The Design Activity Framework: Investigating the Data Visualization Design Process

a dissertation presented
by
Sean Patrick McKenna
to
The School of Computing

in partial fulfillment of the requirements
for the degree of
Doctor of Philosophy
in the subject of
Computing

University of Utah
Salt Lake City, Utah
June 2017

©2017 – Sean Patrick McKenna
all rights reserved.

Advisor: Miriah Meyer Sean Patrick McKenna

The Design Activity Framework: Investigating the Data Visualization Design Process

Abstract

This dissertation establishes a new visualization design process model devised to guide visualization designers in building more effective and useful visualization systems and tools. The novelty of this framework includes its flexibility for iteration, actionability for guiding visualization designers with concrete steps, concise yet methodical definitions, and connections to other visualization design models commonly used in the field of data visualization. In summary, the design activity framework breaks down the visualization design process into a series of four design activities: *understand*, *ideate*, *make*, and *deploy*. For each activity, the framework prescribes a descriptive motivation, list of design methods, and expected visualization artifacts.

To elucidate the framework, two case studies for visualization design illustrate these concepts, methods, and artifacts in real-world projects in the field of cybersecurity. For example, these projects employ user-centered design methods, such as personas and data sketches, which emphasize our teams' motivations and visualization artifacts with respect to the design activity framework. These case studies also serve as examples for novice visualization designers, and we hypothesized that the framework could serve as a pedagogical tool for teaching and guiding novices through their own design process to create a visualization tool.

To externally evaluate the efficacy of this framework, we created worksheets for each design activity, outlining a series of concrete, tangible steps for novices. In order to validate the design worksheets, we conducted 13 student observations over the course of two months, received 32 online survey responses, and performed a qualitative analysis of 11 in-depth interviews. Students found the worksheets both useful and effective for framing the visualization design process. Next, by applying the design activity framework to technique-driven and evaluation-based research projects, we brainstormed possible extensions to the design model. Lastly, we examined implications of the design activity framework and present future work in this space. The visualization community is challenged to consider on how to more effectively describe, capture, and communicate the complex, iterative nature of data visualization design throughout research, design, development, and deployment of visualization systems and tools.

Contents

Acknowledgments

This dissertation would not have been possible without the guidance, support, and critique of my supervisory committee: Miriah, Alexander, Tamara, James, and Nathalie. Miriah has been a huge guiding light throughout my PhD, keeping me focused on the big picture and teaching me how to become a visualization designer, researcher, and practitioner. It has been an honor to work with her over these past five years, and it has been a pleasure to learn and grow under her mentorship. Alexander has provided similar support and mentorship, and it has been a pleasure to work with him as the Visualization Design Lab was established and for the data visualization course. Tamara played a big role in guiding me through the vast topic of research ethics and human research, and her course on human-centered security helped motivate the data sketches design method that I conducted. James taught me, through practice, observation, and discussion, a lot about design and making better software systems for humans to use, and I enjoyed our project together as we redesigned a visualization tool and formulated the groundwork for the design activity framework. Lastly, but certainly not least, Nathalie and I worked together for years, from visualizing set-typed data to establishing requirements for a storytelling authoring tool, and her inspiration, enthusiasm, and brilliance helped us describe and evaluate concepts for visual narrative flow.

I would like to thank my family and friends for all their support throughout the tumultuous time as I pursued a graduate degree. My parents raised me to be an open-minded and driven individual, for which I am very thankful, and my parents and brother have always been a solid rock for me to depend on over time, emotionally and financially, everything from lending an open ear to helping me move. In my extended family, my grandparents, aunts, uncles, and cousins have always been a sounding board of intelligence and wisdom as well as further support. My close friends, Alyssa Lewis, Evan Foster, and Lora Orme, have all provided me with additional support, comfort, and much needed laughter. I am blessed to have had all these wonderful people in my life. I have also been graced with some amazing animals during my graduate career — Evey, my first cat, has provided me with warmth and comfort over the years while also serving as a living alarm clock; my betta fish, Mulberry, adds colorful delight to my day; and three ferrets, Noodle, Flopsy, and Lucy, have been a welcome distraction from the everyday stresses of life.

The pursuit of my graduate career would not have been possible without the dedication of many individuals to my educational and personal experiences. Back in middle school, I was a very quiet and solitary child, and through the concern of my teachers and the help of Christian Ammon, I was able to come out of my shell and embrace my passions outside the classroom, such as tabletop gaming and technical theater, and over time Christian grew from a mentor into a colleague and friend.

These experiences in theater led me to make many other great friends and mentors, from Thomas Bourke to LuAnne Richardt to Mark Weiss to Larry Kirchgaessner. I owe all of them so much for helping hone my attention to detail and troubleshooting skills to be what they are today. For my education, I have to thank my teachers, and William Heeren was one of the most dedicated teachers I have ever had. Despite only taking one chemistry class from him, he mentored me throughout high school in Science Olympiad and SMART Team, and it is through these experiences that my love and passion for computer science grew. The outstanding professors at Cornell College helped further that education in computer science, physics, and chemistry, with the academic excellence and rigor instilled to me by many professors, such as Tony deLaubenfels, Leon Tabak, Lyle Lichty, Kara Beauchamp, Derin Sherman, and Craig Teague. I would not be where I am today without all of these wonderful educators and mentors, and for that I am eternally grateful.

I would like to personally thank the following colleagues of the University of Utah Visualization Design Lab for all their help and feedback on this dissertation: Ethan Kerzner, P. Samuel Quinan, Alex Bigelow, Nina McCurdy, Carolina Nobre, Jimmy Moore, T Cameron Waller, and Kristen Zygmunt. I also thank Jonzy and Dan Bowden for their cooperation and feedback on the visualization data sketches method. The staff members at MIT Lincoln Laboratory played a significant role in the success of the BubbleNet project from design to deployment, which was made possible through the guidance of Diane Staheli. For the redesign project, Dominika Mazur, Matthew Parkin, and James Agutter all contributed significantly to this project as well. On the s-CorrPlot tool, this project would not have been possible without the initial tool written by Samuel Gerber, as well as the additional feedback from Kristi Potter, Wei-Chao Huang, Tom Fletcher, and Kristen Zygmunt. In the narrative flow project, this evaluation would not have been possible without the guiding support of Bongshin Lee and Jeremy Boy, as well as feedback provided from researchers and interns at Microsoft Research. I would like to thank both Michael Sedlmair and Mike Kirby for their feedback and advice when writing the initial design activity framework paper. I would also like to thank Christine Pickett for her many edits, corrections, and comments throughout my dissertation. Lastly, the faculty and staff at the School of Computing and SCI Institute have helped guide my educational career and further my academic research in computer science, and I cannot thank them all enough for this support and dedication.

The work presented in this dissertation would not have been possible without funding support. A majority of this work is sponsored by the Air Force Research Laboratory and the DARPA XDATA program. Additionally, I was provided the Wayne Brown Fellowship by the College of Engineering, which aided in my first years of research and study. During my graduate career, I also was funded through two internships at MIT Lincoln Laboratory as well as one at Microsoft Research. This work is also sponsored in part by the Office of Naval Research award N00014-12-1-0601 and the U.S. Army Research Office under a prime contract issued to Intelligent Automation, Inc. The Lincoln Laboratory portion of this work was sponsored by the Assistant Secretary of Defense for Research &

Engineering under Air Force Contract FA8721-05-C-0002. Opinions, interpretations, conclusions, and recommendations are those of the authors and are not necessarily endorsed by the United States government or Intelligent Automation, Inc. The content of the information does not necessarily reflect the position or the policy of the government, and no official endorsement should be inferred.

This dissertation builds off of a variety of published research for which we obtained permissions for re-use. Reprinted [1], with permission, from McKenna, Sean and Mazur, Dominika and Agutter, James and Meyer, Miriah; "Design activity framework for visualization design"; *Transactions on Visualization and Computer Graphics*; and © 2014 IEEE. Reprinted [2], with permission, from McKenna, Sean and Staheli, Diane and Meyer, Miriah; "Unlocking user-centered design methods for building cyber security visualizations"; *Proceedings of the International Symposium on Visualization for Cyber Security*; © 2015 IEEE. Reprinted [3], by permission of the American Statistical Association (`https://www.amstat.org`), from McKenna, Sean and Meyer, Miriah and Gregg, Christopher and Gerber, Samuel; "s-CorrPlot: An interactive scatterplot for exploring correlation"; *Journal of Computational and Graphical Statistics*; © 2015 Taylor & Francis. Reprinted [4], with permission, from McKenna, Sean and Staheli, Diane and Fulcher, Cody and Meyer, Miriah; "BubbleNet: A cyber security dashboard for visualizing patterns"; *Computer Graphics Forum*; © 2016 John Wiley and Sons.

1 Introduction

This dissertation introduces a new data visualization design process model to guide and support visualization designers through the act of creating effective, useful, and usable visualization systems. Existing visualization design process models prove difficult to use in practice due to a disconnect with design decisions, evaluation methods, design artifacts, and which step a designer is in. We propose the design activity framework to more effectively guide visualization designers through a series of design activities. Each design activity has a descriptive motivation, a collection of generative or evaluative design methods, and the goal of visualization artifacts, such as design requirements, ideas, prototypes, and systems. The development of the design activity framework is grounded in a series of formative and summative data visualization projects, where we encountered limitations of existing models and utilized this framework as a solution. The design activity framework is validated through series of case studies in the fields of biology [3], cybersecurity [1], [2], [4], and data storytelling [5] and also by an external validation using design activity worksheets taught in a data visualization classroom for use in a cumulative project [6].

In order to capture the challenges faced when using existing visualization design process models, we identify five high-level goals for a model that, if met, could effectively aid visualization designers: **achievability**, **flexibility**, **justifiability**, **discoverability**, and **actionability**. Specifically, existing models fail to clearly lay out goals for steps in the design process, whereas our design activity framework includes achievable artifacts and connections to justifiable design decisions. With an activity's clear motivation, list of discoverable design methods, and flexible flow principles, the design activity framework is also more actionable and could help guide visualization designers effectively through the process. By working to capture these five goals, the design activity framework is an improvement upon existing models for data visualization design due to the model's increased descriptive, evaluative, and generative power [7].

