A Nested Model for Visualization Design and Validation

Tamara Munzner, *Member, IEEE*

Abstract—We present a nested model for the visualization design and validation with four layers: characterize the task and data in the vocabulary of the problem domain, abstract into operations and data types, design visual encoding and interaction techniques, and create algorithms to execute techniques efficiently. The output from a level above is input to the level below, bringing attention to the design challenge that an upstream error inevitably cascades to all downstream levels. This model provides prescriptive guidance for determining appropriate evaluation approaches by identifying threats to validity unique to each level. We also provide three recommendations motivated by this model: authors should distinguish between these levels when claiming contributions at more than one of them, authors should explicitly state upstream assumptions at levels above the focus of a paper, and visualization venues should accept more papers on domain characterization.

Index Terms—Models, frameworks, design, evaluation.

1 Introduction

Many visualization models have been proposed to guide the creation and analysis of visualization systems [8, 7, 10], but they have not been tightly coupled to the question of how to evaluate these systems. Similarly, there has been significant previous work on evaluating visualization [9, 33, 42]. However, most of it is structured as an enumeration of methods with focus on *how* to carry them out, without prescriptive advice for *when* to choose between them.

The impetus for this work was dissatisfaction with a flat list of evaluation methodologies in a recent paper on the process of writing visualization papers [29]. Although that previous work provides some guidance for when to use which methods, it does not provide a full framework to guide the decision or analysis process.

In this paper, we present a model that splits visualization design into levels, with distinct evaluation methodologies suggested at each level based on the threats to validity that occur at that level. The four levels are: characterize the tasks and data in the vocabulary of the problem domain, abstract into operations and data types, design visual encoding and interaction techniques, and create algorithms to execute these techniques efficiently. We conjecture that many past visualization designers did carry out these steps, albeit implicitly or subconsciously, and not necessarily in that order. Our goal in making these steps more explicit is to provide a model that can be used either to analyze exisiting systems or papers, or to guide the design process itself.

The main contribution of this model is to give guidance on what evaluation methodology is appropriate to validate each of these different kinds of design choices. We break threats to validity down into four categories. In brief, where *they* is the users and *you* is the designer:

- wrong problem: they don't do that;
- wrong abstraction: you're showing them the wrong thing;
- wrong encoding/interaction: the way you show it doesn't work;
- wrong algorithm: your code is too slow.

The secondary contribution of this paper is a set of three recommendations motivated by this model. We suggest that authors distinguish between these levels when there is a contribution at more than one level, and explicitly stating upstream assumptions at levels above the focus of a paper. We also encourage visualization venues to accept more papers on domain characterization.

We present the base nested model in the next section, followed by the threats and validation approaches for the four levels. We give concrete examples of analysis according to our model for several previous systems, and compare our model to previous ones. We provide recommendations motivated by this model, and conclude with a discussion of limitations and future work.

- *Tamara Munzner is with the University of British Columbia, E-mail: tmm@cs.ubc.ca.*

Manuscript received 31 March 2009; accepted 27 July 2009; posted online 11 October 2009; mailed on 5 October 2009.
For information on obtaining reprints of this article, please send email to: tvcg@computer.org .

2 Nested Model

Figure 1 shows the nested four-level model for visualization design and evaluation. The top level is to characterize the problems and data of a particular domain, the next level is to map those into abstract operations and data types, the third level is to design the visual encoding and interaction to support those operations, and the innermost fourth level is to create an algorithm to carry out that design automatically and efficiently. The three inner levels are all instances of design problems, although it is a different problem at each level.

These levels are nested; the output from an *upstream* level above is input to the *downstream* level below, as indicated by the arrows in Figure 1. The challenge of this nesting is that an upstream error inevitably cascades to all downstream levels. If a poor choice was made in the abstraction stage, then even perfect visual encoding and algorithm design will not create a visualization system that solves the intended problem.

2.1 Vocabulary

The word *task* is deeply overloaded in the visualization literature [1]. It has been used at multiple levels of abstraction and granularity:

- high-level domain: cure disease, provide a good user experience during web search;
- lower-level domain: investigate microarray data showing gene expression levels and the network of gene interactions [6], analyze web session logs to develop hypotheses about user satisfaction [24];
- high-level abstract: expose uncertainty, determine domain parameters, confirm hypotheses [2];
- low-level abstract: compare, query, correlate, sort, find anomalies [1, 40].

In this paper we use the word *problem* to denote a task described in domain terms, and *operation* to denote an abstract task. level. We use *task* when discussing aspects that crosscut these levels.

2.2 Domain Problem and Data Characterization

At this first level, a visualization designer must learn about the tasks and data of target users in some particular target *domain*, such as microbiology or high-energy physics or e-commerce. Each domain usually has its own vocabulary for describing its data and problems, and there is usually some existing workflow of how the data is used to solve their problems. Some of the challenges inherent in bridging the gaps between designers and users are discussed by van Wijk [48].

A central tenet of human-centered design is that the problems of the target audience need to be clearly understood by the designer of

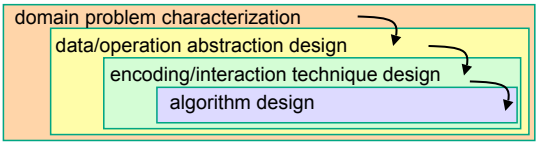

Fig. 1. Our model of visualization creation has four nested layers.

a tool for that audience. Although this concept might seem obvious, sometimes designers cut corners by making assumptions rather than actually engaging with any target users. Moreover, eliciting system requirements is not easy, even when a designer has access to target users fluent in the vocabulary of the domain and immersed in its workflow. As others have pointed out [42], asking users to simply introspect about their actions and needs is notoriously insufficient. Interviews are only one of many methods in the arsenal of ethnographic methodology [9, 39, 42].

The output of domain workflow characterization is often a detailed set of questions asked about or actions carried out by the target users for some heterogeneous collection of data. The details are necessary: in the list above, the high-level domain problem of "cure disease" is not sufficiently detailed to be input to the next abstraction level of the model, whereas the lower-level domain problem of "investigate microarray data showing gene expression levels and the network of gene interactions" is more appropriate. In fact, even that statement is a drastic paraphrase of the domain problem and data description in the full design study [6].

