.

The purpose for creating such an evaluation pattern language is to (a) capture and formalize existing practices for visualization evaluation, (b) disseminate this hard-won experience to researchers, students, and practitioners alike, (c) provide a standardized vocabulary for designing visualization evaluation, and (d) invite the visualization community to design, derive, and discuss new evaluation methods within the context of a growing pattern repository.

This article is a significantly extended version of a research article[13] presented at the BELIV 2012 (Beyond Time and Errors: Novel Evaluation Methods for Visualization) workshop colocated with IEEE VisWeek 2012 in Seattle, Washington. The new material in this version includes a collection of definitions, several new evaluation patterns, and a revised categorization of existing patterns.

Background

In this section, we first establish a common vocabulary for evaluation with human subjects. We then explore evaluation in visualization as well as the patterns' movement in different domains. For more background on evaluation in HCI and visualization, see a general HCI textbook (such as Rogers et al.[14]) as well as prior work by Plaisant,[2] Carpendale,[1] and Lam.[15]

Definitions

Evaluation with human subjects (or, more preferably, human *participants*[16]) is common in HCI and takes many shapes and forms, such as interviews, focus groups, cognitive walkthroughs, expert reviews, and participatory design. One of the most basic classifiers for evaluation methods is whether they are quantitative or qualitative. *Quantitative evaluation* focuses on collecting performance measurements, for example, on time and errors, that can be analyzed using statistical methods. *Qualitative evaluation*, on the other hand, collects more in-depth and free-form data, such as observations, notes, and transcripts, and is often used for more exploratory or explanatory purposes. Additional dimensions exist, like whether using the evaluation for formative or summative purposes, conducting the evaluation in the laboratory or in the field, and at a single time or over a longer time period. Furthermore, some studies do not fit cleanly in any one of these categories; it is certainly possible to collect qualitative data while performing a mainly quantitative experiment (such as free-form comments in a posttest survey) and vice versa (recording mouse interaction while engaging a domain expert in a structured review).

While there exists a bewildering array of terms in HCI evaluation, we will use a basic set of definitions in this article. Please note that we make no claims that these definitions are definite and general beyond the scope of this article. We define an *evaluation study* (also known as just "evaluation," "user evaluation," or "user study"—the term "user" is ubiquitous in HCI evaluation but actually somewhat problematic: first, we are most often evaluating usage as opposed to the users themselves, and second, the term itself has somewhat negative connotations (e.g., drug user)) as an empirical inquiry with the goal of answering one or several research questions. Evaluation studies generally consist of one or several *experiments* (or "user experiments")—an orderly procedure conducted to verify, refute, or derive one or several hypotheses—although this is not always the case (for example, an evaluation study can be entirely observational). We also adopt the notion of expanded boundaries of evaluation studies, which also include exploration and/or problem characterization phases.[3] A *participant* (the term "subject" is commonly used instead of "participant," but its use is generally depreciated) is an individual participating in an evaluation study; most studies involve several participants (from a handful to hundreds).

Experiments generally engage participants in performing certain *tasks*: an activity that the participant is asked to accomplish that is representative of an experimental hypothesis. Quantitative experiments, often called *controlled experiments*, generally involve comparisons in task performance between different participants and different experimental conditions. An *experimental condition* is a complete set of values for the *factors*, also known as *independent variables*, of the experiment. Factors are variables that the experimenter has determined may potentially affect task performance; examples include the size of the dataset, the amount of screen space available, and indeed, the visualization technique used. The instantiation of a task with a particular participant, data, and experimental condition is often called a *trial*. Controlled experiments generally measure quantitative *metrics*, or *dependent variables*, for each trial; for example, the trial completion time, the accuracy, and the number of mouse clicks. Analysis of this data is then often performed using inferential statistical methods that allow for making probabilistic statements as to the influence of any factors on task performance. For qualitative studies, on the other hand, analysis often takes less formal and more interpretative methodology; one example is the grounded evaluation proposed by Isenberg et al.[17]

Evaluation studies, regardless of being qualitative or quantitative in nature, can either take place in a single session or may be spread out in time and span several sessions. The latter is known as a *longitudinal study*. Furthermore, the *validity* of a study or an individual experiment is a measure of its degree of well-foundedness and is often divided into different aspects of the study: (a) *internal validity* being the degree to which the outcome is a function of the controlled parameters of the experiment, (b) *external validity* being the degree to which (internally valid) results can be generalized, and (c) *ecological validity* being the degree to which results can be applied to the real world outside of the research and laboratory setting. External validity can be easily confused with ecological validity, but they are different concepts.

Evaluating visualization

Several visualization systems and techniques have been evaluated using low-level quantitative studies. Examples include work on graphical perception,[18–20] animation,[21–23] and navigation[24–26] for visualization.

However, empirical evaluation for visualization beyond time and error is difficult.[1,2] This is mostly due to the open-ended nature of most visualization tasks, which makes designing relevant quantitative metrics difficult,[27] as well as due to the large individual differences among participants (i.e. the participant's innate and learned ability in understanding visual representations or background knowledge).[28,29] For example, evaluating a canonical visualization task such as investigative analysis has been proven to be especially difficult.[8] Furthermore, it is also not clear that generalizing performance for higher level tasks from many low-level tasks is a valid approach.[4] This is also the reason for the emphasis on qualitative and exploratory user studies of visualization in the literature.

Several important examples of qualitative evaluations exist. Seven common scenarios of evaluation studies have been identified through extensive review of existing literature by Lam et al.,[3] which will provide a good overview. Separate efforts by Bier et al.[30] and Jeong et al.[31] studied professional analysts solving investigative tasks for intelligence and financial domains. Kang et al.[8] conducted a between-subjects study of novice analysts using the Jigsaw[32] system to find a hidden threat in a large dataset of text reports. They used external graders to score results, but focused on qualitative observations rather than quantitative measures when reporting the results from the overall study. Saraiya et al.[33] used free-form insight reports to collect findings for microarray data analysis. Most recently, Kwon et al.[34] adapted insight-based evaluation to an investigative analysis task similar to that of Kang et al.,[8] focusing on the role of time in sensemaking.

An interesting trend in visualization evaluation is to use observations and results on how people manually perform particular tasks to inform visualization design. Both Isenberg et al.[35] and Robinson[36] used qualitative pen-and-paper studies to understand collaboration patterns for analysts working together on a complex task. Similarly, Van Ham and Rogowitz[37] qualitatively studied how people manually organized graph layouts, deriving several recommendations for graph layout algorithms. Dwyer et al.[7] later followed up van Ham and Rogowitz's work by explicitly comparing automatic and user-generated graph layouts based on user performance for several graph tasks using the various layouts generated in an earlier phase of the study.

Patterns

Patterns were originally introduced by architect Christopher Alexander for describing best practices on all levels of scale in urban planning, and a language of some 253 such patterns was assembled in a 1977 book on the topic.[11] The purpose of a *pattern* is to succinctly capture proven solutions to common problems in a reusable form that is accessible even to non-experts; one of the original intentions with Alexander's urban planning pattern language was to give ordinary people, and not just professionals, the means to design their local communities to fit their own needs.

Since their original use, the patterns' concept has been adopted by many domains as a powerful mechanism for capturing and communicating best practices in design; examples include game design,[38] pedagogy, communication policy, and even chess strategy. Perhaps most famously, patterns were adopted by the software engineering community in the 1994 Gang of Four book[12] and has since had a prominent place in computer science practice.[39] Heer and Agrawala[40] extended this tradition in 2006 to visualization software by proposing 12 new design patterns that are prevalent in building visualization software. However, it is important to note that these patterns deal with the mechanics of implementing visualization using programming and has nothing to do with evaluation.

Evaluation patterns

A *visualization evaluation pattern* is a proven solution to a common problem encountered when evaluating a visualization system. These patterns are reusable in the sense that "you can use this solution a million times over, without ever doing it the same way twice" (Alexander et al.,[11] p. *x*). More specifically, the purpose of adopting this concept for visualization evaluation is to provide a catalog of best practices that other researchers can easily adopt in their own work.

In the below treatment, we discuss the anatomy of a pattern, the methodology we followed to identify evaluation patterns, and the repository of patterns that we have created. We then present our list of patterns and their high-level characteristics. The following sections describe each of our 20 patterns in detail.

Anatomy of a pattern

Patterns are generally specified in terms of five basic components regardless of application domain (software engineering, education, design, etc.):[12]

- Name: a handle used to denote the pattern. The name should be in the form of a capitalized noun and should often be one or two words (with some exceptions). An illustrative name helps make the pattern become part of our design vocabulary and eases communication about a pattern between collaborators.
- Problem: a description of the problem and its context where the pattern can be applied.
- Solution: how to solve the problem in a reusable and flexible way. The solution is described in general terms without talking about specific solutions. This is so that the same solution can be applied to the same problem to produce different concrete designs.
- Consequences: applying a specific pattern will always have repercussions on the evaluation on both global and local scale. This section describes some of these consequences and some caveats to keep in mind when applying a specific pattern.
- Examples: one or several concrete examples are also provided to illustrate how to use the pattern.