To be *achievable*, a design process model should clearly highlight the desired visualization outcomes. Existing models outline the design process as a series of steps [8]–[11], but the design activity framework clearly defines and separates activities based on designers' motivations. Additionally, visualization designers are given a clear end product to work toward in each step: a visualization artifact. Commonly, visualization artifacts are tangible software tools, but the visualization design process can involve other kinds of artifacts, such as documents of user needs, a list of software requirements, or sketches. Existing process models do not rigorously capture some of these design artifacts, largely due to how these models consider evaluation as a separate stage of the design process [9]–[11] rather than a part of every activity. Artifacts for visualization design can serve many purposes, from measuring success to being reused in future projects; thus, it is often both beneficial and timesaving to capture and report on a variety of design artifacts. By defining these steps and visualization artifacts, the design activity framework increases achievability for designers to produce visualization artifacts and progress to the next step.

A design process model should support *flexible* iteration along with divergent and convergent methodologies. The design activity framework meets this goal by defining its four steps as activities. These activities can be conducted in various orders, nested, and even conducted in parallel by different members of a visualization design team. To capture the rich, complex, and flexible nature of design [8], [10]–[15], this framing supports capturing the design process in ways many existing visualization design process models fail to support. Additionally, each step differentiates between generative and evaluative methods, where generative methods allow for divergent approaches that encourage exploration, novelty, creativity, and innovation [16], and the evaluative methods winnow and narrow these artifacts to a smaller set.

Since steps can involve evaluative methods, it is important for a design process model to include *justifiable* design decisions as part of visualization designers' rationale, and no existing models connect these design decisions [17], [18] with the design process. The design activity framework includes evaluative methods in each step or activity to promote the validation, justification, and formation of or building upon existing guidelines [18]. Such guidelines can help determine the effectiveness of a technique or encoding [19]–[22] and aid future designers in avoiding common pitfalls [8]. Other common design decisions stem from preexisting guidelines in the visualization community, such as using position along a common scale to encode the most important data visually, over less perceptually effective channels such as angle, area, and saturation of color [23]. These methods and guidelines inform the design process and potential visualization artifacts. It is important to capture visualization designers' decisions and rationale for establishing rigor and transferability [8] of a visualization design process and its various artifacts.

With a broad range of generative and evaluative methods at their disposal, visualization designers could benefit from a *discoverable* approach for finding, utilizing, and reflecting on the use of different user-centered design methods. These design methods can be adopted from a wide variety of fields, from human-computer interaction to design to software engineering. The design activity framework specifically pulls from user-centered design methodologies [24], [25] to organize, suggest, and promote new kinds of design methods for visualization designers to employ. By emphasizing these design methods that focus on identifying and designing for user needs, the methods in this process model are useful for visualization designers conducting applied research with domain experts, such as design studies [8]. The design activity framework was constructed to highlight, correlate, and promote the discovery and inclusion of these user-centered design methods for a wide array of different visualization design projects.

Lastly, a crucial aspect of a visualization design process model is how *actionable* it is, for it to be understood, taught, and utilized by visualization designers. The design activity framework was created with succinct terminology, which has the benefit of clarity for teaching the concepts to visualization design novices [26]. Furthermore, by outlining four concrete steps with evaluation throughout, the framework encourages novices to think about the design rationale and reinforce their visualization knowledge by practicing and applying guidelines across diverse situations and projects. Visualization designers can learn to implement their own design process when activities are further broken down into concrete steps [16], and worksheets for this framework support such a walk-through approach. The notion of iteration in design can also be emphasized, taught, and realized by novices using these worksheets for real-world visualization projects. Moreover, existing visualization models do not explore pedagogical approaches that validate the design process with visualization design novices.

1.1 Contributions

The primary contribution of this dissertation is the **design activity framework**: a structure for how to perform a human-centered, data visualization design process while tracking design methods, visualization artifacts, and design decisions for each design activity. We introduce this framework as a methodology for visualization designers pursuing problem-driven work, such as design studies, but we also recognize its potential usefulness for general visualization design and data visualization pedagogy. The key novelty to this framework is its specificity for visualization design by connecting steps of the design process to the justifiable design decisions made and visualization-specific artifacts obtained. Other novel aspects of this framework include its flexibility for design iteration, actionability to guide visualization designers, and discoverability of user-centered design methods.

To strengthen the contribution of this framework, we evaluated its use and application across several different visualization design projects. Specifically, we utilized this framework to formulate, guide, and reflect on the design process of design studies and research in the fields of biology [3], cybersecurity [1], [2], [4], and data visualization storytelling [5]. For each of these projects, we employed a variety of qualitative and quantitative design methods in order to validate the utility of this framework in real-world projects and reflect on their use in future visualization design projects. We describe two case studies for visualization of cybersecurity datasets that illustrate examples of how to use this framework and provide rich descriptions behind a variety of different visualization design artifacts. To externally validate this framework, we created concrete, step-by-step worksheets for each activity that we evaluated through a series of qualitative surveys and interviews with novice visualization designers in the classroom. As a result of this evaluation, we discovered that students effectively learned how to design and develop a visualization system while justifying their decisions using the design activity worksheets.

1.2 Overview

Chapter 2 presents the necessary background for this dissertation. As the field of visualization matures, theories and models for visualization design have become more prevalent, from evaluation strategies [27]–[31] to the design process itself [8]–[11], [13], [31]. We discuss these different theoretical design models across communities, including models for visualization that focus separately on either the design process or decisions. Lastly, we investigate various pedagogical approaches for teaching the visualization design process [8], [17], [32]–[35].

A core contribution of this dissertation is the design activity framework; we present an overview in Chapter 3. This framework addresses a missing connection between different kinds of visualization design models, providing for a more complete description of a design project in visualization. The framework presents steps of a visualization design process through four design activities: *understand*, *ideate*, *make*, and *deploy*. Each activity contains a motivation or goal, a list of methods, and target visualization artifacts. These concise definitions support novice designers in learning and utilizing the framework to connect their design decisions to visualization artifacts. We showcase a series of design timelines to illustrate how to track and report on activities with their associated artifacts. Lastly, we include a table of possible design methods that visualization designers can employ to generate and evaluate visualization artifacts.

The design activity framework came out of reflections of the successful design project explained in Chapter 4. This project focused on redesigning a cybersecurity tool, resulting in updates to the underlying system as a result of this separate design process. This isolation of the design and development enabled us to reflect on how to best describe and explain our visualization design process to collaborators from different fields. As a result, we describe this project as a case study for how to use the design activity framework, and this chapter presents a variety of design methods, visualization artifacts, and a design timeline.

In Chapter 5, we used the design activity framework to perform a successful design study to create a cybersecurity visualization dashboard. This design study highlights significant challenges for user-centered design, such as limited access to end users and data. We present several design methods that we used to overcome these challenges and produce useful visualization design artifacts. We include a discussion on the design decisions, evaluation, and deployment of this visualization system as a case study that shows the descriptive and generative power of the design activity framework.

To perform an external evaluation of the framework and increase the framework's actionability and achievability, we created concise design activity worksheets, described in Chapter 6. By reflecting on our own experience using a variety of design methods, we identified descriptive steps for each design activity of visualization design. These steps served as a checklist that we taught and utilized in a visualization course with students. We performed a qualitative evaluation of the use of these visualization design worksheets in a cumulative project for which students had to design and develop their own web-based visualization system from scratch. This qualitative evaluation included survey feedback from 32 students and 11 in-depth, semistructured interviews. The design activity worksheets are a new approach to teach the next generation of visualization designers about the data visualization design process, by equipping them with not only the theoretical knowledge but also the practical skills for building better visualization systems and tools.

Chapter 7 explores different applications of the design activity framework for visualization through a perspective on the role of the design process across different kinds of visualization research. For example, a project we encountered before we had the knowledge of this framework involved the creation of a novel technique for exploring correlation, but the resulting technique and tool failed to adequately solve the larger real-world problems faced by our biology collaborators. Another project we conducted was an exploration and evaluation of a design space for interactive, visual data stories, and through this project we tested how readers interact with these stories by building different visual experiences to compare. We recognize that this work was very formative and exploratory, the beginning step in shaping an understanding and setting guidelines for future decisions made by visualization designers in this space. Our reflections on these two projects demonstrate how design can play a subtle yet pivotal role in both technique-driven and evaluation-driven research.

We provide a discussion of the design activity framework for visualization design and outline future work in Chapter 8. In this discussion, we include a call for more pedagogical work and materials for novice visualization designers, based on the results of our evaluation of the design activity worksheets. We explore the connection between visualization design and development, specifically agile software engineering approaches. Additionally, we explore the notion of broader process models, for both research and development, as ways to extend and connect these design models to more effectively guide and support building better visualization tools and systems. Lastly, we conclude this dissertation in Chapter 9 with a summary of the work.

2

Background and Related Work

While research in visualization design has explored many facets of design, the main goals of the design activity framework originate from shortfalls in existing visualization models: achievability, flexibility, justifiability, discoverability, and actionability. In this chapter, we focus on two types of visualization design models, decision and process models [18]. The nested model [17], [18], the primary model for visualization design decisions, addresses justifiability unlike process models. On the other hand, design process models can be grouped into two approaches based on research in the human-computer interaction (HCI) community: creative and engineering [14], [36]–[38]. Together, these two approaches can complement and enrich a design process. Next, we investigate how visualization design process models fail to capture the complex, actionable, and flexible nature of the creative process. Furthermore, we reflect on the role and use of human-centered design methods as a way to promote discoverability in visualization design. Following this, we explore the achievability and actionability of design process models with respect to how design pedagogy is incorporated in the data visualization community. In the next chapter, we will introduce the design activity framework as a way to bridge design activities with the decisions a visualization designer might make, supporting all five goals for an improved data visualization design process model.

2.1 Types of Visualization Design Models

Visualization research often involves the creation of new visual encodings, interaction techniques, and systems. This process of making something new is why design plays an integral role in research [39]. As such, there exist a variety of theoretical models for visualization design and even more that have been adapted, used, and taught by visualization designers. In this work, we focus on two kinds of models for visualization design: decision models and process models [18]. **Decision**

models capture the *what* and *why* of design by characterizing the rationale behind the decisions that a designer makes. This rationale can be useful for tracking decisions with respect to the project's context and for transferring design knowledge and guidelines to other visualization projects. **Process models**, on the other hand, capture the *how* of design, characterizing the actions that a designer takes as a series of steps.

Linking a process model to a decision model enables visualization designers to verify and validate the design decisions they make along each step of the design process, but existing models for visualization design have failed to do so. This link between decision and process models is highlighted by Schön's reflection-in-action concept [40], which emphasizes that the processes of doing and thinking are complementary to each other; thus, the design process and its many design decisions are intricately interconnected. However, existing visualization design process models fail to capture this link and thus do not meet a goal of justifiability. Additionally, we encountered limitations from existing models in terms of actionability since guidance, descriptions, and definitions for steps of the design process are not always outlined.

