The Value of Visualization

Jarke J. van Wijk*
Dept. Mathematics and Computer Science
Technische Universiteit Eindhoven

ABSTRACT

The field of Visualization is getting mature. Many problems have been solved, and new directions are sought for. In order to make good choices, an understanding of the purpose and meaning of visualization is needed. Especially, it would be nice if we could assess what a good visualization is. In this paper an attempt is made to determine the value of visualization. A technological viewpoint is adopted, where the value of visualization is measured based on effectiveness and efficiency. An economic model of visualization is presented, and benefits and costs are established. Next, consequences for and limitations of visualization are discussed (including the use of alternative methods, high initial costs, subjectiveness, and the role of interaction), as well as examples of the use of the model for the judgement of existing classes of methods and understanding why they are or are not used in practice. Furthermore, two alternative views on visualization are presented and discussed: viewing visualization as an art or as a scientific discipline. Implications and future directions are identified.

CR Categories: H.5.2 [Information Interfaces and Presentation]: User Interfaces; I.3.6 [Computer Graphics]: Methodology and Techniques I.3.8 [Computer Graphics]: Applications

Keywords: Visualization, evaluation

1 INTRODUCTION

Modern society is confronted with a data explosion. Acquisition devices like MRI-scanners, large scale simulations on supercomputers, but also stock trading at stock exchanges produce very large amounts of data. Visualization of data makes it possible for researchers, analysts, engineers, and the lay audience to obtain insight in these data in an efficient and effective way, thanks to the unique capabilities of the human visual system, which enables us to detect interesting features and patterns in short time.

Many of us will have written paragraphs like the preceding one, where I attempted to give the standard rationale of our field. In 1987, when the influential ViSC report [16] of the NSF appeared, the expectations were high. Visualization was considered as vital and highly promising for the scientific process. Nowadays, much progress has been made. The advances in graphics hardware are astonishing, most laptop computers are graphics superworkstations according to the standards of just a decade ago. Many new methods, techniques, and systems have been developed. Some of them, such as slices, height-surfaces, and iso-surfaces are now routinely used in practice.

On the other hand, many of these new methods are not used in real-world situations, many research results are nowadays considered as incremental by reviewers, and our prospective users rarely go to our conferences. So, are we, as researchers in visualization, on the right track?

*e-mail: vanwijk@win.tue.nl

In this paper I want to give a contribution to the discussion on the status and possible directions of our field. Rather than to pinpoint specific topics and activities, my aim is to detect overall patterns, and to find a way to understand and qualify visualization in general. This is an ambitious and vague plan, although the basic ground for this is highly practical.

I have to make decisions on visualization in many roles. As a researcher, decisions have to be made ranging from which area to spend time on to which particular solution to implement; as a supervisor, guidance to students must be provided; as a reviewer, new results and proposals for new research must be judged, and opinions are expected if they are worth publishing or funding; as advisor in a start-up company, novel and profitable directions must be spotted. All these cases imply judgement of the value of visualization in varying senses.

How to assess the value of visualization? Visualization itself is an ambiguous term. It can refer to the research discipline, to a technology, to a specific technique, or to the visual result. If visualization is considered as a technology, i.e., as a collection of methods, techniques, and tools developed and applied to satisfy a need, then standard measures apply: Visualization has to be *effective* and *efficient*. In other words, visualization should do what it is supposed to do, and has to do this using a minimal amount of resources. One immediate and obvious implication is that we cannot judge visualization on its own, but have to take into account the context in which it is used .

In section 2 a short overview is given of the background of the topic discussed here. In section 3 an economic model of visualization is proposed. The basic elements are identified first, the associated costs and gains are added next. Various implications of the model are discussed in section 4. In section 5 this model is applied to several cases. In section 6 the model is discussed and alternative views are considered, followed by conclusions in section 7.

Finally, this topic is on one hand very general, high-level, and abstract; on the other hand, it is also very personal, in the sense that it is about values (which are subjective), and valuation of ones own work. To reflect this, I use the first person in this paper, to emphasize that the opinions given are personal. Most examples I use come from my own work, often done together with coworkers. The main reason for this is simply that I am most familiar with it, not only with the techniques and results, but also with the context in which it took place.

2 BACKGROUND

If we use 1987 as the year where visualization started, our discipline celebrates this year its 18th anniversary. In the Netherlands, at this age a person is considered mature. Many things have changed since 1987. Graphics hardware developments are amazing, as well as the large amount of techniques that have been developed to visualize data in a variety of ways.

There are signals that there is a need to reconsider visualization. First of all, there seems to be a growing gap between the research community and its prospective users. Few, if no attendants at the IEEE Visualization conference are prospective users looking for new ways to visualize their data and solve their problems. Secondly, the community itself is getting both more specialized and critical, judging from my experience as paper co-chair for IEEE Visualization 2003 and 2004. In the early nineties, the field lay fallow, and it was relatively easy to come up with new ideas. The proceedings in the early nineties show a great diversity. Nowadays the field is getting more specialized, submitted work consists often of incremental results. This could signal that our field is getting mature. On the other hand, it is not always clear that these incremental contributions have merit, and reviewers are getting more and more critical. Thirdly, some big problems have been solved more or less [14]. For volume rendering of medical data sophisticated industrial packages that satisfy the needs of many users are available.

These trends urge a need to reconsider the field, and to think about new directions. Several researchers have presented [7, 9, 17] overviews of current challenges. Another great overview of the current status of visualization and suggestions for new directions is provided by the position papers [3] contributed by the attendants of the joint NSF-NIH Fall 2004 Workshop on Visualization Research Challenges, organized by Terry Yoo. Many issues are mentioned several times, including handling of complex and large data sets, uncertainty, validation, integration with the processes of the user, and a better understanding of the visualization process itself. One particularly impressive and disturbing contribution is [14], for its title, the name and fame of the author, and the vivid description that indeed the field has changed and new directions are needed.

In this paper no attempt is made to summarize or overview these challenges, but the aim is to find a model or procedure to judge in general if a method is worthwhile or not. In the following sections, a first step towards such a model is presented. Much of it is evident and obvious. As a defense, some open doors cannot be kicked open often enough, and also, if obvious results would not come out, the model and the underlying reasoning would be doubtful. Some statements made are more surprising and sometimes contrary to main stream thinking. To stimulate the debate, I have taken the liberty to present these more extreme positions also, hoping that some readers will not be offended too much.

3 Model

In this section a generic model on visualization is proposed. First, the major ingredients are identified; secondly, costs and gains are associated. The model is abstract and coarse, but it can be used to identify some aspects, patterns and trends.

3.1 Visualization and its context

Figure 1 shows the basic model. Boxes denote containers, circles denote processes that transform inputs into outputs. The aim here is not to position different visualization methods, for which a taxonomy would be a more suitable approach, but rather to describe the context in which visualization operates. No distinction is made, for instance, between scientific visualization and information visualization, at this level there is much more they share than what separates them.

In the following we describe the various steps. We use a mathematical notation for this, merely as a concise shorthand and to give a sense of quantification than as an exact and precise description. Processes are defined as functions, but the domains and ranges of these are ill-defined.

The central process in the model is visualization V:

$$I(t) = V(D,S,t).$$

Data D is transformed according to a specification S into a time varying image $I(t)$. All these should be considered in the broadest sense. The type of data D to be visualized can vary from a single bit to a time-varying 3D tensor field; the specification S includes a specification of the hardware used, the algorithms to be applied

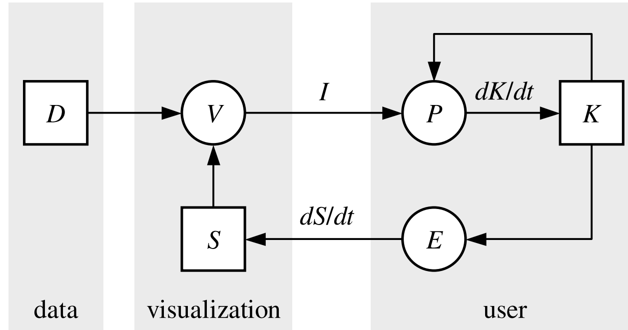


Figure 1: A simple model of visualization

(in the form of a selection of a predefined method or in the form of code), and the specific parameters to be used; the image I will often be an image in the usual sense, but it can also be an animation, or auditory or haptic feedback. In other words, this broad definition encompasses both a humble LED on an electronic device that visualizes whether the device is on or off, as well as a large virtual reality set-up to visualize the physical and chemical processes in the atmosphere. The image I is perceived by a user, with an increase in knowledge K as a result:

$$\frac{dK}{dt} = P(I,K).$$

The amount of knowledge gained depends on the image, the current knowledge of the user, and the particular properties of the perception and cognition P of the user. Concerning the influence of K, a physician will be able to extract more information from a medical image than a lay-person. But also, when already much knowledge is available, the additional knowledge shown in an image can be low. A map showing the provinces of the Netherlands provides more new information to a person from the US than to a Dutch person. Also, the additional value of an image of time-step 321 is probably small when time-step 320 has been studied just before. Concerning the influence of P, a simple but important example is that a colorblind person will be less effective in extracting knowledge from a colorful image than a person with full vision. But also, some people are much better than others in spotting special patterns, structures, and configurations.

The current knowledge $K(t)$ follows from integration over time

$$K(t) = K_0 + \int_0^t P(I,K,t)\,dt$$

where K_0 is the initial knowledge.

An important aspect is interactive exploration, here represented by $E(K)$. The user may decide to adapt the specification of the visualization, based on his current knowledge, in order to explore the data further

$$\frac{dS}{dt} = E(K),$$

hence the current specification $S(t)$ follows from integration over time

$$S(t) = S_0 + \int_0^t E(K)\,dt$$

where S_0 is the initial specification.

3.2 Economic Model

To assess if a visualization method is worthwhile, we must assess its value. We propose to use profitability in an economic sense as a measure for this. We simplify this by assuming that there is a homogeneous user community, consisting of n users which use a certain visualization V to visualize a data set m times each, where each session takes k explorative steps and time T. This is a crude simplification of course. In the real world, the user community will often be highly varied, with different K_0's and also with different aims. The costs associated with using V come at four different levels:

- $C_i(S_0)$*: Initial development costs*. The visualization method has to be developed and implemented, possibly new hardware has to be acquired.
- $C_u(S_0)$*: Initial costs per user*. The user has to spend time on selection and acquisition of V, understanding how to use it, and tailoring it to his particular needs.
- $C_s(S_0)$*: Initial costs per session*. Data have to be converted, and an initial specification of the visualization has to be made.
- C_e*: Perception and exploration costs*. The user has to spend time to watch the visualization and understand it, as well as in modification and tuning of the specification, thereby exploring the data set.

The total costs are now given by

$$C = C_i + nC_u + nmC_s + nmkC_e.$$

The return on these investments consists of the value $W(\Delta K)$ of the acquired knowledge $\Delta K = K(T) - K(0)$ per session, multiplied by the total number of sessions:

$$G = nmW(\Delta K)$$

and hence for the total profit $F = G - C$ we find

$$F = nm(W(\Delta K) - C_s - kC_e) - C_i - nC_u.$$

This gives us a recipe to decide on the value of a visualization method. Positive are high values for n, m, $W(\Delta K)$, and low values for C_s, C_e, C_i, C_u, and k. Or, in other words, a great visualization method is used by many people, who use it routinely to obtain highly valuable knowledge, without having to spend time and money on hardware, software, and effort. Indeed, quite obvious.

4 Implications

Quantification of the elements of the model is hard. In this section we discuss this in more detail, as well as a number of other issues implied by this model.

4.1 Valuable knowledge

Insight is the traditional aim of visualization. The term itself is great, and suggests a high-level contribution to the advance of science. Users are enabled to see things they were not aware of, and this insight helps them to define new questions, hypotheses, and models of their data. However, from an operational point of view, the term insight does not help us much further to assess the value of visualization. One problem is that we cannot directly observe or measure how much insight is acquired, and also, it is difficult to assess what the value of that insight is. In the model we use the term knowledge, but this suffers from the same limitations. Also, there is a strange paradox in the basic paradigm of visualization. We don't know what information is contained in the data, hence we make pictures to get insight. But if we do not know which specific aspects or features should be visible, we cannot assess if we are successful or not.

Nevertheless, we should try to measure or estimate $W(\Delta K)$, if we want to assess the value of visualization, especially because it is the only term in the model for F with a positive sign. An operational approach is to consider the use of visualization as an element in problem solving. The user has a problem, he must decide which action to take, and to make that decision he needs information. The visualization should enable him to extract the relevant information from the data.

Decisions are typically about actions to be taken or not. For instance, should a stock be bought or sold, should a patient be operated or not, which people in an organization are candidates for promotion, etc. Hence, I recommend my students to search for and enumerate possible actions of users after using their prospective tools. If such actions cannot be found or defined, the value of visualization is doubtful. Just claiming that a visualization gives insight is not enough, if we want to offer additional value.

If we know to which actions the visualization should lead to, the next steps are assessment whether the knowledge derived from the visualization does indeed support the decision, and also, to assess the economic value of this decision. This is not easy, but one can try for instance to estimate how much time is saved, or try to quantify the consequences of a wrong decision.

4.2 Alternative methods

Efficiency is relative, an aspect that is not captured explicitly in the model. One could predict a high value for F for a new method, however, if other methods are available to obtain the same knowledge against lower costs, then very likely the value for n is overestimated. Or, stated simply, if a better solution already exists, nobody will use the newer one. The model is too simple here. The effective value of n itself is not a parameter, but a function of, among others, the perceived benefit by potential users.

Developers of new visualization methods should be aware of alternative solutions, and carefully study their advantages and limitations. New methods are not better by definition. Especially when existing methods are heavily used in practice, they have proven to have value. It is often hard to beat straightforward solutions; for instance, in many cases just using a line graph is the best way to show a time-varying signal.

A defense often heard for a lesser performance of new methods compared to existing ones is that the users have not had enough time to get accustomed to them. In some cases this might hold, but an equally viable hypothesis is that an existing method is simply better. For instance, just showing a set of objects in a list enables linear scanning, whereas scanning a fancy 2D or 3D display where the objects are distributed over space is much harder [18].