Many patterns are related in that they target similar problems, have similar solutions, or depend on each other. For this reason, we sometimes also include a "See also" section to list these related patterns.

Identifying patterns

In identifying the patterns found in this article, we drew from our own work as well as from the literature. This naturally means that our selection is based on our own knowledge of the field and is therefore somewhat arbitrary and subjective. We therefore make no claims as to the completeness of our pattern language. We intentionally employed this approach because patterns often capture "dark" knowledge that is not clearly reported in existing literature, or it is reported but not emphasized. Furthermore, some of the patterns presented here are well-known in other fields and are included here because they could be of benefit to the visualization domain. In fact, some patterns are even well-known in visualization evaluation folklore (e.g., Pilot Study), but we have included them here to formalize their existence.

Another step in identifying and validating the patterns in this article was to present an earlier version of this article at the BELIV 2012 workshop on October 14–15 in Seattle, WA. Discussions during the workshop gave us perspective and feedback on the existing patterns (originally 12, now 20) and also gave us insight on new patterns to add. In fact, a common theme discussed at the BELIV workshop was what we characterize as *anti-patterns*:[41] examples of solutions (often straightforward ones) that may initially seem like a good idea to a common problem, but which

Table 1. List of evaluation patterns presented in this article.

Category	Pattern name	Type	Example
Exploration	Factor Mining	Quantitative	Ware et al.[42]
	Trial Mining	Quantitative	Ghani et al.[21]
	Human Blackbox	Quantitative	Dwyer et al.[7]
	Do-It-Yourself	Qualitative	(Common)
	Wizard of Oz	Both	Walny et al.[43]
Control	Luck Control	Quantitative	Pietriga et al.[44]
	Time/Accuracy Elimination	Quantitative	(Common)
	Deadwood Detector	Quantitative	Kim et al.[45]
	Pair Analytics	Qualitative	Arias-Hernandez et al.[46]
Generalization	Complementary Studies	Both	Elmqvist et al.[47]
	Complementary Participants	Both	Andrews et al.[6]
	Expert Review	Qualitative	Tory and Möller[27]
	Paper Baseline	Quantitative	Kang et al.[8]
Validation	Pilot Study	Both	(Common)
	Coding Calibration	Qualitative	Kwon et al.[34]
	Prototype	Qualitative	Henry and Fekete[48]
	Statistics Verification	Quantitative	(Common)
Presentation	Once Upon A Time	Qualitative	Elmqvist et al.[49]
	Case Study	Qualitative	Shneiderman and Plaisant[50]
	Visualizing Evaluation	Both	Kwon et al.[34]

ultimately do not work out. Munzner presents examples of this in her work on pitfalls in writing InfoVis papers.[9] Anti-patterns are not the focus of this work, but their prevalence at the BELIV workshop also lends credibility to our work, and the anti-pattern concept should be studied in the future.

Using the patterns

Design patterns are intrinsically bottom–up,[12] which means that they are not intended to guide the top–down design of an entire evaluation study but rather to serve as components used to solve particular aspects of a study. For this reason, experimenters may find themselves using not just one but potentially several patterns to overcome particular problems or challenges in their study. We recommend that experimenters first acquaint themselves with our evaluation patterns (as well as any additional patterns contributed by others). Then, much like how design patterns are used in software engineering,[12] this basic knowledge should be enough for experimenters to be able to look up relevant patterns based on the practical problems that arise in designing a particular evaluation study.

How to design an entire evaluation study from scratch is outside the scope of this article. A good starting point for top–down evaluation study design is the work by Lam et al.,[3] which takes the high-level goals and research questions as a starting point in describing seven scenarios for visualization evaluation. Standard HCI textbooks such as Rogers et al.[14] may also be useful in this endeavor.

List of patterns

In this article, we propose 20 separate patterns, both new and old, for visualization evaluation (see Table 1). To bring structure to the visualization evaluation pattern language as well as ease navigation in the repository, we have created five broad categories of patterns based on the high-level purpose that the researcher is trying to achieve and the question he or she is trying to answer. However, each category does not necessarily correspond to a type of study (e.g., the Exploration category is not required for an exploratory study). Instead, while designing an evaluation study, the experimenter should take a problem-driven approach to selecting suitable patterns to use. For example, "Exploration" patterns can be used to determine which factors are most important in an evaluation (the "Exploration" category), whereas "Control" patterns can help resolve confounding factors and "Presentation" patterns suggest how to present the collected data to readers.

- Exploration: patterns concerned with exploring the design space of the evaluation study. Are we using the right independent and dependent variables? Are we confident that the study is appropriate? Are we asking the right questions?
- Control: mechanisms for controlling an evaluation study design to achieve high internal validity. Is the study itself sound and appropriate? Are the results going to be conclusive? Are the tasks and data representative?

- Generalization: positioning an evaluation study to achieve high external and ecological validity. Is this study grounded in real-world practices? How can these data be applied outside the laboratory? How trustworthy are these results?
- Validation: finding the right balance, calibration, and parameters for an evaluation to save time, resources, and money. Are we testing the right thing in the right way? Are we wasting time and effort? Do we have all the data and information to analyze the results?
- Presentation: reporting the results of an evaluation correctly and economically. Are the results presented in a way where they can be easily understood? How do we evaluate and analyze higher level tasks and scenarios? How can we communicate our results to our readers?

In the following sections of this article, we review these 20 evaluation patterns in full detail.

Exploration patterns

Exploration patterns are intended for early evaluation design when the experimenter is trying to find appropriate tasks, datasets, factors, and baselines for an evaluation. The goal of this stage is generally to achieve confidence that the evaluation is appropriate for the visualization being evaluated. The below patterns all help in this early design process in various ways.

Factor Mining

Problem. Deciding upon an experimental design is key to any successful controlled experiment, but this is sometimes challenging for complex problem domains. The factors that govern how difficult a trial will be to complete successfully for a participant may be unknown and difficult to control.

Solution. Split the experiment into two phases, where the first phase is an exploratory study used for mining suitable factors, and the second is a straightforward experiment that uses the findings from the first. The exploratory study should use representative trials (possibly generated using Trial Mining, see section "Trial Mining"). For each trial, calculate each of the metrics that are candidates to be used as factors for the follow-up experiment. When statistically analyzing the results, include all the candidate metrics in the model and note which ones have a significant main effect on the main performance metrics. The significant metrics are the ones that should be considered as factors, and the range of values in the tested trials give an indication of which levels to choose for each factor. Interaction effects are particularly interesting to include since they indicate situations where results are split depending on a particular condition.

Consequences. Factor Mining will inevitably add complexity, time, and budget expenditure to a project since it requires an additional phase. Furthermore, in order for the identified factors to be representative, the trials have to be representative as well. This is often problematic: if we knew how to construct a specific trial at a specific level of difficulty, we would likely already know the relevant factors and would not need Factor Mining in the first place. To sidestep this issue, Factor Mining is often used in conjunction with Trial Mining to randomly generate a large number of trials, characterize them, and select representative ones.

Examples. In a study from 2002, Ware et al.[42] investigated the factors influencing the aesthetics of graph aesthetics. Various factors of a graph, such as continuity, number of crossing, number of branches, shortest path length, were measured. Then, the relationship between these measures and the performance outcome (answering the length of the shortest path between two highlight nodes) was analyzed through regression analysis. More recently, Factor Mining was used in a study on the perception of animated node-link diagrams of dynamic graphs.[21] In order to understand which attributes of a dynamic graphs influence the human perception capability, the work enumerates a large number of dynamic graph metrics, such as node and edge speed, angular momentum, and topology change, but there exists no results on the relative significance of these candidate metrics. Therefore, the work used an exploratory study where the important metrics (node speed and target separation) were identified. See also, section "Trial Mining."

Trial Mining

Problem. Generating representative trials is important for ensuring validity, but it is not always possible to generate a trial given specific experimental factor levels such as size, complexity, or density. In other words, the metrics used to characterize a trial may be descriptive rather than generative, and determining how to use them to generate specific trials is too complex or time-consuming.

Solution. Instead of generating a specific trial from parameters, generate a large number (tens of thousands) of entirely random trials and calculate the factor metrics for each random trial. The descriptive statistics for all these generated trials and their calculated metrics will give an idea of important metrics, their data distribution, and their relevant levels (also see section “Factor Mining”).

Once the factor levels have been determined (as intervals for each metric), search the database of random trials and pick trials that meet the criteria. To avoid inadvertently picking outlier trials, consider selecting trials that fall within a specific confidence interval around the mean for each metric.