2.3 Operation and Data Type Abstraction

The abstraction stage is to map problems and data from the vocabulary of the specific domain into a more abstract and generic description that is in the vocabulary of computer science. More specifically, it is in the vocabulary of information visualization: the output of this level is a description of operations and data types, which are the input required for making visual encoding decisions at the next level.

By *operations*, we mean generic rather than domain-specific tasks. There has been considerable previous work on constructing taxonomies of generic tasks. The early work of Wehrend and Lewis also proposes a similar abstraction into operations and data types (which they call objects) [51]. Amar and Stasko have proposed a high-level task taxonomy: expose uncertainty, concretize relationships, formulate cause and effect, determine domain parameters, multivariate explanation, and confirm hypotheses [2]. Amar, Eagan, and Stasko have also proposed a categorization of low-level tasks as retrieve value, filter, compute derived value, find extremum, sort, determine range, characterize distribution, find anomalies, cluster, correlate [1]. Valiati *et al.* propose identify, determine, visualize, compare, infer, configure, and locate [47]. Although many operations are agnostic to data type, others are not. For example, Lee *et al.* propose a task taxonomy for graphs which includes following a path through a graph [25].

The other aspect of this stage is to transform the raw data into the *data types* that visualization techniques can address: a table of numbers where the columns contain quantitative, ordered, or categorical data; a node-link graph or tree; a field of values at every point in space. The goal is to find the right data type so that a visual representation of it will address the problem, which often requires transforming from the raw data into a derived type of a different form. Any data type can of course be transformed into any other. Quantitative data can be binned into ordered or categorical data, tabular data can be transformed into relational data with thresholding, and so on.

Unfortunately, despite encouragement to consider these issues from previous frameworks [8, 10, 43], an explicit discussion of the choices made in abstracting from domain-specific tasks and data to generic operations and data types is not very common in papers covering the design of actual systems. A welcome early exception is the excellent characterization of the scientific data analysis process by Springmeyer *et al.*, which presents an operation taxonomy grounded in observations of lab scientists studying physical phenomena [40].

However, frequently this abstraction is done implicitly and without justification. For example, many early web visualization papers implicitly posited that solving the "lost in hyperspace" problem should be done by showing the searcher a visual representation of the topological structure of its hyperlink connectivity graph [30]. In fact, people do not need an internal mental representation of this extremely complex structure to find a web page of interest. Thus, no matter how cleverly the information was visually encoded, these visualizations all incurred additional cognitive load for the user rather than reducing it.

This abstraction stage is often the hardest to get right. Many designers skip over the domain problem characterization level, assume the first abstraction that comes to mind is the correct one, and jump immediately into the third visual encoding level because they assume it is the only real or interesting design problem. Our guideline of explicitly stating the problem in terms of generic operations and data types may force a sloppy designer to realize that the level above needs to be properly addressed. As we discuss in Section 3.2, this design process is rarely strictly linear.

The first two levels, characterization and abstraction, cover both tasks and data. We echo the call of Pretorius and van Wijk that both of these points of departure are important for information visualization designers [34].

2.4 Visual Encoding and Interaction Design

The third level is designing the visual encoding and interaction. The design of visual encodings has received a great deal of attention in the foundational information visualization literature, starting with the influential work from Mackinlay [26] and Card *et al.* [8] (Chapter 1). The theory of interaction design for visualization is less well developed, but is starting to appear [23, 52]. We consider visual encoding and interaction together rather than separately because they are mutually interdependent. Many problem-driven visualization papers do indeed discuss the design issues for this level explicitly and clearly, especially those written as design studies [29].

2.5 Algorithm Design

The innermost level is to create an algorithm to carry out the visual encoding and interaction designs automatically. The issues of algorithm design are not unique to visualization, and are extensively discussed in the computer science literature [11].

3 Threats and Validation

Each level in this model has a different set of threats to validity, and thus requires a different approach to validation. Figure 2 shows a summary of the threats and validation approaches possible at each level, which are discussed in detail in the rest of this section. A single paper would include only a subset of these validation methods, ideally chosen according to the level of the contribution claims.

In our analysis below, we distinguish between *immediate* and *downstream* validation approaches. An important corollary of the model having nested levels is that most kinds of validation for the outer levels are not immediate because they require results from the downstream levels nested within them. The length of the red lines in Figure 2 shows the magnitude of the dependencies between the threat and the downstream validation, in terms of the number of levels that must be addressed. These downstream dependencies add to the difficulty of validation: a poor showing of a validation test that appears to invalidate a choice at an outer level may in fact be due to a poor choice at one of the levels inside it. For example, a poor visual encoding choice may cast doubt when testing a legitimate abstraction choice, or poor algorithm design may cast doubt when testing an interaction technique. Despite their difficulties, the downstream validations are necessary. The immediate validations only offer partial evidence of success; none of them are sufficient to demonstrate that the threat to validity at that level has been addressed.

3.1 Vocabulary

We have borrowed the evocative phrase *threats to validity* from the computer security domain, by way of the software engineering litera-

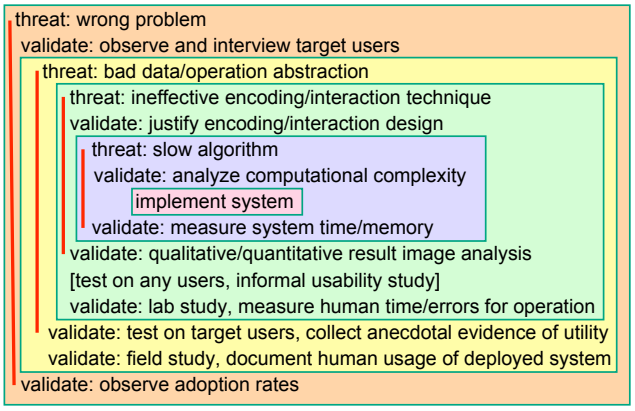

Fig. 2. Threats and validation in the nested model. Downstream levels are distinguished from upstream ones with containment and color, as in Figure 1. Many threats at the outer levels require downstream validation, which cannot be carried out until the inner levels within them are addressed, as shown by the red lines. Usually a single paper would only address a subset of these levels, not all of them at once.

ture. We use the word *validation* rather than *evaluation* to underscore the idea that validation is required for every level, and extends beyond user studies and ethnographic observation to include complexity analysis and benchmark timings. In software engineering, *validation* is about whether one has built the right product, and *verification* is about whether one has built the product right. Our use of *validation* includes both of these questions. In the simulation community, *validation* of the scientific model with respect to real-world observations is similarly considered separately from *verification* of the implementation, and connotes a level of rigor beyond the methods discussed here.