2.1.1 Design Decision Models

Many researchers have explored the general act of decision-making in design. A detailed model by Christiaans and Almendra captures both the mindset and strategies of designers, such as problem-driven (targeting a specific description of a challenge, such as generating software requirements) versus solution-driven (focusing on a small set of possible solutions, such as repurposing an existing tool or technique), as shown in Fig. 2.1 [41]. These strategies are combined with specific operationalizations of that mindset as well as with how decisions get made by an individual or a team, such as autocratic versus autonomic. Similarly, Tang et al. divide design decisions into three groups: planning, problem space, and solution space decisions, in order to better realize the effect decisions have on design [42]. Through studying the process of expert designers, Wu et al. identify three classes of design strategies: forward working (from abstract to concrete), backward working (from concrete to abstract), and problem switching (alternating between the two) [43]. Furthermore, several researchers have broken down decision-making into different kinds of high-level design judgments, e.g., appearance, compositional, navigational, etc. [38], [44]. These decision models are useful for analyzing and comparing general decisions and strategies for design, but they do not capture the specific decisions that visualization designers face when representing and encoding data in an interactive visualization system.

Within the visualization community, the well-cited nested model [17], as seen in Fig. 2.2, is the de facto design decision model. One of the primary motivations of the nested model is to support effective evaluation and validation of different types of decisions that visualization designers make. This model characterizes visualization design decisions as occurring at one of four levels: domain characterization, data and task abstraction, visual encoding and interaction, and algorithm.

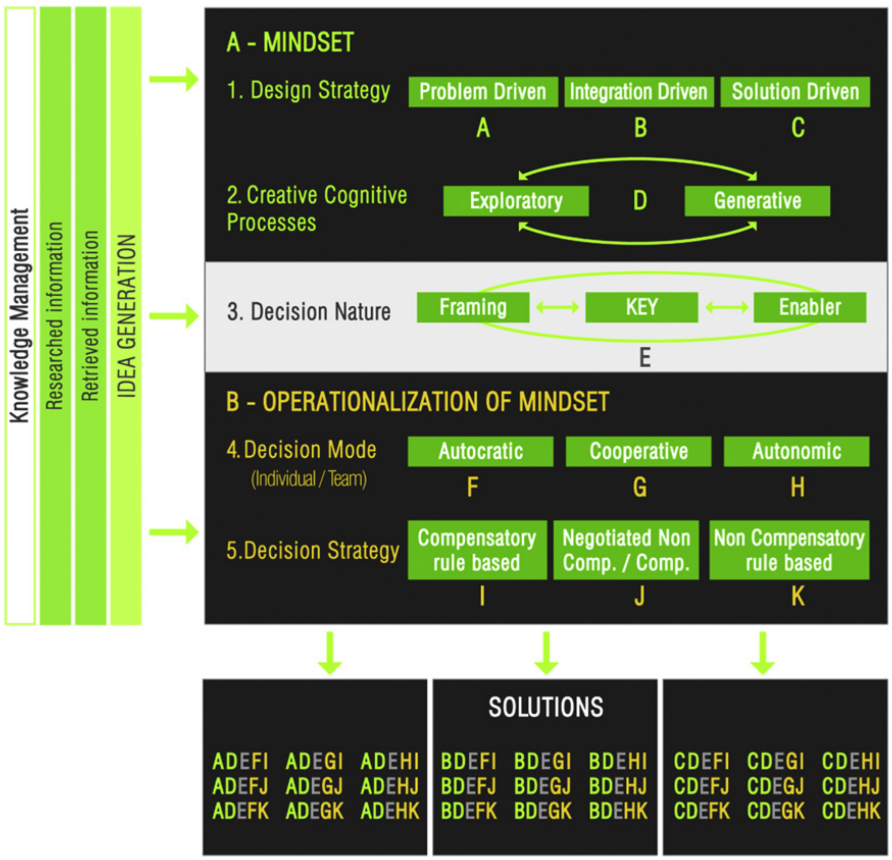


Fig. 2.1. Decision-making framework in design. Christiaans and Almendra identified different forms of design strategies, both problem and solution driven [41]. This decision model aims to capture the mindset and the operationalization of that mindset into strategies in order to describe the decisions that designers make as they produce design solutions. Many other design decision models, outside the visualization field, have similarly identified this separation of the problem and the solution, including the aspects of creative processes, such as exploration and generation, which lead to decisions that produce solutions.

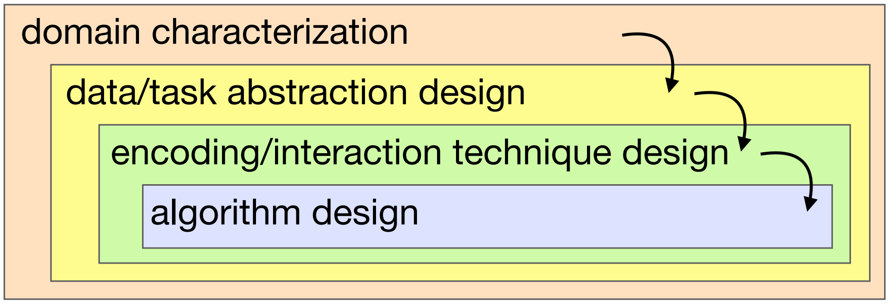


Fig. 2.2. Nested model for visualization design. Munzner produced the nested model with four levels to capture the specific data visualization design decisions [17]. Another goal of the model is to guide visualization designers to find appropriate validation depending on where they are in the model, and this validation guidance can provide warnings about potential errors that can cascade into downstream or deeper levels.

A recent extension to the model, called the nested blocks and guidelines model [18], provides a more fine-grained characterization of individual design decisions as blocks at each level, with guidelines describing the relationships between blocks, as illustrated in Fig. 2.3. Together, blocks and guidelines relate the visualization decisions a designer makes, with regard to finding good blocks in the design of a visualization.

It is important to stress that the nested model, as well as the nested blocks and guidelines model, are not process models; they do not describe *how* to design a visualization, only the types of decisions (*what*) and rationale (*why*) that a visualization designer formulates along the way [18]. Design decisions, rationale, and guidelines are formed from employed evaluation methods. Existing visualization process models fail to incorporate this justifiability because they have no link to decision models. Moreover, numerous existing models capture evaluation as only a step of the design process [9]–[11], [13] rather than a continuing role throughout the process as in design decision models.

2.1.2 Design Process Models

Unlike a decision model, a design process model focuses on describing the specific steps a designer takes over the course of designing a visualization. Whereas design can mean many things to different people, the **data visualization design process** is about the planning, creation, and evaluation of a single data visualization or a multiview, robust visualization system. In this regard, we consider design as a challenge that combines and mixes both engineering and creative design processes [14], [36]–[38], and this balanced mixture is what we sought in the synthesis of the design activity framework. An **engineering design process** begins with a problem definition, where the overall process is largely sequential and convergent toward a single solution [36]. On the other hand, a **creative design process** begins with more gradual problem scoping, and the process has many overlapping activities in which many different possibilities are explored before choosing a single solution [36]. Howard et al. created a design process model to utilize both creative and engineering aspects, displayed in

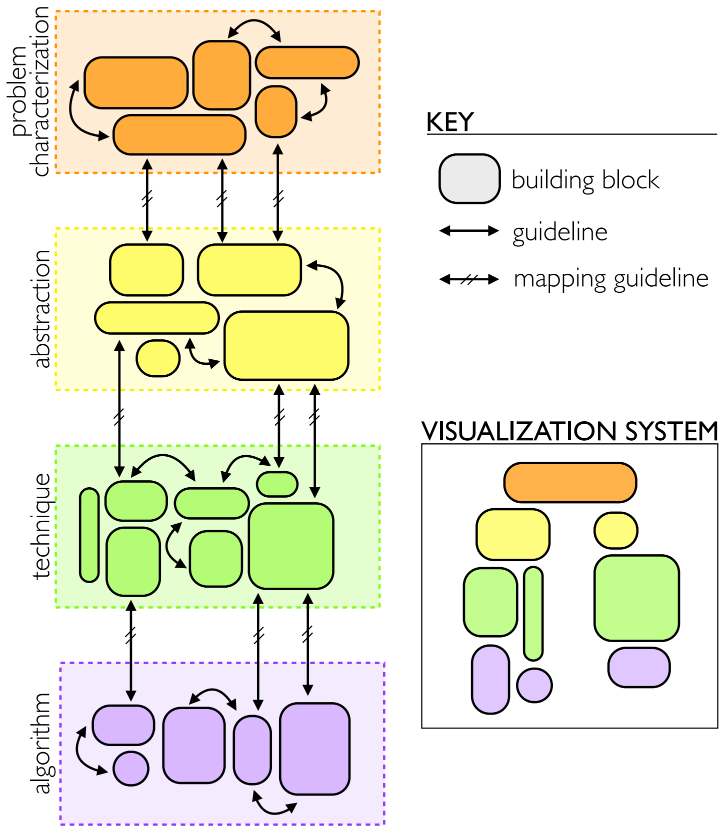


Fig. 2.3. The nested blocks and guidelines model. Meyer et al. introduced an extension to the nested model to more finely capture guidelines [18]. These guidelines stem from design decisions made by visualization designers, and guidelines can be established within or between levels of the nested model. As a visualization system is designed and developed, blocks and guidelines get constructed, compared, and evaluated.

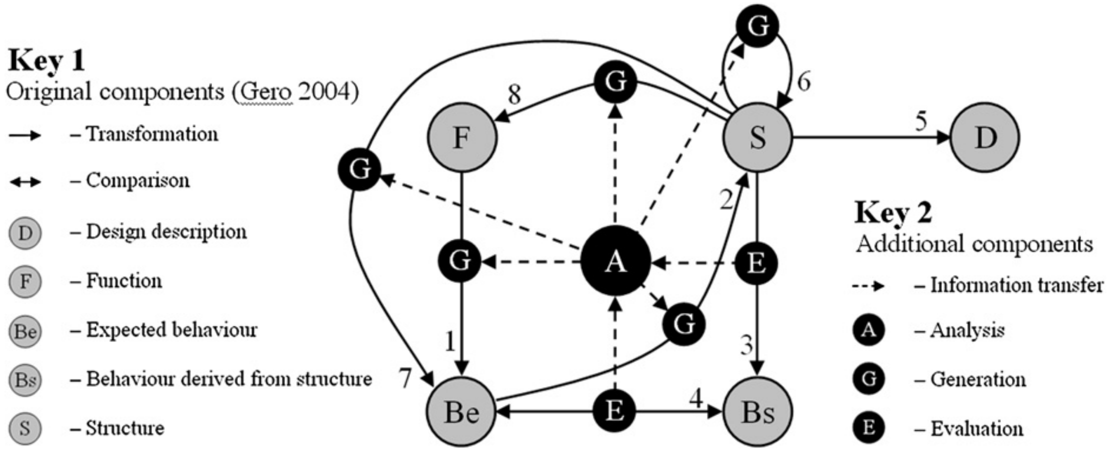


Fig. 2.4. Integrated creative design process model. Howard et al. created a model to encompass both engineering and creative design processes [37]. Components of a previous model are modified and connected in order to emphasize and call out the specific stages of generation, evaluation, and analysis. A key aspect of this model is the incorporation of creative output, which is an idea that must be novel, invented but not obvious, and of use to industry.