Consequences. Using Trial Mining means giving up the ability to generate a representative trial for a specific experimental condition and instead select from a database of randomly generated trials. A lot of random trials may have to be generated, the absolute majority of which will be discarded and never used. All unused trials represent wasted time and effort. This pattern also hinges on being able to generate an unlimited number of random trials, which is not possible for all domains (such as text, images, and audio).

Examples. The Trial Mining pattern was used in a recent study on perception of animated node-link diagrams of dynamic graphs,[21] that is, graphs that change over time. It was unclear what constituted a representative trial for dynamic graphs, so in an initial study, a large number of trials (240,000) were generated. The different graph metrics (node speed, degree, distance, etc.) were calculated for each trial, and when selecting the actual trials to use, trials were picked from within a particular confidence interval (0.7 in this particular example) for each metric.

Human Blackbox

Problem. Objectively measuring the quality of a solution created by a participant can be difficult if the solution is not easily quantifiable and can only be subjectively judged.

Solution. Instead of trying to give a subjective judgment on a solution, which is open to bias (see section “Coding Calibration”), create a follow-up evaluation where new participants use the solutions from the first evaluation to solve a particular task in a way that can be quantified. The participants in the follow-up study essentially become *blackboxes*—objects that can be viewed only in terms of their inputs and outputs without regards to its internal workings—that we do not have to open, just study their outputs given specific inputs. In other words, this pattern takes a highly pragmatic approach to judging an artifact: instead of trying to qualitatively assess the artifact, we simply measure participant performance in using the artifact to solve an information task.

Consequences. This pattern requires adding a second evaluation, which is both costly and time-consuming. It also requires designing a new task for the second evaluation that uses the output from the first and yields a result that can be easily quantified (i.e. completion time, accuracy, and number of interactions).

Examples. To our knowledge, this pattern was first used by Dwyer et al.[7] in work that builds on an earlier study of user-generated graph layouts.[37] However, Dwyer et al. added a second experiment where participants performed several graph tasks using the user-generated layouts from the first experiment. In other words, the performance of participants solving these tasks in the second experiment became robust metrics of the quality of each user-generated layout from the first. In van Ham and Rogowitz’s[37] original work, this second experiment was not present, forcing the authors to make more or less subjective judgments of the quality of the user-generated layouts. Another example of the Human Blackbox pattern was a set of three graph revisitation experiments conducted by Ghani and Elmqvist[51] where the visual encodings selected as quantitatively optimal by participants in the first two experiments fed into the encodings used in the third and final experiment. See also, section “Coding Calibration.”

Do-It-Yourself

Problem. Visualization and interaction design comprises countless decisions on a wide array of aspects ranging from color scheme, user interface, transitions, selection techniques, and visual encodings. Attempting to empirically validate all these design decisions using human subjects is not practical. In fact, sometimes the visualization system being evaluated is too complex—for example, the expertise and time requirements may be too high—or the intended user group is impossible to access for evaluation.

Solution. In the time-honored tradition of scientists experimenting on themselves, Do-It-Yourself (DIY) engages a single individual—the designer herself or himself—to serve as a human participant in a single, continuously running evaluation on the design of a visualization system or technique. This allows the designer to make rapid progress based on their own expertise and experience, only deferring key and important questions to large-scale empirical evaluation. Less vital decisions can be made at the discretion of the designer. However, since it can be difficult to determine whether a decision is vital or not, even for an experienced designer, it is very important that the designer takes disciplined and structured notes while using the DIY pattern.

Consequences. Applying DIY means that many decision designs can be made quickly and without the cost of a human subject evaluation. Furthermore, utilizing project members may virtually be the only way to find participants for an evaluation that spans a very long time or requires very specific expertise or background. However, successfully applying the pattern typically requires long experience in making the necessary design decisions; a novice researcher may not be in a position to reliably make these decisions. Furthermore, even with an experienced experimenter, a major weakness of DIY is the threat of lack of objectivity and integrity: the participant may become too wrapped up in the project or the system to be able to reliably find flaws or make the right decisions. In other words, DIY is generally *not* a replacement for formal user experiments (although the examples below do use them as such), but should only be used to focus actual experiments on important questions.

Examples. DIY is routinely used by both visualization practitioners and researchers alike when designing new techniques and systems. However, the pattern can also arguably be used in place of a large-scale empirical evaluation if the experimental data are sufficiently large or detailed. For example, MyLifeBits[52] is an ongoing DIY evaluation where one of the co-authors—Gordon Bell—continually logs data about his personal life using an automated database system. Wigdor et al.[53] present a longitudinal study of a single participant using a digital tabletop as a replacement for a desktop computer over the course of 13 months. Similarly, a recent article[54] presents the author's own experiences with using an interactive desk over the duration of a full year. All three of these examples are successful because they take a disciplined and structured approach to DIY evaluation by using careful subjective observations and surveys paired with quantitative measurements. See also, sections "Paper Baseline," "Prototype," and "Wizard of Oz."

Wizard of Oz

Problem. Many interesting research questions require significant new technological advances in order to be answered, but it is sometimes difficult to predict whether this development effort will be worthwhile. In fact, a positive outcome of the evaluation may be necessary to motivate even pursuing a speculative technical advance in the first place. This leads to a chicken-and-egg problem: to evaluate the idea we need an implementation, but to build an implementation, we must first evaluate the idea to motivate the development effort.

Solution. Conduct an evaluation where the participants interact with a computer system that is partially or fully operated by an experimenter (also known as the Wizard). The experimenter (Wizard) manually performs the computationally challenging tasks and feeds the desired output back to the participant. Participants are generally not informed of the existence of the experimenter posing as the computer system and believe that the system is fully autonomous. Rather than being a complete computer system, however, a Wizard of Oz visualization platform is merely a more or less hollow interface that forwards requests to the experimenter (often located in another room) and returns the experimenter's actions as output to the participant.

The Wizard of Oz pattern is well-known in the general HCI community[55] (see below for more examples), but has so far seen little use in the visualization community. For this reason, it is worthwhile to highlight in this treatment.

Consequences. The Wizard of Oz pattern allows for evaluating new or even speculative techniques and mechanisms without the time and cost of implementing them (such as speech recognition, gesture detection, and high-level reasoning). Participants believe that the system is autonomous, thus encouraging natural behavior when interacting with it. However, as with any method involving deception, there is a risk that the participant realizes that there is another human being involved, which may affect their performance.

Examples. A recent article by Walny et al.[43] uses the Wizard of Oz pattern to support robust pen and touch recognition for data exploration on interactive whiteboards. The wizard received participant input on a separate, networked computer and used a dedicated control panel to issue system commands in response to this input. In general HCI, Wizard of Oz protocols have been used for similar computationally difficult problems, such as natural language processing[55] and pen input recognition.[56] See also, sections "Pair Analytics" and "Prototype."

Control patterns

Internal validity is widely defined as the degree to which the outcome is a function of the controlled parameters of an evaluation. In other words, an experiment with high internal validity is designed in such a way that irrelevant parameters have little or no impact on the results. The purpose of Control patterns is to achieve high internal validity by controlling or eliminating such irrelevant parameters.

Luck Control

Problem. Chance is sometimes a major factor for certain tasks, such as visual search. For example, if the participant is searching for a target in a collection of distractors by clicking on each potential target to find out if it is the correct one, they may get lucky and immediately pick the correct target (or unlucky and always pick the correct target last). This renders the distractors largely ineffective.

Solution. The common solution is to leave issues such as this to random chance in the knowledge that they will even out over the course of the evaluation, but sometimes luck may have too large an impact to be ignored. In such situations, the solution is to limit the impact of luck by explicitly controlling discovery order without the knowledge of the participants. For example, if there are five potential targets (doors to open) to pick and only one of them is the correct answer (one door opens to the object the participant is looking for), add an experimental factor D with values 1–5 that says which of the five targets is the correct one. Whether a potential target is the right one or not is determined when the participant actually clicks on it (given earlier clicks). Each participant will thus be lucky ($D = 1$, i.e. on the first click) and unlucky ($D = 5$, i.e. on the last click) once per condition, and all other levels of chance in-between.

Consequences. Using Luck Control requires an additional factor to be added to the experimental design, which can sometimes be problematic for experiments that already have a large number of conditions. In addition, sometimes the number of possible outcomes is too large to model directly using a factor; in such situations, define intervals of outcomes as "easy," "medium," and "difficult" in terms of the impact of random chance (corresponding to, for example, discovery order 1–5, 6–10, and 11–15). Finally, Luck Control can only be used in situations where the determination of what potential target is the correct one can be performed on the spot, and not when setting up the trial.

A danger with this approach is that participants may suspect that luck is being controlled. However, if done correctly, this knowledge should not impact performance; the decision of which potential target is the correct one is lazily resolved.