3.2 Iterative Loops and Rapid Prototyping

Although this model is cast as four nested layers for simplicity, in practice these four stages are rarely carried out in strict temporal sequence. There is usually an iterative refinement process, where a better understanding of one layer will feed back and forward into refining the others, especially with user-centered or participatory design approaches. The intellectual value of separating these four stages is that we can separately analyze whether each level has been addressed correctly, no matter in what order they were undertaken.

Similarly, the discussion below is simplified by implying that the only way to address nested layers is to carry out the full process of design and implementation. Of course, there are many rapid prototyping methodologies for accelerating this process by creating low-fidelity stand-ins exactly so that downstream validation can occur sooner. For example, paper prototypes and wizard-of-oz testing [12] can be used to get feedback from target users about abstraction and encoding designs without addressing the algorithm level at all.

3.3 Domain Threats

At the domain problem and data characterization level, the assertion is that particular problems of the target audience would benefit from visualization tool support. The primary threat is that the problem is mischaracterized: the target users do not in fact have these problems. An immediate form of validation is to interview and observe the target audience to verify the characterization, as opposed to relying on assumptions or conjectures. These validation approaches are mostly qualitative rather than quantitative [9, 14], and appropriate methodologies include ethnographic field studies and semi-structured interviews, as also advocated by Shneiderman and Plaisant [39]. Isenberg *et al.* propose the term *grounded evaluation* for this class of pre-design exploratory approaches [20].

A downstream form of validation is to report the rate at which the tool has been adopted by the target audience. We do note that adoption rates can be considered to be a weak signal with a large rate of false negatives and some false positives: many well-designed tools fail to be adopted, and some poorly-designed tools win in the marketplace. Nevertheless, the important aspect of this signal is that it reports what the target users do of their own accord, as opposed to the approaches below where target users are implicitly or explicitly asked to use a tool.

3.4 Abstraction Threats

At the abstraction design level, the threat is that the chosen operations and data types do not solve the characterized problems of the target audience. The key aspect of validation against this threat is that the system must be tested by target users doing their own work, rather than an abstract operation specified by the designers of the study.

A common downstream form of validation is to have a member of the target user community try the tool, in hopes of collecting anecdotal evidence that the tool is in fact useful. These anecdotes may have the form of insights found or hypotheses confirmed. Of course, this observation cannot be made until after all three of the other levels have been fully addressed, after the algorithm designed at the innermost level is implemented. Although this form of validation is usually qualitative, some influential work towards quantifying insight has been done [37].

A more rigorous validation approach for this level is to observe and document how the target audience uses the deployed system as part of their real-world workflow, typically in the form of a longer-term field study. We distinguish these field studies of deployed systems, which are appropriate for this level, from the exploratory pre-design field studies that investigate how users carry out their tasks before system deployment that are appropriate for the characterization level above. We do echo the call of Shneiderman and Plaisant [39] for more field studies of deployed systems. Although a few exist [15, 28], they are still far too rare given that they are the main validation method to address the threat at a critical design level. We conjecture that this shortage may be due to the downstream nature of the validation, with two levels of dependencies between the design choice and its testing.

3.5 Encoding and Interaction Threats

At the visual encoding and interaction design level, the threat is that the chosen design is not effective at communicating the desired abstraction to the person using the system. One immediate validation approach is to justify the design with respect to known perceptual and cognitive principles. Methodologies such as heuristic evaluation [53] and expert review [44] are a way to systematically ensure that no known guidelines are being violated by the design. A less structured approach is a free-form discussion of choices in a design study paper.

A downstream approach to validate against this threat is to carry out a formal user study in the form of a laboratory experiment. A group of people use the implemented system to carry out assigned tasks, usually with both quantitative measurements of the time spent and errors made and qualitative measurements such as preference. The size of the group is chosen based on the expected experimental effect size in hopes of achieving statistically significant results.

Another downstream validation approach is the presentation of and qualitative discussion of results in the form of still images or video. This approach is downstream because it requires an implemented system to carry out the visual encoding and interaction specifications designed at this level. This validation approach is strongest when there is an explicit discussion pointing out the desirable properties in the results, rather than assuming that every reader will make the desired inferences by unassisted inspection of the images or video footage. These qualitative discussions of images sometimes occur in a case study format, supporting an argument that the tool is useful for a particular task-dataset combination.

A third appropriate form of downstream validation is the quantitative measurement of result images created by the implemented system. For example, many measurable aesthetic criteria such as number of edge crossings and edge bends have been proposed in the subfield of graph drawing [41], some of which have been empirically tested [50].

Informal usability studies do appear in Figure 2, but are specifically not called a validation method. As Andrews eloquently states: "Formative methods [including usability studies] lead to better and more

usable systems, but neither offer validation of an approach nor provide evidence of the superiority of an approach for a particular context" [4]. They are listed at this level because it is a very good idea to do them upstream of attempting a validating laboratory or field study. If the system is unusable, no useful conclusions can be drawn from these methods. We distinguish usability studies from informal testing with users in the target domain, as described for the level above. Although the informal testing with target users described at the level above may uncover usability problems, the goal is to collect anecdotal evidence that the system meets its design goals. In an informal usability study, the person using the system does not need to be in the target audience, the only constraint is that the user is not the system designer. Such anecdotes are much less convincing when they come from a random person rather than a member of the target audience.

3.6 Algorithm Threats

At the algorithm design level, the primary threat is that the algorithm is suboptimal in terms of time or memory performance, either to a theoretical minimum or in comparison with previously proposed algorithms. An immediate form of validation is to analyze the computational complexity of the algorithm. The downstream form of validation is to measure the wall-clock time and memory performance of the implemented algorithm. Again, the methodology for algorithm analysis and benchmark measurements is so heavily addressed in the computer science literature that we do not belabor it here.

