The Nested Blocks and Guidelines Model

Miriah Meyer
University of Utah
miriah@cs.utah.edu

Michael Sedlmair
University of Vienna
michael.sedlmair@univie.ac.at

P. Samuel Quinan
University of Utah
psq@cs.utah.edu

Tamara Munzner
University of British Columbia
tmm@cs.ubc.ca

ABSTRACT

We propose the nested blocks and guidelines model (NBGM) for the design and validation of visualization systems. The NBGM extends the previously proposed four-level nested model by adding finer grained structure within each level, providing explicit mechanisms to capture and discuss design decision rationale. Blocks are the outcomes of the design process at a specific level, and guidelines discuss relationships between these blocks. Blocks at the algorithm and technique levels describe design choices, as do data blocks at the abstraction level, whereas task abstraction blocks and domain situation blocks are identified as the outcome of the designer's understanding of the requirements. In the NBGM, there are two types of guidelines: *within-level* guidelines provide comparisons for blocks within the same level, while *between-level* guidelines provide mappings between adjacent levels of design. We analyze several recent papers using the NBGM to provide concrete examples of how a researcher can use blocks and guidelines to describe and evaluate visualization research. We also discuss the NBGM with respect to other design models to clarify its role in visualization design. Using the NBGM, we pinpoint two implications for visualization evaluation. First, comparison of blocks at the domain level must occur implicitly downstream at the abstraction level; and second, comparison between blocks must take into account both upstream assumptions and downstream requirements. Finally, we use the model to analyze two open problems: the need for mid-level task taxonomies to fill in the task blocks at the abstraction level, as well as the need for more guidelines mapping between the algorithm and technique levels.

Categories and Subject Descriptors

H.5.2 [**Information Systems Application**]: User Interfaces—*Evaluation/methodology*

Keywords

Nested model, validation, design studies, visualization

1. INTRODUCTION

Creating visualization tools and techniques is a *design* process. To guide and inform design, many different models have been proposed in many different design disciplines including visualization [12, 27, 57, 58], software engineering [2, 8, 13, 23, 32, 42, 50, 52] and design itself [10, 19, 51, 53, 59, 60]. In visualization, researchers have mainly focused on **process models** that describe stages with concrete actions a designer should engage in, and **architecture models** that focus on the structure of a software system in terms of its components. An example of a process model is the nine-stage framework for visualization design studies that provides practical guidance for collaborative visualization projects [57]; an example of an architecture model is the venerable visualization pipeline model [12]. Visualization research to date, however, has paid little attention to **design decision models** that allow for describing and capturing design decisions. Other design-related communities have noted the importance of such models: these models help designers to better structure the often ill-defined design problem, support the exploration of different design alternatives, and allow understanding, discussion and reasoning about a particular design [50].

In this paper we propose a new design decision model for visualization called the **nested blocks and guidelines model** (**NBGM**). This model is based on the nested model proposed by Munzner in 2009 [43] which provides a framework for thinking about the design and validation of visualization systems at four levels. The original nested model focuses on providing guidance in terms of choosing appropriate validation techniques at each level, and stresses the negative and cascading implications of poor design decisions. It has provided guidance, motivation, framing, and ammunition for a broad range of visualization papers, including problem-driven design studies [14, 18, 45, 49, 56, 62], technique-driven work [33], evaluation [1, 61], models [15, 17, 30, 35, 55, 57, 66], and systems [5, 20].

While we and others have used the nested model extensively as a way to guide decisions about evaluation and design, we have found that it falls short in capturing and describing design decisions at the level of detail needed for thorough analysis. To overcome this deficiency, we opt to leverage the popularity of the original nested model and propose the NBGM as an extension. The NBGM supports design and analysis at a finer grain by explicitly considering components within each of the four levels, as well as the relationships between them. This extension provides a direct mechanism for capturing the rationale behind visualization design decisions, thereby providing context for decision making.
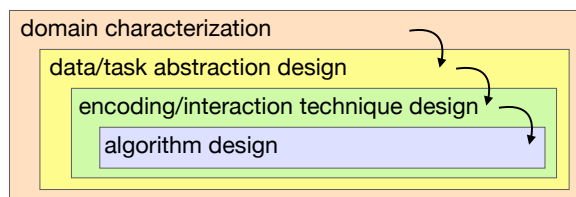
Figure 1: Original depiction of the nested design model [43], with arrows indicating the cascading effects of decisions made at higher levels.

The NBGM proposes *blocks* as a generic term for components that are the outcomes of the design process at each level. Concrete examples of blocks are a network as a data abstraction block, a node-link diagram as a visual encoding technique block, and a specific force-directed layout approach such as GEM [22] as an algorithm block. We then define *guidelines* as statements about the relationships between blocks. Concrete examples of guidelines are that a node-link diagram is a good visual encoding of small graphs, or one specific force-directed layout algorithm is faster than another.

The primary contribution of this paper is the nested blocks and guidelines model (NBGM), presented in Section 2. In Section 3 we provide several concrete examples of how a researcher can use the NBGM for analysis to describe and evaluate visualization research. Section 4 features a characterization of three types of design models that we identified through surveying related work across several fields, and situates the NBGM within this characterization in order to clarify its role in visualization design. We also use the NBGM in a generative way to illuminate and discuss several open issues, which leads to two secondary contributions. First, we derive a set of implications for visualization evaluation in Section 5.1: the necessity of abstraction blocks to compare domain level findings, and the challenge of directly comparing blocks due to the inherent reliance on both upstream assumptions and downstream requirements. Second, in Section 5.2, we present an analysis of two open problems in our field using the NBGM: the need for more work and clarity at the abstraction level, and the need to establish a more complete set of guidelines that map up from the algorithm level to the technique level. This analysis illustrates the descriptive capabilities of the NBGM for reasoning about visualization research challenges.

This work builds on that presented in a previous workshop paper [40]. The major additions in this paper are further definitions for blocks at each level of the model in Section 2; an analysis of three papers from the IEEE InfoVis'12 conference proceedings using the NBGM in Section 3; and an examination of design models in previous work and the role of the NBGM in visualization design in Section 4.

2. ADDING BLOCKS AND GUIDELINES

The original description of the nested model [43] breaks down the design and evaluation of a visualization project into four nested levels, shown in Figure 1. The highest level is the characterization of the domain of interest; the next level is the design of the data and task abstractions for that characterization; the third level is the design of the visual encoding and interaction techniques for those abstractions; and the lowest level is the design of the algorithms that implement those techniques programmatically. The focus of the original nested model is on the cascading implications of design decisions made at different levels, where the decisions made
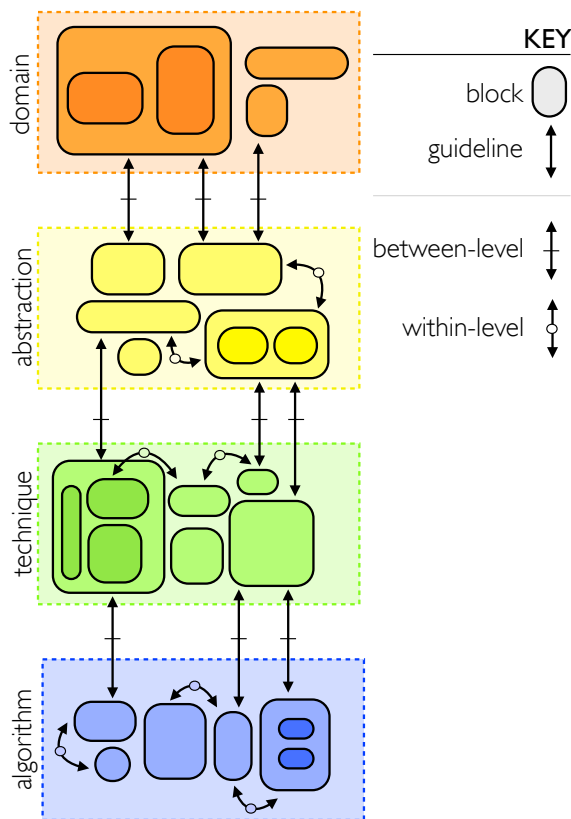

Figure 2: The NBGM has *blocks* that are the outcomes of the design process within a level, represented by individual shapes within each level, and *guidelines* for making choices between these blocks, represented by arrows. Between-level guidelines map blocks at adjacent levels, and within-level guidelines compare blocks within a level. Nested blocks indicate hierarchical relationships.

at one level become the assumptions at the next level. The arrows in Figure 1 represent these implications.