Fig. 2.4 [37].

An example of a creative model is Pugh's process, a design funnel in Fig. 2.5 [45] that begins with concept generation and controlled convergence iterating over time until a final concept is reached [46]. Additionally, ideation and design activities often involve sketching as a method that is a crucial aspect for creative design processes, as sketching is not simply the act of drawing but rather is an activity involving generation, brainstorming, learning, reasoning, and design thinking [46]. As recognized by researchers in the design [37], HCI [36], [38], and visualization [14] communities, the combination and balanced mixture of both creative and engineering process models is useful for characterizing the design process for visualization designers. Specifically, this combination supports goals of flexibility and actionability.

Visualization-specific design process models describe unique aspects for designing and evaluating visualization systems; however, they largely do not connect to visualization design decisions and do not explicitly incorporate aspects of a creative design process, such as the goals of flexibility and discoverability. The seminal research method of multidimensional longitudinal case studies [31] proposes a process and specific methods for assessing and evaluating visualization systems deployed in the wild. This model, however, does not cover the creation and development of a visualization system. More abstracted design process models for visualization have also been proposed in a variety of forms — waterfall, cyclical, and spiral — to perform user-centered design [10], [11], [13], but they emphasize convergence as in an engineering design process model. The design process model used by both Lloyd and Dykes [9] (shown in Fig. 2.6) and Goodwin et al. [47] is drawn from an international standard on human-centered design, ISO13407, which has recently been updated, ISO9241-210 [12]. This standard's model, presented in Fig. 2.7, describes different design activities

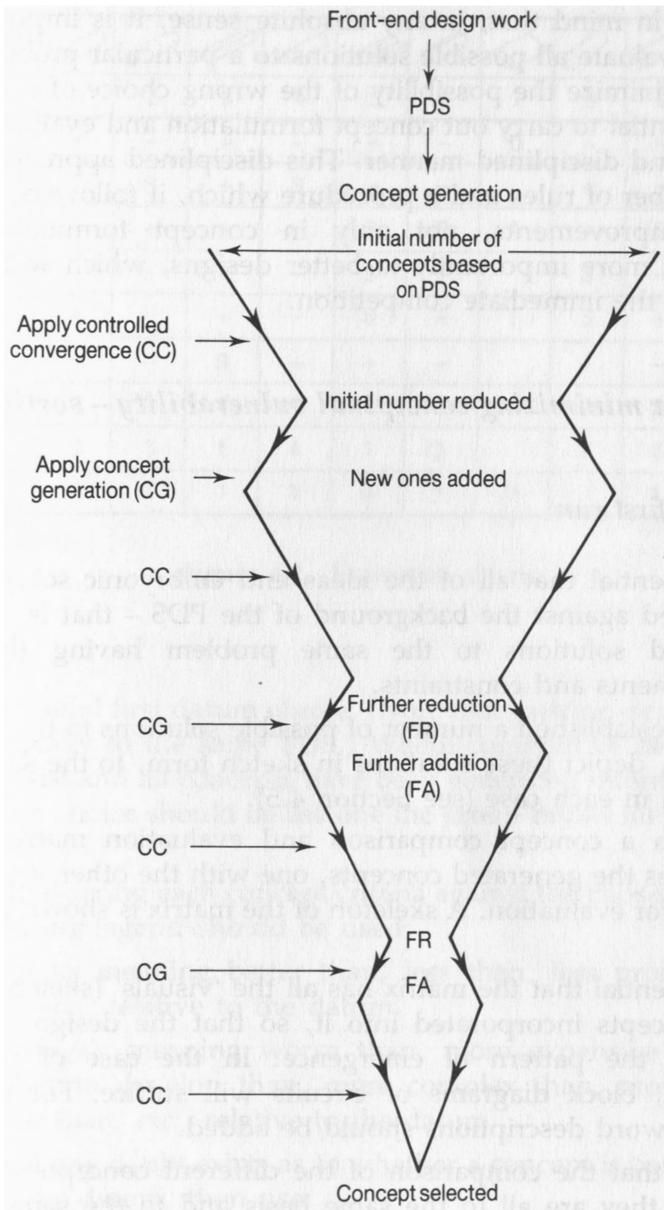


Fig. 2.5. Design funnel model. For product engineering, Pugh laid out the popular notion of a design funnel [45], where concept generation (CG) is interspersed with controlled convergence (CC). This funnel is foundational to the concepts of generation and evaluation for design.

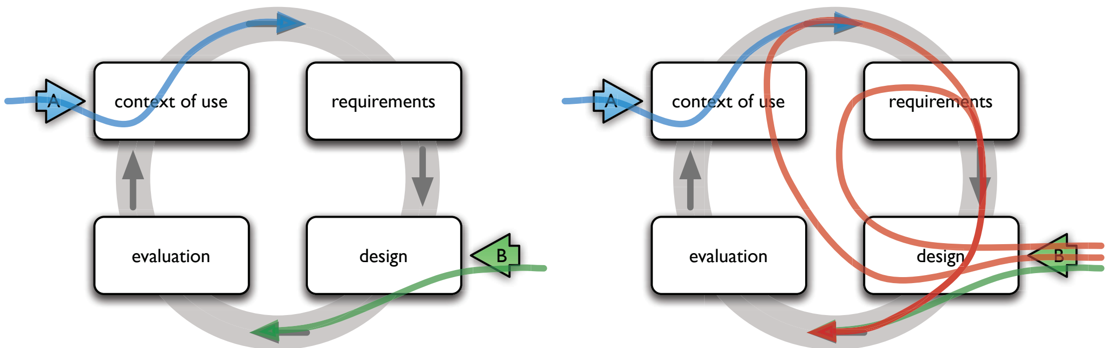


Fig. 2.6. Model for the human-centered design process. Lloyd and Dykes adapted a process model [9] to illustrate traditional engineering design processes (left), as well as more creative processes (right, in red). This development cycle further highlights the need for a model that clearly emphasizes the flexibility of a creative design process instead of implying strictly cyclical movement.

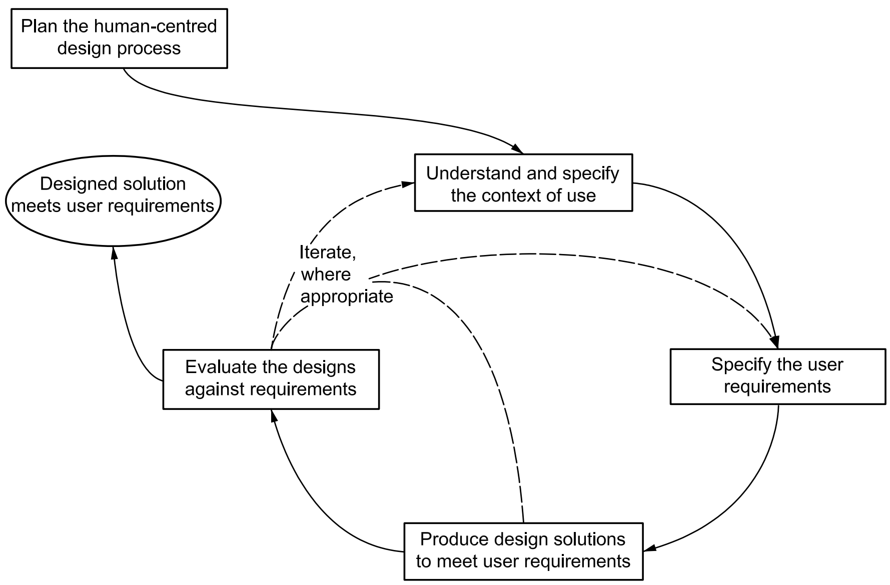


Fig. 2.7. Process model where design activities are interconnected. The standard ISO9241-210 includes this design process model [12] that breaks apart the stages of the design process into overlapping, human-centered design activities. However, the iteration and interconnection of these activities is not as clearly illustrated in the model or the standard.

as a cycle, emphasizing an engineering approach.

Goodwin et al. accompany this engineering process model with specific methods for eliciting creativity from end users [47], a step toward including aspects of a creative design process. Vande Moere and Purchase further characterize the role of design in visualization [14], and, although no design process model is outlined, their assertions on design emphasize the importance of creative aspects for visualization design. By embracing concepts from action design research (ADR), McCurdy et al. applied ADR to a design study to illustrate how intervention with collaborators shaped the final visualization design and how they reflected and learned throughout this process [48]. Although the

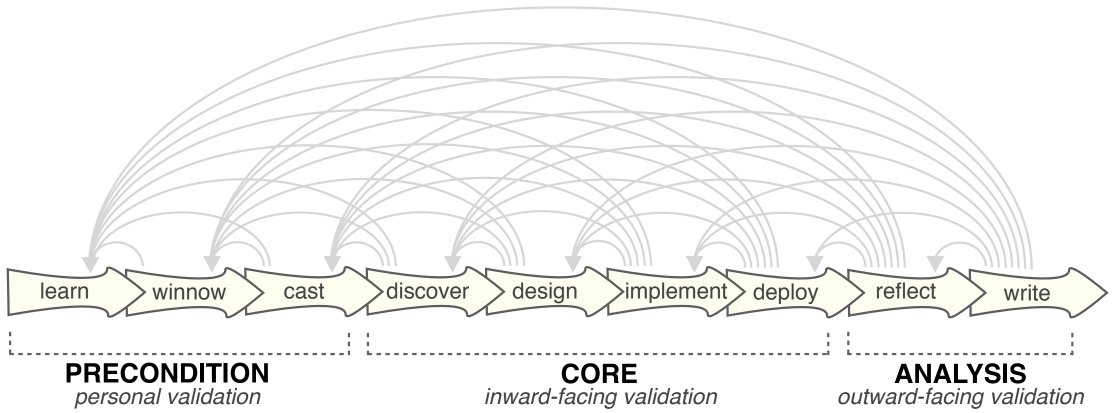


Fig. 2.8. Nine-stage framework process model for design studies. Sedlmair et al. introduced nine stages of the design process, characterized by three high-level categories [8]. The gray arrows indicate iterative and nonlinear aspects. The design activity framework extends the four core stages of this model — discover, design, implement, and deploy — in order to increase the model's achievability, flexibility, justifiability, discoverability, and actionability.

visualization community recognizes that creative aspects and design decisions are important, none of the visualization process models explicitly incorporate creative aspects or link back to visualization design decisions.