Another threat that is often addressed implicitly rather than explicitly is that the algorithm could be incorrect; that is, it does not meet the specification for the visual encoding or interaction design set at the level above. Presenting still images created by the algorithm or video of its use is also a validation against this threat, where the reader of a paper can directly see that the algorithm correctness objectives have been met. Usually there is no need for an explicit qualitative discussion of why these images show that the algorithm is in fact correct.

3.7 Mismatches

A common problem in weak visualization papers is a mismatch between the level at which the contribution is claimed and the validation methodologies chosen. For example, the contribution of a new visual encoding cannot not be validated by wall-clock timings of the algorithm, which addresses a level downstream of the claim. Similarly, the threat of a mischaracterized task cannot be addressed through a formal laboratory study where the task carried out by the participants is dictated by the study designers, so again the validation method is at a different level than the threat against the claim. This model explicitly separates the visualization design problem into levels in order to guide validation according to the unique threats at each level.

4 Examples

We now analyze several previous visualization papers in terms of our model, to provide concrete examples.

4.1 Genealogical Graphs

McGuffin and Balakrishnan present a system for the visualization of genealogical graphs [27]. They propose multiple new representations, including one based on the *dual-tree*, a subgraph formed by the union of two trees. Their prototype features sophisticated interaction, including automatic camera framing, animated transitions, and a new widget for ballistically dragging out subtrees to arbitrary depths.

This exemplary paper explicitly covers all four levels. The first domain characterization level is handled concisely but clearly: their domain is genealogy, and they briefly discuss the needs of and current tools available for genealogical hobbyists. The paper particularly shines in the analysis at the second abstraction level. They point out that the very term *family tree* is highly misleading, because the data type in fact is a more general graph with specialized constraints on its structure. They discuss conditions for which the data type is a true tree, a multitree, or a directed acyclic graph. They map the domain problem of recognizing nuclear family structure into operations about subgraph structure, and discuss the *crowding* problem at this abstract level. At the third level of our model, they discuss the strengths and weaknesses of several visual encoding alternatives, including using connection, containment, adjacency and alignment, and indentation. They present in passing two more specialized encodings, fractal node-link and containment for free trees, before presenting in detail their main proposal for visual encoding. They also carefully address interaction design, which also falls into the third level of our model. At the fourth level of algorithm design, they concisely cover the algorithmic details of dual-tree layout.

Three validation methods are used in this paper, shown in Figure 3. There is the immediate justification of encoding and interaction design decisions in terms of established principles, and the downstream method of a qualitative discussion of result images and videos. At the abstraction level, there is the downstream informal testing of a system prototype with a target user to collect anecdotal evidence.

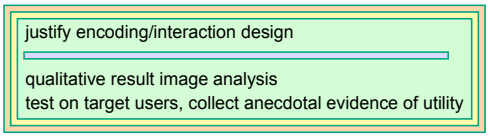

Fig. 3. Genealogical graphs [27] validation levels.

4.2 MatrixExplorer

Henry and Fekete present the MatrixExplorer system for social network analysis [17]. Its design comes from requirements formalized by interviews and participatory design sessions with social science researchers. They use both matrix representations to minimize clutter for large and dense graphs, and the more intuitive node-link representations of graphs for smaller networks.

All four levels of the model are addressed, with validation at three of the levels, shown in Figure 4. At the domain characterization level, there is explicit characterization of the social network analysis domain, which is validated with the qualitative techniques of interviews and an exploratory study using participatory design methods with social scientists and other researchers who use social network data. At the abstraction level, the paper includes a detailed list of requirements of the target user needs discussed in terms of generic operations and data types. There is a thorough discussion of the primary encoding design decision to use both node-link and matrix views to show the data, and also of many secondary encoding issues. There is also a discussion of both basic interactions and interactive reordering and clustering support. In both cases the authors use the immediate validation method of justifying these design decisions. There is also an extensive downstream validation of this level using qualitative discussion of result images. At the algorithm level, the focus is on the reordering algorithm. Downstream benchmark timings are mentioned very briefly.

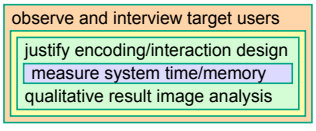

Fig. 4. MatrixExplorer [17] validation methods.

4.3 Flow Maps

Phan *et al.* propose a system for creating flow maps that show the movement of objects from one location to another, and demonstrate it on network traffic, census data, and trade data [32]. Flow maps reduce visual clutter by merging edges, but most previous instances were hand drawn. They present automatic techniques inspired by graph layout algorithms to minimize edge crossings and distort node positions while maintaining relative positions.

This paper has a heavy focus on the innermost algorithm design level, but also covers the encoding and abstraction levels. Their analysis of the useful characteristics of hand-drawn flow maps falls into the abstraction level of our model. At the visual encoding level, they have a brief but explicit description of the goals of their layout algorithm, namely intelligent distortion of positions to ensure that the separation distance between nodes is greater than the maximum thickness of the flow lines while maintaining left-right and up-down ordering relationships. The domain characterization level is addressed more implicitly than explicitly: there is no actual discussion of who uses flow maps and why. However, the analysis of hand-drawn flow maps could be construed as an implicit claim of longstanding usage needs.

Three validation methods were used in this paper, shown in Figure 5. At the algorithm level, there is an immediate complexity analysis. There is also a brief downstream report of system timing, saying that all images were computed in a few seconds. There was also a more involved downstream validation through the qualitative discussion of result images generated by their system. In this case, the intent was mainly to discuss algorithm correctness issues at the innermost algorithm level, as opposed to addressing the visual encoding level.

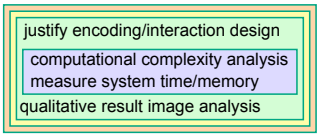

Fig. 5. Flow Map [32] validation methods.

4.4 LiveRAC

McLachlan *et al.* present the LiveRAC system for exploring system management time-series data [28]. It uses a reorderable matrix of charts with stretch and squish navigation combined with semantic zooming, so that the chart's visual representation adapts to the available space. They carry out an informal longitudinal field study of its deployment to operators of a large corporate web hosting service. Four validation methods were used in this paper, shown in Figure 6.