Examples. A form of Luck Control is used in many experiments that include factors to model the difficulty of a trial. However, to our knowledge, explicitly controlling discovery was first proposed by Pietriga et al.[44] in their operationalization of multiscale search. Javed et al.[25] use a similar approach when evaluating a multi-focus technique called PolyZoom for exploring two-dimensional (2D) multiscale spaces. Giving participants a choice of four possible target areas on a map, Javed balances which of the four areas actually contain the target (without the participants' knowledge) instead of relying on random chance. See also, section "Time/Accuracy Elimination."

Time/Accuracy Elimination

Problem. Some evaluations come down to time and error performance. However, it is often difficult to balance both measurements given the individual differences of participants (the so-called time/accuracy tradeoff). One participant may be very thorough and score few errors at the cost of high completion times, whereas another participant may quickly solve tasks while incurring many errors. This remains an issue even with very specific instructions.

Solution. Design experimental tasks so that one of time or error measurements are eliminated. An example of eliminating error would be for a visual search task, where the task can be designed so that the participant is not allowed to answer with an incorrect target; the trial only ends when the correct target is selected. Analogously, to eliminate timing, either give a specific time limit (say, 10 s) or give no time limit at all to find the target, but allow only one click.

Consequences. Eliminating time or error enables analyzing only one metric and will give more definite answers on the impact of the conditions. On the other hand, it may nuance tradeoffs between time and error to be overlooked. Furthermore, one of the dangers with eliminating one of these factors is that in forcing the participant to be correct, he or she can sometimes get stuck and not be able to complete the trial.

Examples. In a recent study on occlusion management for tabletop interfaces, Javed et al.[57] used a task where participants were asked to recall a sequence of images while their completion time was recorded. However, out-of-order selections were not possible; if the participant selected the wrong image, a brief error message flashed on the screen and an error counter was increased. The participant was still required to make the correct choice to progress to the next image in the sequence. In other words, errors were eliminated and only timing was analyzed.

Similarly, standard text entry evaluation stipulates three error correction conditions: *none*, *recommended*, and *forced*.[58] Of the three, the "forced" condition is an example of error elimination in that it does not allow typing mistakes; only the correct key for the next letter to type will result in output. See also, section "Luck Control."

Deadwood Detector

Problem. Crowdsourcing participants for studies is a great way to collect lots of data quickly and economically.[18,59] However, many participants are "deadwood" in that they are simply looking for the monetary compensation and are not paying sufficient attention to the evaluation tasks.

Solution. Various approaches have been proposed to motivate crowdsourced workers (often called Turkers) and filter out those who did not pay proper attention.[60–64] However, many of these approaches require additional steps (e.g., adding dummy tasks) or damage the validity of the study (e.g., removing outliers based on task performance). An effective and universally applicable approach is to measure the randomness of a crowdsourced worker's performance while completing tasks. This approach is based on the assumption that deadwood Turkers randomly select responses in order to quickly get through the whole evaluation, yielding more or less random responses, that is, which follow the uniform distribution. Thus, filtering out participant whose performance is not consistent over time (i.e. $p > p_{threshold}$) effectively filters out deadwood from the collected data.

Consequences. By identifying deadwood Turkers and eliminating their data from the evaluation, crowdsourcing-based approaches become a viable option to collect data from a large number of study participants.

Examples. This approach was used in a recent crowdsourced study to eliminate deadwood from collected data.[45] A more detailed procedure of this approach was presented by Kim et al.[65] at the BELIV 2012 workshop.

Pair Analytics

Problem. Understanding the cognitive process of a participant using a visualization tool is difficult. Conducting an interview after an evaluation session may reveal some major points, but the results will be mostly summative and will not capture details encountered on the fly. A think-aloud protocol may help capture such information, but this approach may affect the behavior of the participant, and the collected data are often somewhat random and difficult to understand. Participants often cannot articulate what they think, or provide too much information that is not necessarily helpful. Finally, the tool itself often serves as a barrier against effective sensemaking since most visualization evaluations do not provide proper, long-term training for the tool, making the participant less than fluent in using it.

Solution. The basic idea of Pair Analytics[46] is to form a team consisting of an experimenter (often a visualization expert) and a participant (often a subject matter expert) to explore the dataset and perform the required tasks. The pair complements each other since the experimenter is well-versed with the tool and will "drive" it, and the participant is well-versed with the problem domain for the dataset to analyze. Furthermore, in solving the task together, the driver and the domain expert will be verbally externalizing their cognitive processes when they communicate to investigate the data. This verbal communication provides natural insight into the sensemaking process, compared to think-aloud protocols where the verbal communication is easily perceived as artificial by the participant.

Consequences. The Pair Analytics pattern requires that the experimenter who is driving the tool remains objective. Nevertheless, a possible consequence of this pattern is that the mere presence of the experimenter will influence the domain expert in an unforeseen way.

Examples. The VAST 2007 competition[66] included a special session where the winners were invited to use their visual analytics tools on a smaller dataset and working together with a professional analyst. Similarly, Grammel et al.[67] used a human mediator to pilot visualization construction software in order to determine how novice users create new visualizations. The mediator insulated the novice participants from the complexities of the software, yet allowed the researchers to study the thought process and reasoning going into this task. Additional examples of pair analytics can be found at http://tinyurl.com/pair-analytics.

See also, sections "Complementary Participants," "Expert Review," and "Wizard of Oz."

Generalization patterns

External, or ecological, validity is loosely defined as an estimate of the degree to which the results of an evaluation can be applied to realistic situations. In contrast to control patterns, the purpose of generalization patterns is to achieve high ecological validity by introducing different sets of environments, participants, and real-world examples.

Complementary Studies

Problem. In designing a visualization user study, we are often faced with a choice of a rigorous and unrealistic study, or a realistic but ad hoc one. Achieving both in the same study is often impossible: for a rigorous study, we need to be able to generate balanced trials, which means that the data cannot be truly real. For realistic data, on the other hand, we run into learning effects, variability in the trials, and difficulty controlling all aspects of the task and dataset. In other words, the rigorous toy study lacks *ecological validity* (conformance to a realistic situation), whereas the ad hoc study lacks *internal validity* (confidence of the measured results actually being a function of the factors).

Solution. The obvious solution for remedying the above problem is to include *both* kinds of studies in a paper and have them complement each other. The rigorous toy study will probably be the backbone of proving that the system or technique actually works in the general (but unrealistic) case. The realistic ad hoc study, on the other hand, will serve as a much-needed sanity check and help to convince the reader that the work is applicable to the real world.

Consequences. Using the Complementary Studies patterns essentially requires twice the resources in time, money, and preparation of conducting just one of the two possible studies. Beyond that, describing the details of two studies in the same research article may be costly in terms of space. Furthermore, conducting two or more studies evaluating the same phenomenon may result in contradictory results; the researcher must be ready to handle this case.

Examples. In evaluating visual search performance for the Color Lens technique,[47] which dynamically adapts a color scale to fit the range of data values within a magic lens, there was an option between searching for a variable-strength feature (a circle) in a random noise background and a named feature in a real photograph (i.e. "find the deer in this picture of a forest"). Instead of choosing just one option, both studies were conducted and reported on. Andrews et al.,[6] in a study of space layout practices for sensemaking, perform two separate and complementary evaluations, one engaging professional analysts and another engaging graduate students (this is also an example of Complementary Participants, see section "Complementary Participants"). See also, section "Complementary Participants."

Complementary Participants

Problem. Many visualization systems are designed for a particular expert user population, but getting access to this population for evaluation purposes is often very difficult. For example, a visual analytics system such as Jigsaw[32] is intended for expert analysts, but finding a good number of actual analysts that are willing to invest the time to help evaluate the system is difficult.

Solution. Run two versions of the evaluation: a smaller version with a small number of expert analysts and a larger version with non-expert participants selected from the general population. The tasks and datasets for the two versions can be radically different. Similar to Complementary Studies, the few expert participants allow for retaining ecological validity and may be able to offer deep insights on the visualization, whereas the larger pool of general participants provide internal validity and information on human motor, perceptual, and cognitive abilities not specific to experts.

Consequences. The Complementary Participants does not entirely remove the need to engage expert participants for an evaluation, but it does ease the burden by radically reducing the number of such participants needed. This pattern may also require more money and time than with one participant group. Finally, it is possible, maybe even likely, that the two participant group give rise to different and even contradictory results.

Examples. The visual analytics tool Jigsaw[32] has primarily been evaluated using general non-expert participants (often university students), such as in the qualitative evaluation performed by Kang et al.[8]

However, Jigsaw has also been utilized by professional analysts (although not reported in the same article). Andrews et al.[6] used Complementary Participants and Complementary Studies in evaluating their Analyst's Workstation tool, engaging five professional analysts in one study, and eight students in another. See also, section "Complementary Studies."

Expert Review

Problem. Using study participants recruited from the general population is impractical if the visualization system being evaluated requires very specialized knowledge and skills. The experimenter may be looking for deep and informed insights that no layperson can provide. At the same time, expert users are often protected and have little or no availability (or even interest) to participate in a large-scale user evaluation.