The model closest to the design activity framework is the nine-stage framework for conducting design studies [8], illustrated in Fig. 2.8, that captures the steps from initial planning through the reflective analysis of a complete project. The middle core stages of the model describe the steps involved with designing a visualization system, with four stages that, at a high level, are similar in motivation to the proposed design activity framework. In some of these middle stages, the levels of the nested model are mentioned; however, an explicit description of what types of artifacts should be expected at each step is not provided. Furthermore, the model as a whole only loosely captures the overlapping and iterative nature of visualization design, as well as the role of evaluation throughout, which McCurdy et al. argue is crucial and should occur concurrently with building a system [48]. As such, we consider the design activity framework as an extension of the four-core design stages in Sedlmair's model for design studies, focused on helping conduct general visualization design projects.

The nine-stage framework, although the first model of its kind to provide guidance for conducting design studies, provides less actionable advice for visualization designers, such as knowing what design stage they are in, what kinds of methods to employ, or the specific artifacts and decisions they should make. The design activity framework is largely inspired by the nine-stage framework but focuses on providing more actionable guidance for visualization designers, which is not currently available within the nine-stage framework, and linking to design decisions to support justifiability.

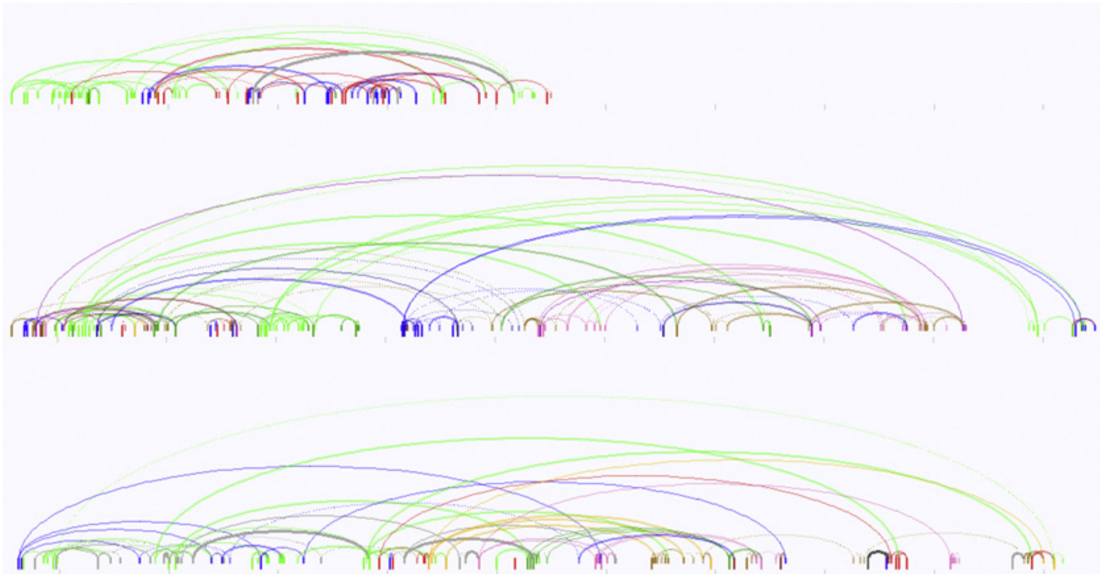


Fig. 2.9. Designer ideas tracked over time. Baker and van der Hoek captured key ideas that designers generated from three companies: Intuit, AmberPoint, and Adobe (top to bottom) [49]. During a structured session with designers from these three companies, the researchers categorized the different types and connections between ideas as they evolved in the design session. Each rectangular node represents a specific idea, and the nodes are connected by arcs based on relationships of the user needs or idea repetition. Colors denote different subjects discussed by the designers. This visualization emphasizes that ideas and their connections played a nonlinear role throughout the process.

2.2 Flexibility in Visualization Design

In a visualization design project in which we worked with two general designers and a psychologist (discussed in Chapter 4), we discovered a disparity with how existing visualization design models support flexibility throughout the design process. Although some visualization researchers argue that design and research methods can elicit creativity in visualization design [9], [14], [47], design practitioners explicitly emphasize and highlight the complex nature of the design process [49], [50] as well as the role of design constraints [51], [52]. Baker and van der Hoek observed designers from Intuit, AmberPoint, and Adobe [49]. The researchers tracked the designers' ideas over time, as seen in Fig. 2.9, which shows the complex, iterative, and nonlinear nature of the design process. In addition, a model used by Kumar to demonstrate design methods also shows the nonlinear nature of the design process, as in Fig. 2.10, and he further cautions against models that imply linearity [24]. Design constraints and complex ordering of design activities were not explicitly captured in many existing visualization process models which led to a lack of flexibility when trying to track, describe, and document our own design processes. Furthermore, by failing to connect to design decisions or levels of the nested model [17], these visualization models do not emphasize the importance of design rationale for decision-making, which can play a critical role in how a visualization design process unfolds.

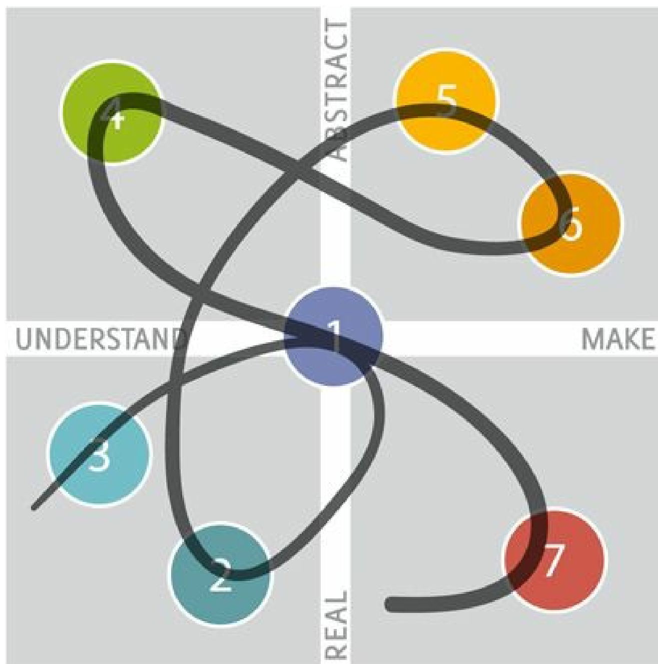


Fig. 2.10. Design innovation process model. Kumar presents this model with seven activity modes for design innovation [24] — sense intent, know context, know people, frame insights, explore concepts, frame solutions, and realize offerings. The curved line represents the design process and highlights the importance of capturing nonlinearity for design. This model's concept motivated the representation of the design activity framework, to embrace creative process models by steering away from a linear or cyclic diagram.

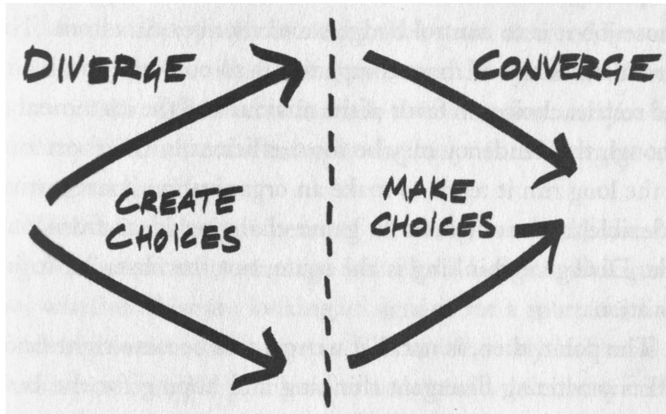


Fig. 2.11. Design thinking funnel. For design thinking, Brown emphasizes the importance of a design funnel [57], which diverges and converges choices over time. First, divergent thinking creates many choices or possible solutions. Brown emphasizes that analysis is an important step for convergent thinking, and we incorporated this concept in the design activity framework.

By reflecting on our own design process in Chapter 4, we identify a need for a process framework that balances the flexibility and actionability of models from the design community with the explicit artifacts and decisions necessary for visualization design. Bigelow et al. further emphasize this need that designers have for design flexibility, specifically for using visualization systems to broadly explore visual encodings [15]. We developed the design activity framework to overcome shortcomings in existing visualization design process models [8]–[13], [47] and to incorporate ideas from a broad range of models in HCI [38], [53], [54] and design [35]–[38], [43], [55]–[63].

Several creative design process models emphasize the importance of a design funnel, where ideas and concepts are generated and evaluated over time [45], [57]. These concepts are illustrated in Fig. 2.5 and Fig. 2.11, and we incorporated aspects of these models into the design activity framework. By utilizing this design funnel and providing succinct definitions, activity motivations, and desired visualization artifacts, such as design constraints, the design activity framework achieves greater actionability and flexibility over existing visualization design process models. Additionally, principles that show a flow of activities that is complex, iterative, and multilinear increase the flexibility supported by the design activity framework.

2.3 User-Centered Design Methods

User-centered design methods provide a promising approach to build better and more effective visualization tools, and thus promoting the discoverability of such design methods can benefit visualization design process models. By focusing on users' needs, wants, and limitations, user-centered design methods can result in more useful, usable, and enjoyable tools that enable users to achieve their goals more effectively, efficiently, and with increased satisfaction, thus providing benefits such as increased productivity, better accessibility, reduced stress and risk of harm, and an improved sense of well-being [12]. User-centered design and its methods have become widely accepted within the HCI community [13] and have become more popular and accepted within the visualization community as well. Many different visualization practitioners have illustrated potential phases and methods of a user-design process [9]–[11], [13], but each lacks the connection to design decisions or justifiability. Furthermore, these phases do not list a broad set of possible set of design methods to support discoverability of new methods to employ. To further complicate matters, many process models contain differing phases of visualization design without a clear definition of the phases of this process and how they connect to specific design methods for generation or evaluation.