Although the four levels of this model are crucial for structuring how we reason about the design of a visualization systems, we find that considering finer grained structure within each level leads to even more fruitful analysis. We thus propose the concepts of blocks and guidelines as an extension to the original model, illustrated in Figure 2. In the rest of this section, we define blocks at each level of the model, give a definition for guidelines in terms of these blocks, and discuss how to use the NBGM.

2.1 Blocks

A **block** is the outcome of the design process at a specific level: an algorithm, a visual encoding and/or interaction technique, a data or task abstraction, or a specific domain situation. The term *block* allows us to refer to these different kinds of outcomes in a generic way at any level — Figure 2 shows these blocks as individual shapes at each level. Blocks exist at different granularities, leading to hierarchical relationships among blocks at the same level. For example, a treemap block at the technique level often includes the finer grained blocks of color and size encodings.

The increased specificity of finer grained blocks expands their generality. For example, the same color and size encoding blocks that compose the more complex treemap block can also be used in composing a bubble chart block. Fine-grain blocks often exhibit one-to-many relationships with their hierarchical superiors. Conversely, complex blocks that share a fine-grained block can be characterized as *overlapping* or *intersecting*.

We chose the term *blocks* as an allusion to the experience of playing with building blocks. When building something up, some blocks fit together nicely while others do not. When breaking something down, complex structures are decomposed into simpler pieces, with many different possible structures emerging from the same base set of individual blocks.

The visualization designer might decide to **use** an existing block, or **contribute** a new one. The exact meaning of *contribute* depends on the level in the NGBM. At the lower levels, blocks are **designed**: that is, they are created, produced, or invented by the designer. Designed blocks are the outcomes of the designer's creativity. At the upper levels, however, blocks are **identified**. Here, blocks are the outcome of the designer's understanding of the requirements of users in a specific situation. In the rest of this section we further refine the definition of blocks at each of the four levels of the NBGM.

2.1.1 Domain Level

Blocks at the domain level describe a specific **situation**, which encompasses a group of target users, their field of interest, their questions, and their measurements or data. One example of a situation block is a computational biologist working in the field of comparative genomics, using genomic sequence data to ask questions about the genetic source of adaptivity in a species [39]. Another example is members of the general public making medical decisions about their healthcare in the presence of uncertainty [41]. At this level, situation blocks are *identified* rather than designed because the outcome of the design process is an understanding that the designer reaches about the needs of the user. The methods typically used by designers to identify domain situation blocks are interviews, observations, or careful research about target users within a specific domain.

Many problem-driven visualization projects are aimed at target users working on a specific problem in a particular application domain, thus motivating the name of *domain* characterization for this level in the original nested model. Our term *situation*, however, emphasizes a broader and more nuanced scope, namely that a specific set of users whose questions about data arise from their work within particular field of study is just one kind of situation. This perspective aligns with the concept of holistic consideration from the design thinking literature [10], a topic we discuss in more detail in Section 4. Situations are a more general way to consider a group of people that is not directly tied to formal field of study.

As with all blocks, situation blocks exist at different granularities, organized in a hierarchical structure. Coarse-grained blocks at the top of a hierarchy might depict an entire problem domain such as bioinformatics or finance, within which might be subdomains such as sequence assembly or stock market analysis. Even these subdomains are usually insufficient for informing the visualization design process, necessitating the identification of more specific, fine-grained situations. For example, the MizBee design study [39] has a table of specific low-level questions asked by researchers in comparative genomics, such as "What are the differences between individual nucleotides of feature pairs?" and "What is the density of coverage and where are the gaps across a chromosome?". This situation block hierarchy imposes a natural set of conditions and constraints on the downstream design decisions. Thus, the greater the specificity of the situation block, the more useful it is for guiding the design and evaluation of blocks at lower levels of the NBGM.

2.1.2 Abstraction Level

The abstraction level consists of task blocks and data blocks. Examples are the tasks of finding outliers and trends [3]; the dataset types of tables, networks, and text; and the attribute types of categorical, ordered, and quantitative [44]. Task blocks are *identified* by the designer as being suitable for a particular domain situation block, just as the situation blocks themselves are identified at the level above.

Data blocks, however, are *designed*. Selecting a data block is a creative design step rather than simply an act of identification. The designer often chooses to transform the original data from the form in which it was identified in the upstream situation into something different. Sometimes, however, the designer may decide to use the data in exactly the way that it was measured in the domain situation — we argue that applying the identity transformation is (and should be) an active design decision.

The work of Nielsen et al. [46] in designing the ABySS-Explorer visualization tool for genome sequence assemblies is an instructive example of a data transformation design. In this design study, the authors initially identified a graph structure for the measured data that came directly from the domain situation, where unique sequence strands are represented as nodes and overlapping strands as edges — this structure comes from the algorithms used by the domain experts to create sequence assemblies. Observing these experts, however, revealed that they often swapped the nodes and edges when reasoning about genome sequences. The visualization designers therefore decided to create a new data abstraction graph that also swapped the nodes and edges, which they then visualized using a node-link diagram. The authors present arguments for why the transformed data block is a better design choice than simply using the original data format. Analyzing this work according to the NBGM establishes these two design decisions — the two graphs — as different data abstraction blocks.

The NBGM helps us draw a sharp distinction between identifying task abstractions and designing data abstractions. We can now differentiate between a failure of misinterpretation by the designer in mapping a domain situation to an abstract task, and a failure due to a poor design with respect to data abstraction choices. The interwoven nature of task identification and data abstraction design is further discussed in Section 5.2.1.

2.1.3 Technique Level

Technique blocks are *designed*, in that they are the outcomes of a decision by the designer about both visual encoding and interaction. While some technique blocks might solely focus on one or the other, many reflect an intrinsic combination of both visual encoding and interaction aspects. Thus, we do not distinguish visual encoding blocks from interaction blocks at this level in the way that we distinguish data and task blocks at the abstraction level above.

Consider, for instance, Wattenberg and Viégas' word tree [68] that combines a hierarchical tree representation of keywords-in-context

with the interaction mechanisms of searching, hovering and browsing through different key words. Complementary to word trees, other example technique level blocks for visualizing text are phrase nets [64] and wordles [65].

2.1.4 Algorithm Level

Algorithm blocks are also *designed*. While the main focus of technique blocks is on the *what* to draw on the screen, the focus of algorithm blocks is on the *how* to do so programmatically. Some examples of algorithm blocks for, say, direct volume rendering are: ray casting [36], splatting [69], and texture mapping [11]. Algorithm blocks are often intrinsically connected to technique blocks. A force-directed layout algorithm such as GEM [22], for instance, produces a specific node-link visual encoding. These inherent connections result in an inseparable stack of blocks, an idea we explore in more detail in Section 5.1.2.

2.2 Guidelines

A **guideline** is a statement about the relationships between blocks. Guidelines help designers make choices about which blocks are appropriate versus which blocks are a mismatch with their requirements. **Within-level guidelines** relate choices within a particular level, specifying things such as the fastest algorithm among several contenders within the algorithm level. **Between-level guidelines** designate choices about how to move between levels, such as selecting which visual encoding technique to use for a particular data and task abstraction.

The arrows in Figure 2 represent guidelines that connect individual blocks, in contrast to the arrows in Figure 1 that represent dependencies and go between entire levels. For visual clarity, the NBGM's depiction orders the levels vertically rather than explicitly nesting them.