Solution. In HCI, an expert review is a structured evaluation of an interactive system using a small set of usability experts that explore the system with an eye toward usability problems. Anecdotal evidence shows that only five usability experts can find up to 75% of all usability problems in a system.[14] Tory and Möller[27] propose the use of expert reviews as a method to evaluate visualization, not just on usability issues but also on additional aspects. In fact, coupled with a pattern such as Pair Analytics, an Expert Review (with the domain expert serving as the participant), this pattern allows for evaluating a new visualization system even in an early formative stage where usability issues have not yet been resolved.

Since Expert Reviews are structured evaluations, it is often useful to provide the expert participant with some form of written task sheet to follow. This sheet should contain both simple, straightforward questions, to get the participant up to speed with using the tool, as well as more open-ended questions, to promote deep insight.

Consequences. Using an Expert Review to evaluate a visualization system significantly reduces the time and cost investment of evaluation while still exposing the system to human subjects for validation. Many times, only a small number of expert participants are needed for the review. Furthermore, the insights collected during Expert Review will be of higher quality than those given by laypersons. However, as observed by Tory and Möller,[27] Expert Reviews should not replace user studies because different evaluation mechanisms test different things (see the Complementary Participants pattern). For example, an Expert Review will not allow for comparing two different techniques or interface designs. In fact, expert participants are still going to be different and have different opinions, so an Expert Review is no guarantee to achieve consistent results across all participants.

Examples. Yi et al.[68] employ the Expert Review pattern to evaluate the TimeMatrix visualization tool for dynamic graphs using three social scientists trained in social network analysis (SNA). Similarly, Elmqvist et al.[49] use two visualization researchers to validate the DataMeadow system for multivariate visual analytics.

See also, sections "Complementary Participants" and "Pair Analytics."

Paper Baseline

Problem. Determining how people make sense of data "in the wild" can often be obscured by the visualization tools themselves. The participants may not be fully fluent in using the tool, and the tools—being research prototypes—may not have an optimal interaction design. In fact, first developing a prototype tool with the intention of supporting people's "natural" mechanism of interacting with data without knowing this mechanism is actually somewhat counterproductive.

Solution. Instead of designing a visualization system to use as an evaluation platform, conduct an entirely paper-based evaluation. Rather than using interactive computer displays, use paper printouts of the displays to be studied and give them to each participant. Participants can still be asked to explore data and solve tasks, but they will be drawing on paper printouts instead of a computer system. If appropriate, combine the paper-based study with a visualization-based study to compare the two.

Consequences. This pattern reduces the need for costly and time-consuming software development. However, it is most suitable for formative design and will obviously not yield results for interactive behavior. Furthermore, while a computer-based study is easy to instrument, this is not the case for a paper-based one; the experimenter may have to resort to videotaping participants, or keep careful observation logs.

Examples. Kang et al.[8] include a Paper Baseline condition in their qualitative study of Jigsaw-[32] where the participant only receives paper printouts of the reports that other participants use the Jigsaw tool to analyze. Isenberg et al.[35] and Robinson[36] both base their studies of collaboration data analysis solely on paper printouts. See also, sections "Prototype" and "Wizard of Oz."

Validation patterns

Validation patterns are intended for early confirmation that the design of an evaluation study or an analysis scheme is appropriate, thereby identifying problems before wasting time and resources. The purpose of Validation patterns is to increase the efficiency of such evaluation processes.

Pilot Study

Problem. Evaluations often contain many parameters specific to the visualization technique, such as the difficulty of the trials, task formulation and design, training sessions, blocking and order, data measurement, and overall study balance. Such parameters often have a large impact on the outcome of the evaluation. Therefore, it is not scientifically valid to arbitrarily set appropriate values for these parameters, and using the study itself to calibrate the values is costly in time and resources.

Solution. Perform several dry runs of the evaluation study with unbiased participants. Each dry run (or pilot) should mimic a real evaluation session as closely as possible, but changes to the study may be made after each pilot to improve its design. Pilot participants should be objective and unbiased to yield the most benefit (in other words, involving a project member as a participant is a bad idea), but having expertise in human subject evaluation is helpful since it allows the participant themselves to give informed advice on how to improve the study. Determining how many pilots to run is open to debate; one or two is often too few, whereas three or more allows for achieving stability in the changes made to the study. It is also advisable to conduct the planned statistical or qualitative analysis using the collected pilot data, which often helps the experimenter to identify any errors and mistakes in the data collection. However, Pilot Study data should never be included in the final analysis.

Consequences. Running one or several Pilot Studies is a very common practice in human subjects evaluation, but deserves being highlighted here as a pattern in recognition of its prominence. Even though Pilot Studies add to the time investment of performing the evaluation, they are truly invaluable in detecting problems early, and therefore often repay themselves many times over the course of a study. Furthermore, Pilot Studies can be used to inform and motivate design decisions for evaluation studies (they improve the validity of the study) and should therefore be reported in the article. A pilot can sometimes be run in combination with the Expert Review (section "Expert Review"), Prototype (section "Prototype"), and Wizard of Oz (section "Wizard of Oz") patterns.

Examples. Pilot Studies are typically not highlighted in research articles, but are nevertheless used to calibrate most reported evaluations. One concrete example is the Pilot Study used in a graphical perception experiment by Javed et al.[25] where findings from the pilots were used to find suitable levels for the factors included in the experiment. See also, sections "Expert Review," "Prototype," and "Wizard of Oz."

Coding Calibration

Problem. When analyzing qualitative data (e.g., interview results and insight reports), the data are often coded to impose structure on large, unstructured data by multiple coders (or raters). Unless the coding scheme is determined by prior literature (rare in visualization) or open coding[69] is used, multiple coders often need to construct a coding scheme while analyzing the data (closed coding). This process is iterative and often causes painful re-coding of the entire dataset due to changes in coding scheme.

Solution. Having multiple meetings among coders to calibrate a coding scheme while coding randomly selected subsets (about 10%) of data is crucial. During calibration, codebooks should be compared and discrepancies between results discussed. The discussion often leads to refining codebooks, and clarified definitions should be written on a shared document. Calibration meetings should be continued until no major disagreement is found. Even after the coding scheme is stabilized, if any coder identifies unclear cases, new meetings should be called. Inter-coder reliability[70] can be calculated after the coding scheme is stabilized to clarify definitions and prevent minor errors (although high inter-coder reliability cannot guarantee similar analyses by all coders[71]).

Consequences. While it may require an additional investment of effort, Coding Calibration ultimately saves resources by establishing a code scheme as early as possible.

Examples. Recent work by Kwon et al.[34] uses a similar approach and reports on the calibration process to some degree while coding insight reports, including the number of coders, inter-coder reliability, coding calibration processes, and scoring schemes. In their study of paper-based practices for visual analysis, Isenberg et al.[35] report on their methodology for coding notes and video captures of evaluation sessions. Their description of having two separate coding passes involving two coders would seem to indicate that Coding Calibration was used between the two passes. This coding approach is further described in another article on "grounded evaluation."[17]

Prototype

Problem. Software development is costly, both in time and resources, and this is particularly true for interactive systems such as visualization tools where the developer has to make countless design decisions on interface, visual encodings, layout, and so on. Fully implementing all design alternatives and comparing their performance is therefore not practical.

Solution. To solve this issue, the field of HCI has long promoted the use of prototypes[14] of varying fidelity. These prototypes are often built using cheap and readily available resources, such as paper, Post-its®, colored pens, scissors, and glue. They can then be used in user-centered or participatory design sessions with domain experts to determine which of several alternatives is optimal, and what changes should be made to them. Prototypes can be made increasingly more complex by using digital tools or interface mockups.

Prototypes serve an additional purpose: because domain experts are generally not well-versed in software development and visualization design, the prototypes give a tangible example of what is possible using the new technology (this use actually encroaches on the Wizard of Oz pattern described in section "Wizard of Oz").

Consequences. Applying the Prototype pattern requires additional effort to create prototypes, potentially several different ones, during the design phase of a project. This is often costlier than simply starting software development of the visualization tool itself. On the other hand, having prototypes allows for collecting early formative feedback from intended users. This may improve the quality of the visualization tool that is ultimately developed based on this feedback.

Examples. While the Prototype pattern currently does not yet appear to be widely used in the visualization community, we are convinced that several researchers use it without necessarily reporting this in their articles. A concrete example is Henry and Fekete,[48] who report on participatory design sessions for their MatrixExplorer tool where paper prototypes were used. Similarly, Walny et al.[43] use a high-fidelity prototype of an interactive whiteboard for a Wizard of Oz study where an administrator translates participant handwriting input to system commands. See also, sections "Paper Baseline," "Do-It-Yourself," and "Wizard of Oz."