For visualization design, many publications report the use and benefit of user-centered design methods, but they do not always state the connections to the larger design process which limits both the discoverability and actionability of these methods. In many design studies, researchers interview users to derive requirements for a visualization tool [19]–[22], [64], [65]. Other methods for deriving user requirements include the personas design method [66], qualitative coding, and data sketches [2]. Others promote similar design-first, co-creation, or creative approaches to help find innovative visualization solutions, emphasizing visual concepts before user requirements [47], [67], [68]. Several researchers have used iterative usability studies and user feedback to improve upon the design of a visualization prototype [4], [69], [70]. Many of the visualization models capture more of an engineering design process, with a testing or evaluation phase [9]–[11], [13] that does not support the role of evaluation and decisions throughout the process. User-centered design methods can be evaluative at any step or phase, to limit or narrow the choice of potential artifacts going forward. This gap stems from a lack of emphasis on evaluative methods in existing visualization design process models, but there exist a plethora of design methods that get utilized in real-world projects and that could be introduced to visualization designers if a model supported their discoverability.

2.4 Pedagogy of Data Visualization Design

Another crucial aspect for visualization design models is how easily they can be introduced, taught, and understood by novice visualization designers. Few visualization process models have been studied and reported on their use in a classroom setting, which is one approach to evaluate a model. Existing process models for visualization could be improved to more clearly outline the desired, achievable visualization artifacts. Furthermore, the actionability of such models has not been studied or explored. For example, actionable guidance might provide a more step-by-step walk-through [16] for the design process, and illustrative examples of the design process can further add to the actionability of a model.

Over time, pedagogy for data visualization has shifted from more theoretical concepts to emphasizing more actionable skills, such as design critique and critical analysis [71]–[73]. As educators incorporate aspects such as active learning [72], [74], [75] and design workshops [73], [76], [77]

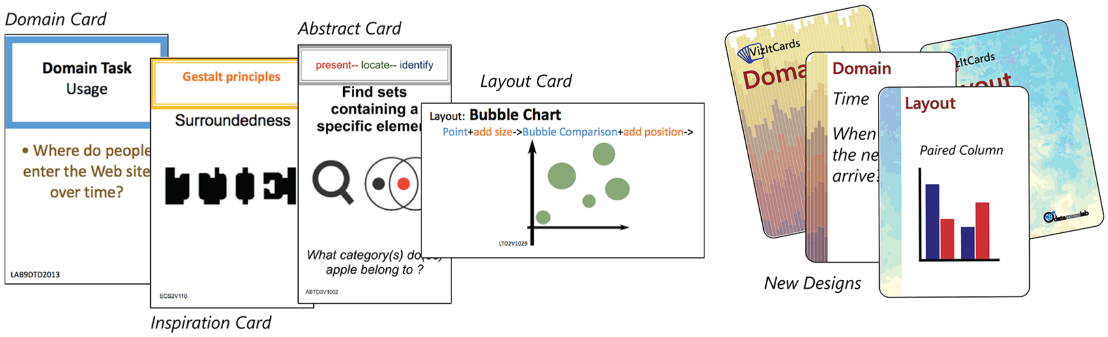


Fig. 2.12. Example VizItCards for use in design workshops. He and Adar incorporated the cards in design workshops to teach and reiterate core visualization concepts in the classroom [73], such as the importance of identifying domain tasks, finding inspiration, and selecting appropriate abstractions and encodings. These cards and this workshop approach could work in tandem with the design activity framework and worksheets when designing a visualization.

into the classroom, it is important to teach concepts like design thinking to students [73]. To this purpose, He and Adar utilized a card-based toolkit, highlighted in Fig. 2.12, to teach visualization design thinking to students in class workshops [73]. Moreover, teaching design thinking to students emphasizes how to generate ideas broadly and how to avoid refining ideas too early [16].

However, novices may struggle in visualization design projects since existing design models outlined in textbooks [34], [35] and research papers [8], [33] utilize high-level terminology that is often theory-based and less actionable out of context. Simplifying such terminology and focusing on comprehensive aspects [26] help students understand and apply concepts more readily. Furthermore, steps for the ideation process have been outlined by the five design-sheet methodology with worksheets shown in Fig. 2.13 [16], but broader steps beyond just ideation would be beneficial for visualization design pedagogy. The design activity framework and worksheets provide such a step-by-step description that increases not only its achievability but actionability as a visualization design process model.

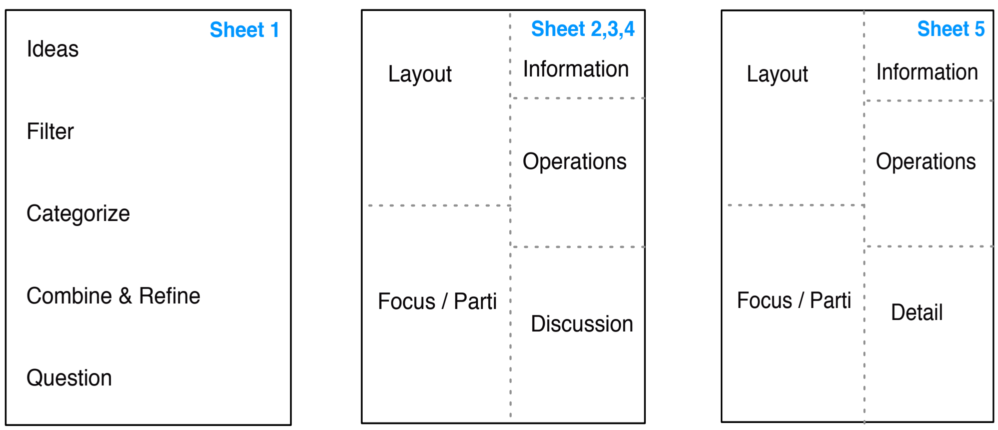


Fig. 2.13. Five design-sheet methodology worksheets. Roberts et al. present a method for teaching the ideation process to students using five worksheets [16] that encourage sketching visualization ideas and comparing them before realizing a final visualization design. This work motivated the creation of design activity worksheets for students to utilize and guide their design process for a cumulative project.

3
Design Activity Framework

A design process consists of *activities* or steps taken to achieve a given outcome, such as a visualization system. Design *decisions* are the reasons and justification behind a choice made for a visualization, such as encoding with either a pie or bar chart. Design *artifacts*, like a system prototype, are the goal for each step of the design process and result from making design decisions. As explained in Chapter 2, no work currently connects all three of these aspects for visualization design. By connecting these components, the design activity framework more comprehensively captures and describes the design process with achievability through artifacts, flexibility in principles of flow, justifiability by linking to design decisions, discoverability of different user-centered design methods, and actionability for use by visualization designers.

In this chapter, we present an overview of the **design activity framework** [1], a flexible structure meant to guide a visualization designer through the real-world, iterative, and multilinear process of developing a visualization for a specific problem or application domain. We envision the framework as a lens that visualization designers could use to orient themselves within the design process, to choose useful methods, to make appropriate design decisions, and to analyze and summarize the process itself. The design activity framework makes use of the nested model [17] to explicitly link the actions visualization designers take with the visualization decisions they make along the way, leading to what we believe is a more actionable visualization process model than those that currently exist.

We developed the framework over the course of several real-world design processes, as a result of reflection on a previous research project (Chapter 7) and more successful case studies (Chapter 4 and Chapter 5). Next, we discuss the methodology by which we initially formulated the design activity framework. Then, we present the idea of a design activity, which forms the basis of the new framework. Following that, we describe the four activities contained in the framework: *understand*, *ideate*, *make*, and *deploy*. For each activity, we articulate the motivation, possible design methods, and the visualization artifacts that relate to decisions and have an explicit link back to levels of the nested model. This framework supports an iterative, human-centered visualization design process that we characterize with the introduction of design timelines. Lastly, we provide guidance on how to choose effective design methods with a table of 33 selected design methods of interest to visualization designers to employ in their own projects.

3.1 Methodology

Through a series of reflective discussions, we formulated the groundwork for the design activity framework [1]. Our design team was tasked with a redesign of a cybersecurity visualization tool. Our team was composed of two visualization practitioners, a psychologist, and several designers. During this multidisciplinary project, we were cognizant of our design process, visualization artifacts, motivations, and design methods. The two visualization practitioners on the team attempted to use the nine-stage framework [8] to guide and model this process, but the model was unable to capture and track our design constraints, artifacts, and decisions in such a way that promoted team collaboration and the final project communication.

As a result, our team conducted a literature review of numerous design models, presented in the previous chapter. This review utilized a characterization between two types of design process models, creative and engineering models. Our team recognized the importance of a visualization design process model to incorporate both creative and engineering aspects to describe, capture, and prescribe processes effectively, such as our redesign project and other past experiences. However, in this literature review, we noted that no existing visualization design process models had combined these types of models comprehensively.

In order to create a new design process model for data visualization, we utilized our team's combined experience in visualization design and general design, in addition to terminology and concepts from existing models, as a guide to identify different stages and components of the process. As a redesign team, we iteratively codified the different stages performed, goals identified, artifacts created, and methods employed in our own visualization projects. As a result, we established a consensus for the terminology and these concepts which comprise the design activity framework. Next, we introduce these concepts.

3.2 A Design Activity

At the core of the design activity framework is the concept of a design **activity**, a group of actions a visualization designer takes to work toward a specific artifact or set of artifacts. Many creative process models tend to avoid breaking a process into sequential steps, stages, or phases but, rather,

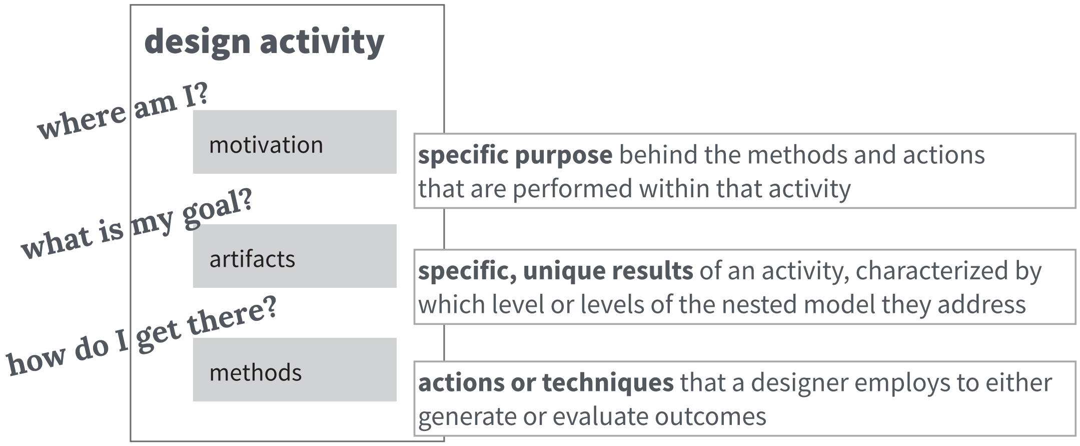


Fig. 3.1. Breakdown of a *design activity*. The design activity framework is composed of a series of design activities, each with a motivation, visualization artifacts, and design methods. We characterize these three components to answer high-level questions about the design process, thus increasing the actionability of the framework.

use the term activities [43], [54], [55], [59], [62], [63], which are not necessarily linear, and they are often overlapping. We characterize the composition of each activity using several key components: a *motivation*; clear, tangible visualization *artifacts* related to design decisions; and a collection of design *methods*. We illustrate an overview of a design activity in Fig. 3.1.