2.2.1 Within-level Guidelines

Within-level guidelines are about choices made between blocks on one level, however they often come with inherent upstream dependencies. These upstream dependencies become clear when considering that the NBGM inherits the original nested model's emphasis on dependencies between levels. For example, at the technique level Tory et al. propose within-level guidelines that drawing simple points is a more effective visual encoding choice than drawing landscapes whose height encodes the density of the points [63]. In this paper there is a clear statement about the dependency of the within-level guideline on an upstream decision made at the abstraction level: this guideline applies to dimensionally reduced data.

In some cases, however, it is safe to state a within-level guideline as having little or no dependency on upstream blocks. An example of such an unrestricted within-level guideline at the algorithm level is to choose the newer Voronoi treemap algorithm of Nocaj and Brandes [48] over the original algorithm of Balzer and Deussen [6] because it is independent of display resolution and faster to compute. This particular guideline is unlikely to depend on dataset characteristics. In many cases, though, within-level guidelines *do* have upstream dependencies that are not made explicit. Analysis with the NBGM is one way to shed some light on these unacknowledged dependencies.

2.2.2 Between-level Guidelines

Movement from one level to another is guided by between-level guidelines. These guidelines map blocks at one level to those at
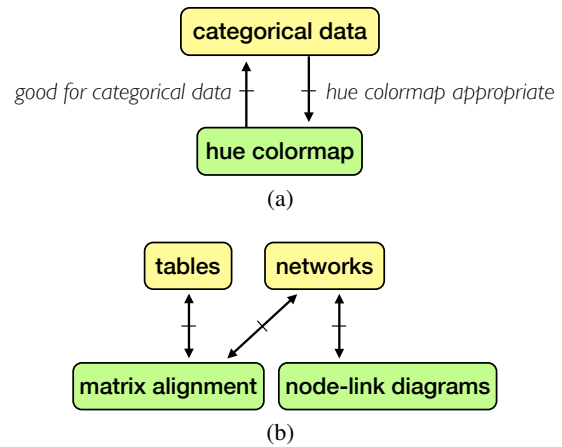

(a)

(b)

Figure 3: a) Example of a simple pairwise between-level guideline that can be inverted to go upwards or downwards. b) Guidelines can also be many-to-one or many-to-many; these more complex guidelines are not trivially invertible. The example shows a many-to-many between-level guideline.

an adjacent level. Figure 3(a) shows a well-established guideline between hue colormaps and categorical data [67]. In the literature, the term *characterization* is sometimes used to describe moving upward from a lower to a higher level, and the term *guideline* for moving downward from higher to lower — we consider these concepts to simply be two sides of the same coin. The simple upward characterization "hue-based colormaps are appropriate for categorical data" can be trivially restated as the downward guideline "if your data is categorical, then hue-based colormaps are appropriate". We propose guidelines as a more generic word to describe any stated relationship between blocks, and show guidelines in all subsequent figures with bidirectional arrows. Moreover, we note that the term guideline is used extensively in the visualization literature without a clear definition. Here we provide an explicit definition for guidelines in the context of visualization design decisions.

2.2.3 Guidelines and Complexity

Although Figures 2 and 3(a) illustrate one-to-one pairwise guidelines for visual simplicity, these guidelines can be many-to-one or many-to-many. Most of these more complex guidelines are not trivially invertible, such as the between-level guideline example shown in Figure 3(b): matrix alignment is a block at the technique level that is suitable for multiple blocks at the abstraction level, both tables and networks. The technique-level block of node-link diagrams, however, does not match up with the abstraction-level table block.

A particular visualization system can be decomposed into a stack of blocks, with one or a small set of blocks chosen at each of the different levels. Guidelines help a designer make these choices, and also help an evaluator analyze whether a designer's choices are reasonable. Figure 4 illustrates this idea. Figure 4(a) shows two simple visualization system designs, where different choices of blocks are made at each level. In keeping with the original nested model, the choice of blocks at a higher level constrains the suitable choices of blocks at the lower ones. Figure 4(b) shows a more representative real-world example, where the domain problem is decomposed into multiple abstract subtasks and thus the full system includes multiple stacks of blocks.
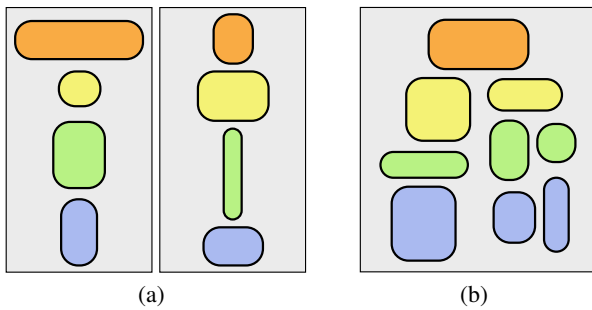
Figure 4: **Constructing visualization systems with blocks. a)**
A designer stacks one or more blocks at each level, making
choices based on guidelines; the two simple examples show that
the choices at higher levels constrain the choices beneath. b)
A real-world visualization system is likely to include multiple
stacks of blocks.

Figure 5: **The nine-stage framework for conducting design**
studies [57]. The eighth stage is to reflect on lessons learned
by confirming, refining, rejecting, or proposing guidelines. The
NBGM clarifies the scope of this stage.

2.3   Using the NBGM

We can use the NBGM in all three of the ways proposed by Beaudouin-
Lafon to assess the power of a model [7]: it has **descriptive** power
to describe a range of existing systems, **evaluative** power to assess
multiple design alternatives, and **generative** power to help design-
ers create new designs.

In the next section we use the NBGM to describe and evaluate pre-
viously proposed systems. That is, we identify blocks and describe
guidelines between the blocks as a form of post-hoc analysis, even
though the designers themselves did not explicitly consider their
work in these terms. We find that being forced to consider the de-
sign outcomes at each level leads to a much richer analysis. For
example, even though most paper authors do articulate new guide-
lines as contributions, they do not necessarily state the upstream
and downstream assumptions and dependencies of these guidelines.
Because the NBGM requires an analyst to state the outcomes at
each level as fine-grained blocks, the scope of guidelines are more
clearly understood.

The NBGM model is also intended to be a generative model to
help shape early design thinking. For example, an evaluator with
the goal of contributing new guidelines might first analyze existing
work in terms of current blocks, as describing blocks is a precondi-
tion for generating guidelines. In another example, a designer who
sets out to create a new block might explicitly consider in advance
how it should connect to existing blocks at levels above and below.
On the other hand, a design study researcher might first create a
visualization system for an identified problem without considering
this model at all, then analyze the resulting system using the model
descriptively to break the system down into blocks, and finally use
the model generatively to create guidelines about the use of these
blocks.

One of the motivations for developing the NBGM was to achieve
more clarity when considering our own recent work, the nine-stage
framework for conducting design studies [57]. The concerns of
the nine-stage framework cross-cut those of the NGBM. While the
NBGM (and the original nested model) focuses on the *outcomes*
of decisions made by the designer, the nine-stage framework in-
stead focuses on classifying the *actions* of a designer into separa-
ble stages. Figure 5 shows this framework, with the penultimate
reflect stage highlighted. We claim that at this stage a visualiza-
tion researcher should reflect on how lessons learned in a partic-
ular project relate to the larger research area of visualization by
confirming, refining, rejecting, or proposing new guidelines. Us-
ing the NBGM, we can now state that the reflection stage can and
should involve both between- and within-level guidelines. Through
the elucidation of guidelines, the NBGM provides explicit mecha-

2.2.4   Guidelines and Evaluation
One goal of visualization evaluation is to provide *actionability*,
which is guidance for a particular audience to take action based
on research results [26]. Guidelines are one such form of action-
ability for visualization designers, resulting from validation in both
technique-driven work and design studies.

Within-level guidelines often arise from validation efforts when
proposing a new block is the main research contribution, at either
the technique or algorithm design level. Between-level guidelines
are often the result of validation in design studies, where the em-
phasis is on justifying choices from the set of existing blocks by
showing that blocks match up properly with each other across lev-
els. Guidelines that map between levels are thus a central concern
in design studies, and most existing design study papers do indeed
emphasize them.