At the domain characterization level, the paper explains the roles and activities of system management professionals and their existing workflow and tools. The immediate validation approach was interviews with the target audience. Their phased design methodology, where management approval was necessary for access to the true target users, makes our use of the word *immediate* for this validation a bit counterintuitive: many of these interviews occurred after a working prototype was developed. This project is a good example of the iterative process we allude to in Section 3.2.

At the abstraction level, the choice of a collection of time-series data for data type is discussed early in the paper. The rationale is presented in the opposite way from our discussion above: rather than justifying that time-series data is the correct choice for the system management domain, they justify that this domain is an appropriate one for studying this data type. The paper also contains a set of explicit design requirements, which includes abstract operations like search, sort, and filter. The downstream validation for the abstraction level is a longitudinal field study of the system deployed to the target users, life cycle engineers for managed hosting services inside a large corporation.

At the visual encoding and interaction level, there is an extensive discussion of design choices, with immediate validation by justification in terms of design principles. Algorithms are not discussed.

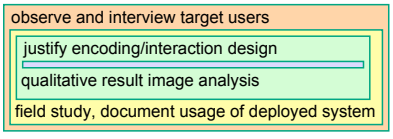

Fig. 6. LiveRAC [32] validation methods.

4.5 LinLog

Noack's LinLog paper introduces an energy model for graph drawing designed to reveal clusters in the data, where clusters are defined as a set of nodes with many internal edges and few edges to nodes outside the set [31]. Energy-based and force-directed methods are related approaches to graph layout, and have been heavily used in information visualization. Previous models strove to enforce uniform edge lengths as an aesthetic criterion, but Noack points out that to create visually distinguishable clusters requires long edges between them.

Although a quick glance might lead to an assumption that this graph drawing paper has a focus on algorithms, the primary contribution is in fact at the visual encoding level. The two validation methods used in the paper are qualitative and quantitative result image analysis, shown in Figure 7.

Noack clearly distinguishes between the two aspects of energy-based methods for force-directed graph layout: the energy model itself, versus the algorithm that searches for a state with minimum total energy. In the vocabulary of our model, his LinLog energy model is a visual encoding design choice. Requiring that the edges between clusters are longer than those within clusters is a visual encoding using the visual channel of spatial position. One downstream validation approach in this paper is a qualitative discussion of result images, which we consider appropriate for a contribution at the encoding level. This paper also contains a validation method not listed in our model, because it is relatively rare in visualization: mathematical proof. These proofs are about the optimality of the model results when measured by quantitative metrics involving edge lengths and node distances. Thus, we classify it in the quantitative image analysis category, another appropriate method to validate at the encoding level.

This paper does not in fact address the innermost algorithm level. Noack explicitly leaves the problem of designing better energy-minimization algorithms as future work, using previously proposed algorithms to showcase the results of his model. The top domain characterization level is handled concisely but adequately by referencing previous work about application domains with graph data where there is a need to see clusters. For the second abstraction level, although the paper does not use our model vocabulary of *operation* and *data type*, it clearly states the abstraction that the operation is finding clusters for the data type of a node-link graph.

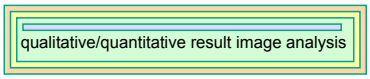

Fig. 7. LinLog [31] validation methods.

4.6 Lab Studies

Many laboratory studies are designed to validate and invalidate specific design choices at the visual encoding and interaction level by measuring time and error rates of people carrying out abstracted tasks, as Figure 8 shows. For instance, Robertson *et al.* test the effectiveness of animation compared to both traces and small multiples for showing trends [36]. They find that animation is the least effective form for analysis; both static depictions of trends are significantly faster than animation, and the small multiples display is more accurate. Heer *et al.* compare line charts to the more space-efficient horizon graphs [16]. They identify transition points at which reducing the chart height results in significantly differing drops in estimation accuracy across the compared chart types, and find optimal positions in the speed-accuracy tradeoff curve at which viewers performed quickly without attendant drops in accuracy.

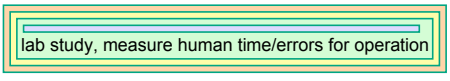

Fig. 8. Lab studies as a validation method.

5 Comparison to Other Models

We now discuss how our model fits within the context of previous work on visualization models and evaluation techniques.

5.1 Visualization Models

As we discuss above, previous pipeline models have been proposed to guide the creation and analysis of visualization systems [7, 8, 10]. Our model is heavily influenced by them: our Abstraction level corresponds to the Data and Visualization Transformation stages, and our Visual Encoding level corresponds to the Visual Mapping Transformation stage. The limitation that we address is that these previous models were not tightly coupled to the question of how to evaluate them. Similarly, the importance of tasks is clearly articulated in Shneiderman's taxonomy [38], but there is no guidance on evaluation methodology. A recent chapter presents four theoretical models for information visualization [35], but again none of these models offer explicit guidance on how to tightly couple design and evaluation.

Some of the issues we discuss at the Abstraction level were also addressed in Tory and Möller's discussion of the transformation from user model into design model [43]. The task taxonomies [1, 2, 51] are also an important guide at this level, but the goal of our model is to address a broader scope.

As we also discuss above, there has been significant previous work on evaluating visualization [9, 33, 42], but mostly with the focus on how to use the methods rather than when to use them. One welcome exception is an article by Kosara *et al.* [22], which explicitly discusses not only how to do user studies but also why and when to do them. Their discussion is a good first step, but does not provide the framework of a formal model. Another is the multi-dimensional in-depth long-term case studies (MILCs) approach, advocated by Shneiderman and Plaisant [39], which does partially address the question of when to use what kind of evaluation. While we agree with many of their ideas at a high level, one of the ways we differ is by drawing clearer lines between levels, with the goal of providing guidance that can be used outside of long-term case studies, in a broad spectrum of different visualization design contexts.