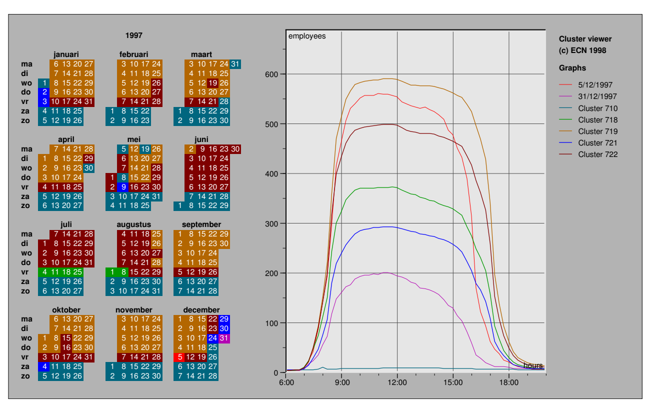


Figure 2: Visualization of daily patterns [28], an example of the combined use of conventional statistical and graphical methods.

Alternative methods are not limited to visualization methods. For instance, when an automatic method exists to extract the relevant information, visualization is useless. Visualization is not 'good' by definition, developers of new methods have to make clear why the information sought cannot be extracted automatically. One reason could be that such automated methods are not fullproof. In this case, integration of automated methods, for instance from statistics or data-mining, and visualization is a great idea, see for instance the work underway and led by Jim Thomas in the Visual Analytics arena [19].

Figure 2 shows an example where we used standard methods in a new combination [28]. For the analysis of a time-series of one year, daily patterns were clustered, i.e., finding similar daily patterns was automated. The results are shown using two conventional representations: average daily patterns of clusters are shown as graphs, and the days per cluster are shown on a calendar. The approach is straightforward and conventional, and very effective.

4.3 High initial costs

One important reason that new visualization techniques are not used in practice is the high initial cost per user $C_u(S_0)$ involved. Let us consider a potential customer for visualization, for instance a researcher doing complex simulations. First, he has to realize that maybe visualization can help him to understand his data. This is not obvious, he already uses some methods to extract information from his results in a condensed form. For instance in molecular dynamic simulations, one typical aim is to derive large scale quantities (temperatures, porosity, etc.) via simulation from the properties on a small scale (size of ions, fields, etc.). Such large scale quantities can be calculated fairly easily from the raw data. Mathematicians working in Computational Fluid Dynamics are often not interested in particular flow patterns, but rather in convergence of numerical methods and conservation of quantities, which again can be calculated easily and summarized in a few numbers.

The easiest way to visualize data is to use post-processing capabilities that are integrated with the software used. Commercial packages for, for instance, computational fluid dynamics or finite element simulation offer these. From a visualization point of view, the techniques offered are far from state of the art: Usually just options like iso-surfaces, color mapping, slicing, streamlines and arrow plots are provided. But if these meet the demands of our user, then this is a highly cost-effective way.

Suppose that this option is not available or falls short. The next step is to find alternatives. Our researcher has to get acquainted with possible solutions. Unfortunately, there are no books that present and compare novel visualization techniques (like volume rendering or topology based flow visualization) at an introductory level. So he has to study research papers, or search and get in contact with an expert in the field.

Next steps are also costly. Maybe he can get a research prototype to work with, or else he has to (or let somebody) implement the novel techniques. Often additional software has to be developed to convert his data to a suitable format.

This all takes much time and effort, while it is unclear whether the new method will indeed solve his problem. Hence, a rational decision is to abstain from this.

There are of course ways to share the initial costs with others. A group of researchers can take advantage of an initial investment by one of them. Also, providers of simulation software can be asked to integrate new methods. Visualization does not seem to have a high priority here however. For an impression of what providers think to be important for their customers, we can have a look at web-sites of companies like MSC or Fluent, and observe that features like advanced simulation capabilities and tight integration are promoted much more than visualization, which is just mentioned in passing by under the header of post-processing.

4.4 Visualization is subjective

In the ideal case, one would hope that extraction of knowledge from data is an objective process, in the sense that the outcome does not depend on who performs it, and that the analysis can be repeated afterwards by others, with the same outcome. Statistics aims at this, a typical pattern is the use of statistical tests to validate hypotheses on the data. Such tests make assumptions on the data (such as a normal distribution) and have free parameters (like the confidence level), but furthermore, they do meet the criteria for objectiveness.

Unfortunately, visualization often does not meet this aim. Consider

$$\frac{dK}{dt} = P(V(D,S,t),K).$$

This simply means that the increase in knowledge using visualization not only depends on the data itself, but also on the specification (for instance, which hardware has been used, which algorithm has been used and which parameters), the perceptual skills of the observer, and the a priori knowledge of the observer. Hence, the statement that visualization shows that a certain phenomenon occurs is doubtful and subjective.

An even harder case is the statement that a certain phenomenon does not occur. I have often spent hours visualizing data, searching for patterns and structure. Sometimes some result could be produced using a particular setting of the parameters, in other cases I failed to do so. When a visualization does not show clear patterns, it is hard to decide if this is a limitation of the visualization method, or that the setting of the parameters was wrong, or that the data simply does not contain significant patterns.

This does not mean that visualization is useless. If there are no better alternatives to inspect complex data, visualization has to be used. Another line of defense is that visualization should not be used to verify the final truth, but rather to inspire to new hypotheses, to be checked afterwards. Part of the subjectiveness can be eliminated by simply showing the visualization to the audience, so that they can view and judge it themselves. However, this does not take away the subjectiveness inherent in S, as a second hand viewer we do not know how sensitive the ultimate visualization is to changes in scales and/or selections of the data.

4.5 Negative knowledge

In the previous subsection we considered subjective aspects of visualization. There is another problem: Visualizations can be wrong and misleading. Or, in the terminology introduced here, negative knowledge ($|\Delta K| < 0$) can be produced. Tufte has introduced the *lie-factor* [23], which he defined as the ratio of the size of an effect shown in the graphic to the size of the effect in the data.

Here, I just want to give an example of my own experience with this. A long time ago I visualized the waves produced by ships for a maritime research institute. The data were the result of simulations. Figure 3 (a) shows the result of bilinear interpolation of the data. I found these results unclear, hence I decided to use an interpolating spline, thereby smoothing the surface while remaining faithful to the data. Figure 3 (b) shows clearly that two sets of waves are generated: the standard waves as well as a set of waves orthogonal to this. I proudly presented this discovery to the researcher, who immediately replied that this was physically totally impossible. A much better visualization is shown in figure 3 (c), where an approximating spline is used. The artifacts in the middle image are the result of aliasing. The data orthogonal to the ship are sampled close to the Nyquist frequency, interpolation gives rise to aliases, which corresponding waves have in this 2D case a different direction than the original wave. A smoothing interpolating spline smoothes away the high frequencies, but the first aliases survive and give rise to wrong interpretations. I learned from this that interpolation is not by definition better than approximation, and also that the judgement

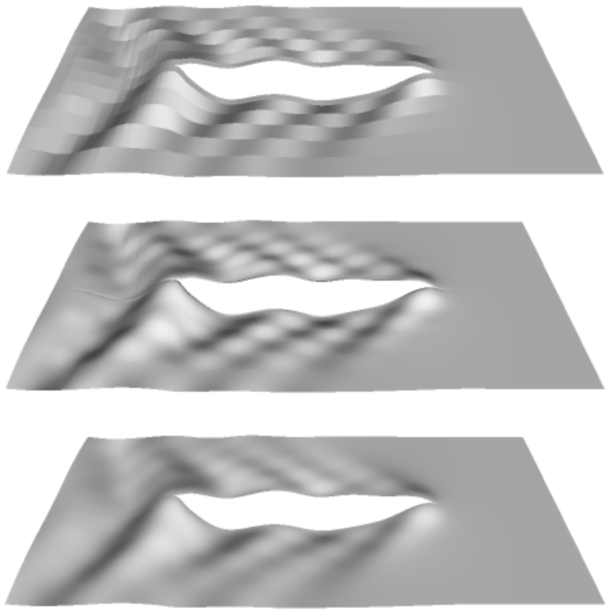


Figure 3: Wave surface, from top to bottom (a) bilinear interpolation, (b) cubic interpolation, (c) cubic approximation. Incorrect interpolation leads to artifacts.

of an expert, with a high K_0, is vital for proper interpretation and validation. I never published this, and also, articles on limitations and pitfalls of visualization are scarce. For an advancement of the field, more such reports would be highly beneficial.

4.6 Interaction

Interaction is generally considered as 'good'. One could advocate the opposite: Interaction should be avoided, and well for two reasons. First of all, as mentioned before, allowing the user to modify S freely will lead to subjectiveness. It is tempting to tune the mapping so that the desired result comes out strongly, but this can be misleading. Also, high customization can make it hard to compare different visualizations. Secondly, interaction is costly, and leads to a high C_e. Rerendering the image after a change of the mapping or the point of view taken requires often a few seconds, viewing it again also. If many options are available to modify the visualization, trying them all out can take hours. A developer of a new method therefore should think carefully about good defaults, or automatic ways to set the visualization parameters, so that as much knowledge is transferred as possible.

Obviously, in many cases interaction strongly enhances the understanding of the data. The most important case is simply when the amount of data to be shown does not fit on the screen, or is too large to be understood from a single image. In this case, navigation and selection of the data has to be supported. Ideally, the user has to be provided with cues that will lead him quickly to images where something interesting can be seen. Another case is during development of new methods. I stimulate my students to make every aspect of their new methods customizable via user interface widgets, so that the total solution space can be explored. However, for the final versions of their prototypes I recommend them to offer suitable presets under a few buttons, so that a good visualization can be obtained with little effort.

5 Examples

In this section a number of (classes of) techniques are considered and the cost model is used to explain their adoption in practice.

5.1 Texture based flow visualization

The use of texture to visualize fluid flow has been introduced in the early nineties. The idea is that dense textures enable viewers to judge the direction of flow at all locations of the plane, whereas the standard arrows and streamlines only give discrete and hard to interpret samples. The topic has been studied heavily in the visualization community, a recent non-exhaustive overview [13] has 90 references. The progress made in this decade is great. The early Spot Noise technique [24] was an interesting first attempt, in 1993 Cabral and Leedom introduced Line Integral Convolution (LIC), which gave high quality renderings of 2D fluid flow [5]. Many other variations and additions have been presented since then, for instance to handle flow on surfaces and in volumes, and also to boost the performance, using software or hardware acceleration [13]. Nowadays, high quality 2D texture images of flow fields can easily be generated on standard hardware at 50 or more frames per second [25]. This seems a success story, but on the other hand, these methods are not integrated in commercial software, users of Computational Fluid Dynamics (CFD) are typically completely unaware of their existence, let alone that they routinely use them to solve their problems. Here I use texture based flow visualization because I am most familiar with it, but for other classes of methods, such as topology based flow visualization and feature based flow visualization, similar patterns seem to apply.

How can we explain this? We consider the parameters of the cost model. The number of users n is not too great. CFD is vital for some areas, but there are few cases where CFD is routinely used for screening, compared to for instance medical applications. The frequency of use m is also not very high. Often, CFD-users spend much time on defining the model, simulations can also take a long time. By then, they are very familiar with their models (high K_0). For the analysis of the results many alternative options are available, including composite quantities (such as lift of an airfoil) and straightforward cross-sections and arrow plots, with low costs. The use of texture based visualization incurs at least a high value for C_u (see section 4.3). The additional ΔK that texture based visualization offers is unclear. Laidlaw et al. [12] have compared different vector visualization methods. LIC turned out to yield better results for critical point detection, but worse results for other aspects, such as estimation of the angle of the flow. Also, standard LIC does not give the sign of the direction of the flow. Hence, we can doubt about the value of ΔK. And finally, it is not clear what the real value is of this ΔK, in the sense that better visualization leads to better decisions. At least, so far there does not seem to be such a strong need for better visualization methods in the CFD community that they have attempted to integrate these methods into their packages.

5.2 Cushion treemaps

Also in the early nineties, Johnson and Shneiderman introduced the concept of a treemap [8] to visualize large hierarchical data sets. The base algorithm is straightforward: A rectangle is recursively subdivided according to the hierarchical data, in such a way that the size of each rectangle corresponds to the size of each leaf element. In the late nineties we proposed to use hierarchical cushions to show the underlying hierarchical structure more clearly [26]. We packaged this technique in 2000 in SequoiaView [1], a tool for the visualization of the contents of a hard disk (figure 4), and made this publicly available as freeware. Since then, SequoiaView has been downloaded about 400,000 times from our site. Also, it has been

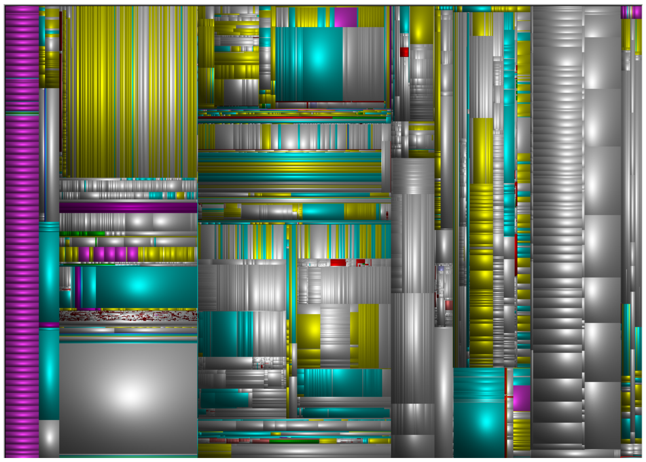


Figure 4: Visualization hard disk using SequoiaView [1, 26, 27], an example of an application that has found an audience.

distributed three times via CD with the German computer magazine C't. This is an example how visualization has reached an audience.

The economic model helps to explain this result. First, the number of (potential) users is very large, in principle equal to the number of PC users. Typically, such a tool is used several times per year, which is not very high, but not neglectable. Alternative solutions for this problem are scarce (SpaceMonger, using also treemaps is an example), and getting an overview of a hard disk is hard using Windows Explorer.

Information can be derived fairly easy from the visualization. It is easy to spot large files, large directories, and large collections of files. Furthermore, this information is directly valuable for the user: The tool can help (and many users have confirmed this) to delay buying a new hard disk. The action is clear here: removal of files. We offer an option to start up Windows Explorer from SequoiaView to remove files manually. The initial costs per user are low: The tool itself is freeware, it only has to be downloaded and installed. The costs per use case are minimal as well. By default, the tool starts to collect data from the last folder specified, and an image is shown automatically. Exploration is easy: Extra information per file can be obtained by hovering the pointer over the rectangles.

In summary, F is high in this case. We would like to think that this is a result of our visualization method, however, the main reasons are probably that our tool meets a real need of real users, and that the costs, in all respects, are minimal.