Both kinds of guidelines may arise from evaluation papers. For ex-
ample, Heer, Kong, and Agrawala provide within-level guidelines
on how to choose between horizon graphs and bar charts based
on the available screen space [29]. These guidelines are based on
reflections from extensive empirical evaluation of different visual
encoding techniques across a range of resolutions. In another eval-
uation paper, Heer and Bostock provide between-level guidelines
on how to choose visual encoding techniques appropriate for differ-
ent abstract data types [28], following in the footsteps of Cleveland
and McGill [16]. These between-level guidelines are also based on
empirical evaluation.

A paper might contribute a new block without explicitly contribut-
ing any guidelines about its use. The NBGM's emphasis on both
blocks and guidelines suggests that this choice is a lost opportu-
nity: even a preliminary set of guidelines establishing mappings to
blocks at the levels above and below the contributed block would
make the newly contributed block much more valuable.

In summary, guidelines encapsulate a relationship between two or
more blocks, where within-level guidelines compare blocks within
a single level and between-level guidelines map blocks at adjacent
levels. Without this extension to the model, we had difficulty in
reasoning about the role of guidelines within visualization design;
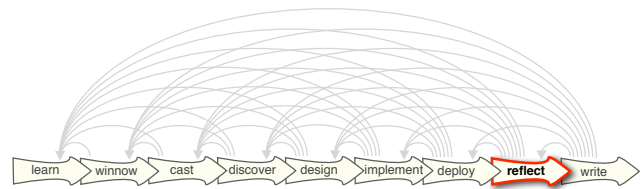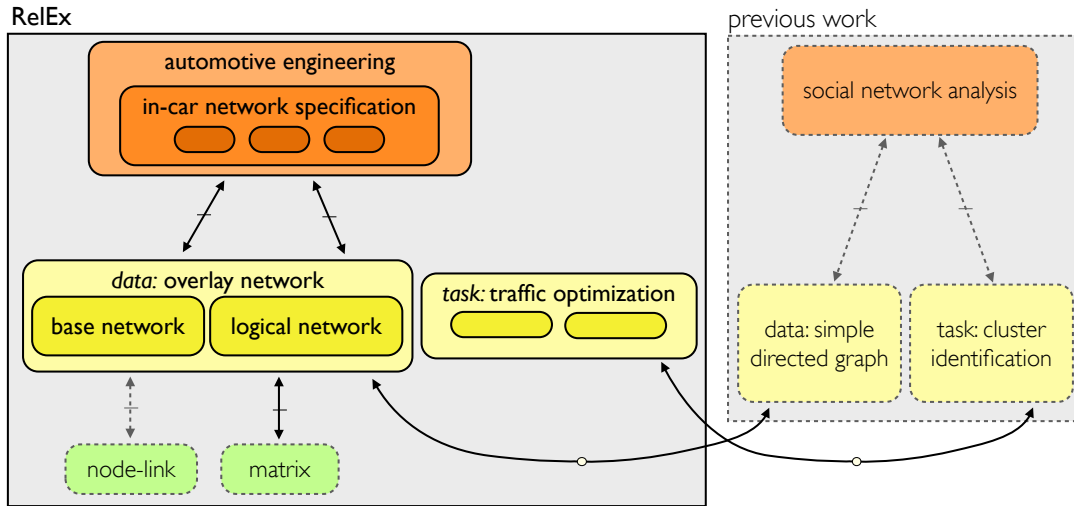these terms allow us to express ideas about guidelines crisply.

Figure 6: NBGM analysis of RelEx design study subset [54]. Continuous lines mark blocks and guidelines for which active contributions were made, and dashed outlines mean that existing blocks or guidelines were simply used. Nesting reflects hierarchical relations between blocks, and block stacks are enclosed in grey. Empty boxes indicate that further levels of detail are discussed in the paper but are omitted here for brevity.

nisms to more fully capture the rationale and knowledge associated with visualization design decisions in a way that was not supported before.

3.   ANALYSIS EXAMPLES

In this section we use the NBGM to analyze three papers from the IEEE InfoVis'12 conference proceedings. To give a broad view of the types of analysis the extension supports we analyze one design study paper, one technique paper, and one evaluation paper. We chose not to analyze examples of system or model papers because these paper types are typically not focused on the sorts of design decisions characterized in the nested model.

We use the NBGM to reason about the contributed blocks and guidelines presented in the papers. One central question we consider in this analysis is whether the researchers *used* existing blocks and guidelines, or actively *contribute* them as new ideas. We find the suite of four verbs used in the nine-stage framework [57] — *propose, refine, confirm, reject* — helpful for analyzing and discussing contributions in terms of the NBGM. In terms of blocks, a contribution is either proposing a new block or refining an existing one; the line of proposing and refining is admittedly fuzzy. Guidelines, on the other hand, characterize situations in which one block is better than another, or when a block appropriately maps to another one. The set of potential guideline contributions therefore not only includes propose and refine, but also confirm or reject existing guidelines. These analyses illustrate the formalisms afforded by the NBGM for capturing and elucidating visualization design decision rationale.

In the diagrams accompanying each paper analysis we use solid lines to indicate a contribution and dashed lines to indicate the usage of an existing block or guideline. A stack of blocks from different levels is indicated with a grey background, and hierarchical blocks are shown using containment.

3.1   Design Study: RelEx

Sedlmair et al. [54] present a design study focused on supporting automotive engineers tasked with specifying and optimizing traffic for in-car communication networks. The authors discuss the design and deployment of the RelEx visualization tool, with observations of its usage in the field providing anecdotal evidence of the speedup and simplification of engineer's daily practices. A major focus of their work is a characterization of the problem and questions at the domain level, the creation of appropriate data and task at the abstraction level, and a discussion of the differences between these abstractions and those previously proposed in other network visualization projects.

Figure 6 shows the result of our breakdown of the RelEx paper contributions and discussion in terms of blocks and guidelines according to the NBGM. Only a subset of the blocks discussed in the paper are covered in this analysis for brevity. This example shows that the stacks of blocks can be both broad and deep.

At the domain level there is one situation block for the domain of automotive engineering, and another for the subdomain of in-car network specification. The paper includes an in-depth analysis of this subdomain in terms current practices, challenges and the needs of experts; these further details are indicated by the additional fine-grain situation boxes in the diagram.

Building on the problem analysis, the paper provides a thorough justification and discussion of the abstraction choices for data and tasks. The abstract task of optimizing network traffic is indicated with a coarse-grain block, which is then refined into a more specific set of tasks. In terms of the underlying data, the authors characterize a course-grain block abstractly as an *overlay network*; that is, a *logical network* with a *base network* specified on top. The dense logical network is shown in the RelEx tool with a matrix representation, while the small and sparse base network is visually encoded as a node-link diagram. Both between-level decisions follow previously identified design guidelines by Ghoniem et al. [25]. The paper's contribution includes confirming the guidelines for matrix representations through a field study of the post-deployment use of the matrix display, shown with a solid line in Figure 6.
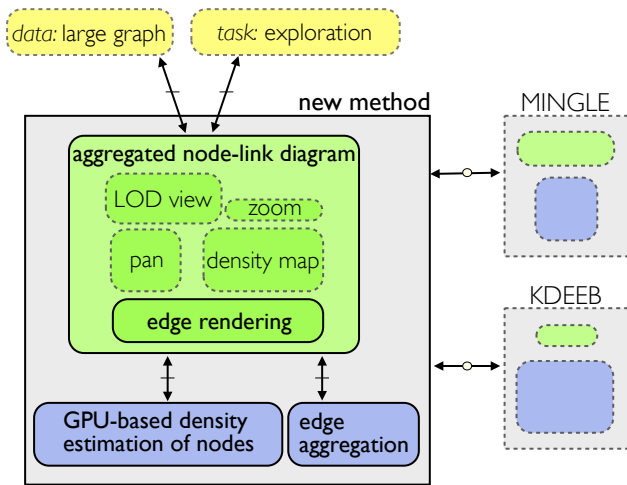
Figure 7: NBGM analysis of a new, LOD technique for rendering large graphs interactively on the GPU [71]. The proposed method is a stack of algorithm and technique blocks, where the technique block is a combination of both previously proposed and new blocks. The new method is compared to several other systems, establishing new within-level guidelines.