5.2 Formative, Summative, and Exploratory Evaluation

Significant previous work has been devoted to answering the question of *when* to use what kind of evaluation in the larger context of human-computer interaction. The three-level classification of evaluation into formative, summative, and exploratory methods is very relevant. *Formative* evaluations are intended to provide guidance to the designers on how to improve a system and answer the question "can I make it better?" [3, 13]. Informal usability studies are the classic example. Other evaluation methodologies typically considered formative include cognitive walkthroughs [42], expert reviews [44], and heuristic evaluations [53]. In contrast, formal laboratory studies and post-deployment field studies are *summative* evaluations, intended to measure the performance of a system and answer the question "is it right?" [3, 13]. A third kind of evaluation is *exploratory*, intended to answer the question "can I understand more?" [4, 13]. One example of these are the ethnographic pre-design field studies. (We note that any of these three types can involve qualitative or quantitative methodology, and similarly that both field and laboratory studies can involve either methodology.)

Ellis and Dix argue convincingly that even laboratory studies often end up being used for formative purposes, despite an original summative intent [13]. Post-deployment field studies can also be done with exploratory intent rather than, or in addition to, summative intent. We would like to make a similar argument in the opposite direction. We suggest that expert reviews and heuristic evaluations can be used with the intent of summative evaluation, as an immediate validation approach for the encoding and interaction design choices. We reiterate that it would be dangerous to stop there and declare victory; downstream validation with real users is certainly called for as well.

The previous work of Andrews [4] and Ellis and Dix [13] is perhaps the closest in spirit to this paper, explicitly addressing the question of when to use what evaluation methods for visualization in particular. However, our model provides a tightly coupled connection between design and evaluation at four distinct stages, as opposed to their more general discussion about three of the stages. Moreover, they do not include algorithm-level threats to validity in their analysis, whereas our model does. As we discuss in Section 6, the line between visual encoding and algorithms can be surprisingly murky, so untangling contributions between these two levels will be an aid to clear discussion.

6 Recommendations

Our model gives rise to three recommendations. First, we suggest that authors who make contributions at multiple levels should clearly distinguish between them. Second, we suggest that authors should clearly state what assumptions they make for levels upstream of their focus. Both of these recommendations are intended to help readers synthesize new work into a coherent larger picture more easily, and to help subsequent researchers build on the work more easily. Third, we argue that the visualization community would benefit from more papers that focus on problem characterization, and thus that they should be encouraged at visualization venues.

6.1 Distinguish Between Levels

For papers that have contributions at multiple levels, we advocate clearly distinguishing between claims according to the level at which they occur. For example, a hypothetical paper might claim as a contribution both a domain problem characterization validated by observational study and a new visual encoding technique validated only by qualitative arguments about result images. There are no claims at the other two levels because it relies on a previously proposed abstraction approach, and technique is so straightforward that only a very high-level algorithm needs to be described so that the work is replicable.

The value of making these distinctions clearly is that readers can more easily synthesize a coherent picture of how new work fits together with the existing literature to advance the state of the field. It also allows subsequent authors to more easily build on the work. In the case above, it would be clear to potential authors of a follow-on paper that validating the encoding technique with a formal laboratory study would address an open problem. If the author is the one to distinguish between the levels, then all subsequent readers will have a shared and clear idea of this characterization. When future readers must create individual interpretations, there will be less consensus on what aspects remain as open problems, versus as partial solutions that can be further refined, versus as closed problems with comprehensive solutions.

Making these distinctions is not always straightforward. The problem of murky entanglement between the visual encoding and algorithm levels occurs in papers throughout information visualization. We illustrate the difficulty with another example from the subfield of graph drawing. Archambault *et al.* present the TopoLayout system for drawing multilevel graphs [5]. Topological features such as trees, biconnected components, and clusters are recursively detected, and collapsed into nodes to create a multilevel hierarchy atop the original base graph. Each feature is drawn with an algorithm appropriate for its structure. All drawing algorithms are area-aware, taking the space required to draw lower-level features into account at higher levels of the graph hierarchy.

The paper may appear at first glance to have a heavily algorithmic focus. There is a thorough discussion of the algorithm design, and validation for that level includes immediate complexity analysis and downstream benchmark timings against several competing systems. The second abstraction level is expressed reasonably clearly, namely that the operation is seeing structure at multiple scales for the data type of a node-link graph.

However, the paper also uses the validation method of an extensive qualitative discussion of result images, with an emphasis on visual quality rather than algorithm correctness. Looking through the lens of our model, we interpret this choice to mean that the paper is also staking a claim at the visual encoding level. However, considerations at the visual encoding level are not discussed very explicitly. The somewhat implicit visual encoding claim is that a good spatial layout should have

as little overlap as possible between interesting substructures. This visual encoding choice is intriguing, and the qualitative image discussion makes a good case for it. If this visual encoding choice had been described more clearly and explicitly in the paper itself, it would perhaps be easier for subsequent researchers to build on the work. For example, they could compare this choice with other visual encodings, or to create faster algorithms to accomplish the same encoding.

6.2 State Upstream Assumptions

In the common case where the focus of a paper is on only a subset of the four levels, we advocate explicitly reporting assumptions about levels upstream of the focus. The value of doing so, as above, is to create a clear foundation for subsequent authors and to help readers understand how the work fits with respect to the existing literature. This reporting can be very brief, especially when it includes a citation to previous work that does address the level in question. As discussed above, Noack's LinLog paper handles domain characterization adequately with a single sentence.

The level most often neglected in visualization paper is the abstraction level. We conjecture that guiding authors to include even a few sentences about the chosen abstraction may encourage designers to consider their choices more carefully.

6.3 Encourage Problem Characterization Papers

This model highlights the importance of problem characterization, which is showcased as one of only four levels. However, hardly any papers devoted solely to analysis at this level have been published in venues explicitly devoted to visualization: we are aware of only one [45]. Isenberg *et al.* [20] note that these kinds of papers are common in the computer supported cooperative work [46] and HCI communities [19]. We echo their call to arms that the visualization community would benefit from more such papers. The domain problem characterization stage is both difficult and time consuming to do properly. People who have not had the experience of doing so may be tempted to assume it is trivial. We argue against this mistake.

To use the language of paper types [29], we note that while Design Studies often include a discussion of this problem characterization level as just part of a larger contribution, a full-fledged exploratory study that characterizes the workflow in a problem domain can be a paper-sized contribution in its own right, in the Evaluation category.