5.3 Presentation vs. exploration

Next we consider a more general case. The main use cases for visualization are exploration (where users do not know what is in the data), and presentation (where some result has to be communicated to others). It is hard to quantify this, but my impression is that many researchers in visualization consider exploration as the major raison d'être for visualization, whereas presentation is considered as something additional and not too serious. However, from my own experience, presentation is at least just as important as exploration. Many users find videos and images attractive for presenting their work at conferences; the popularity of visualization tools and demos often rises sharply just before open days. For years I had a pleasant and fruitful cooperation with Flomerics Ltd. in the UK. This company develops CFD-based tools for, amongst others, thermal assessment for the electronics industry. My major contact there was the marketing manager, who could use visualization to show the benefits of the CFD tools to managers.

In a broader sense, we can view visualization everywhere. Commercial television uses visualization to show the chemical miracles of new cosmetics, the ingenuity of vacuum-cleaners, and why a new fitness device does not harm your back. Obviously, such visualizations are probably not the result of visualizing data, but rather the result of fantasy of advertisement agencies. Selling stuff is not only the realm of business, but also of science itself. Once I heard someone state: *The purpose of visualization is funding, not insight*. We can explain the value of visualization for presentation with the cost model. If we consider the viewers of such visualizations as the users, we see that n is high; K_0 is low (the viewers know little about the topic, so much can be gained); the action to be taken is clear (buy a product, fund research) and has direct economic consequences; the costs for the viewers are low (they just have to watch the visualization), although they can be high for the presenter. And furthermore, for these purposes there are almost no alternative or competing techniques. Pure facts (product X saves Y percent of time) can be convincing, but to make plausible why, and also to show that this is all Scientifically Sound, visualization is the way to go.

6 Discussion

In the preceding sections a number of questions were raised and various disturbing statements were made. There are many objections that can be made, and in this section some of them are given. One important distinction is to consider visualization either as technology, art, or as science. Associated with these are a number of routes for future work.

6.1 Technology

In the cost model, visualization is considered as a technology, to be measured for utility. In this context, research in visualization should lead to new solutions that are useful in practice. Not all the work done is successful in this respect, but we can find a number of reasons to explain this.

First of all, innovation is a merciless process, where only few new solutions survive. A rule of thumb in product development is that thousand ideas lead to hundred prototypes, which lead to ten products, out of which just one is successful. The visualization research community operates in the start of this pipeline, hence it should come as no surprise that not everything finds its way. We can see it as a mission to develop inspiring new ideas, which are a primary fuel in the innovation process.

Creativity however consists of two parts: creation of new ideas as well as selection of the best ones. The first task is fulfilled properly by the visualization community, the second is not. The number of careful validations of visualization methods is still low, although this seems to be improving in the last years.

Secondly, innovation is a long chain. Developing new methods is quite different from turning these into products and marketing them. There is a gap between our prospective users and the research community. Both do not have the proper stimuli to bridge this gap: individual researchers are too busy increasing the number of publications they are judged on, and for the end-users implementing new methods is far too costly. The gap can be filled in different ways. One way is via commercial companies (spin-off companies, or companies that integrate visualization in their simulation packages), an alternative is via open source and academic development and maintenance, funded by government agencies. VMD [2] is an example of the latter category. As a corollary, if we think that visualization is useful and that this gap causes the lack of adoption, we should aim at increasing funding for more practical activities. Or we should start up companies.

Thirdly, one could state that all this is a matter of time. It takes time before new ideas penetrate, before new users become aware of new methods, before initiatives are taken to integrate new methods into existing systems. This might be true in some cases, however, it is also too easy to use this as an excuse. It could be used for any method, hence it does not help us to distinguish between good and bad ones.

Fourthly, the focus in the model is on large numbers of users and use cases. One can also consider cases where the number of users is small, but where the value of the result is very large. In the books of Tufte some great cases are presented, such as Snow's discovery of the cause of a cholera epidemic in 1854 [21]. Are there recent cases for new visualization methods? Cases that enabled the researcher to obtain a major scientific insight, to save many lives, or to solve a crucial technological problem? One would like to read more case studies in this spirit, which show that visualization is worthwhile and can make a difference.

Finally, one defense is that maybe we are not doing too bad, compared to other disciplines. Many disciplines (for instance, in mathematics) do not care about practical usability at all, for some computer science fields that do claim to have practical relevance it is also hard to see the adoption in practice. Why should we bother? This notion is explored further in the next subsection.

6.2 Art

One could claim that visualization has value in its own right, and for its own purposes. One part of this is in the results: Some of the images we produce have a clear aesthetic value. But the art of visualization can also be found in the ideas, methods, and techniques developed. We can consider ourselves as a group of puzzle solvers, and the challenge is to develop new, simple, and elegant solutions, which provide us all with intellectual and aesthetic satisfaction.

This is not a line of defense that can help us to convince our prospective users and sponsors. Nevertheless, I do want to mention it, because it can give a powerful thrust (and obviously also because results of this possibly will find applications in the real world). In the early nineties, I worked hard on using texture for visualization – not to satisfy users, but simply because the puzzle was tough, challenging, and hard to crack. The work of our student Ernst Kleiberg on botanically inspired tree visualization (figure 5, [10]) was not driven by user requests, but just an experiment to find out if it could be done at all. At the Information Visualization Symposium in 2004 we got two messages back. Alfred Kobsa found the usability limited, compared to other methods [11]; on the other hand, Stuart Card showed this image in his keynote speech as an example of a nice visualization. Is this a good visualization or not?

Finally, in my own work, I found aesthetic criteria on new methods to be guiding and effective. Sometimes, each link of the chain from idea, mathematical model, algorithm, implementation to visual result is clean, simple, elegant, symmetric, etc. It is amazing how much effort is required to reach this. Developing great ideas is simple, rejection of bad ideas takes all the time.

6.3 Science

Apart from considering visualization as a technology, or as an art for its own sake, we could consider visualization research as a scientific discipline. If there is something like a Science of Visualization, what should it bother about? Loosely defined, a scientific discipline should aim at a coherent set of theories, laws, and models that describe a range of phenomena, have predictive power, are grounded in observations, and that can be falsified.

If we look at the field now, many algorithms and techniques have been developed, but there are few generic concepts and theories. One reason for the lack of fundamental theories is that visualization is intrinsically complex, has many aspects, and can be approached from different perspectives. In terms of the model proposed, visualization can be observed from the point of view of the data D to be visualized, the various solutions proposed (S and V), from the ΔK aimed at, i.e., the purpose or discipline for which it is applied, the images I themselves, or from aspects such as perception P or exploration E. Also, developing good visualization solutions is intrinsically a design problem, and closed form solutions for the optimalization problem "Given D find V such that ΔK is optimal" cannot be expected.

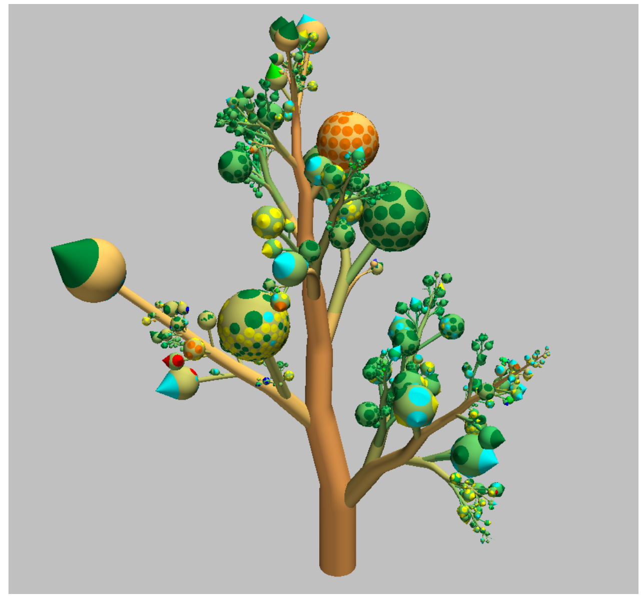


Figure 5: Botanic visualization contents of a hard disk [10, 27]. Useful or just a nice picture?

Nevertheless, we could and should aim at more generic insights, at several levels. First of all, a *descriptive* approach can be pursued further. Methods are analyzed and categorized, leading to taxonomies that show how they relate to and differ from each other. Such taxonomies span up the current solution space, and can lead to insight where new opportunities are. Some examples of good overview papers are [30, 6, 13], a great example of a taxonomy is given in [4], where a variety of different marching cube style algorithms are brought under one umbrella using computational group theory. Even if it were only because the field is still developing and overviews are quickly outdated, more work in this area should be encouraged. Taxonomies need not be confined to methods, also taxonomies on different kinds of data and especially on different types of knowledge that are relevant for end users are useful.

Secondly, *evaluation* and *validation* are important. Assessment of the effectiveness and efficiency of different methods and techniques is vital from a technological point of view (which method to use), but also as a base for more generic statements on visualization. A science of visualization should be empirical, in the sense that concrete measurements of the phenomena studied are done, which in our case concern people making and watching images that depict data. Tory and Möller [20] give a good overview of the current status of the use of human factors research in visualization, and identify areas for future research.

Thirdly, in line with the previous, we should ultimately aim at *generic results* (models, laws) that enable us to understand what goes on and to predict why certain approaches do or don't work. In the end, explanations should be based on properties of the environment of visualization, especially the end user. The value of visualization is ultimately determined by his perceptual abilities, his knowledge on the data shown, the value he assigns to various insights, and the costs he is willing to spend.

Ware's book on Information Visualization [29] is a rich source of insights on perception and how these can be used to improve visualization, Tufte gives many useful guidelines and recommendations in his books [23, 21, 22]. However, many of these are not quantitative, and also, do not explain how to handle conflicting requirements. One operational and practical criterium on guidelines is that they should allow for automated implementation, such that the user gets a good, if not optimal view on the data without costs. The early work of Mackinlay [15] on automated generation of visualizations is great, and it is surprising that the state of the art in this area does not seem to have advanced much further since then.

Finally, *methodological* issues have to be studied further. This concerns questions like how to design visualizations and how to measure and evaluate the effectiveness of various solutions. And also, how to assess the value of visualization in general.

7 Conclusion

In the preceding sections, I have tried to answer the question how the value of visualization can be assessed. As a conclusion, I think there is not a single answer, but that it depends on the point of view one adopts. One view is to consider visualization purely from a technological point of view, aiming for effectiveness and efficiency. This requires that costs and benefits are assessed. The simple model proposed enables us to get insight in various aspects of visualization, and also to understand why certain classes of methods have success and others not. Another view is to consider visualization as an art, i.e., something that is interesting enough for its own sake, and finally a view on visualization as an empiric science was discussed.

Obviously, these three different views, schematically depicted in fig. 6, are strongly related, and results from one view can stimulate work according to the other views. Finally, each view that is adopted does imply playing a different game, and if we want to win, we should play those games according their own rules: aim for provable effectiveness and efficiency, aim for elegance and beauty, and aim at generic laws with predictive power.

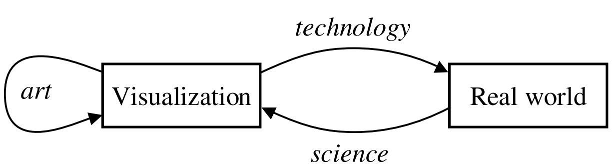


Figure 6: Views on visualization

References

[1] www.win.tue.nl/sequoiaview.

[2] www.ks.uiuc.edu/Research/vmd.

[3] Position papers NIH/NSF fall 2004 workshop on visualization research challenges, 2004. visual.nlm.nih.gov/evc/meetings/vrc2004.

[4] D.C. Banks and S. Linton. Counting cases in marching cubes: Toward a generic algorithm for producing substitopes. In *Proceedings IEEE Visualization 2003*, pages 51–58, 2003. (Best Paper award).

[5] B. Cabral and L. C. Leedom. Imaging vector fields using line integral convolution. *Computer Graphics*, 27:263–270, 1993. Proceedings SIGGRAPH'93.

[6] I. Herman, G. Melanon, and M.S. Marshall. Graph visualisation in information visualisation: a survey. *IEEE Transactions on Visualization and Computer Graphics*, 6(1):24–44, 2000.

[7] B. Hibbard. Top ten visualization problems. *SIGGRAPH Computer Graphics Newsletter*, 33(2), 1999.

[8] B. Johnson and B. Shneiderman. Treemaps: a space-filling approach to the visualization of hierarchical information structures. In *Proceedings of IEEE Visualization '91*, pages 284–291. IEEE Computer Society Press, 1991.

[9] C. Johnson. Top scientific visualization research problems. *IEEE Computer Graphics and Applications*, 24(4):13–17, 2004.

[10] E. Kleiberg, H. van de Wetering, and J.J. van Wijk. Botanical visualization of huge hierarchies. In *Proceedings IEEE Symposium on Information Visualization (InfoVis'2001)*, pages 87–94. IEEE Computer Society Press, 2001.

[11] A. Kobsa. User experiments with tree visualization systems. In *Proceedings 2004 IEEE Symposium on Information Visualization (InfoVis 2004)*, pages 9–16, 2004.

[12] D.H. Laidlaw, R.M. Kirby, J.S. Davidson, T.S. Miller, M. da Silva, W.H. Warren, and M.J. Tarr. Quantitative comparative evaluation of 2d vector field visualization methods. In *Proceedings IEEE Visualization 2001*, pages 143–150. IEEE Computer Society Press, 2001.

[13] R.S. Laramee, H. Hauser, H. Doleisch, B. Vrolijk, F.H. Post, and Daniel Weiskopf. The state of the art in flow visualization: Dense and texture-based techniques. *Computer Graphics Forum*, 23(2):203–221, 2004.

[14] B. Lorensen. On the death of visualization, 2004. In [3].

[15] J.D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, 1986.

[16] Bruce H. McCormick, Thomas A. DeFanti, and Maxine D. Brown (eds.). *Visualization in Scientific Computing*. ACM SIGGRAPH, New York, 1987.

[17] T.-M. Rhyne, B. Hibbard, C. Johnson, C. Chen, and S. Eick. Can we determine the top unresolved problems of visualization? In *Proceedings IEEE Visualization 2004*, pages 563–565, 2004.

[18] B. Shneiderman, 2004. Comment during session at InfoVis 2004.

[19] J.J. Thomas and K.A. Cook (eds.). *Illuminating the Path: Research and Development Agenda for Visual Analytics*. IEEE, 2005.

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

[21] E.R. Tufte. *Envisioning Information*. Graphics Press, 1990.

[22] E.R. Tufte. *Visual Explanations: Images and Quantities, Evidence and Narrative*. Graphics Press, 1997.

[23] E.R. Tufte. *The Visual Display of Quantitative Information (2nd edition)*. Graphics Press, 2001.