The **motivation** of an activity is the specific purpose behind the methods and actions that are performed within that activity. For example, a motivation can be to brainstorm new ideas to solve a specific problem or to test the efficacy of an aspect of a specific visualization for a given task. By matching a real-world motivation to those specified for each activity in the framework, visualization designers can place themselves within a specific design activity, which helps them choose appropriate methods and identify visualization artifacts.

Next, visualization **artifacts** are the specific, unique results of a design activity, characterized by which level or levels of the nested model they address. Artifacts are closely connected with design **methods**, which are actions or techniques that a designer employs to either generate or evaluate artifacts. It is in the application of methods to the broad space of all visualization design options, particularly methods for evaluation, that design **decisions** are made between artifacts.

We highlight two distinct kinds of methods used in each design activity: *generative* versus *evaluative*, as shown in Fig. 3.2. Generative methods are largely meant to be divergent and create many artifacts, such as methods for brainstorming [37], [78] or increasing creativity [47], [79]. Evaluative methods, on the other hand, are convergent and filter artifacts, such as methods that elicit feedback from domain experts [28], [80] or user studies [81], [82]. This distinction between generation and evaluation is common within the design community [41], [57], [83], [84]. For example, Pugh's design funnel includes both concept generation and controlled convergence [45]. Interestingly, some

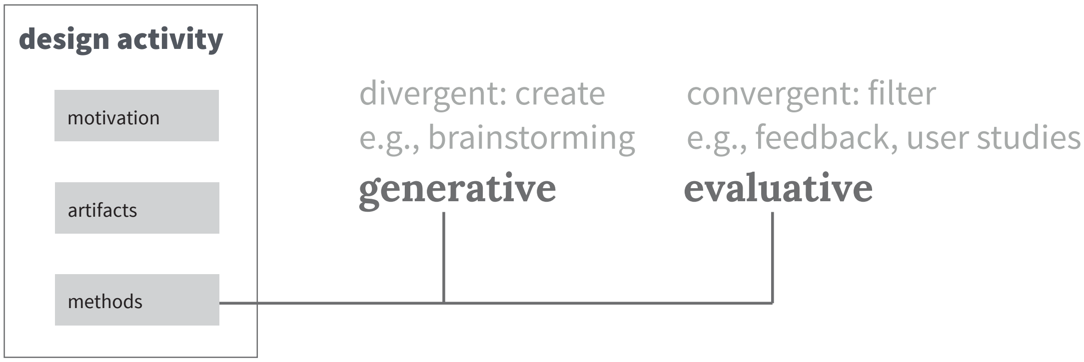


Fig. 3.2. Two types of design methods. For visualization design, these methods are typically generative (divergent) or evaluative (convergent) in nature as visualization artifacts get created and justified. Design decisions play a critical role in this process to select and winnow different visualization artifacts.

methods can be both generative and evaluative, such as observation and interviewing. In the design activity framework, we consider generative and evaluative design methods as vital components of each activity, unlike process models that capture evaluation as a single, unique stage in the design process [9]–[13], [47]. This emphasis on evaluation methods encourages visualization designers to consider evaluation early, and often, for each design activity.

The design activity framework further characterizes the methods based on two spectrums. First, generative methods can be used *narrowly to broadly*. For example, a novice visualization designer may narrowly consider only a single idea in the ideate stage, as opposed to specifically applying brainstorming methods to generate many different ideas broadly. Second, evaluative methods can be applied *informally to formally*, such as a visualization designer informally choosing a prototype based on personal preferences versus formally comparing multiple prototypes through a controlled user study. Characterizing the use of methods in each activity is important for two reasons: 1) for elucidating missed opportunities throughout the design process for further investigation and work; and 2) for providing a mechanism to thoughtfully incorporate real-world project constraints, such as time and budget considerations, into the design process.

We identify four overlapping, critical activities for designing visualizations for real-world problems and applications: *understand*, *ideate*, *make*, and *deploy*. An overview of these activities, with the unique letter and coloring scheme used throughout this dissertation, is shown in Fig. 3.3. Next, we articulate these unique motivations and visualization artifacts for all four design activities of the design activity framework.

3.3 Understand Activity

The first activity in the design activity framework for visualization projects is to **understand** the problem domain and target users. The motivation for this activity is *to gather, observe, and research*

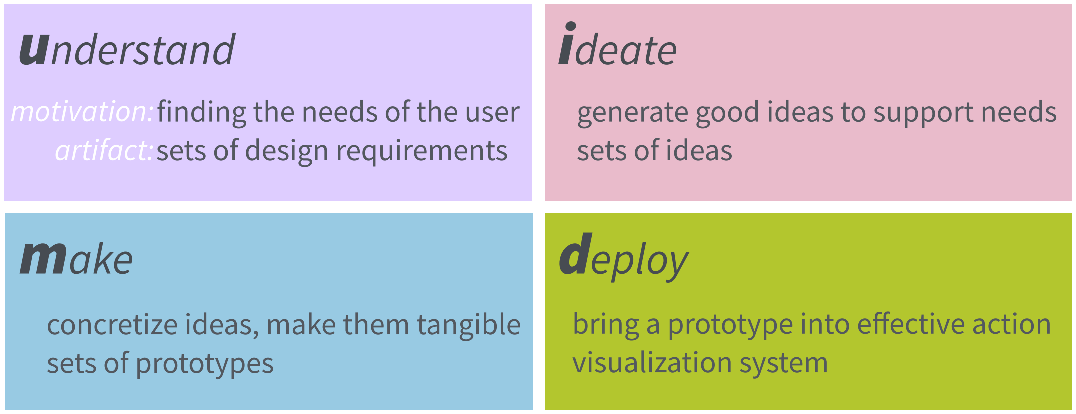


Fig. 3.3. Four core design activities. We identified four visualization design activities: *understand*, *ideate*, *make*, and *deploy*. Additionally, we detail the motivation and expected visualization artifacts within each activity. Throughout this dissertation, we refer to activities using the first initial of each activity and its associated color.

available information to find the needs of the user. The artifacts of this activity are commonly referred to as design requirements [9], [12], [47], [56], [85], [86]. These design requirements are often tailored to help users solve the problems or challenges they face in visualizing data. Not all challenges that users or domain collaborators may face will use or need a visualization system. For example, sometimes statistics, machine learning, or existing visualization tools such as Microsoft Excel or Tableau can help solve certain problems. In a visualization project, it is important for a visualization designer to identify early on if there is a need and a set of design requirements that cannot easily be solved using existing tools.

We break down visualization design requirements of the *understand* activity into three classes: *opportunities*, *constraints*, and *considerations*. Visualization design **opportunities** encompass the data and task abstraction artifacts that have a potential to impact the work and field of the target users. It is important to uncover data and task abstractions that cannot or cannot easily be solved using existing visualization tools. These opportunities may also include higher level themes discovered through the domain characterization, such as workflow inefficiencies when a collaborator may use static visualization tools in a pipeline that is slow and not interactive, thus making comparison tasks more difficult. Design **constraints** are rigid limitations from the project that the visualization designer must work with, such as rigid deadlines, limited hardware or computing systems, and access to expert users and their time. **Considerations**, however, are a looser, more flexible form of constraints that a designer should strive to consider, such as the importance of a final visualization tool's aesthetics, usability, or adherence to a set of domain visualization standards. These considerations may and sometimes should be discounted, but careful analysis and justification need to underlie such design decisions. Together, these three classes of visualization artifacts for the *understand* activity play a crucial role in all following activities, and they often get reconsidered, adjusted, and prioritized throughout the design process. For example, data and task abstractions can be changed based on new data needed by a collaborator or a user expressing a need for a new type of task not previously considered when performing an evaluation of a deployed visualization system.

3.4 Ideate Activity

The second activity in the framework is the **ideate** activity, which has the motivation *to generate a plethora of concepts and then winnow these into good ideas that meet the needs of a user.* The visualization artifacts of the *ideate* activity are a set of ideas often externalized in a variety of forms, from sketches to wireframes to low-fidelity prototypes. Generation and evaluation are two very important steps for ideation. For example, many creative designers strive to generate ideas free of limitations, constraints, or considerations because early judgment and decision-making can limit the range of possible ideas and concepts produced. Divergent design thinking is valued and recommended here, especially with visualization design, because such thinking is where innovation occurs and new visualization designs, techniques, and algorithms are created.

Beyond the creation of ideas, it is also important to compare, evaluate, and winnow the broad set of ideas into ones that have a greater potential for impact on a visualization tool. For example, a common choice faced by visualization designers is whether to encode a network graph as a set of nodes and links drawn between them or as a matrix to organize and restrict their position. Commonly, this decision can be motivated by the types of tasks a user wants to perform, but it could also be influenced by other factors such as the characteristics of the data (e.g., how many nodes or links?) or what role this tool needs to play in a larger system. Furthermore, to communicate and share ideas within a design team or to evaluate them with users, it is often necessary to externalize these ideas by sketching them on paper or through more refined versions such as wireframes or low-fidelity prototypes. As a designer externalizes these ideas onto some medium such as paper, it is common that more details of an idea must be fleshed out and concretized [39]. Another suggested technique in the visualization community is data sketching [9], which incorporates data into the ideation process, to discover the results and limitations encountered when using real data to realize an idea. For both concept sketching and data sketching, it is also possible that more ideas may spin out of this process, as a variation upon an existing idea or new idea combining several together, so it is often recommended for visualization designers to externalize their ideas early and often to help them generate more and better ideas.