7 Discussion and Limitations

While this model does emphasize a problem-driven approach to visualization design, it also applies to technique-driven research. In particular, our recommendations to state upstream assumptions and distinguish between levels are offered in hope of creating a more unified visualization literature where these two approaches interleave usefully.

A clear limitation of this model is that it errs on the side of oversimplifying the situation. This choice was deliberate, as the goal of providing very clear guidance took priority over that of presenting a more subtle and sophisticated discussion. Many nuances of evaluation methodology are glossed over here. Moreover, the reductionist assumption that complex sensemaking tasks can be broken down into low-level components is unlikely to be completely true.

This model is by no means the only way to approach the design and development process. For example, van Wijk urges that designers first set up requirements, then generate multiple solutions, then match the solutions to the requirements and pick the best one [49]. Our model combines well with his process, the three design levels could each be addressed with that approach. Also, even very different process approaches could be analyzed post hoc with this model in mind, to ensure that each of the specific stages that we identify has been adequately addressed.

We do not describe how to carry out any of the validation methodologies discussed here, as a great deal of previous work already covers that material. We also deliberately leave out some kinds of user studies from our discussion, such as the psychophysical style of characterizing human perceptual and cognitive capabilities, because their intent is not to validate a particular design or application.

The examples and vocabulary of this paper arise from information visualization (infovis) rather than scientific visualization (scivis). The two subfields have enough methodological differences that adapting this model to reflect the concerns of the scivis process would require significant future work. In scivis the visual encoding for spatial position is typically intrinsic to the given dataset, so is not available as a degree of freedom in the visualization design. Similarly, the abstraction stage may be highly constrained by the input data and task, and thus the scope of its validation may be similarly constrained. However, there are some interesting correspondences. For example, *feature-based* approaches in the scivis literature often involve nontrivial decisions at both the abstraction and visual encoding stages [18], and transfer function design [21] also falls into the visual encoding stage.

Our list of threats and validation methodologies is not exhaustive. We conjecture that other threats and validation approaches could also be usefully classified into one of the four levels of the existing model. However, others may argue for cleaving the process into more or different levels. For example, does it make sense to separate domain task and problem characterization from operation and data abstraction? On the one hand, it is useful to be able to distinguish them at the level of validation, by treating exploratory field studies separately from post-deployment field studies. On the other hand, a characterization of a domain with no attempt at abstraction may not be very useful, so perhaps collapsing them into one level would be more apt.

8 Conclusion

We have presented a model that classifies validation methodologies for use at only one of four separate levels, in a unified approach to visualization design and evaluation. We offer it in hopes of spurring further discussion about the interplay between design and evaluation.

Acknowledgments

We thank James Ahrens, Gordon Kindlmann, Heidi Lam, Miriah Meyer, Melanie Tory, Jarke van Wijk, and the anonymous reviewers for helpful comments on previous drafts of this paper. We thank Hamish Carr for the conversation that started this train of thought.

References

[1] R. Amar, J. Eagan, and J. Stasko. Low-level components of analytic activity in information visualization. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 111–117, 2005.

[2] R. Amar and J. Stasko. A knowledge task-based framework for the design and evaluation of information visualizations. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 143–149, 2004.

[3] K. Andrews. Evaluating information visualizations. In *Proc. AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM, 2006. Article 1.

[4] K. Andrews. Evaluation comes in many guises. AVI Workshop on BEyond time and errors (BELIV) Position Paper, 2008. `www.dis.uniroma1.it/~beliv08/pospap/andrews.pdf`.

[5] D. Archambault, T. Munzner, and D. Auber. TopoLayout: Multilevel graph layout by topological features. *IEEE Trans. on Visualization and Computer Graphics*, 13(2):305–317, March/April 2007.

[6] A. Barsky, T. Munzner, J. Gardy, and R. Kincaid. Cerebral: Visualizing multiple experimental conditions on a graph with biological context. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis '08)*, 14(6):1253–1260, 2008.

[7] S. K. Card and J. Mackinlay. The structure of the information visualization design space. In *Proceedings of the 1997 IEEE Symposium on Information Visualization (InfoVis)*, pages 92–99, 1997.

[8] S. K. Card, J. Mackinlay, and B. Shneiderman. *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.

[9] S. Carpendale. Evaluating information visualizations. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization: Human-Centered Issues and Perspectives*, volume 4950, pages 19–45. Springer LNCS, 2008.

[10] E. H. Chi and J. T. Riedl. An operator interaction framework for visualization systems. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 63–70, 1998.

[11] T. H. Cormen, C. E. Leiserson, and R. L. Rivest. *Introduction to Algorithms*. MIT Press, 1990.

[12] S. Dow, B. MacIntyre, J. Lee, C. Oezbek, J. D. Bolter, and M. Gandy. Wizard of Oz support throughout an iterative design process. *IEEE Pervasive Computing*, 4(4):18–26, Oct-Dec 2005.

[13] G. Ellis and A. Dix. An explorative analysis of user evaluation studies in information visualisation. In *Proc. AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM, 2006. Article 2.

[14] S. Faisal, B. Craft, P. Cairns, and A. Blandford. Internalization, qualitative methods, and evaluation. In *Proc. AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM, 2008. Article 5.

[15] V. González and A. Kobsa. A workplace study of the adoption of information visualization systems. *Proc. Intl. Conf. Knowledge Management (I-KNOW)*, pages 92–102, 2003.

[16] J. Heer, N. Kong, and M. Agrawala. Sizing the horizon: The effects of chart size and layering on the graphical perception of time series visualizations. In *Proc. ACM Human Factors in Computing Systems (CHI)*, pages 1303–1312, 2009.

[17] N. Henry and J.-D. Fekete. MatrixExplorer: a dual-presentation system to explore social networks. *IEEE Trans. Visualization and Computer Graphics (Proc. Infovis '06)*, 12(5):677–684, 2006.

[18] C. Henze. Feature detection in linked derived spaces. In *IEEE Conf. Visualization (Vis)*, pages 87–94, 1998.