[24] J.J. van Wijk. Spot noise: Texture synthesis for data visualization. *Computer Graphics*, 25:309–318, 1991. Proceedings SIGGRAPH'91.

[25] J.J. van Wijk. Image based flow visualization. *ACM Transactions on Graphics*, 21(3):745–754, 2002. Proceedings of ACM SIGGRAPH 2002.

[26] J.J. van Wijk and H. van de Wetering. Cushion treemaps. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis'99)*, pages 73–78. IEEE Computer Society Press, 1999.

[27] J.J. van Wijk, F. van Ham, and H.M.M. van de Wetering. Rendering hierarchical data. *Comm. ACM*, 46(9ve):257–263, September 2003.

[28] J.J. van Wijk and E. van Selow. Cluster and calendar-based visualization of time series data. In *Proceedings 1999 IEEE Symposium on Information Visualization (InfoVis'99)*, pages 4–9. IEEE Computer Society Press, 1999.

[29] C. Ware. *Information Visualization: Perception for Design. (2nd Edition)*. Morgan Kaufman, 2004.

[30] P.C. Wong and R.D. Bergeron. 30 years of multidimensional multivariate visualization. In G.M. Nielson, H.H. Hagen, and H. Mueller, editors, *Scientific Visualization – Overviews, Methodologies, and Techniques*. IEEE Computer Society Press, 1997.

A Framework of Interaction Costs in Information Visualization

Heidi Lam

Abstract—Interaction cost is an important but poorly understood factor in visualization design. We propose a framework of interaction costs inspired by Norman's Seven Stages of Action to facilitate study. From 484 papers, we collected 61 interaction-related usability problems reported in 32 user studies and placed them into our framework of seven costs: (1) Decision costs to form goals; (2) System-power costs to form system operations; (3) Multiple input mode costs to form physical sequences; (4) Physical-motion costs to execute sequences; (5) Visual-cluttering costs to perceive state; (6) View-change costs to interpret perception; (7) State-change costs to evaluate interpretation. We also suggested ways to narrow the gulfs of execution (2–4) and evaluation (5–7) based on collected reports. Our framework suggests a need to consider decision costs (1) as the gulf of goal formation.

Index Terms—Interaction, Information Visualization, Framework, Interface Evaluation

1 INTRODUCTION

Even though interaction is vital to interface success, the information visualization community has generally focused more on visual encoding than on interaction. Ideally, visualization designers should be able to weigh costs and benefits of interaction based on empirical results. Before we can effectively evaluate interaction, we need to first understand how it can contribute to visualization use and how designs can fall short in supporting these roles. While taxonomies of interaction techniques exist to study roles of interaction in interface use (see [69] for a survey), we still need a framework to study interaction costs.

Similar to understanding interaction techniques, one of the first steps in understanding interaction costs is to identify instances. To better design, we need to take an holistic approach and study interaction during interface use [3] instead of focusing on individual technique in isolation and with abstract tasks (e.g., [27]). Typical user studies seldom explicitly measure or even identify interaction costs, but reports on recorded usage patterns, participant strategies, and interface choice sometimes provide insights.

In this paper, we propose a framework of seven interaction costs based on cost reports gathered using a qualitative review. From 484 papers, we identified 32 that reported interaction-related usability issues. Reports collected were placed into a framework of action cycles in visualization use inspired by Norman's Seven Stages of Action [39]. We highlight interaction-design considerations to narrow Norman's gulfs of execution and evaluation, and propose adding a gulf of goal formation to study decision costs in establishing data-analysis focus.

After related work in Section 2, we summarize our framework in Section 3. Section 4 describes our review method and Section 5 elucidates our framework. Section 6 puts forth design considerations to mitigate some of the interaction costs.

2 BACKGROUND AND RELATED WORK

For the purpose of this paper, we define **interaction** as actions from users that cause visible changes in the visualization, and **interaction techniques** as those actions. Since we gather our cost instances from papers, we only considered observable interaction in typical evaluations, ignoring non-action communications (e.g., eye-gazes) and unsolicited system actions (e.g., alerts). To us, an **interaction cost** is when the dialogue between users and system breaks down, or where users face enough difficulty accomplishing tasks to become aware of the user interfaces as obstacles to be overcome [67].

• *Heidi Lam is with the University of British Columbia, E-mail: hllam@cs.ubc.ca.*

Manuscript received 31 March 2008; accepted 1 August 2008; posted online 19 October 2008; mailed on 13 October 2008.
For information on obtaining reprints of this article, please send e-mailto:tvcg@computer.org.

One approach in framework development is to extend existing interaction taxonomies (see [69] for a survey) to cover costs. However, interaction costs can be implementation dependent. For example, navigation implemented in pan-and-zoom interfaces may fail to support object constancy, while navigation implemented in overview+detail interfaces may incur view-coordination problems.

Instead, we prefer a user-centric model depicting action steps. Researchers have developed models of user interaction with different foci and abstraction levels. Spence's navigation framework [61] and Card *et al.*'s knowledge crystallization task [8, p.10] are cognitive models with insufficient user-visualization interaction focus for our purpose. Chi and Riedl's operator framework classified operators into stages between raw data (value) and visual representation (view) [11], but our focus is on view only. Jankun-Kelly *et al.*'s P-Set Model describes the exploration process at the interaction-technique level (e.g., zoom, rotate), but does not cover low-level motion (e.g., mouse drag) or high-level cognitions (e.g., result interpretation) that can also incur costs [28]. We therefore based our framework on Norman's Seven Stages of Action [39] for its comprehensive coverage of user actions.

3 A FRAMEWORK OF INTERACTION COSTS

We adapted Norman's Seven Stages of Action [39, p.46–53] to visualization use and classified interaction-related publication statements, or **reports**, based on the stage of interaction at which they occur. Our framework has seven costs (Fig 1):

1. *Decision costs to form goals*: When interfaces become more powerful and display more data points, users usually need to decide to focus on a subset of data (Section 5.1.1) and interface options (Section 5.1.2).
2. *System-power costs to form system operations*: Once users have a question in mind, they need to translate it into operations. Deciding on the correct operation sequences may be difficult especially for powerful systems (Section 5.2).
3. *Multiple input mode costs to form physical sequences*: When the input device offers multiple modes, translating system operations to device operations may be difficult due to inconsistent mode operations on multiple views (Section 5.3.1), mode change with inadequate visual feedback (Section 5.3.2), and overloaded input controls (Section 5.3.3).
4. *Physical-motion costs to execute sequences*: Even for young and healthy users over short time spans, motions such as mouse position (Section 5.4.1) and mouse drag (Section 5.4.2) can incur costs. Even low-cost motions can collectively cumulate into usability problems (Section 5.4.3).
5. *Visual-cluttering costs to perceive state*: Interaction such as mouse hovering can cause visual cluttering that makes state perception difficult (Section 5.5).

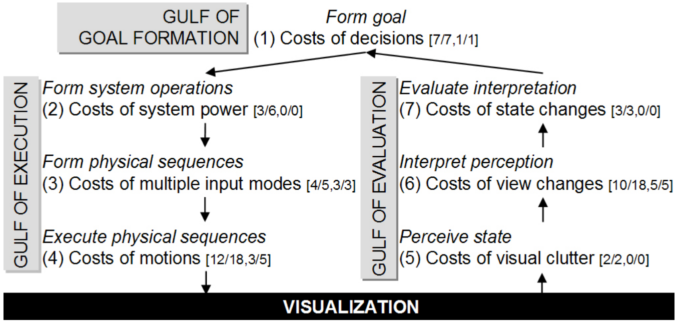


Fig. 1. A framework of interaction costs inspired by Norman's Seven Stages of Action (in italics) [39]. We added a gulf of goal formation to Norman's gulfs of execution and evaluation. Bracketed numbers are report and recommendation counts (included-in-this-paper/reviewed). Two general-observation reports are excluded in this figure.

6. *View-change costs to interpret perception*: Interaction usually result in view changes that requires re-interpretation based on expectations, which may not be met in automated systems (Section 5.6.1). Interpretation requires object association of: (1) temporal objects, as in zooming (Section 5.6.2); (2) spatial objects, as in view coordination (Section 5.6.3); and (3) local and global objects, as in navigation (Section 5.6.4).

7. *State-change costs to evaluate interpretation*: Data analysis often requires reflection on multiple data views or analysis states. Section 5.7 looks at the need for refinding in visualizations.

Norman's model has two gulfs. The **gulf of execution** is "the difference between the [user] intentions and the allowable [system] actions" [39, p.51], which covers costs (2–4) in our framework. The **gulf of evaluation** "reflects the amount of effort that the person must exert to interpret the physical state of the system and to determine how well the expectations and intentions have been met" [39, p.51], which covers costs (5–7). To emphasize data analysis and discovery supported in information visualization, we added a **gulf of formation** to cover cost (1), or the amount of effort required to formulate suitable intents. Section 6 discusses design considerations to narrow these gulfs.

Since Norman's model is an approximate model, behaviours are not required to involve all stages in sequence and each may take drastically different time durations [39, p.48]. For example, in information visualization, the steps may be carried out in rapid successions as in the case of dynamic queries [2]. In general, this paper treats cognitive-decision costs as gulf of goal formation, physical-motion costs as gulf of execution, and cognitive-interpretation costs as gulf of evaluation.

4 Methodology and Scope

Our work differs from most systematic reviews such as Chen & Yu's [10] as we did not start with specific questions. Instead, we took a bottom-up and qualitative approach to identify interaction costs from coded individual study reports. Our approach has three stages: venue selection; paper coding and selection; report and cost identification.

4.1 Venue selection

We reviewed papers from the IEEE Symposium on Information Visualization (InfoVis) proceedings, Palgrave's Journal of Information Visualization (IVS), Elsevier's International Journal of Human-Computer Studies (IJHCS), and ACM's Transactions of Computer-Human Interface (TOCHI). Our resource constraints limited our venue selection: we focused on journal publications as we believe researchers have more room to report observations.

4.2 Paper coding and selection

We coded all publications for InfoVis (1995–2007), IVS (2002–2007), and TOCHI (1994–2007). For IJHCS, we first filtered the papers with the query term "visualization" for title, abstract, or keywords and retrieved 49 papers between the years 1994 and 2007.[1]

	Category/Pub	InfoVis	IVS	IJHCS	TOCHI
	# papers	283	117	35	49
1.	No evaluation	178	50	8	12
2.	Case studies	48	32	1	4
3.	Informal studies (iSys|iReport)	18 (17|**2**)	5 (5|0)	3 (3|0)	1 (1|0)
4.	Qual study (iSys|iReport)	5 (5|**2**)	10 (10|**3**)	6 (6|**1**)	5 (4|0)
5.	Quant expt (iSys|iReport)	34 (20+2?|**8**)	20 (9+1?|**3**)	17 (11|**6**)	27 (20|**7**)

Table 1. Coded paper counts. For categories (3–5), we further classified if the study interface is interactive (iSys), and if the paper reported interaction use (iReport). '?' denotes cases where we were uncertain if the study interfaces were interactive. Qual study = qualitative evaluations with users and tasks; Quant expt = quantitative factorial experiments. Bolded numbers represent the 32 papers used in this review.

For IJHCS and TOCHI, the two non-infovis specific venues, we included papers that focused on infovis subjects[2] and reported at least one visualization technique or system. We were then left with 26 IJHCS and 49 TOCHI papers, along with 283 InfoVis and 117 IVS papers. We coded the papers by their evaluation methods, as shown in Table 1. In short, we started with 484 infovis-related papers and ended up with 32 for our review.

4.3 Report and cost identification

From the 32 papers that reported interaction-related usability issues, we collected 61 reports of problems and 14 design recommendations. We took a bottom-up approach by grouping reports into costs and putting costs to action stages. Due to space constraints, we only included 42 reports and 12 recommendations from 27 papers here to illustrate our framework (Fig 1). The full list is at `http://www.cs.ubc.ca/~hllam/res_icost.htm`.

5 Interaction Costs

For each of the seven costs, we provide background literature and commonly-used solutions, followed by reports as examples and when appropriate, as test cases for these solutions. We included report authors' recommendations. Report-paper citations are denoted with '*'.

5.1 Decision costs to form goals

While command-line interfaces have been criticized for lack of cues to prompt user actions, modern flexible visualization systems may also share this problem. We found reports in two areas that require decision making: finding a data subset to explore (Section 5.1.1) and choosing amongst interface options (Section 5.1.2).

5.1.1 Choosing a data subset

When visualizations become more powerful and display more data points, users need to decide on data focus. Visualization can guide users by providing visual cues or information scent. Pirolli and Card developed the information foraging theory [44] that has been applied to designs of web-searching support tool designs (e.g., [43]) and to model web usage behaviours (e.g., [7, 12]). Their later Sensemaking process [45] incorporated Russell *et al.*'s sensemaking loop [52] with their foraging loop to depict intelligence analysis.

However, to provide information scents, the software needs to address the challenge in predicting user interests and intent (Section 5.4.1). One workaround is to simply provide more information

[1] Last query performed on December 12, 2007

[2] In our analysis, we excluded subjects such as theory (e.g., cognitive models, design architectures, and evaluation methods), human-behavioural studies (e.g., ethnographic studies), non-visual interfaces (e.g., audio), virtual or augmented reality, computer-assisted collaborative work focusing on human-human communications (e.g., designs of avators), and special issue introductory papers.

such as usage statistics (e.g., scented widgets [66]) and metadata to faciliate data reordering and selection (e.g., Table Lens [48]).