The paper includes a discussion of the relationship between the domains and abstractions proposed by the authors and those used in other visualization work. On the right of Figure 6 are three blocks discussed extensively in previous work. While it is not meaningful to directly compare the situation of automotive engineering on the left to that of social network analysis on the right, there are new within-level contributions that compare the proposed abstraction blocks on the left to those already existing at the abstraction level on the right.

3.2 Technique: Rendering Large Graphs

Zinsmaier et al. [71] present a technique paper proposing an image-space method for rendering large graphs interactively on the GPU. The method creates a summarized view of the graph by deriving a density estimation of the nodes that results in clusters, and aggregating edges by maintaining inter-cluster edges only. Exploration of the graph is supported with pan and zoom interaction, where the amount of summarization of the graph is determined by the current view using a level-of-detail (LOD) approach.

Figure 7 shows our breakdown of this work into blocks and guidelines according to the NBGM. This paper is also an example where the contribution includes the proposal of new blocks, but at the technique and algorithm levels rather than the domain and abstraction levels.

At the bottom of Figure 7 are two algorithm blocks. One algorithm computes a density estimation of the nodes using an image-based GPU approach. The second algorithm is a straightforward method for aggregating edges using results from the density estimation of the nodes. These algorithm blocks map to a complex, hierarchical technique block that renders the graph as a form of a node-link diagram that makes use of the density estimation of the nodes and aggregated edges. This technique block includes previously proposed visual encoding and interaction sub-blocks for: LOD viewing; the interaction mechanisms of pan and zoom; and a visual encoding of the density estimation of the nodes using transparency. The block
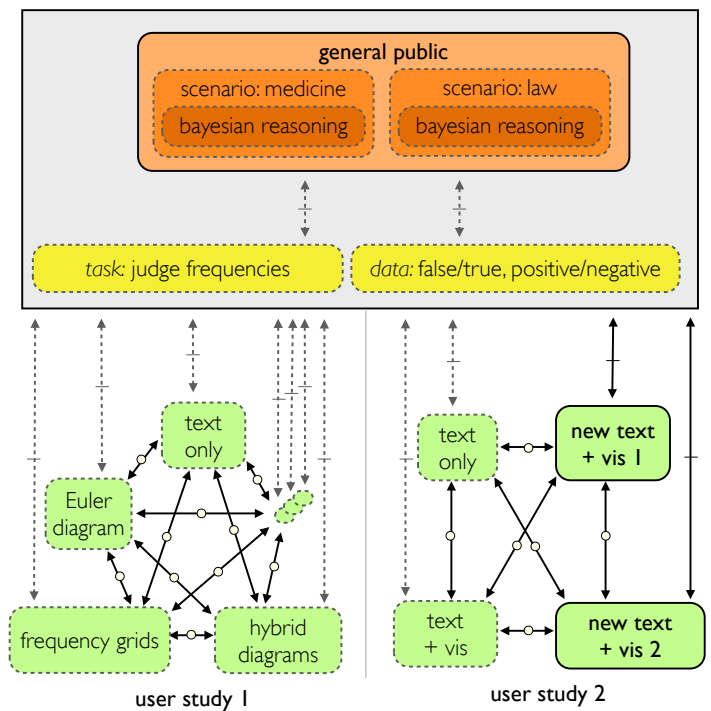

Figure 8: NBGM analysis of a paper evaluating visualization for Bayesian reasoning [41]. In a first study, the authors evaluate seven previously proposed encoding technique blocks, four of which are explicitly shown in the diagram. In a second study, they refine two previously proposed blocks and compared them to two representative blocks from the first study.

also contains a new technique for careful rendering of transparent, overlapping edges. At the top of the diagram are two abstraction blocks that are described as upstream assumption in the paper. The data abstraction block consists of large graphs, defined to be those that "fit into video memory but cannot be rendered as a node link diagram without significant over-plotting". The task abstraction block is stated as the exploration of large graphs.

The proposed method is validated against several others that also render large graphs, establishing within-level guidelines. These guidelines compare the new stack of technique and algorithm blocks to those stacks in the MINGLE [24] and KDEEB [31] systems. All three of these methods have tightly woven technique and algorithm blocks, so comparison can only occur between entire stacks of blocks that encompass both levels, rather than individual blocks at each level.

3.3 Evaluation: Effects on Reasoning

In their evaluation paper, Micallef et al. [41] present the results of two user studies on the effect of visualization on Bayesian reasoning, carried out through crowdsourcing. The goal of this work is to overcome the generalizability limitations of previous studies on this topic.

Figure 8 shows our analysis of this paper in terms of the NBGM. This work is a typical example of an evaluation paper in which the goal is to compare different visual encoding technique blocks. For evaluation papers, it is crucial to state the problem characterization and abstraction explicitly. The problem addressed by Micallef et

al. is Bayesian reasoning, which has been well characterized in previous work as the task of judging probabilities, using probabilistic data that could be true/false or positive/negative. These abstraction blocks are shown as yellow blocks with a dashed outline, indicating that they are used rather than contributed. The authors describe two domains where this abstraction commonly occurs — medicine and law — to ground the abstraction in real-world problems. These situations are shown as orange domain blocks. The authors further characterize the problem, and thus their major goal, as a problem that is faced by the general public. Previous studies do not account for this target user group, thus this paper identifies the general public as a new situation block for this context.

The major contributions of the paper, however, are the results of the two studies designed for this situation which reject an existing guideline and lead to the proposal of a new one. The authors conduct a first study that compares seven visual encoding blocks to each other. These blocks include Euler diagram representations, frequency grids, hybrid approaches, and purely textual problem descriptions, shown in the diagram as green blocks. Although within-level guidelines already existed for these blocks, the authors wanted to confirm, refine or reject these previously proposed guidelines for the general public. The authors could not replicate previously reported effects despite careful experimental design, and so they rejected the previous guidelines, going on to propose new ones based on their findings.

The findings of the first study also prompted the authors to design two technique blocks that combine textual descriptions and visualizations in novel ways by refining existing methods that do so. In a second study, they compare these two new blocks against a pure textual description and a standard combination of text and visualization. They were able to characterize situations in which one of the new blocks is superior to the others, leading to the proposal of new within-level guidelines.

4. TYPES OF DESIGN MODELS

By surveying design-related literature in software engineering, cognitive science, visualization, and design itself, we identified three common types of design models that are closely related to our goals: *process models*, *architecture models*, and *design decision models*. The NBGM is an example of this last category. These models capture knowledge about different aspects of design. In this section we discuss and characterize each model type based on their intent in order to clarify the role of the NBGM within visualization design. Although we also investigated a thread of previous work on "research through design" that falls at the intersection of the HCI and design literatures as a potential source of ideas [21, 34, 70], we determined that it did not directly address the concrete questions of design models that is our focus in this paper.

Process models capture the *how* of design; they classify the actions of the designer according to a set of stages, offering practical guidance on the order and the transition criteria for progressing through those stages. Visualization-related process models like multi-dimensional in-depth long-term case studies [58] and the nine-stage framework for conducting design studies [57] strive to minimize the time and energy of visualization researchers by providing practical guidance for designing and evaluating visualization systems. In software engineering, many more process models exist to help developers avoid costly late-stage, large-scale development changes [8, 13, 52]. Neither the NBGM nor the original nested model are process models because they do not characterize the *how*

of visualization design. Instead, the NBGM strives to make explicit the design decisions and assumptions that process models tend to omit.

Another design model found in the visualization literature is the **architecture model**. Sometimes referred to *reference models* [12], these models provide context-independent solutions to core classes of design problems encountered during the implementation of a software system. High-level examples of architecture models include the various iterations of the visualization pipeline [12, 42]. Examples of lower level models are found in the literature on *design patterns*. A design pattern describes a demonstrated software solution to a common design problem that is customizable to many different contexts [2, 23]. Although design patterns are heavily used in the fields of computer architecture and software engineering, Heer and Agrawala [27] provide design patterns specifically for visualization. The NBGM does not fit into this class of design models as it does not address implementation issues.