Statistics Verification

Problem. Statistical data analysis of study results is sometimes postponed until the data collection is finished due to various reasons (e.g., researchers are not comfortable with statistical data analysis). Since the researcher did not have a firm idea of the required statistical analysis in advance, the collected data tend to have many issues: (1) unnecessary data were collected, (2) confounding factors are not properly controlled, and (3) a required statistical test is too complicated or does not provide sufficient power.

Solution. Design the statistical tests before the data are actually collected. Statistical tests should be ready before evaluation begins. Even writing the scripts for a statistical package (e.g., R, SAS, and SPSS) and running the test with fabricated data will help the experimenter focus on how to design the evaluation study. It will also minimize any problems with analyzing the data at a later stage, such as not having the right data, not performing a representative task, or not being able to use a specific statistical analysis (the data may not be normally distributed, for example). If a statistical consultant is available, getting his or her assistance while designing the study would be instrumental as well.

However, this does not mean that an experimenter can simply delegate the whole statistical work to a statistician. The experimenter should be in charge of the final decision of experimental designs since the only experimenter clearly knows what he or she wants to do. Thus, learning basic statistics is mandatory. Several online materials specifically designed for HCI researchers are also available (e.g., http://yatani.jp/HCIstats in R and http://depts.washington.edu/aimgroup/proj/ps4hci/in JMP and IBM SPSS[72]).

Consequences. Adopting this pattern will minimize various adversary situations as discussed above. Since data collection is often costly, collecting the wrong data or missing out important design elements can cause serious delay of a research project and/or be costly in terms of resources (e.g., participants' compensation). Furthermore, considering the statistical analysis up front will let the researcher store the data in a format amenable to this analysis, avoiding tedious and time-consuming reformatting.

Examples. It is difficult to know whether experimental designs in the literature were driven by statistical tests or not. The quality of the statistical analysis is often an indicator of whether or not the authors let their experimental design be guided by the analytical methods. For example, some articles include a subsection called "Data Analysis" under the "Method" section. This might allude that data analysis approach was planned while designing the experimental design. Two recent studies on dynamic graph visualization[51] and animated transitions[73] use this pattern and include careful descriptions of their data collection methods. See also, section "Pilot Study."

Presentation patterns

Evaluations are meaningless if their results are not presented to an external audience. Presentation patterns guide an experimenter in how to communicate the evaluation results clearly and efficiently.

Once Upon A Time

Problem. Proposing a novel visual representation, interaction technique, or visualization system is often a substantial contribution in itself, and also performing an in-depth human participant's evaluation can sometimes be too much for a single research article. Meanwhile, simply listing the features of the technique or explaining the underlying algorithms may be overwhelming to readers.

Solution. Provide a fictional usage scenario to demonstrate the utility of the new technique or system. This scenario is basically a story. There is a character with a problem, motivation, or question. The character gradually solves a series of problems using different features of the novel technique. The scenario is fictional but should be believable, so readers can feel empathy with the character. The scenario should be accompanied by clear screenshots that highlight how the technique helps the character solve the problem, potentially step by step. Screenshots may even be annotated so that the reader can easily follow the narrative. When an interaction technique is being described, the authors can use a series of screenshots (small multiples) or create a companion video to show the action. In fact, basing a companion video on the written scenario in the article makes the presentation even stronger.

Consequences. A usage scenario gives the reader a concrete example of how a user may use the proposed techniques to solve a problem. It validates the work without requiring an actual user evaluation to be performed, which is cost-effective in terms of time and resources. However, usage scenarios do not expose the proposed technique or system to actual human participants, which means that the narrative is going to be limited by the viewpoint of the author. For this reason, this pattern is best used as a complement with actual user evaluation studies.

Examples. One of the earliest notable examples of the Once Upon A Time pattern is in the article describing the GRASPARC system by Brodlie et al.,[74] and Yi et al.[75] present a fictional scenario of choosing a breakfast cereal using Dust & Magnet. Similarly, Elmqvist et al.[76] explain how the ScatterDice system can be used using a story of a person buying a digital camera.

Case Study

Problem. Realistic tasks are often complex and high level to the point that they cannot be isolated and studied in a quantitative laboratory setting,[2] and a bottom–up model of assembling higher level tasks from low-level ones has questionable value.[4] In general, laboratory studies tend to be one-off, simplistic, and lack ecological validity by virtue of taking place in a laboratory rather than a real work environment. Furthermore, conducting a quantitative laboratory study involving intended users is sometimes impractical; they may be too busy, located at a remote site, or deal with sensitive data. In other words, the users are willing to use the visualization tool in their real working environment, but cannot commit to a dedicated user study solely for the purpose of scientific evaluation.

Solution. Conduct a case study using a small set of participants. Researchers collect whatever information available and report individual cases. Since the environment around the cases is not controllable, the resulting insights cannot be generalized. However, its ecological validity is very high given the particular context used in the case study. Furthermore, the resulting stories are not fictional as in the Once Upon A Time pattern, but factual. Because researchers often report a limited number of cases, the author should be careful in reporting the outcomes. The outcomes should not be too generalized. Instead, each individual case should be analyzed deeply with rich details.

A special case of a case study is the Multi-Dimensional In-Depth Case Studies (MILCS) methodology proposed by Shneiderman and Plaisant.[50] MILCS is a disciplined approach to conducting case studies with a small set of participants using a wide range of methods over an extended period of time (months to years).

Consequences. Running a Case Study yields realistic and believable narratives of real users interacting with the visualization tool without requiring massive time and effort on behalf of the researcher. Even though the results of a case study cannot be easily generalizable, they may provide in-depth insights about how the visualization techniques are used in a realistic situation.

Examples. Liu et al.[77] developed SellTrend, a visualization system for airline travel purchase requests, while working with a global travel information service provider. They presented their experiences designing and evaluating the tool with their intended user group as a case study. Shneiderman and Plaisant[50] give several additional examples of case studies in visualization and HCI research.

Visualizing Evaluation

Problem. Data collected from an evaluation study are often complicated. The data may be multidimensional and even temporal. Traditional approaches to report statistical analysis (e.g., *p*-values and bar charts with confidence intervals) may not be sufficient to communicate the complexity of such data.

Solution. Use visualizations to report the evaluation data (colloquially speaking, "eat your own dog food" or "dogfooding," a term that refers to a company routinely using its own products to demonstrate their quality). Since the audience of a visualization paper should be able to make sense of (even novel) visualizations, researchers should actively exploit the benefits of visualization techniques in their own articles. Of course, this does not mean that one can dump all the data into figures. Beyond bar charts (with confidence intervals) and boxplots, which are useful for showing performance data for different conditions in quantitative experiments, the most practically useful visualizations for Visualizing Evaluation are likely event timelines (see examples below). These give the reader an indication of temporal trends, outliers, and patterns in how a group of participants abstractly used a visualization system.

The authors still need to carefully select only the relevant data, choose the most appropriate visualization technique, and provide easy-to-understand instructions as well as comprehensive legends. The visualizations also should resonate with the storyline of the article. Furthermore, the most appropriate way to explaining the data could very well be more traditional methods.

Consequences. One of the most notable benefits of this pattern is not necessarily showing more data in a article, but that one can invite readers into the sense-making process. Besides the findings and implications reported by the authors, readers can delve into the evaluation data and find interesting details. This pattern can also increase the credibility of the work by showing a more holistic and complete view of the collected data. However, it is important to note that graphical representations do not replace the need for inferential statistics for quantitative experiments.

Examples. Kwon et al.[34] reported insights and view usage over time in a visualization tool for 12 participants using a single timeline visualization included as a figure in the article. This figure was inspired by similar figures in Isenberg et al.,[35] Robinson,[36] and Kang et al.[8]

Discussion

Our selection of patterns in this article is not exhaustive and is limited by our own work, our knowledge of the field, and our personal experience in performing visualization evaluation. More work is needed to expand and develop this pattern language. Toward this end, we have created a Wiki to serve as a repository for visualization evaluation patterns. This Wiki can be found on the following URL: http://visevalpatterns.wikia.com/. The intention is for any member of the visualization community to contribute their own evaluation patterns to the repository. Furthermore, all new patterns should be discussed, evaluated, and compared with existing patterns before they are adopted as canonical evaluation patterns. For this reason, the Wiki is editable by anyone willing to contribute. To get the ball rolling, we have added the 20 evaluation patterns presented in this article to the repository.

Why a pattern language, which by its very design is bottom–up and may not give the high-level guidance needed for newcomers to the field? Existing work has already studied a more top–down scenario-based approach,[3] and we find that a bottom–up catalog of patterns is a good complement to such work. Patterns are essentially "experience in a can," ready to be opened and used by anyone, regardless of their personal expertise and experience. For this reason, we think that this work fills an important gap in the literature.

Having said that, our treatment is not complete and is far from a full pattern language. There is certainly space for many more patterns, such as on interaction logs, evaluation platforms, and visual search. An important activity in the future will thus be to expand this language to include such aspects. Furthermore, the concept of anti-patterns may give rise to equally useful examples of what *not* to do in visualization evaluation. Finally, an important point of improvement will be to tie together all patterns into a complete language by discussing their relationships, how to combine them, and how to choose between them given a specific situation.