We found reports where users were lost in data due to insufficient information displayed. Kobsa reported that with Eureka,

> Since attributes in Eureka are vertically aligned, there is not very much room for attribute labels if data has more than, say, 20 dimensions. In this case, users have troubles making sense of the data and finding the attributes they need since the attribute labels on top of the columns are largely hidden. [*32, p.127]

Similarly, Abello *et al.* found that users did not know where to go when exploring a large unknown dataset using ASK-Graph:

> Users would be able to navigate around just fine, but had no idea where they should start to look for interesting features. As a result they sometimes stumbled upon something interesting, but spent most of their time randomly browsing the data. [*1, p.67]

In the context of menu design, Hornbæk and Hertzum recommended increasing information scent by expanding ahead:

> Perhaps hierarchical menus could also benefit from some of the ideas used for increasing the number of simultaneously visible menu items in nonhierarchical menus. Submenus could, for example, reveal more of their contents by dynamically expanding two levels of the menu structure in response to cursor movements. [*22, p.28]

5.1.2 Choosing amongst interface options

In some cases, users may form their goals based on available interface options. Even though human intuition may want more choices, decisions require mental and visual concentration [55]. Indeed, participants have expressed their preference to have less interface options, such as in Chung's evaluation of SBizPort:

> The subjects also liked the user-friendliness and ease of navigation of SBizPort. Nine subjects commented positively on it. For instance, a subject said that SBizPort's "categories make it easier to browse" and another subject said not having too much (many) options (in SBizPort) is helpful to the user because it makes it easy to look for the information. [...]
>
> In comparison, subjects were less satisfied with YahooES than SBizPort. They had difficulty obtaining precise information from YahooES. Ten subjects said that YahooES had too many options to choose from and hence distracted them from finding relevant information. [*13, p.821]

Participants may also express their preference by choosing less interactive but suboptimal interfaces:

> We believe this interesting choice [for single-level interfaces] was due to the cost of interface interaction complexity, which may also explain the lack of performance benefits over the optimal single-[level] interface for each task. Although seemingly tedious and laborious, using the high-[detail] plots has a low cognitive load: the only navigation available is scrolling, a relatively passive exercise, and the answer will usually be apparent sooner or later. In contrast, navigation in a multiple-[level] interface is complex, as it involves active selection of potential target candidates, an action that requires mental and visual concentration. [*34, p.1284–1285]

5.2 System-power costs to form system operations

Once users have a question in mind, they need to translate it into operations. Deciding on the correct sequences of operations may be difficult especially when the visualization offers a large set of operations. In evaluating Spotfire, Kobsa reported a cognitive set-up cost:

> While Spotfire offers several representations in parallel, in many cases not all of them are suitable for solving a given problem. It took users considerable time to decide on the right representation and to correctly set the coordinates and the parameters, particularly when the solutions required several steps. This seems to be caused both by the wealth of visualizations that the system offers, but also by the restrictions each of them imposes once it has been selected. [*32, p.127]

Without a clear set of available operations, users may have expectations based on standards, as found by Kang *et al.*'s NetLens study:

> [users observed an inconsistency that] while multiple histogram bars are selectable, using Shitft+Click is not supported [in NetLens]. [*30, p.29]

Users therefore assumed the burden to find the correct operations:

> Users must therefore plan in advance what variables should be used and how they should be represented. This planning must be performed without assistance from a visualization and takes up considerable time. [*32, p.129]

5.3 Multiple input mode costs to form physical sequences

Having multiple input modes or overloaded controls can lead to mode errors, where "the action appropriate for one mode has different meanings in other modes" [39, p.110].

The guideline to eliminate multiple input modes is impossible to follow when the input device needs to provide more actions than the number of available controls. The consequence may be costs in inconsistent mode use (Section 5.3.1), imperceptible mode changes (Section 5.3.2), or when overloading a single control (e.g., for parallel panning and zooming; Section 5.3.3).

5.3.1 Inconsistent mode use on multiple views

Users form habits with interface use. One interaction cost is due to inconsistent mode use on different views. For example, when studying zoomable user interfaces, Hornbæk *et al.* reported that,

> Subjects' habit formation highlighted some limitations in the interfaces. At least eight subjects tried to use a way of navigating from the overview window in the detail window or vice versa. Some subjects tried to click on the detail window, probably with the intention of jumping to the place where they clicked. This way of navigating seemed to be taken from the overview window, where clicking on a point centers the field-of-view box on that point. Similarly, some subjects tried to zoom in and out while they had the mouse over the overview window. This way of interacting seemed to be mimicked after the interaction with the detail view. [*23, p.381]

Hornbæk *et al.* suggested eliminating view-specific commands:

> We believe that interfaces with an overview should eliminate navigation commands that are specific only to the overview window or to the detail window, that is, they should aim at unifying navigation. All zoom and pan actions should therefore be similar across windows. [*23, p.384]

5.3.2 Imperceptible mode changes

Sometimes, mode change may not be intended or even noticed by users, especially with inadequate visual feedback. Hornbæk and Hertzum, in their study of Bederson's fisheye menu [4], reported that,

> [W]hen in focus-lock mode, many participants wanted to keep scrolling up or down toward an item they had seen before entering this mode. However, in the fisheye and multifocus menus, items visible in small font moved out of the menu as items in the transition region expanded. [...] Although visually indicated, these modes seemed to confuse participants. Especially when in focus-lock mode and accidentally crossing the center of the menu, at least seven participants expressed confusion when the focus area was consequently dynamically recentered to the mouse position. [*22, p.19]

Hornbæk and Hertzum suggested making the change continuous:

> Our data show that the binary nature of this mode caused participants problems. A simple idea would be to use a continuum instead. When the mouse is moved toward the righthand side of the menu, the selection height of menu items would increase toward a maximum of their visual height.[*22, p.28]

Or to make the mode change explicit by requiring user actions:

> [With the] quasimode [the user enters the] focus-lock mode when [he] presses the mouse button. This would lessen the possible confusion of modes by turning the focus lock on when users are about to complete their selection, thereby enabling any final adjustments of mouse position to be made at maximum selection height before users release the mouse button to select the target menu item. [*22, p.28]

5.3.3 Overloaded controls

Overloading input controls can lead to confusion and errors, even when users are aware of the operations. Hornbæk *et al.* found that with zoomable user interfaces,

> Many subjects experienced occasional problems with the combined zoom and pan button. Even though subjects practiced this combination button during the training tasks, 18 subjects zoomed at least one time when they verbally indicated that they were trying to pan. The delay before zooming began was sometimes too short. This appeared to happen when subjects began initiating a pan action without having made up their minds about which direction to pan. [*23, p.381]

Buering *et al.* also found similar problems with zoomable user interfaces in PDA-sized devices:

> Seven subjects mentioned that they had problems with the sliding technique of the ZUI. [...] As a result some subjects accidentally triggered a zoom operation when actually trying to slide. [*6, p.834–835]

How to best provide parallel zooming and panning remains unclear:

> Research is needed to find a method for interacting with zoomable user interfaces using a two-dimensional input device that is intuitive and supports habit formation. [...] Ideally, zooming and panning should be allowed to take place in parallel. [*23, p.384]

5.4 Physical-motion costs to execute sequences

Physical motion can be hard when the display is very large [14] or very small [49], or for specific user populations such as children [25] and the elderly [60]. Even with desktop-sized displays and physically-apt users tested over limited periods of time, we still found reports of user dissatisfaction with commonly-used mouse interaction such as positioning (Section 5.4.1) and dragging (Section 5.4.2). Furthermore, even low-cost simple motions can accumulate (Section 5.4.3).

5.4.1 Costs in Mouse Position

Fitts' Law models the ease of target selection [16]:

$$MT = a + b\log_2(A/W + 1) \tag{1}$$

where MT is average movement time, A is separation between the two targets, W is target width, and a and b are experiment constants. Fitts' Law implies that the less distance traveled and the bigger the target, the easier it is to reach. Despite its simplicity, applying Fitts' Law to interaction design often involves tradeoffs such as the need to predict user intents to optimize A; trading display capacity for larger W; and causing targets to move by improving A or W.

1. Reducing travel distance needs user intent
The challenges to minimize A are to predict user intent and preserve target locations, since moving targets are hard to select (see below). Predicting user intent can be difficult even for 1D lists. Sears and Shneiderman's Split menu, a menu added with a separate frequency ordering on top of the default alphabetical ordering, showed benefits over alphabetical menus only when frequently selected items were at the bottom of the list [*56].

2. Sacrificing target size for display capacity
Display capacity is a design priority especially for space-limited devices (as in mobile devices) or for large-data displays. We found reports when W was reduced too far and resulted in usability problems.

To accommodate up to six months' of calendar information on PDA-sized screens, Bederson *et al.* used a fisheye technique to create DateLens [*5]. With semantic zooming, users of DateLens can view each day in the calendar at various levels of detail. However, their study found small targets as the most serious usability issue:

> Experienced Pocket PC users often use their fingers to tap on targets in the user interface [...] Even with the stylus, users often invoked incorrect behaviors and actions accidentally when attempting to scroll or make UI selections in DateLens. [*5, p.115]

Indeed, we found similar reports from researchers that tested distortion-based desktop interfaces. Hornbæk and Hertzum examined the usability of Bederson's fisheye menu, where the menu-item font is reduced to increase display capability and reduce traveling distance [4]. Hornbæk and Hertzum found a cost with this optimization,

> the larger the selection height of menu items, the lower the selection time. [*22, p.27]

Li and North looked at dynamic query (DQ) sliders and brushing histograms and found that DQ sliders were more efficient than brushing histograms in range and criteria tasks, possibly because,

> The targets [in the brushing histograms] for clicking were narrower and smaller compared with DQ sliders. Thus it was easier for users to make incorrect or accidental selections [using the brushing histograms]. [*35, p.152]

In addition to ensuring selectable target sizes, Bederson *et al.* proposed another general guideline,

> allow the user to adjust the font size [...] in novel visualizations [*5, p.117]

3. Increasing target size may cause targets to move
Reducing travel distance A and target width W in Fitts' Law (Eqn 1) may cause targets to move. For single isolated targets, McGuffin and Balakrishnan showed that increasing W resulted in faster selection time [*36]. For multiple expanding targets, especially when the targets are closely packed together as in most visualization, there are tradeoffs. For example, distortion-based visualization technique such as fisheye can cause target expansion as the pointer moves closer:

> The problem that occurs when focus-targeting in fisheye views is that targets appear to move in the opposite direction to the motion of the magnifying lens. This means that a focus target will move towards an approaching pointer, and away from a retreating one, making it more difficult to precisely position the focus point relative to the underlying visualized data.
>
> Moving targets are always more difficult to hit—but to make matters worse, the gradually increasing magnification of a fisheye lens makes targets move faster and faster the closer the focus comes to them. In fact, targets move at their highest rate of apparent speed at the exact moment that the pointer nears the target, making it difficult for a user to precisely position the pointer over the target. [18, p.267–268]

Gutwin suggested speed-coupled flattering as a solution to ensure that "relatively static [view] during the acquisition phase of targeting, in order to simplify precise positioning" [18, p.269]. While Gutwin showed the effectiveness of speed-coupled flattering [18], focus lock in Bederson's fisheye menu [4] was found to be a substantial usability problem in Hornbæk and Hertzum's 2007 study [*22]. Focus lock is a mechanism where users,

> move the pointer to the right side of the menu, which locks the focus on the item the cursor is over. Then, when users move the pointer up and down, the focus stays fixed, but individual menu elements can still be selected. The focus region on the right side of the menu gets highlighted to indicate that the menu is in focus lock mode. [4, p.220]

In practice, Hornbæk and Hertzum observed difficulty in its use,

> One reason for the higher selection times [for the fisheye-based menus] appears to be an occasionally ineffective use of the focus lock. [...] This finding accords with the observation that participants sometimes had to leave the focus-lock mode because an item of interest was pushed out of view as a result of the expansion of menu items close to the mouse. Another reason is that participants may choose to enter the focus-lock mode only after having faced difficulties in acquiring the target. The time penalty associated with the focus lock suggests that the best time to make the shift to focus-lock mode was not obvious to participants. [*22, p.23]

In short, actual pixel space does not directly translate to selection space. Even though expanding targets may facilitate coarse navigation, actual target acquisition is more difficult with moving targets. Hornbæk and Hertzum thus concluded that,

> [Since i]tems are moving, the number of pixels in the motor space from which an item must be selected is lower than that of the hierarchical menu. [...] Stable position of menu items are central to the usability of the hierarchical menu.[*22, p.24]

Indeed, McGuffin and Balakrishnan also concluded that,

> The model [of expected benefit in multiple expanding targets] presented indicates that a net reduction in selection time with tiled expanding targets may be possible, however, in practice, the benefit may be negligibly small. [*36, p.419]

5.4.2 Costs in Mouse Drag

Mouse drags are almost ubiquitous on modern interfaces. Nonetheless, reports suggest limiting its use.

In a study to evaluate the PDQ Tree browser, Kumar *et al.* found that the panning action on the detail view, achieved by dragging on the field-of-view in the overview, was not universally welcomed:

> One subject emphasized the need to always fit the overview into one screen only, so that no scrolling of the overview is required. [...]
>
> Another subject suggested that users should be able to click anywhere in the overview and have the field-of-view jump to that position. This would enable fast coarse navigation [without mouse drag]. Fine-tuning could then be accomplished by dragging the field-of-view. [*33, p.119–120]

In PDQ's dynamic query panel, users filter by selecting tree-node attributes from a list using drag-and-drop. While some participants enjoyed the drag-and-drop mechanism, one obviously did not: "Drag-and-drop becomes a drag for experienced users, so drop it!" This sentiment was shared by others:

Some other subjects also echoed the feeling that it might be easier and faster to just replace each drop area with a menu of attributes at that level. [*33, p.120]

Dragging is also used to specify bounding boxes for zooming. In a study of fisheye and zoomable interfaces on PDA-sized devices, Buering *et al.* reasoned that since,

the fisheye interface required far fewer actions but, since task times were similar, it seems that they required more time to execute. Hence we assume that drawing a bounding box is cognitively more demanding than the more direct zooming of the ZUI. [*6, p.834]

5.4.3 Costs in accumulated motions

Even for simple movements such as mouse clicks, repeated actions can accumulate into measurable costs. In evaluating visualization systems to display microarray data, Saraiya *et al.* found that for GeneSpring,

even though users tended to focus on a small number of basic visualization features, usability issues (such as the higher quantity of clicks required to accomplish tasks) reduced their overall insight performance. [*53, p.7]

We found a similar report from Hetzler *et al.* in their In-Spire study:

The ability for a user to track a theme over time, while quite doable, involved too much manipulation and user intervention. This prompted the addition of the Keep Current capability [...]. [*20, p.94]

5.5 Visual-cluttering costs to perceive state

Interaction can cause visual distraction and occlusion that make perception difficult. For example, mouse hovering, despite providing tool-tip guidance, can cause unwanted visual distraction. Granitzer *et al.* noticed that in InfoSky when mouse hovering

near the bottom of the hierarchy, where collections contained many documents, users were confused by the 'jumping around' of document titles. The prototype displayed the titles of those documents which were 'near' to the cursor. [*17, p.130]

In a study of Spotfire, Saraiya *et al.* found that,

Spotfire's parallel coordinates view employs a poorly designed selection mechanism. Selected lines in its parallel coordinates results in an occluding visual highlight that made it very difficult for users to determine which genes were selected. [*53, p.7]

5.6 View-change costs to interpret perception

Interaction often causes change in the visual display. Users therefore need to associate objects in the old view to those in the new based on their expectations. Augmented interaction using machine intelligence may fail to meet these expectations and causes user confusion or even distrust (Section 5.6.1). Interpretation of changes is dependent on implementations. We found reports in three cases: costs in object association between between temporal frames (Section 5.6.2), between simultaneously displayed views (Section 5.6.3), and between local and global objects (Section 5.6.4).