Instead, we classify the NBGM as a **decision model**. Stemming from research on *design rationale systems* [50], decision models aim to capture the rationale behind design decisions [32]. These models support the design process by relating the current design decision to the larger design space, providing the context for enabling more informed decisions by making design rationale transparent to others. We also classify the original nested model as a decision model, but it is a coarse-grained one. In the NBGM the context surrounding any given visualization design decision is modeled by the set of blocks and guidelines comprising upstream assumptions and downstream implications, as well as by the known alternative design paths. In this way, the NBGM represents a knowledge-based framework for reasoning about and discussing individual design decisions. Blocks and guidelines provide an explicit mechanism to capture and reason about context and design rationale that the original nested model does not support.

We can describe the NBGM more abstractly as a model that contains nodes and links. In this framing, we encountered other models built on a similar concept, however none deal with the contextualization of design decisions. Electronic-Tool-Integration platforms [38], for example, abstractly model data transformation routines from different tools as nodes, while using edges to denote datatype compatibility for inputs and outputs. The low level tool-coordination tasks that such systems support are primarily concerned with implementation rather than design rationale. Similarly, the process models we encountered [8, 13, 52, 57, 58] are often modeled as a set of nodes representing distinct development stages and links representing recommended progressions between those stages.

It is tempting to view the NBGM as a cookbook for visualization design, where practitioners trace paths through the set of known blocks and guidelines. This viewpoint, however, breaks down because the field of design is never static; rather, it is continually innovating new solutions through the consideration of problems in the larger context of people, systems, materials, and environment [10]. The continual expansion of solutions stems from the open-ended nature of both the wicked [10, 51] and ill-structured [59] problems tackled by designers. Furthermore, these problems imply a lack of optimality in the design space for visualizations, that the search for a solution always involves *satisficing* [53, 60]. Different underlying philosophies create subtle distinctions that can add significant complexity to the concept of a satisficing condition [19, 53, 60],

but the core idea is always the same: there is no *best* solution, only verification of whether a solution is *sufficient* or *good enough*. We do not claim that the notion of visualization design entailing satisficing is new: for example, the same argument appears in previous work on design study methodology [57]. The contribution of the NBGM is that it assists the visualization designer in making satisficing decisions through the formalization of context.

5. DISCUSSION

The combination of the NBGM and our analysis examples have led us to identify implications for visualization evaluation. The NBGM also supports an explicit analysis of several visualization research challenges. Our discussion of these issues is a secondary contribution of this work.

5.1 Implications for Visualization Evaluation

Analyzing papers with the NGBM in Section 3 leads to a set of implications for visualization evaluation. We believe that these implications are important for guiding evaluation endeavors and that they are difficult to express without the NBGM.

Following the original nested model [43], we continue to use the term *evaluation* in a very broad sense. We consider evaluation that is: both quantitative and qualitative; both formative and summative; and done at every level of the model, from domain characterization through algorithm comparison.

5.1.1 Comparing Domains via Abstractions

Analyzing the similarities and differences between visualization systems targeted to different domains is a key step to providing generalizable solutions. Figure 2, however, does not include arrows between blocks at the domain level. By definition, domain situation blocks tackle different fields, different problems, and different data. Attempting to directly compare between them is not helpful as the only useful statement to make is that they differ.

The original nested model articulates the idea that the abstraction level is the correct place to do such comparisons. The NBGM emphasizes this point further by describing mappings between domain situation blocks and data and task abstraction blocks. When the same abstraction block maps to multiple situation blocks then a common solution is feasible, but when two situations map to different abstractions the NGBM provides a useful way to distinguish between them. A key feature of the NBGM is a formalism to describe visualization design rationale.

The analysis of the RelEx design study in Section 3.1 is an example of this principle. It is relatively clear that the situation of engineers developing in-car networks is different than sociologists studying how children interact with each other in an elementary-school classroom. But the situation of film critics analyzing how acting styles spread across genres also differs from that of the sociologists, yet previous work argues that the same abstraction-level blocks are suitable for both of the latter examples, and many other situations as well. As illustrated in the RelEx analysis in Section 3.1 a useful way to confirm domain similarities and differences is at the abstraction level, when the authors identified and designed new data and task blocks that were indeed different than those used before.

We stress the importance of rich data and task abstractions for creating effective and generalizable visualization systems. The goal of establishing a complete set of task and data abstraction blocks to which the much larger set of possible domain situations could be mapped is the long-term grand challenge of applied visualization research. Although a succession of calls for work along these lines has been made by advocates of problem-driven work for well over a decade [37, 57], the NBGM allows a more precise and thus more actionable statement of how to make progress towards this goal.

5.1.2 Comparing Stacks of Blocks

The goal of many evaluations found in visualization research papers is the establishment of within-level guidelines: that is, to illustrate that a new or previously proposed block is superior to others at that level. These within-level guidelines are important for characterizing how and when a visualization designer should choose one block over another at a specific level of design. Within-level guidelines focusing on blocks at a particular level, however, most often intrinsically rely on both upstream assumptions made at the levels above and on downstream decisions made at the levels below. Thus, within-level guidelines must be understood as existing within the context of a stack of blocks, not in a vacuum consisting only of the blocks in a particular level.

To help others understand when certain within-guidelines hold, and when they do not, it is imperative to be clear about upstream assumptions as they can critically influence the results of a study. The Bayesian reasoning study analyzed in Section 3.3 rejects previous guidelines that are not specific about upstream assumptions by showing that they do not hold in a stack with a situation block for the general public. The running time of an algorithm might be heavily influenced by the choices made for data blocks at the abstraction level; changing the scale or type of datasets tested might even reverse results. The NBGM's emphasis that blocks occur within the context of a specific stack helps emphasize the importance of the upstream assumptions and downstream decisions.

Establishing within-level guidelines also has an inherent dependency on downstream blocks, at least in some cases. For example, comparing two different data abstractions for a specific problem necessitates choosing a visual encoding in order to evaluate the effectiveness of the abstractions. Similarly, evaluating an interaction technique block often requires one or more algorithm blocks to sufficiently test it. The original nested model emphasizes these dependencies in its discussion of evaluation methods.

Thus, although in some cases blocks may be compared within one level, in other cases choices across levels are so interwoven that a more reasonable choice is to compare between stacks of blocks. These comparisons still give rise to the equivalent of within-level guidelines, but for a stack that spans multiple levels. The LOD graph rendering technique analyzed in Section 3.2 is an example where the the visual encoding techniques are inherently bound to their algorithmic realizations. The authors therefore compare stacks of technique and algorithm blocks against each other. Likewise, because of the downstream dependencies mentioned earlier, comparing abstractions always necessitates comparing stacks of blocks.

5.2 Knowledge Gaps

Extending the nested model with the concepts of blocks and guidelines clarifies why certain parts of the design process are particularly difficult by affording an explicit description of gaps in our visualization knowledge. These gaps present rich opportunities for future work.

5.2.1 Untangling the Abstraction Level

Considering the four levels in terms of their constituent blocks uncovers a dearth of identified blocks at the abstraction level. We have many blocks for visual encodings and interaction techniques, and perhaps even more blocks at the algorithm level, but we have far fewer established blocks for data and task abstractions. Without blocks we cannot have guidelines; blocks at both levels are a precondition for guidelines that map between them. We believe that this lack of both blocks and between-level guidelines at least partially accounts for the difficulty of going from a domain characterization to an abstraction when conducting a design study.

In particular, our knowledge of tasks is deeply incomplete. The well-cited work of Amar, Eagan, and Stasko [3] describes a taxonomy of low-level tasks such as *retrieve value*, *filter*, and *sort*, and was intended to provide a foundation for the development of richer task taxonomies at higher levels. Amar and Stasko [4] also presented a taxonomy of very high-level tasks such as *expose uncertainty*, *formulate cause/effect*, and *confirm hypothesis*. However, we lack task blocks at the middle level. There is a pressing need to propose and validate mid-level task taxonomies that bridge the gap between *finding the minimum value of an attribute* and *confirming or refuting a hypothesis*. The very recent work from Brehmer et al. is a first step [9].