In spite of its incompleteness, the collection of patterns in this article provides us with some insights. In structuring our work, we found that the collected patterns can be neatly categorized into five different categories of patterns: Exploration, Control, Generalization, Validation, and Presentation patterns. We found that each category represents what we have striven to achieve in our evaluation studies. Some categories basically confirm what a research method course may cover, such as striking a balance between internal validity (Control patterns) and external validity (Generalization patterns). Other categories (Exploration and Validation) actually shed light on some of the "dark" practices that we and other authors implicitly use in our evaluation studies. For example, Exploration patterns are useful to determine what to evaluate, whereas Validation patterns help us confirm that we are evaluating the right things in the right way. Additional patterns added to each of these categories will only serve to illuminate these dark practices even further.

Conclusion and future work

We have presented a pattern language for data visualization evaluation. While many of these patterns are known (or even well-known) in the community, we think that they provide a powerful lens for looking at evaluation and will help to disseminate experience, provide a standard vocabulary, and invite contributions by the community as a whole. Future work on this topic will be to continually expand and evolve the language of patterns based on current practices in the domain. Furthermore, as observed above, identifying anti-patterns for visualization evaluation is another worthy future research goal.

Acknowledgements

We thank the attendees of the BELIV 2012 workshop as well as the anonymous reviewers for their detailed and constructive feedback, which substantially improved the quality of this paper.

Funding

This material is based upon work supported by the U.S. Department of Homeland Security's VACCINE Center under award no. 2009-ST-061-CI0001.

References

1. Carpendale S. Evaluating information visualizations. In: Kerren A, Stasko J, Fekete JD, et al. (eds) *Information visualization: human-centered issues and perspectives* (Vol. 4950, Lecture Notes in Computer Science). Berlin: Springer, 2008, pp. 19–45.
2. Plaisant C. The challenge of information visualization evaluation. In: *Proceedings of the ACM conference on advanced visual interfaces*, Gallipoli, Italy, 25–28 May 2004, pp. 109–116. New York: ACM.
3. Lam H, Bertini E, Isenberg P, et al. Empirical studies in information visualization: seven scenarios. *IEEE T Vis Comput Gr* 2012; 18(9): 1520–1536.
4. North C. Toward measuring visualization insight. *IEEE Comput Graph* 2006; 26(3): 6–9.
5. Gaines BR. Modeling and forecasting the information sciences. *Inform Sciences* 1991; 57–58: 3–22.
6. Andrews C, Endert A and North C. Space to think: large, high-resolution displays for sensemaking. In: *Proceedings of the ACM conference on human factors in computing systems*, Atlanta, GA, 10–15 April 2010, pp. 55–64. New York: ACM.
7. Dwyer T, Lee B, Fisher D, et al. A comparison of user-generated and automatic graph layouts. *IEEE T Vis Comput Gr* 2009; 15(6): 961–968.
8. Kang Y, Görg C and Stasko J. Evaluating visual analytics systems for investigative analysis: deriving design principles from a case study. In: *Proceedings of the IEEE symposium on visual analytics science and technology*, Atlantic City, NJ, 12–13 October 2009, pp. 139–146. New York: IEEE.
9. Munzner T. Process and pitfalls in writing information visualization research papers. In: Kerren A, Stasko JT, Fekete JD, et al. (eds) *Information visualization: human-centered issues and perspectives* (Vol. 4950, Lecture Notes in Computer Science). Berlin: Springer, 2008, pp. 134–153.
10. Munzner T. A nested process model for visualization design and validation. *IEEE T Vis Comput Gr* 2009; 15(6): 921–928.
11. Alexander C, Ishikawa S, Silverstein M, et al. *A pattern language: towns, buildings, construction*. New York: Oxford University Press, 1977.

12. Gamma E, Helm R, Johnson R, et al. *Design patterns: elements of reusable object-oriented software*. Boston, MA: Addison Wesley, 1994.
13. Elmqvist N and Yi JS. Patterns for visualization evaluation. In: *Proceedings of beyond time and errors: novel evaluation methods for visualization*, Seattle, WA, 14–15 October 2012, article no. 12. New York: ACM.
14. Rogers Y, Sharp H and Preece J. *Interaction design: beyond human-computer interaction*. 3rd ed. Chichester: John Wiley & Sons, 2011.
15. Lam H. A framework of interaction costs in information visualization. *IEEE T Vis Comput Gr* 2008; 14(6): 1149–1156.
16. Boytnton PM. People should participate in, not be subjects of, research. *BMJ* 1998; 317(7171): 1521.
17. Isenberg P, Zuk T, Collins C, et al. Grounded evaluation of information visualizations. In: *Proceedings of beyond time and errors: novel evaluation methods for information visualization*, Florence, Italy, 5 April 2008. New York: ACM.
18. Heer J and Bostock M. Crowdsourcing graphical perception: using Mechanical Turk to assess visualization design. In: *Proceedings of the ACM conference on human factors in computing systems*, Atlanta, GA, 10–15 April 2010, pp. 203–212. New York: ACM.
19. Heer J, Kong N and Agrawala M. Sizing the horizon: the effects of chart size and layering on the graphical perception of time series visualization. In: *Proceedings of the ACM conference on human factors in computing systems*, Boston, MA, 4–9 April 2009, pp. 1303–1312. New York: ACM.
20. Lam H, Munzner T and Kincaid R. Overview use in multiple visual information resolution interfaces. *IEEE T Vis Comput Gr* 2007; 13(6): 1278–1285.
21. Ghani S, Elmqvist N and Yi JS. Perception of animated node-link diagrams for dynamic graphs. *Comput Graph Forum* 2012; 31(3): 1205–1214.
22. Heer J and Robertson G. Animated transitions in statistical data graphics. *IEEE T Vis Comput Gr* 2007; 13(6): 1240–1247.
23. Robertson G, Fernandez R, Fisher D, et al. Effectiveness of animation in trend visualization. *IEEE T Vis Comput Gr* 2008; 14(6): 1325–1332.
24. Elmqvist N, Henry N, Riche Y, et al. Mélange: space folding for multi-focus interaction. In: *Proceedings of the ACM conference on human factors in computing systems*, Florence, Italy, 5–10 April 2008, pp. 1333–1342. New York: ACM.
25. Javed W, Ghani S and Elmqvist N. Polyzoom: multiscale and multifocus exploration in 2D visual spaces. In: *Proceedings of ACM conference on human factors in computing systems*, Austin, TX, 5–10 May 2012, pp. 287–296. New York: ACM.
26. Nekrasovski D, Bodnar A, McGrenere J, et al. An evaluation of pan & zoom and rubber sheet navigation with and without an overview. In: *Proceedings of ACM conference on human factors in computing systems*, Montreal, QC, 22–27 April 2006, pp. 11–20. New York: ACM.
27. Tory M and Möller T. Evaluating visualizations: do expert reviews work? *IEEE Comput Graph* 2005; 25: 8–11.
28. Yi JS. Implications of individual differences on evaluating information visualization techniques. In: *Proceedings of beyond time and errors: novel evaluation methods for visualization*, Atlanta, GA, 10–11 April, 2010.
29. Peck E, Yuksel B, Harrison L, et al. Towards a 3-dimensional model of individual cognitive differences: position paper. In: *BELIV*, Seattle, WA, #13, 14–15 October 2012, article no. 6.
30. Bier EA, Card SK and Bodnar JW. Entity-based collaboration tools for intelligence analysis. In: *Proceedings of IEEE symposium on visual analytics science & technology*, Columbus, OH, 19–24 October 2008, pp. 99–106. New York: IEEE.
31. Jeong DH, Dou W, Stukes F, et al. Evaluating the relationship between user interaction and financial visual analysis. In: *Proceedings of the IEEE symposium on visual analytics science & technology*, Columbus, OH, 19–24 October 2008, pp. 83–90. New York: IEEE.
32. Stasko JT, Görg C, Liu Z, et al. Jigsaw: supporting investigative analysis through interactive visualization. *Inf Vis* 2008; 7(2): 118–132.
33. Saraiya P, North C, Lam V, et al. An insight-based longitudinal study of visual analytics. *IEEE T Vis Comput Gr* 2006; 12(6): 1511–1522.
34. Kwon B, Javed W, Ghani S, et al. Evaluating the role of time in investigative analysis of document collections. *IEEE T Vis Comput Gr* 2012; 18(11): 1992–2004.
35. Isenberg P, Tang A and Carpendale S. An exploratory study of visual information analysis. In: *Proceedings of the ACM conference on human factors in computing systems*, Florence, Italy, 5–10 April 2008, pp. 1217–1226. New York: ACM.
36. Robinson AC. Collaborative synthesis of visual analytic results. In: *Proceedings of the IEEE symposium on visual analytics science & technology*, Columbus, OH, 19–24 October 2008, pp. 67–74. New York: IEEE.
37. Van Ham F and Rogowitz BE. Perceptual organization in user-generated graph layouts. *IEEE T Vis Comput Gr* 2008; 14(6): 1333–1339.
38. Björk S and Holopainen J. *Patterns in game design*. Newton Center, MA: Charles River Media, 2004.
39. Gamma E. Design patterns—past, present & future. In: Nanz S (ed.) *The future of software engineering*. Berlin Heidelberg: Springer, 2011, 72 pp.
40. Heer J and Agrawala M. Software design patterns for information visualization. *IEEE T Vis Comput Gr* 2006; 12(5): 853–860.
41. Koenig A. Patterns and antipatterns. *JOOP* 1995; 8(1): 46–48.
42. Ware C, Purchase H, Colpoys L, et al. Cognitive measurements of graph aesthetics. *Inf Vis* 2002; 1(2): 103–110.
43. Walny J, Lee B, Johns P, et al. Understanding pen and touch interaction for data exploration on interactive whiteboards. *IEEE T Vis Comput Gr* 2012; 18(12): 2779–2788.
44. Pietriga E, Appert C and Beaudouin-Lafon M. Pointing and beyond: an operationalization and preliminary