5.6.1 Augmented interaction

To offload cognitive costs in dealing with large data, one design option is to offer automatic data processing. However, users have expectations as to what should happen after an interaction. Failure to meet expectations can lead to confusion or even distrust.

We found such reports in Siirtola and Mäkinen's study on the automatic reorderable matrix [*59]. Their participants rated the subjective satisfaction question, *Overall, the experimental application was easy to use and performed as expected, and did not do unexpected things*, one out of five, possibly because,

When interviewed, four of the participants said that the reason for the rating was the uncomfortable mouse behavior—especially the proportional acceleration was different from what they were used to. The other reason was the heuristic nature of the reordering algorithm. For some participants, it was difficult to accept that the same setting of the slider would sometimes produce a new ordering. [*59, p.46]

Siirtola and Mäkinen also reported confusion in slider effects on matrix ordering based on their algorithm:

Some of the participants commented during the experiment and in the interview that the continuous reordering feels a bit disturbing, although this does not show in the questionnaire results. They felt that a small change in the slider should not result a major change in the matrix ordering. This was the initial reaction, and most of the participants were able to accept this later in the experiment as a characteristic of the user interface. This behavior is due to the nature of the barycenter heuristic and cannot be avoided. [*59, p.47]

5.6.2 Temporal-frame association

When visual objects change with time, users need to keep objects in memory for association. Smooth animation has been proposed as a solution to connect different temporal views in zooming interfaces to preserve object constancy [51] and applied with success (e.g., [31]). Others solutions include minimizing visual changes [38] and providing visual cues such as background grids or landmarks [9, 70].

Our review indicates that animation alone may be inadequate. Siirtola and Mäkinen reported in a study of automatic reorderable matrix,

Based on post-discussion, it was determined that subjects found the feedback to be inadequate during the row and column movements. There should be an outline of row or column visible during the move operation to indicate what is moving. The current implementation updates the matrix view as the mouse moves, but does not indicate the current selection. [*59, p.42–43]

McGuffin *et al.* suggested minimizing changes to allow landmarking:

Features that allow the user to manually position or lock down. the relative placement of nodes would help alleviate the detrimental effects of rearrangement and allow for better landmarking and more consistent displays, thus reducing the time necessary to visually scan for nodes. [*37, p.125]

5.6.3 Multiple-view association

Interfaces may display different views of the same data, either at multiple visual levels as in overview+detail or focus+context interfaces [*34], or in different forms, as in multiform interfaces [50]. Traditionally, designers have used interaction techniques of brushing and linking to coordinate between the different views [64].

For example, in their Snap-together visualization study, North and Shneiderman found that coordination between overview and detail view enabled the use of the overview:

If only the overview information is needed, then naturally coordination is not necessary. But for the important cases where access to details is needed, then coordination is critical. [*42, p.737]

However, we also found reports that suggest linking and brushing alone may be insufficient. In studying zoomable user interfaces, Hornbæk *et al.* reported problems in associating between the overview and the zoomable views, which resulted in slower task time for the interface with an overview because,

switching between the detail and the overview window required mental effort and time moving the mouse. Our data modestly supports this explanation, since the number of transitions between overview and detail window were positively correlated with task completion time. [*23, p.382]

Another consideration is the amount of space needed for interaction:

Coarse [navigation] and [...] resizing the field-of-view box could be difficult at low zoom factors. Subjects commented that the overview was hard to resize. In support of those comments, we note that the overview window used in the experiment occupied 256x192 pixels. When a zoom factor of 20 was reached, the field-of-view box was only 13x10 pixels, which was probably hard for most users to resize and move using the mouse. [*23, p.382]

These observations led to a design recommendation:

To obtain the benefit of easy navigation provided by overviews, designers should use overviews at least one-sixteenth the size of the detail window (in area). For overviews coupled to a detail view less than the size of one screen or for screens on small devices, the overview might need to be larger to support navigation. For systems where much navigation is expected on the overview, for example, in support of monitoring tasks, a larger overview should be provided. For systems with zoom factors over 20 as used in our system, more usability problems will occur when using the overview, and consequently a larger overview will be necessary. [*23, p.384]

Another recommendation is landmarking as semantic linking between views in Crampes *et al.*'s KMap:

There should be different KMaps according to the tasks, but that all KMaps should be semantically linked to maintain users' mental map. [*15, p.222]

5.6.4 Local-global association

When the display space contains little or no information for navigation, the problem of desert fog occurs [29]. Navigation becomes difficult once users have lost association between local and global objects.

In studying zoomable user interfaces, Hornbæk *et al.* observed that,

> At least six subjects repeatedly experienced what has been called desert fog, that is, they zoomed or panned into an area of the map that contained no map objects. [*23, p.381]

Kumar *et al.* observed in their PDQ Tree-browser study that,

> subjects get somewhat disoriented when the level of the tree was changed. This is because the layout algorithm generates a fresh layout whenever the tree structure changes, i.e. whenever more or less levels are requested to be seen. [*33, p.119]

In contrast, simpler navigation led to better interface use. Westerman *et al.*, in a document-browsing study with 2D and 3D display, hypothesized that,

> participants found the process of navigation less effortful in the two-dimensional condition, and therefore were prepared to adopt a more exploratory' strategy. [*65, p.731]

Simple navigation of the hierarchical menu in Hornbæk and Hertzum's study was also found to support better performance:

> Participants perform well with the hierarchical menu [as] it simplifies navigation. With fisheye and overview menus, participants made longer fixations, suggesting increased mental activity, compared with the hierarchical menu. Also, participants' scanpaths were longer with the multifocus and overview menus, indicating more visual search. Reasons for this could include: (a) the need with nonhierarchical menus to determine or remember which part of the menu structure one is currently in [...] [*22, p.25]

In addition to navigation simplicity, current focus can be used as landmarks to aid navigation in temporal view changes:

> It is felt that this [navigation] problem can be significantly alleviated by retaining the same current focus. For example, if the user asks to see the University level while the state Florida is near the center, the new view should be initialized to show universities within Florida. [*33, p.119]

Landmarks may also be useful aids in multiple spatial views:

> a detail-only interface could include cues about the current zoom factor, cues about the current position in the information space, and aids for avoiding desert fog. If such cues are integrated into the detail view, the mental and motor effort associated with shifting to the overview might be reduced, as would the screen real estate lost due to the presence of an overview. [*23, p.384]

5.7 State-change costs to evaluate interpretation

Data exploration often involves comparing between previously viewed data projections. Refinding has been studied in the context of web search (e.g., [63]). Lack of refinding support may inhibit exploration.

In our review, we found reports to suggest that in general, overview+detail interfaces are better than zooming or fisheye interfaces in supporting refinding, since participants explored more when using overview+detail interfaces. One such report is Hornbæk and Frokjær's document-reading study, where,

> The overview pane supports jumping directly to targets; it helps returning to previously visited parts of the document; and it invites and supports further explorations. Subjects using the fisheye interface depend extensively on the algorithm that determines which sections to collapse initially, even though subjects do not trust this algorithm. [*24, p.142–143]

Plumlee and Ware also found different interface-use strategies between the zooming and the multiple-window (WM) interfaces because of their differing support on refinding:

> The results show that subjects made dramatically more visits with the eye between windows than they made with the zooming interface. In addition, subjects made more eye-visits (in the multiple-window condition) than the model predicted would be necessary to achieve perfect performance.
>
> This suggests a kind of satisficing strategy with visual working memory as a limited-capacity, cognitively critical resource. When visits are cheap in time and cognitive effort, for example when they are made via eye movements, they are made frequently and people make a separate eye movement to check each component of the two patterns they are comparing. Thus their visual WM capacity relating to the task is effectively one. However, when visits are expensive in time and cognitive effort, for example when zooming is required, subjects attempt to load more information into visual WM and they also quit the task after fewer visits, which results in many more errors. [*47, p.205]

When the interface does not support refinding well, users may find it difficult to return to a state. Yi *et al.* reported this problem in their study of Dust & Magnet (DnM):

> Because DnM allows users a high degree of freedom in adjusting and manipulating dimensions, it is challenging to explicitly document certain clustering schemes. Yet, if the user keeps applying the 'Center Dust' feature, it is possible to regenerate similar clustering. However, the dimensions of DnM can move all over the main view, so DnM does not have the reproducibility that other visualization techniques have. [...] [T]he problem of reproducibility is really a trade-off with the heightened ability for users to freely explore and manipulate the data with a high degree of freedom. [*68, p.255]

6 DISCUSSION: NARROWING THE GULFS

Interaction is vital to the success of modern visualization. Interaction has been found to benefit users by allowing multiple data views, for example, in controlling rotation [26] and object transformations [57]. Also, interaction can be fun for users, as seen in the use of Vizster social situations [19]. However, even though visualization evaluations seldom focus on interaction, interaction costs do impact usability. Indeed, in discussing their study results of testing five visualization tools for microarray analyses, Saraiya *et al.* commented that,

> The design of interaction mechanisms in visualization is critically important. Usability can outweigh the choice of visual representation. [*53, p.7]

In our review, we noticed a number of interaction costs that impact usability. In some cases, researchers attributed those interaction costs to inferior results obtained in their studies. We further discuss interaction design considerations based on our framework (Fig 1) as suggestions to narrow the gulfs of execution (Section 6.1) and evaluation (Section 6.2). We also raise the issue of whether visualizations should also address the gulf of goal formation (Section 6.3).

6.1 Narrowing the Gulf of Execution: Less is more

Similarly to visual cluttering, complex interaction is detrimental. Interfaces that provide too many choices, both in the form of input modes (Section 5.3) and as interface options (Section 5.1.2) may deter effective use by inducing unnecessary cognitive loads. In executing physical sequences, reports show users were annoyed by repetitive drag-n-drop movements in PDQ (Section 5.4.2) and in In-Spire (Section 5.4.3), were less effective in insight generation using GeneSpring (Section 5.4.3), made more errors with the Brushing histogram (Section 5.4.3), and produced worse time and accuracy performances with the Drill-Down method over the Distortion method in tree-node searches (Section 5.4.3).

It is unclear if training can mitigate these costs. Buering *et al.* believed so (Section 5.3.3):

> Seven subjects mentioned that they had problems with the sliding technique of the ZUI. We assume that this was mainly caused by the users' unfamiliarity with this kind of panning. [*6, p.834–835]

We also wondered if insufficient experience with novel and complex interfaces may have influenced interface choice where participants chose simpler but suboptimal interfaces (Section 5.1.2):

> Switching from a multiple-[level] mode to a single [level] can thus provide an easily perceived short-term benefit of lower cognitive load, despite potentially increasing the total time required to complete the task. Our study training for the users required them to demonstrate proficiency in the use of all four interfaces, as is usual in single-session laboratory settings. We conjecture that users trained to demonstrate proficiency in a multiple-[level] interface may still not have internalized confidence in its use: that is, may not have adequately understood the longer-term cost of these short-term choices. [*34, p.1285]

Nonetheless, we believe designs should aim for a small set of simple and predictable interaction, even at the cost of reduced user control. Simpler interaction can reduce errors, as seen in Li and North's study,

> simpler interactions of DQ [dynamic-query] sliders (only the slider thumbs were interactive) [as it] helped avoid mistakes. In contrast, all bars in the brushing histograms were interactive. [...] Thus it was easier for users to make incorrect or accidental selections [*35, p.152]

Simpler interaction can lead to better user performance, as reported by Shi *et al.* in their study with a tree-map like distortion technique for visual search tasks,

> The main reason that users perform better at the distortion techniques is due to the fact that in the drill-down approach the user has to drill-down and roll-up during several iteration[s] until the node is found. [*58, p.88]

Designers can replace sequences of actions by a single action to simplify interaction. The tradeoff is reduced interaction flexibility offered by intermediate steps. Our reviewed studies offer two examples: Hetzler *et al.* added a Keep Current capability in In-Spire to replace the more interactive approach to keep track of document themes over time (Section 5.4.3), and filtering with the DQ sliders instead of the more flexible brushing histograms. Designers can also use interaction techniques that are less physically demanding, such as using mouse clicks over drag-and-drops as suggested by participants in Kumar *et al.*'s PDQ Tree browser study (Section 5.4.2), or provide keyboard shortcuts such as Ctrl+S to avoid the more costly menu selections.

6.2 Narrowing the Gulf of Evaluation

To evaluate interaction outcomes, users need to first understand visual changes caused by interaction. While we believe interface intelligence can offer benefits especially in large-data exploration, we believe users should be kept in the control loop (Section 6.2.1). As for object associations, our reports suggest that commonly-used solutions are effective but may be inadequate (Section 6.2.2), and visualization should provide better support for reflection in analysis (Section 6.2.3).

6.2.1 Steerable and predicable augmentation

It is well known that interface intelligence can result in user confusion or even mistrust. Given the large quantity of data under visual analysis, users should be able to benefit from machine intelligence. Perhaps the solution is to emphasize augmentation rather than automation, allowing the human operator to understand and diagnose unexpected system behaviours, and to modify algorithm parameters [41].

6.2.2 Commonly-used solutions inadequate

Reports collected in this review indicate that while standard solutions for object associations such as animation, and linking and brushing are effective, users may need additional help to understand view changes. Lack of training may account for some of the troubles. For example, Saraiya *et al.* reasoned that the lack of multiple-view use was due to insufficient participant experience:

> Since the participants were novice users, they were also not experienced with performing data analysis on multiple views simultaneously.[...]
>
> Also, giving participants a longer training period on brushing and linking might have been helpful for them to better utilize the reverse brushing direction in which the parallel coordinate view is used to query the graph view. [*54, p.231]

Landmarking has been proposed as a visual aid (Sections 5.6.2–4). An open research question is to identify the number and type of visual or interaction aids required for object associations.

6.2.3 Support reflection in analysis

Our reports prompted us to consider the need for reflective cognition support in visualizations. While well-designed interaction supports experiential cognition as users effortlessly respond to incoming information without conscious reflection [40, p.22–31], reflective cognition is needed in analysis as users need to compare and contrast effects of different hypotheses [40, p.205]. Section 5.7 lists reports where users explored more with interfaces that supported refinding. Supporting refinding by allowing users to save visualization state is therefore not an implementation detail but an important design consideration.