The abstraction level itself might also benefit from more study. The NBGM already introduces the distinction between designed data and identified task abstraction blocks, as discussed in Section 2.1.2. In our own experience the process of designing data and identifying task blocks is interwoven because of a cycle where progress with one leads to further progress with the other. Identified input data from a domain situation typically allows the designer to identify an initial task abstraction, which informs the design decision of how to transform the data — but using the new derived data often allows the task abstraction to be further refined, and the cycle continues. This interplay between task identification and data derivation may illustrate the need for further refinement of the NBGM at the abstraction level, to better support analysis of how data is transformed and derived.

5.2.2 From Algorithms To Techniques

Establishing guidelines from algorithms up to techniques is sometimes straightforward, but other times remarkably subtle. In many cases the literature does not concisely describe the result of a specific algorithm in terms of a visual encoding technique. For example, different algorithms in the general family of force-directed network layout can give rise to quite different visual encodings in terms of the spatial proximity of classes of nodes to each other. Very few authors explicitly discuss these characteristics; Noack is a notable exception [47]. Fully characterizing the mappings up from specific algorithms to the visual encodings that they produce remains an important open problem. Until this goal is attained, problem-driven visualization designers will have a difficult time in making wise decisions about which algorithms to use.

5.3 Limitations and Future Work

The NBGM is a companion to visualization process models, such as the nine-stage framework for conducting design studies. These models provide much needed guidance and framing for reasoning about, and capturing, knowledge about visualization design. As the visualization community embraces problem-driven research, there is a need for more formalization of what this means in terms of process, decisions, knowledge, and contributions. The NBGM is a

step in this direction, but with limitations: the concepts of *blocks* and *guidelines* as described here may not encompass all the complexities inherent in visualization design; further taxonomies at all levels of the NBGM may point to new or refined levels of design decisions; and further validation of the NBGM through field work could refine aspects of the NBGM or lead to the development of new design decision models. While we have already found the concepts in the NBGM to be useful mechanisms in our own research, we see these potential limitations as avenues for rich, future research about visualization design.

6. CONCLUSIONS

We present an extension of the four-level nested model, the nested blocks and guidelines model. The NBGM identifies blocks within each level that are specific outcomes of the design process, allowing us to define guidelines expressing relationships between blocks either within a single level or between two adjacent levels. In this paper we use the NBGM as a framework for analyzing visualization research contributions, leading us to several implications for visualization evaluation, as well providing a new framing for open gaps in knowledge.

The NBGM supports a more detailed description of each level of design found in the original nested model, especially the domain and abstraction levels. At the domain level we identify situation blocks that encompass a field of study, target users, and their questions and measurements. Situation blocks are significantly different from blocks on the other levels, as they are identified rather than designed. The abstraction level is a hybrid of both identified task blocks and designed data blocks. We also differentiate between blocks that are contributed vs. blocks that are simply used. We see our clarifications of the domain and abstraction levels as a first step towards a better understanding of the entire visualization design problem as outlined by the original nested model.

The NBGM also allows us to derive two implications for visualization evaluation. First, abstraction blocks are an imperative device for comparing different domain problems. Comparing domains via the abstraction level leads to interesting and actionable insights for visualization researchers and leads to a broader, more general understanding of the problems data analysts face. Second, evaluators need to carefully characterize upstream assumptions, as well as necessary downstream decisions, when creating within-level guidelines — changing these stacks of blocks can have drastic results on guideline studies. Being aware of these assumptions and requirements helps in carefully accounting for these effects as a confounding variable in a study.

Finally, we use the NBGM to analyze two gaps: the need for further work and clarification at the abstraction level, and the need for a more complete set of guidelines between techniques and algorithms. We are not the first to identify these gaps, but this NBGM-based analysis allows us to more clearly articulate what avenues of future work might most benefit our research community.

7. ACKNOWLEDGMENTS

8. REFERENCES

[1] Y. ah Kang and J. Stasko. Characterizing the intelligence analysis process: Informing visual analytics design through a longitudinal field study. In *IEEE Conf. Visual Analytics Science and Technology (VAST)*, pages 21–30, 2011.

[2] C. Alexander, S. Ishikawa, and M. Silverstein. *A Pattern Language: Towns, Buildings, Constructions*. Center for Environmental Structure Berkeley, Calif: Center for Environmental Structure series. Oxford University Press, 1977.

[3] 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.

[4] 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.

[5] D. Auber, D. Archambault, R. B. A. Lambert, M. Mathiaut, P. Mary, M. Delest, J. Dubois, and G. Melançon. The tulip 3 framework: A scalable software library for information visualization applications based on relational data. Technical report, INRIA Research Report 7860, 2012.

[6] M. Balzer and O. Deussen. Voronoi treemaps. In *Proc. IEEE Symposium on Information Visualization (InfoVis)*, pages 49–56, 2005.

[7] M. Beaudouin-Lafon. Designing interaction, not interfaces. In *Proc. ACM Conf. Advanced Visual Interfaces (AVI)*, 2004.

[8] B. W. Boehm. A spiral model of software development and enhancement. *Computer*, 21(5):61–72, 1988.

[9] M. Brehmer and T. Munzner. A multi-level typology of abstract visualization tasks. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 2013. in press.

[10] R. Buchanan. Wicked problems in design thinking. *Design Issues*, 8(2):pp. 5–21, 1992.

[11] B. Cabral, N. Cam, and J. Foran. Accelerated volume rendering and tomographic reconstruction using texture mapping hardware. In *Proc. Symp. Volume Visualization (VolVis)*, pages 91–98, 1994.

[12] S. K. Card, J. D. Mackinlay, and B. Shneiderman, editors. *Readings in information visualization: using vision to think*. Morgan Kaufmann Publishers Inc., 1999.

[13] E. Céret, S. Dupuy-Chessa, G. Calvary, A. Front, and D. Rieu. A taxonomy of design methods process models. *Information and Software Technology*, 55(5):795 – 821, 2013.

[14] C.-Y. Chiu and A. D. Russell. Design of a construction management data visualization environment: A top-down approach. *Automation in Construction*, 20(4):399–417, 2011.

[15] J. Chuang, D. Ramage, C. D. Manning, and J. Heer. Interpretation and Trust: Designing Model-Driven Visualizations for Text Analysis. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 443–452, 2012.

[16] W. S. Cleveland and R. McGill. Graphical perception: Theory, experimentation, and application to the development of graphical methods. *Journal of the American Statistical Association (JASA)*, 79(387):531–554, 1984.

[17] S. Conversy, S. Chatty, and C. Hurter. Visual scanning as a reference framework for interactive representation design. *Information Visualization*, 10(3):196–211, 2011.

[18] I. Dillingham, B. Mills, and J. Dykes. Exploring road incident data with heat maps. In *Conf. Geographic Information Science Research UK (GISRUK)*, 2011.

[19] D. Fallman. Design-oriented human-computer interaction. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 225–232. ACM, 2003.

[20] A. G. Forbes, T. Höllerer, and G. Legrady. behaviorism: A framework for dynamic data visualization. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 16(6):1164–1171, 2010.

[21] C. Frayling. *Research in art and design*. Royal College of Art London, 1993.

[22] A. Frick, A. Ludwig, and H. Mehldau. A fast adaptive layout algorithm for undirected graphs. In *Proc. Graph Drawing (GD)*, volume 894 of *LNCS*, pages 388–403. Springer, 1995.

[23] E. Gamma, R. Helm, R. Johnson, and J. Vlissides. *Design patterns: elements of reusable object-oriented software*. Addison-Wesley Longman Publishing Co., Inc., 1994.

[24] E. Gansner, Y. Hu, S. North, and C. Scheidegger. Multilevel agglomerative edge bundling for visualizing large graphs. In *Pacific Visualization Symposium (PacificVis), 2011 IEEE*, pages 187 –194, 2011.

[25] M. Ghoniem, J.-D. Fekete, and P. Castagliola. On the readability of graphs using node-link and matrix-based representations: a controlled experiment an d statistical analysis. *Information Visualization*, 4(2):114–135, 2005.