[19] P. Isenberg, A. Tang, and S. Carpendale. An exploratory study of visual information analysis. In *Proc. ACM Conf. on Human Factors in Computing Systems (CHI)*, pages 1217–1226, 2008.

[20] P. Isenberg, T. Zuk, C. Collins, and S. Carpendale. Grounded evaluation of information visualizations. In *Proc. AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM, 2008. Article 6.

[21] G. Kindlmann. Transfer functions in direct volume rendering: Design, interface, interaction. SIGGRAPH 2002 Course Notes. `http://www.cs.utah.edu/~gk/papers/sig02-TF-notes.pdf`.

[22] R. Kosara, C. G. Healey, V. Interrante, D. H. Laidlaw, and C. Ware. Thoughts on user studies: Why, how, and when. *IEEE Computer Graphics and Applications*, 23(4):20–25, 2003.

[23] H. Lam. A framework of interaction costs in information visualization. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis '08)*, 14(6):1149–1156, 2008.

[24] H. Lam, D. Russell, D. Tang, and T. Munzner. Session Viewer: Visual exploratory analysis of web session logs. In *Proc. IEEE Symposium on Visual Analytics Science and Technology (VAST)*, pages 147–154, 2007.

[25] B. Lee, C. Plaisant, C. S. Parr, J.-D. Fekete, and N. Henry. Task taxonomy for graph visualization. In *Proc. AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM, 2006.

[26] J. D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Trans. on Graphics (TOG)*, 5(2):111–141, 1986.

[27] M. J. McGuffin and R. Balakrishnan. Interactive visualization of genealogical graphs. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 17–24, 2005.

[28] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC - interactive visual exploration of system management time-series data. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 1483–1492, 2008.

[29] T. Munzner. Process and pitfalls in writing infovis research papers. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization: Human-Centered Issues and Perspectives*, volume 4950 of *LNCS*, pages 133–153. Springer-Verlag, 2008.

[30] T. Munzner and P. Burchard. Visualizing the structure of the world wide web in 3D hyperbolic space. In *Proc. Virtual Reality Modeling Language Symposium (VRML)*, pages 33–38. ACM SIGGRAPH, 1995.

[31] A. Noack. An energy model for visual graph clustering. In *Proc. Graph Drawing (GD'03)*, volume 2912 of *LNCS*, pages 425–436. Springer-Verlag, 2003.

[32] D. Phan, L. Xiao, R. Yeh, P. Hanrahan, and T. Winograd. Flow map layout. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 219–224, 2005.

[33] C. Plaisant. The challenge of information visualization evaluation. In *Proc. Advanced Visual Interfaces (AVI)*, pages 109–116. ACM Press, 2004.

[34] A. J. Pretorius and J. J. van Wijk. What does the user want to see? what do the data want to be? *Information Visualization Journal*, 2009. advance online publication Jun 4, doi:10.1057/ivs.2009.13.

[35] H. C. Purchase, N. Andrienko, T. Jankun-Kelly, and M. Ward. Theoretical foundations of information visualization. In A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, editors, *Information Visualization: Human-Centered Issues and Perspectives*, volume 4950 of *LNCS*, pages 46–64. Springer-Verlag, 2008.

[36] G. Robertson, R. Fernandez, D. Fihser, B. Lee, and J. Stasko. Effectiveness of animation in trend visualization. *IEEE Trans. Visualization and Computer Graphics (Proc. Infovis '08)*, 14(6):1325–1332, 2008.

[37] P. Saraiya, C. North, and K. Duca. An insight-based methodology for evaluating bioinformatics visualizations. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 11(4):443–456, 2005.

[38] B. Shneiderman. The eyes have it: A task by data type taxonomy of information visualization. In *Proc. IEEE Visual Languages*, pages 336–343, 1996.

[39] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: Multi-dimensional in-depth long-term case studies. In *Proc. AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM, 2006. Article 6.

[40] R. R. Springmeyer, M. M. Blattner, and N. L. Max. A characterization of the scientific data analysis process. In *Proc. IEEE Visualization (Vis)*, pages 235–242, 1992.

[41] K. Sugiyama, S. Tagawa, and M. Toda. Methods for visual understanding of hierarchical system structures. *IEEE Trans. Systems, Man and Cybernetics*, 11(2):109–125, 1981.

[42] M. Tory and T. Möller. Human factors in visualization research. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 10(1):72–84, 2004.

[43] M. Tory and T. Möller. Rethinking visualization: A high-level taxonomy. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 151–158, 2004.

[44] M. Tory and T. Möller. Evaluating visualizations: Do expert reviews work? *IEEE Computer Graphics and Applications*, 25(5), Sept. 2005.

[45] M. Tory and S. Staub-French. Qualitative analysis of visualization: A building design field study. In *Proc. AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM, 2008. Article 7.

[46] M. Tory, S. Staub-French, B. Po, and F. Wu. Physical and digital artifact-mediated coordination in building design. *Computer Supported Cooperative Work (CSCW)*, 17(4):311–351, Aug. 2008.

[47] E. Valiati, M. Pimenta, and C. M. Freitas. A taxonomy of tasks for guiding the evaluation of multidimensional visualizations. In *Proc. AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM, 2006. Article 15.

[48] J. J. van Wijk. Bridging the gaps. *IEEE Computer Graphics & Applications*, 26(6):6–9, 2006.

[49] J. J. van Wijk. Views on visualization. *IEEE Trans. Visualization and Computer Graphics*, 12(2):421–432, 2006.

[50] C. Ware, H. Purchase, L. Colpys, and M. McGill. Cognitive measurements of graph aesthetics. *Information Visuazation*, 1(2):103–110, 2002.

[51] S. Wehrend and C. Lewis. A problem-oriented classification of visualization techniques. In *Proc. IEEE Visualization (Vis)*, pages 139–143, 1990.

[52] J. S. Yi, Y. A. Kang, J. T. Stasko, and J. A. Jacko. Toward a deeper understanding of the role of interaction in information visualization. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis '07)*, 13(6):1224–1231, 2007.

[53] T. Zuk, L. Schlesier, P. Neumann, M. S. Hancock, and S. Carpendale. Heuristics for information visualization evaluation. In *Proc. AVI Workshop on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM, 2008. Article 9.