evaluation of multi-scale searching. In: *Proceedings of the ACM conference on human factors in computing systems*, San Jose, CA, 28 April–3 May 2007, pp. 1215–1224. New York: ACM.

45. Kim SH, Dong Z, Xian H, et al. Does an eye tracker tell us the truth about visualizations?: findings while investigating visualizations for decision making. *IEEE T Vis Comput Gr* 2012; 18(12): 2421–2430.
46. Arias-Hernandez R, Kaastra LT, Green TM, et al. Pair analytics: capturing reasoning processes in collaborative visual analytics. In: *Proceedings of the Hawaii international conference on system sciences*, Kauai, HI, 4–7 January 2011, pp. 1–10. New York: IEEE.
47. Elmqvist N, Dragicevic P and Fekete JD. Color lens: adaptive color scale optimization for visual exploration. *IEEE T Vis Comput Gr* 2011; 17(6): 795–807.
48. Henry N and Fekete JD. MatrixExplorer: a dual-representation system to explore social networks. *IEEE T Vis Comput Gr* 2006; 12(5): 677–684.
49. Elmqvist N, Stasko JT and Tsigas P. DataMeadow: a visual canvas for analysis of large-scale multivariate data. *Inf Vis* 2008; 7(1): 18–33.
50. Shneiderman B and Plaisant C. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In: *Proceedings of the AVI workshop on beyond time and errors: novel evaluation methods for information visualization*, Venice, Italy, 23 May 2006, pp. 1–7. New York: ACM.
51. Ghani S and Elmqvist N. Improving revisitation in graphs through static spatial features. In: *Proceedings of the graphics interface conference*, St John's, NL, 25–27 May 2011, pp. 175–182. New York: ACM.
52. Gemmell J, Bell G and Lueder R. MyLifeBits: a personal database for everything. *Commun ACM* 2006; 49(1): 88–95.
53. Wigdor D, Penn G, Ryall K, et al. Living with a tabletop: analysis and observations of long term office use of a multi-touch table. In: *Proceedings of IEEE tabletop*, Newport, RI, 10–12 October 2007, pp. 60–67. New York: IEEE.
54. Hardy J. Experiences: a year in the life of an interactive desk. In: *Proceedings of the ACM conference on designing interactive systems*, Newcastle, UK, 11–15 June 2012, pp. 679–688. New York: ACM.
55. Dahlbäck N, Jonsson A and Ahrenberg L. Wizard of Oz studies—why and how. In: *Proceedings of the international workshop on intelligent user interfaces*, Orlando, FL, 4–7 January 1993, pp. 193–200. New York: ACM.
56. Davis RC, Saponas TS, Shilman M, et al. SketchWizard: wizard of Oz prototyping of pen-based user interfaces. In: *Proceedings of the ACM symposium on user interface software and technology*, Newport, RI, 6–10 October 2007, pp. 119–128. New York: ACM Press.
57. Javed W, Kim K, Ghani S, et al. Evaluating physical/virtual occlusion management techniques for horizontal displays. In: *Proceedings of INTERACT*, Lisbon, Portugal, 5–9 September 2011, pp. 391–408. Heidelberg: Springer-Verlag.
58. Arif AS and Stuerzlinger W. Analysis of text entry performance metrics. In: *Proceedings of the IEEE symposium on human factors and ergonomics*, Toronto, ON, Canada, 26–27 September 2009, pp. 100–105. New York: IEEE.
59. Kosara R and Ziemkiewicz C. Do Mechanical Turks dream of square pie charts. In: *Proceedings of beyond time and errors: novel evaluation methods for information visualization*, Atlanta, GA, 10–11 April 2010, pp. 63–70. New York: ACM.
60. Callison-Burch C. Fast, cheap, and creative: evaluating translation quality using Amazon's Mechanical Turk. In: *Proceedings of the conference on empirical methods in natural language processing*, Singapore, 6–7 August 2009, pp. 286–295. ACL and AFNLP.
61. Downs JS, Holbrook MB, Sheng S, et al. Are your participants gaming the system?: screening Mechanical Turk workers. In: *Proceedings of the ACM conference on human factors in computing systems*, Atlanta, GA, 10–15 April 2010, pp. 2399–2402. New York: ACM.
62. Ipeirotis PG. Analyzing the Amazon Mechanical Turk marketplace. *XRDS: Crossroads, The ACM Magazine for Students*, Winter 2010, 17(2), pp. 16–21.
63. Rogstadius J, Kostakos V, Kittur A, et al. An assessment of intrinsic and extrinsic motivation on task performance in crowdsourcing markets. In: *Proceedings of the AAAI conference on weblogs and social media*, Barcelona, 17–21 July 2011, pp. 321–328. Palo Alto, CA: AAAI.
64. Shaw AD, Horton JJ and Chen DL. Designing incentives for inexpert human raters. In: *Proceedings of the ACM conference on computer-supported cooperative work*, Hangzhou, China, 19–23 March 2011, pp. 275–284. New York: ACM.
65. Kim SH, Yun H and Yi JS. How to filter out random clickers in a crowdsourcing-based study? In: *Proceedings of beyond time and errors: novel evaluation methods for visualization*, Seattle, WA, 14–15 October 2012. New York: ACM.
66. Plaisant C, Grinstein GG, Scholtz J, et al. Evaluating visual analytics at the 2007 VAST symposium contest. *IEEE Comput Graph* 2008; 28(2): 12–21.
67. Grammel L, Tory M and Storey MA. How information visualization novices construct visualizations. *IEEE T Vis Comput Gr* 2010; 16(6): 943–952.
68. Yi JS, Elmqvist N and Lee S. TimeMatrix: visualizing temporal social networks using interactive matrix-based visualizations. *Int J Hum: Comput Int* 2010; 26(11–12): 1031–1051.
69. Creswell JW. *Qualitative inquiry and research design: choosing among five traditions*. Thousand Oaks, CA: SAGE, 1997.
70. Tinsley HE and Weiss DJ. Interrater reliability and agreement of subjective judgments. *J Couns Psychol* 1975; 22(4): 358–374.
71. Armstrong D, Gosling A, Weinman J, et al. The place of inter-rater reliability in qualitative research: an empirical study. *Sociology* 1997; 31(3): 597–606.
72. Wobbrock JO. Practical statistics for human-computer interaction: an independent study combining statistics theory and tool know-how. In: *Annual workshop of the human-computer interaction consortium*, Pacific Grove, CA, 14–18 June 2011.

73. Dragicevic P, Bezerianos A, Javed W, et al. Temporal distortion for animated transitions. In: *Proceedings of the ACM conference on human factors in computing systems*, Vancouver, BC, Canada, 7–12 May 2011, pp. 2009–2018. New York: ACM.
74. Brodlie K, Poon A, Wright H, et al. GRASPARC: a problem solving environment integrating computation and visualization. In: *Proceedings of the IEEE conference on visualization*, San Jose, CA, 25–29 October 1993, pp. 102–109. New York: IEEE.
75. Yi JS, Melton R, Stasko J, et al. Dust & Magnet: multivariate information visualization using a magnet metaphor. *Inf Vis* 2005; 4(4): 239–256.
76. Elmqvist N, Jean-Daniel PD and Fekete JD. Rolling the dice: multidimensional visual exploration using scatterplot matrix navigation. *IEEE T Vis Comput Gr* 2008; 14(6): 1141–1148.
77. Liu Z, Stasko J and Sullivan T. SellTrend: inter-attribute visual analysis of temporal transaction data. *IEEE T Vis Comput Gr* 2009; 15(6): 1025–1032.