[26] M. Gleicher. Position paper: Why ask why? considering motivation in visualization evaluation. In *Proc. VisWeek Wkshp. BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM, 2012.

[27] J. Heer and M. Agrawala. Software design patterns for information visualization. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 12(5):853–860, 2006.

[28] J. Heer and M. Bostock. Crowdsourcing graphical perception: using mechanical turk to assess visualization design. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 203–212, 2010.

[29] 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 Conf. Human Factors in Computing Systems (CHI)*, pages 1303–1312, 2009.

[30] J. Hullman and N. Diakopoulos. Visualization rhetoric: Framing effects in narrative visualization. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 17(12):2231–2240, 2011.

[31] C. Hurter, O. Ersoy, and A. Telea. Graph bundling by kernel density estimation (proc. eurovis). *Computer Graphics Forum*, 31(3pt1):865–874, 2012.

[32] A. Jansen and J. Bosch. Software architecture as a set of architectural design decisions. In *Working IEEE/IFIP Conf. Software Architecture (WICSA)*, pages 109–120, 2005.

[33] P. Kok, M. Baiker, E. A. Hendriks, F. H. Post, J. Dijkstra, C. W. Löwik, B. P. Lelieveldt, and C. P. Botha. Articulated planar reformation for change visualization in small animal imaging. *IEEE Trans. Visualization and Computer Graphics (Proc. Vis)*, 16(6):1396–1404, 2010.

[34] I. Koskinen, T. Binder, and J. Redström. Lab, field, gallery, and beyond. *Artifact*, 2(1):46–57, 2008.

[35] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information

visualization: Seven scenarios. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 18(9):1520–1536, 2012.

[36] M. Levoy. Display of surfaces from volume data. *IEEE Comput. Graph. Appl.*, 8(3):29–37, 1988.

[37] J. D. Mackinlay. Presentation, visualization, what's next? InfoVis 2000 Keynote Address, 2000.

[38] T. Margaria. Web services-based tool-integration in the eti platform. *Software and Systems Modeling*, 4(2):141–156, 2005.

[39] M. Meyer, T. Munzner, and H. Pfister. MizBee: A multiscale synteny browser. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 15(6):897–904, 2009.

[40] M. Meyer, M. Sedlmair, and T. Munzner. The Four-Level Nested Model Revisited: Blocks and Guidelines. In *Proc. VisWeek Wkshp. on BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*. ACM Press, 2012.

[41] L. Micallef, P. Dragicevic, and J. Fekete. Assessing the effect of visualizations on bayesian reasoning through crowdsourcing. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 18(12):2536 –2545, 2012.

[42] K. Moreland. A survey of visualization pipelines. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 19(3):367–378, 2013.

[43] T. Munzner. A Nested Model for Visualization Design and Validation. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 15(6):921–928, 2009.

[44] T. Munzner. Visualization. In *Fundamentals of Graphics*, pages 675–707. AK Peters, 3rd edition, 2009.

[45] T. Munzner, A. Barsky, and M. Williams. Reflections on QuestVis: A visualization system for an environmental sustainability model. In *Scientific Visualization: Interactions, Features, Metaphors*, volume 2, pages 240–259. Dagstuhl Follow-Ups, 2011.

[46] C. B. Nielsen, S. D. Jackman, I. Birol, and S. J. Jones. ABySS-Explorer: Visualizing genome sequence assemblies. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 15(6):881–8, 2009.

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

[48] A. Nocaj and U. Brandes. Computing voronoi treemaps: Faster, simpler, and resolution-independent. *Computer Graphics Forum*, 31(3pt1):855–864, 2012.

[49] H. Piringer, W. Berger, and J. Krasser. HyperMoVal: interactive visual validation of regression models for real-time simulation. *Computer Graphics Forum (Proc. EuroVis)*, 29(3):983–992, 2010.

[50] W. C. Regli, X. Hu, M. Atwood, and W. Sun. A survey of design rationale systems: Approaches, representation, capture and retrieval. *Engineering with Computers*, 16(3-4):209–235, 2000.

[51] H. W. Rittel and M. M. Webber. Dilemmas in a general theory of planning. *Policy Sciences*, 4(2):155–169, 1973.

[52] W. W. Royce. Managing the development of large software systems: concepts and techniques. In *Proc. International Conference on Software Engineering (ICSE)*, pages 328 – 338. IEEE Computer Society Press, 1987.

[53] D. A. Schön. *The Reflective Practitioner: How Professionals Think in Action*. Basic Books, 1983.

[54] M. Sedlmair, A. Frank, T. Munzner, and A. Butz. Relex: Visualization for actively changing overlay network specifications. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 18(12):2729 –2738, 2012.

[55] M. Sedlmair, P. Isenberg, D. Baur, and A. Butz. Information Visualization Evaluation in Large Companies: Challenges, Experiences and Recommendations. *Information Visualization*, 10(3):248–266, 2011.

[56] M. Sedlmair, P. Isenberg, D. Baur, M. Mauerer, C. Pigorsch, and A. Butz. Cardiogram: Visual Analytics for Automotive Engineers. In *Proc. ACM Conf. Human Factors in Computing Systems (CHI)*, pages 1727–1736, 2011.

[57] M. Sedlmair, M. Meyer, and T. Munzner. Design Study Methodology: Reflections from the Trenches and the Stacks. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 18(12):2431–2440, 2012.

[58] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proc. AVI Wkshp. BEyond time and errors: novel evaLuation methods for Information Visualization (BELIV)*, pages 1–7. ACM, 2006.

[59] H. A. Simon. The structure of ill structured problems. *Artificial Intelligence*, 4(3–4):181 – 201, 1973.

[60] H. A. Simon. *The Sciences of the Artificial*. MIT Press, third edition, 1996.

[61] D. Sprague and M. Tory. Exploring how and why people use visualizations in casual contexts: Modeling user goals and regulated motivations. *Information Visualization*, 11(2):106–123, 2012.

[62] A. S. Spritzer and C. M. D. S. Freitas. Design and evaluation of magnetviz – a graph visualization tool. *IEEE Trans. Visualization and Computer Graphics (TVCG)*, 18(5):822–835, 2012.

[63] M. Tory, D. W. Sprague, F. Wu, W. Y. So, and T. Munzner. Spatialization design: Comparing points and landscapes. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 13(6):1262–1269, 2007.

[64] F. van Ham, M. Wattenberg, and F. B. Viegas. Mapping text with phrase nets. *IEEE Trans. on Visualization and Computer Graphics (Proc. InfoVis)*, 15:1169–1176, 2009.

[65] F. B. Viegas, M. Wattenberg, and J. Feinberg. Participatory visualization with wordle. *IEEE Trans. on Visualization and Computer Graphics (Proc. InfoVis)*, 15(6):1137–1144, 2009.

[66] X. Wang, W. Dou, T. Butkiewicz, E. A. Bier, and W. Ribarsky. A Two-stage Framework for Designing Visual Analytics System in Organizational Environments. In *IEEE Conf. Visual Analytics Science and Technology (VAST)*, pages 249–258, 2011.

[67] C. Ware. *Information Visualization: Perception for Design*. Morgan Kaufmann/Academic Press, 2nd edition, 2004.

[68] M. Wattenberg and F. B. Viegas. The word tree, an interactive visual concordance. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 14(6):1221–1228, 2008.

[69] L. A. Westover. Splatting: A parallel, feed-forward volume rendering algorithm. Technical report, University of North Carolina at Chapel Hill, 1991.

[70] J. Zimmerman, E. Stolterman, and J. Forlizzi. An analysis and critique of research through design: towards a formalization of a research approach. In *Proc. ACM Conf.*

Designing Interactive Systems (DIS), DIS '10, pages 310–319. ACM, 2010.

[71] M. Zinsmaier, U. Brandes, O. Deussen, and H. Strobelt. Interactive level-of-detail rendering of large graphs. *IEEE Trans. Visualization and Computer Graphics (Proc. InfoVis)*, 18(12):2486 –2495, 2012.