6.3 Thinking about the Gulf of Goal Formation

Interface experimenters tend to provide study tasks and data for the participants. In real life, however, users have to come up with their own tasks and data focus with real goals. Even though the interface may be well designed at any point, users may not be able to use the visualization if they fail to form analysis questions. Large-data analysis is in itself a difficult task that required nontrivial amount of training and experience. Visualization perhaps should not be expected guide users who do not know what they are looking for, or are unable to articulate a question. Yi *et al.*, in evaluating the Dust-and-Magnet (DnM) interface, were ambivalent about adopting this role for visualization,

> One might question whether DnM can be effective for people who do not know how to articulate a question. This is a fair question because DnM does not supply any systematic approaches to producing this query initially. Therefore, the user should pose the question in advance in order to find answers using DnM effectively. Conversely, as shown in the user evaluation, the fact that DnM is easy and interesting to use might encourage users to explore data sets more vigorously, which smoothly lead them to pose proper questions. [*68, p.255]

On the other hand, users did get lost without guidance. In their evaluation of the Integrated Thesaurus-Results Browser, Sutcliffe *et al.* recommended system guidance:

> The visualization did appear to be comprehensible to users; however, it was hindered by lack of guidance on search strategies and possibly by the manipulations we provided for exploring the thesaurus and results browser visualizations. Basing visualization design on user tasks and data models has been advocated by others and demonstrated in successful products; however, in more complex tasks further research on visualization design methods that integrate active system guidance with visual browser and exploration tools is required. [*62, p.760]

Our framework therefore suggest a need to consider the gulf of goal formation, or if visualization should guide users in data explorations.

7 Limitations of Study

While we aimed to provide an objective and in-depth review to understand impacts of interaction costs on interface usability, our framework is limited by our venue scope and reported costs—we can only discuss techniques that had been studied and costs reported. Unlike study-statistics results, reporting on interface interactivity has not been standardized in publications. Our reports are therefore possibly subjective as they are observations and conjectures by paper authors.

Also, reasons behind the lack of reports are impossible to discern. Researcher may not report interaction costs for at least three reasons: first, study participants did not exhibit significant ill effects from interface interaction; second, interaction costs observed were considered to be due to study participants' inexperience with the interface and could be overcome with use; third, the study did not measure interaction costs, either with objective measurements or subjective observations.

Given these limitations, we took a qualitative route. A more quantitative approach, for example to assess the prevalence and severity of interactions costs, remains an open problem. Consequently, we can only conjecture rather than derive claims from our analysis, and our framework of seven costs is necessarily incomplete and perhaps non-representative. Also, our framework is only one interpretation of reports collected. With these limitations and the broad scope of our review, we can only provide broad general guidelines in Section 6.

8 Conclusion

We performed a systematic review on 484 InfoVis, IVS, IJHCS, and TOCHI publications and isolated 32 that reported on interaction uses. From these publications, we identified 61 interaction-related usability reports and grouped them into seven main costs inspired by Norman's Seven Stages of Action [39]. In addition to the Norman's gulfs of execution and evaluation, we proposed adding a gulf of goal formation to cover decision costs. Our framework is an initial step to study interaction costs.

Results of our review suggest a need to focus on interaction costs in visualization evaluations. We were surprised that even in journal publications where page constraints are less severe than in conference proceedings, less than 30% of user studies of interactive interface mentioned interaction. Even though cognitive costs seem to be more prevalent in modern visualization, we found that 30% of all reports are motion costs, perhaps since they are more observable. Our framework suggests a need to better identify cognitive costs in interaction. For example, our surveyed papers only identified costs in data and system option selections to form goals (Section 5.1), which are insufficient to capture the essence of sensemaking in visual analysis. Our framework therefore suggests a need to diversify from our traditional focus on visual encoding (cost 5) to cover the entire action cycle.

In terms of study methodologies, given the challenges in evaluating visualizations (e.g., [46, 21]), we believe recording observations in laboratory or field studies is a good starting point, as by collecting statements on interaction during interface use, we can begin to identify possible factors in interaction before we can develop more objective usability metrics, and perhaps new technologies, to capture and quantify these costs. Only then can we truly quantify the prevalence and severity of interaction costs on interface usability.

References

[1] *J. Abello *et al.* ASK-Graph View: A Large Scale Graph Visualization System. *IEEE TVCG*, 12(5):669–676, 2006.

[2] C. Ahlberg and B. Shneiderman. Visual Information Seeking: Tight Coupling for Dynamic Query Filters with Starfield Displays. In *Proc ACM CHI*, pages 313–317, 1994.

[3] M. Beaudouin-Lafon. Designing Interaction, Not Interfaces. In *Proc ACM AVI*, pages 15–22, 2004.

[4] B. Bederson. Fisheye Menus. In *Proc ACM UIST'00*, pages 217–226, 2000.

[5] *B. Bederson *et al.* DateLens: A Fisheye Calendar Interface for PDAs. *ACM TOCHI*, 11(1):90–119, March 2004.

[6] *T. Buring *et al.* User Interaction With Scatterplots on Small Screens—A Comparative Evaluation of Geometric-Semantic Zoom and Fisheye Distortion. *IEEE TVCG*, 12(5):829–836, 2006.

[7] S.K. Card *et al.* A Morphological Analysis of the Design Space of Input Devices. *ACM TOIS*, 9(2):99–122, 1991.

[8] S.K. Card *et al.* *Readings in Information Visualization: Using Vision to Think*. Morgan Kaufmann, 1999.

[9] M.S.T. Carpendale *et al.* Three-dimensional pliable surfaces: for the Effective Presentation of Visual Information. In *ACM UIST)*, pages 217–226, 1995.

[10] C. Chen and Y. Yu. Empirical Studies of Information Visualization: A Meta-Analysis. *Intl. J. of Human-Computer Studies (IJHCS)*, 53:851–866, 2000.

[11] E.H. Chi and J.T. Riedl. An Operator Interaction Framework for Visualization Systems. In *Proc IEEE InfoVis*, pages 63–70, 1998.

[12] E.H. Chi *et al.* Using Information Scent to Model User Information Needs and Actions and the Web. In *Proc ACM CHI*, pages 490–497, 2001.

[13] *W. Chung. Studying Information Seeking on the Non-English Web: An Experiment on a Spanish Business Web Portal. *IJHCS*, 64(9):811–829, 2006.

[14] M. Collomb *et al.* Improving Drag-and-Drop on Wall-Size Display. In *Proc GI*, pages 25–32, 2005.

[15] *M. Crampes *et al.* Concept Maps for Designing Adaptive Knowledge Maps. *Information Visualization*, 5(3):192–224, 2006.

[16] P. Fitts. The Information Capacity of the Human Motor System in Controlling the Amplitude of Movement. *J. Expt. Psych.*, 47:381–391, 1954.

[17] *M. Granitzer *et al.* Evaluating a System for Interactive Exploration of Large, Hierarchically Structured Document Repositories. In *Proc IEEE InfoVis*, pages 127–134, 2004.

[18] C. Gutwin. Improving Focus Targeting in Interactive Fisheye Views. In *Proc ACM CHI*, pages 267–274, 2002.

[19] J. Heer and D. Boyd. Vizster: Visualizing Online Social Networks. In *Proc IEEE InfoVis*, pages 33–40, 2005.

[20] *E.G. Hetzler *et al.* Turning the Bucket of Text into a Pipe. In *Proc IEEE InfoVis*, pages 89–94, 2005.

[21] K. Hornbæk. Current Practice in Measuring Usability: Challenges to Usability Studies and Research. *IJHCS*, 64:79–102, 2006.

[22] *K. Hornbæk and M. Hertzum. Untangling the Usability of Fisheye Menus. *ACM TOCHI*, 14(2), 2007.

[23] *K. Hornbæk *et al.* Navigation Patterns and Usability of Zoomable User Interfaces with and without an Overview. *ACM TOCHI*, 9(4):362–389, 2002.

[24] *K. Hornbæk *et al.* Reading Patterns and Usability in Visualization of Electronic Documents. *ACM TOCHI*, 10(2):119–149, 2003.

[25] J.P. Hourcade *et al.* Differences in Pointing Task Performance between Preschool Children and Adults using Mice. *ACM TOCHI*, 11(4):357–386, 2004.

[26] G.S. Hubona *et al.* The Effects of Motion and Stereopsis on Three-dimensional Visualization. *IJHCS*, 47(5):609–627, 1997.

[27] R.J.K. Jacob *et al.* Integrality and Separability of Input Devices. *ACM TOCHI*, 1(1):3–26, 1994.

[28] T.J. Jankun-Kelly *et al.* A Model and Framework for Visualization Exploration. *IEEE TVCG*, 13(2):357–369, 2007.

[29] S. Jul and G.W. Furnas. Critical Zones in Desert Fog: Aids to Multiscale Navigation. In *Proc ACM UIST*, pages 97–106, 1998.

[30] *H. Kang *et al.* NetLens: Iterative Exploration of Content-Action Network Data. *Information Visualization*, 6(1):18–31, 2007.

[31] C. Klein and B.B. Bederson. Benefits of Animated Scrolling. In *Extended Abstracts of CHI*, pages 1965–1968, 2005.

[32] *A. Kobsa. An Empirical Comparison of Three Commerical Information Visualization Systems. In *Proc IEEE InfoVis*, pages 123–130, 2001.

[33] *H.P. Kumar *et al.* Browsing Hierarchical Data with Multi-level Dynamic Queries and Pruning. *IJHCS*, 46(1):103–124, 1997.

[34] *H. Lam *et al.* Overview Use in Multiple Visual Information Resolution Interfaces. *IEEE TVCG*, 13(6):1278–1283, 2007.

[35] *Q. Li and C. North. Empirical Comparison of Dynamic Query Sliders And Brushing Histograms. In *Proc IEEE InfoVis*, pages 147–153, 2003.

[36] *M.J. McGuffin and R. Balakrishnan. Fitts' Law and Expanding Targets: Experimental Studies and Designs for User Interfaces. *ACM TOCHI*, 12(4):388–422, 2005.

[37] *M.J. McGuffin *et al.* Expand-ahead: A Space-Filling Strategy for Browsing Trees. In *Proc IEEE InfoVis*, pages 119–126, 2004.

[38] K. Misue *et al.* Layout Adjustment and the Mental Map. *J. of Visual Languages and Computing*, 6(2):183–210, 1995.

[39] D.A. Norman. *The Design of Everyday Things*. Doubleday/Currency, 1990.

[40] D.A. Norman. *Things that Make Us Smart*. Basic Books, 1993.

[41] D.A. Norman. *The Design of Future Things*. Basic Books, 2007.

[42] *C. North and B. Shneiderman. Snap-Together Visualization: Can Users Construct and Operate Coordinated Visualizations. *IJHCS*, 53(5):715–739, 2000.

[43] C. Olson and E.H. Chi. ScentTrails: Integrating Browsing and Search on the Web. *ACM TOCHI*, 10(3):177–197, 2003.

[44] P. Pirolli and S.K. Card. Information Foraging. *Psych. Review*, 106:643–675, 1999.

[45] P. Pirolli and S.K. Card. Sensemaking Processes of Intelligence Analysis and Possible Leverage Points as Identified Through Cognitive Task Analysis. In *Proc Intl. Conf. on Intelligence Analysis*, 2005.

[46] C. Plaisant. The Challenge of Information Visualization Evaluation. In *Proc ACM AVI*, pages 109–116, 2004.

[47] *M. Plumlee and C. Ware. Zooming versus Multiple Window Interfaces: Cognitive Costs of Visual Comparisons. *TOCHI*, 13(2):179–209, 2006.

[48] R. Rao and S.K. Card. Table Lens: Merging Graphical and Symbolic Representations in An Interactive Focus Plus Context Visualization for Tabular Information. In *Proc ACM CHI*, pages 318–322, 1994.

[49] X. Ren and S. Moriya. Improving Selection Performance on Pen-Based Systems: A Study of Pen-Based Interaction for Selection Tasks. *ACM TOCHI*, 7(3):384–416, 2000.

[50] J.C. Roberts. Multiple-View and Multiform Visualization. In *Visual Data Exploration and Analysis VII. Proc SPIE 3960*, pages 176–185, 2000.

[51] G.G. Robertson *et al.* The Cognitive Coprocessor Architecture for Interactive User Interfaces. In *Proc ACM UIST*, pages 10–18, 1989.

[52] D.M. Russell *et al.* The Cost Structure of Sensemaking. In *Proc ACM CHI*, pages 269–276, 1993.

[53] *P. Saraiya *et al.* An Evaluation of Microarray Visualization Tools for Biological Insight. In *Proc IEEE InfoVis*, pages 1–8, 2004.

[54] *P. Saraiya *et al.* Visualization of Graphs with Associated Timeseries Data. In *Proc IEEE InfoVis*, pages 225–232, 2005.

[55] B. Schwartz. *The Paradox of Choice: Why More Is Less*. HarperCollins, 2004.

[56] *A. Sears and B. Shneiderman. Split Menus: Effectively Using Selection Frequency to Organize Menus. *ACM TOCHI*, 1(1):27–51, 1994.

[57] K. Sedig *et al.* Application of Information Visualization Techniques to the Design of a Mathematical Mindtool: A Usability Study. *Information Visualization*, 2(3):142–159, 2003.

[58] *K. Shi *et al.* An Evaluation of Content Browsing Techniques for Hierarchical Space-Filling Visualizations. In *Proc IEEE InfoVis*, pages 81–88, 2005.

[59] *H. Siirtola and E. Mäkinen. Constructing and Reconstructing the Reorderable Matrix. *Information Visualization*, 4(1):32–48, 2005.

[60] M.W. Smith *et al.* Aging, Motor Control, and the Performance of Computer Mouse Tasks. *Human Factors*, 41(3):589–596, 1999.

[61] R. Spence. A framework for Navigation. *IJHCS*, 51(5):919–945, 1999.

[62] *A.G. Sutcliffe *et al.* Evaluating the Effectiveness of Visual User Interfaces for Information Retrieval. *IJHCS*, 53(5):741–763, 2000.

[63] J. Teevan. The Re:Search Engine: Simultaneous Support for Finding and Re-Finding. In *Proc ACM UIST*, pages 23–32, 2007.

[64] M.O. Ward. XmdvTool: Integrating Multiple Methods for Visualizing Multivariate Data. In *Proc IEEE Vis*, pages 326–333, 1994.

[65] *S.J. Westerman *et al.* Browsing a Document Collection Represented in Two- and Three-Dimensional Virtual Information Space. *IJHCS*, 62(6):713–736, 2005.

[66] W. Willett *et al.* Scented Widgets: Improving Navigation Cues with Embedded Visualizations. *IEEE TVCG*, 13(6):1129–1136, 2007.

[67] T. Winograd and C.F. Flores. *Understanding Computers and Cognition*. Ablex Publishing Corp., 1986.

[68] *J.S. Yi *et al.* Dust & Magnet: Multivariate Information Visualization Using a Magnet Metaphor. *Information Visualization*, 4(4):239–256, 2005.

[69] J.S. Yi *et al.* Toward a Deeper Understanding of the Role of Interaction in Information Visualization. In *Proc IEEE InfoVis*, pages 1224–1231, 2007.

[70] A. Zanella *et al.* On the Effects of Viewing Cues in Comprehending Distortions. In *ACM CHI*, pages 119–128, 2002.

* denotes publications with listed reports

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.

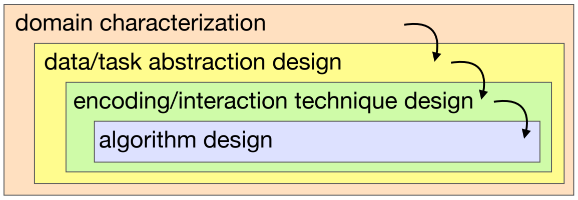


Figure 1: Original depiction of the nested design model [43], with arrows indicating the cascading effects of decisions made at higher levels.

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

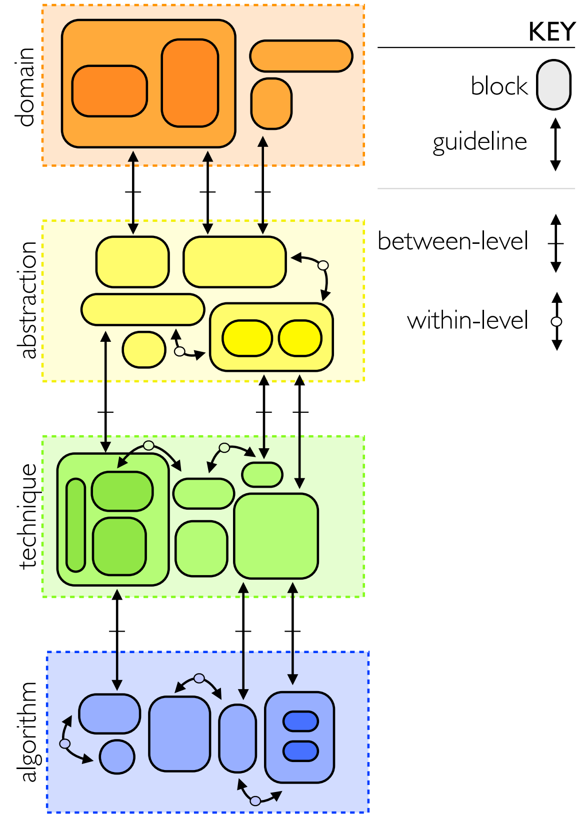


Figure 2: The NBGM has *blocks* that are the outcomes of the design process within a level, represented by individual shapes within each level, and *guidelines* for making choices between these blocks, represented by arrows. Between-level guidelines map blocks at adjacent levels, and within-level guidelines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

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

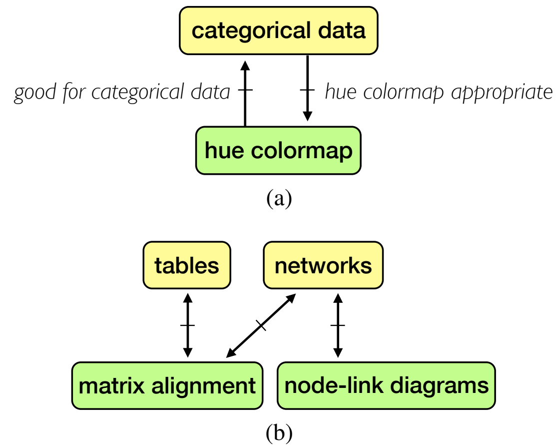


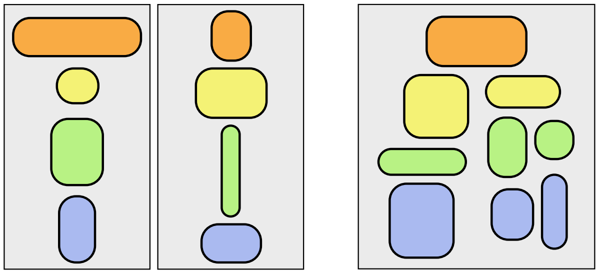
Figure 3: a) Example of a simple pairwise between-level guideline that can be inverted to go upwards or downwards. b) Guidelines can also be many-to-one or many-to-many; these more complex guidelines are not trivially invertible. The example shows a many-to-many between-level guideline.

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.



(a) (b)

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.

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 actionability 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 emphasis is on justifying choices from the set of existing blocks by showing that blocks match up properly with each other across levels. 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 example, 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 evaluation paper, Heer and Bostock provide between-level guidelines on how to choose visual encoding techniques appropriate for different 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 contributing any guidelines about its use. The NBGM's emphasis on both blocks and guidelines suggests that this choice is a lost opportunity: 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.

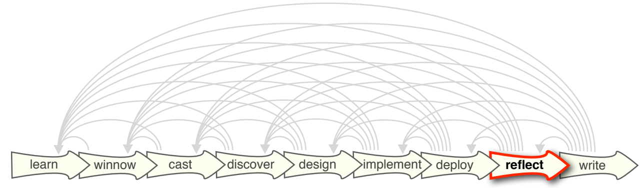


Figure 5: The nine-stage framework for conducting design studies [57]. The eighth stage is to reflect on lessons learned by confirming, refining, rejecting, or proposing guidelines. The NBGM clarifies the scope of this stage.

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 designers create new designs.

In the next section we use the NBGM to describe and evaluate previously 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 design outcomes at each level leads to a much richer analysis. For example, even though most paper authors do articulate new guidelines 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 precondition 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 instead focuses on classifying the *actions* of a designer into separable stages. Figure 5 shows this framework, with the penultimate *reflect* stage highlighted. We claim that at this stage a visualization researcher should reflect on how lessons learned in a particular project relate to the larger research area of visualization by confirming, refining, rejecting, or proposing new guidelines. Using 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-

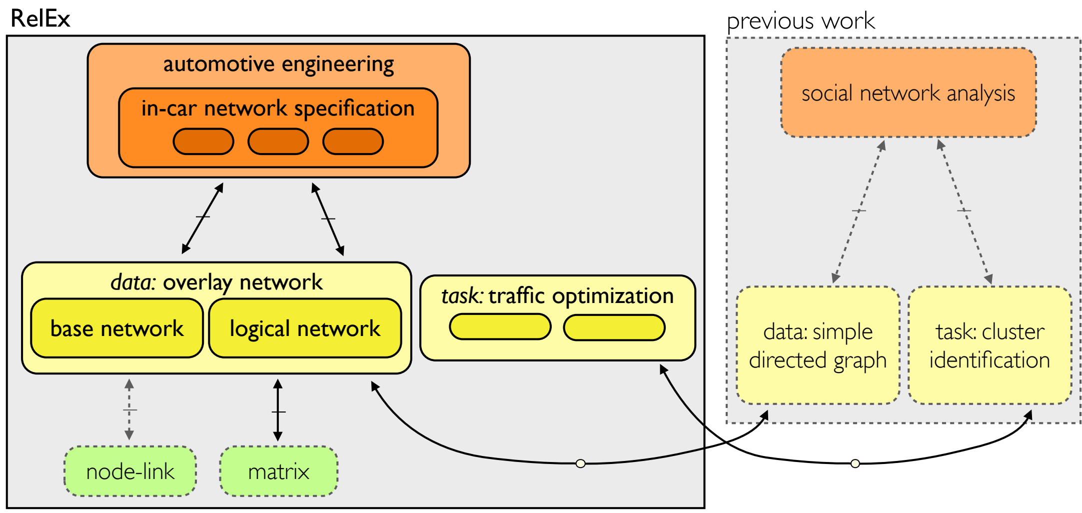


Figure 6: NBGM analysis of RelEx design study subset [54]. Continuous lines mark blocks and guidelines for which active contributions were made, and dashed outlines mean that existing blocks or guidelines were simply used. Nesting reflects hierarchical relations between blocks, and block stacks are enclosed 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.

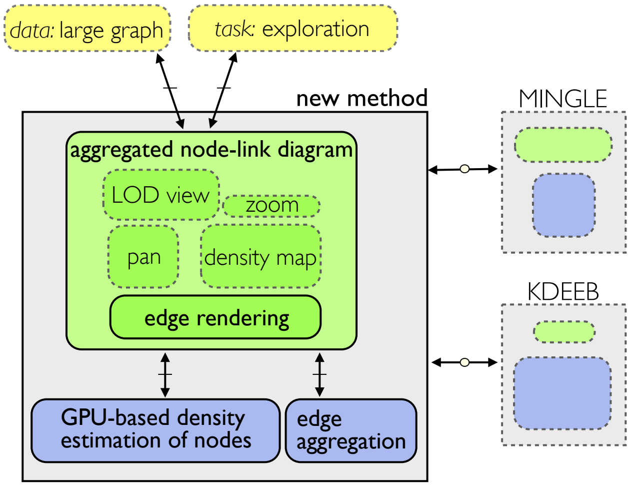


Figure 7: NBGM analysis of a new, LOD technique for rendering large graphs interactively on the GPU [71]. The proposed method is a stack of algorithm and technique blocks, where the technique block is a combination of both previously proposed and new blocks. The new method is compared to several other systems, establishing new within-level guidelines.

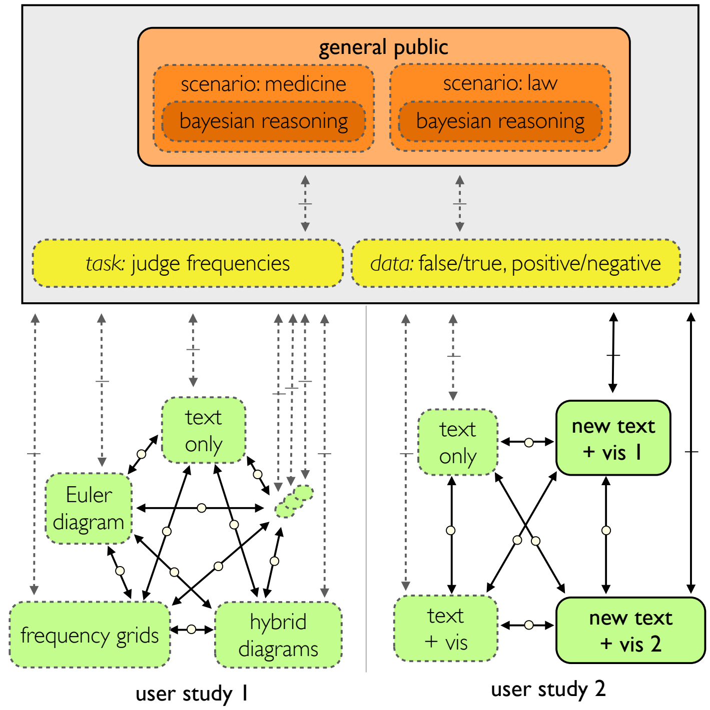


Figure 8: NBGM analysis of a paper evaluating visualization for Bayesian reasoning [41]. In a first study, the authors evaluate seven previously proposed encoding technique blocks, four of which are explicitly shown in the diagram. In a second study, they refine two previously proposed blocks and compared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

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 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

We greatly appreciate feedback from the infovis groups at Utah and UBC that helped in clarifying the ideas in this paper: Alex Bigelow, Matt Brehmer, Jessica Dawson, Joel Ferstay, Stephen Ingram, and Sean McKenna. This work was funded in part by an NSERC Strategic Grant, a Microsoft Research Faculty Fellowship, and NSF grant IIS-1212806.

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.

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. 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

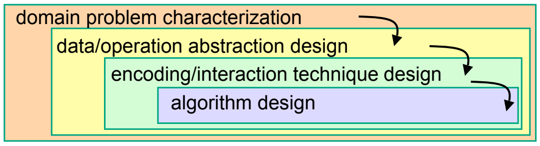


Fig. 1. Our model of visualization creation has four nested layers.

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-

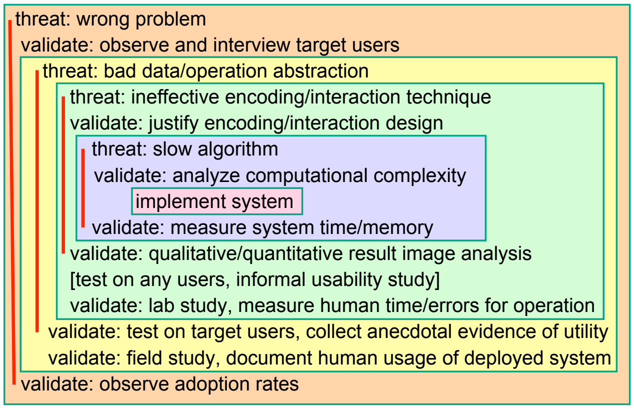


Fig. 2. Threats and validation in the nested model. Downstream levels are distinguished from upstream ones with containment and color, as in Figure 1. Many threats at the outer levels require downstream validation, which cannot be carried out until the inner levels within them are addressed, as shown 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.

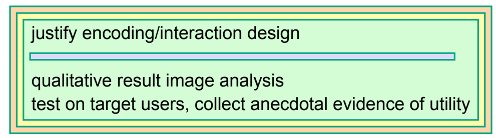


Fig. 3. Genealogical graphs [27] validation levels.

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.

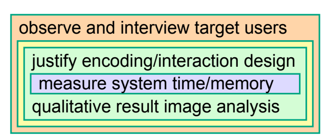


Fig. 4. MatrixExplorer [17] validation methods.

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.

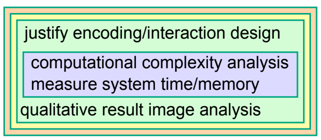


Fig. 5. Flow Map [32] validation methods.

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.

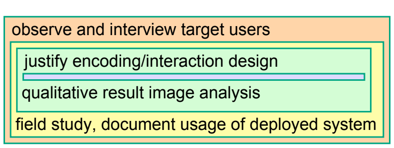


Fig. 6. LiveRAC [32] validation methods.

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.

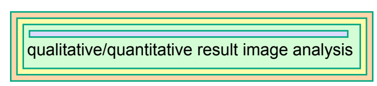


Fig. 7. LinLog [31] validation methods.

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.

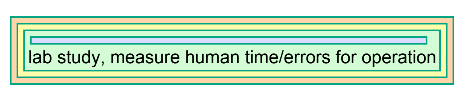


Fig. 8. Lab studies as a validation method.

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. Fisher, 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.