3.5 Make Activity

Next, a visualization designer must start to build a visualization system in the **make** activity. This activity's motivation is *to concretize ideas into tangible prototypes.* The visualization artifacts from the *make* activity are a set of prototypes, with prototypes defined as *"approximations of a product* *along some dimensions of interest"* [87]. These prototypes must be built to handle and visualize real datasets, and it is common that, as prototypes get constructed, more design requirements or ideas may be explored and discovered, highlighting the iterative nature of visualization design. Another aspect of the *make* activity goes beyond design: visualization designers need to employ software engineering and development techniques for writing code and programs to build visualizations to meet the needs of the users. This could be as simple as writing scripts or automating processes for generating visualizations and tying them together using a variety of existing tools, but designers can also use other visualization frameworks (e.g., D3.js, Vega, Processing) or graphics toolkits (e.g., OpenGL, WebGL, Canvas) to build and generate interactive visualizations from the ground up.

Most engineering design process models couple the ideate and make activities. We believe that these two activities have related, but different, motivations and artifacts for visualization design, making their separation important for a careful consideration of all types of visualization design decisions. Although low-fidelity prototypes can exist in the *ideate* activity, prototypes for the *make* activity are of a higher fidelity and typically involve encoding of real data in order to evaluate the efficacy of the visualization technique for a specific problem. The *ideate* activity is meant to free the designer from focusing on low-level design decisions in order to broadly consider more abstract ones. The *make* activity, on the other hand, focuses the designer on the low-level design decisions necessary to actualize an idea into a concrete, testable prototype, such as the details of how to encode a data item or which algorithms to utilize.

3.6 Deploy Activity

The final design activity in the visualization framework is the **deploy** activity, with the motivation *to construct a visualization system and bring it into effective action in a real-world setting in order to support the target users' work and goals.* The overall visualization artifact of this activity is a usable visualization system. This activity is the ultimate goal of problem-driven visualization design since it supports real-world users in their own work environments. Constructing a visualization system often involves considerations and steps not necessary for early visualization prototypes. For example, it is less common to focus on usability, aesthetics, or scalability issues in a prototype system, but these critical aspects of a final system can impact domain collaborators' ability or desire to effectively use the produced visualization tool for their set of tasks.

Another important step in the *deploy* activity is to consider optimizations needed by users to use a tool to solve their problems efficiently. As an example, imagine domain collaborators generate their data using an industry standard tool and want to be able to use that specific data format inside a newly designed visualization system. For building an initial prototype of the system, visualization designers may require the collaborators to export their data or use a script to get it into a standard format in order to visualize it. However, for daily tasks, this data process would be cumbersome, require significant training, and limit the ability of the collaborators to use the tool. A key aspect

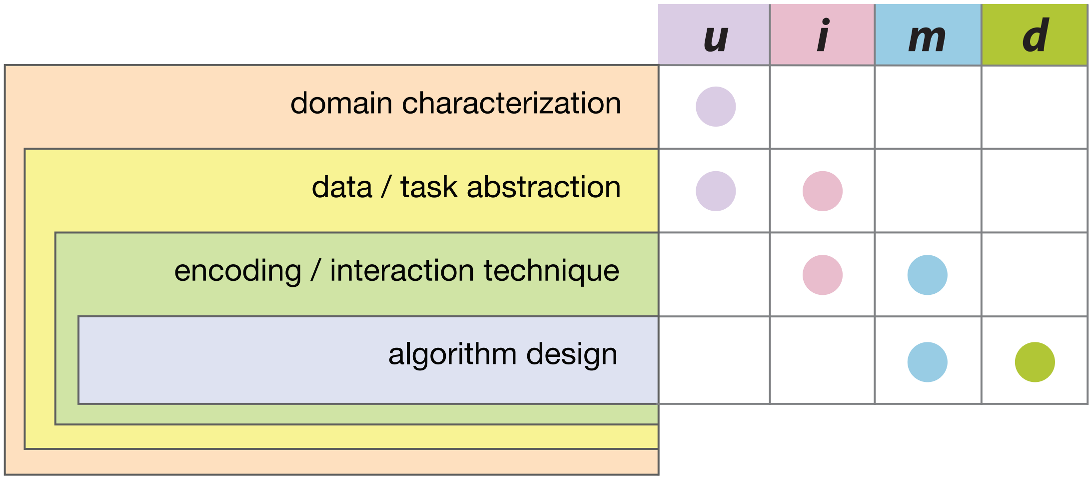


Fig. 3.4. Connections to the nested model. We illustrate the overlap of the design activity framework with respect to the levels of the nested model [17]. It is important to note that each of the three inner levels of the nested model exists across two activities in the framework; thus, a visualization designer must think carefully about with which levels of the nested model any process artifact corresponds.

of deploying a visualization system is improving these challenging aspects by having the system support domain collaborators' native data formats once the tool is deployed, so that no special steps are needed for them to use the visualization tool to solve their domain tasks.

3.7 Justifying Design Decisions

Both novice and expert visualization designers can utilize the design activity framework to reflect on the design decisions they made by tracking the visualization artifacts they produce in each activity. As shown in Fig. 3.4, three of the four design activities and their associated visualization artifacts map to two levels of the nested model, implying that a specific visualization artifact can result from different types of visualization design decisions. Conversely, a designer focusing on just one type of design decision (e.g., which data abstraction to use) will often move through different design activities to pick the right one; thus, the culmination of a complete visualization could involve moving through this framework in a complex, iterative, and *multilinear* fashion. By multilinear, we mean that a process combines forward, linear movement with cyclic, backward, and parallel movements. We discuss more about movement and design timelines in the following section.

Artifacts for the *understand* activity fall into the outer two levels of the nested model, the domain characterization and abstraction levels. These visualization artifacts of design requirements consist of acquired knowledge about the target set of users, their domain-specific questions and goals, their workflows, and the types of measurements or data they have acquired — these artifacts are referred to as situation blocks in the nested blocks and guidelines model [18]. Furthermore, design requirements can also include contextual information about the project itself, such as real-world project considerations, i.e., time, budget, expertise, etc. Visualization artifacts can touch on the abstraction level of design decisions through an identification of the tasks that users need to perform to reach their goals, as well as an initial data abstraction that describes the users' measurements in a structured way.

In the *ideate* activity, ideas encompass design decisions made at both the abstraction and technique levels of the nested model. More specifically, at the abstraction level, ideas reflect decisions made about how to structure the data or derive new data types that will best support the needs of the users. At the technique level, the visualization artifacts reflect high-level design decisions about visual encoding and interaction technique choices based on the abstraction decisions, such as choosing a specific visualization technique, while ignoring lower level decisions about the details of that technique; exploring these low-level decisions is the function of the *make* activity described next. Thus, the *ideate* activity supports a very broad exploration of the high-level design space for investigating a specific problem, leaving more detailed design decisions to later activities. Ideation is commonly considered as a separate activity in the design community [8], [35], [53], [55], [57]–[59], and this separation highlights the different kinds of design decisions made within the visualization design process.

Visualization prototypes from the *make* activity can explore aspects of design decisions made at the inner two levels of the nested model, the technique and algorithm levels. Visualization prototypes explicitly explore the design decisions related to actualizing a specific visualization or interaction technique using code, so they typically involve implementing a given visualization encoding or interaction technique and the necessary algorithms that make the prototype work. This activity is not just about implementing a given design; rather, the activity, including development or coding, also involves critical visualization design decisions [14]. For example, when using a map-based encoding for a dataset, a visualization designer might discover that several data points are right on top of each other and has to make low-level decisions on how the algorithm or encoding handles these overlapping points so that they are shown completely and without error to the user.

The *deploy* activity and its final visualization system are often constructed using methods from the field of software engineering and user experience engineering, with a focus on supporting target users utilizing the tool in a real-world situation. Thus, the visualization system must touch on decisions made at the algorithm level of the nested model, in addition to other decisions that are not necessarily about the visualization design itself, such as integration with existing software, databases, etc. As such, these additional decisions are not captured by the nested model. However, the algorithm level is important to consider, since there may be issues of scalability or the interactive speeds of the system when using the real, potentially larger datasets that collaborators use in their day-to-day workflows. These types of optimizations can even be published on their own as an algorithm improvement to speed up the visualization encoding or interactive technique in other visualization systems and tools.

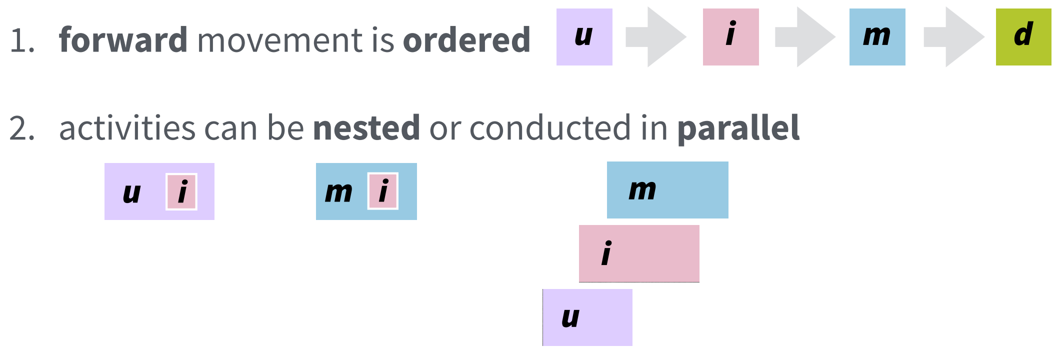


Fig. 3.5. Two movement principles for visualization design. We synthesized two ordering principles for design activities. The first principle is that moving forward must advance only to the adjacent activity (backward movement is unrestricted), and the second principle is that activities may be conducted in parallel or in a hierarchical fashion based on the design team's focus and division of work. These principles support flexible, iterative, and multilinear design across each of the different activities.

3.8 Timelines of the Design Process

In our experience, a visualization design process rarely progresses neatly through a set of designated stages; this fact motivated our synthesis of the design activity framework, which can be pieced together by designers in many different ways to best suit the needs of their visualization design project. This complex motion aligns with creative process models from the design community that already emphasize that design is messy, iterative, and multilinear [24], [61], [88]. These creative process models advocate that there is no one right way in which to engage in the design activities of a framework.

We identified two basic principles for the design activity framework when it comes to the flow of the visualization design process. First, the activities are ordered when moving forward: *understand*, *ideate*, *make*, *deploy*. A project can start with any activity, as with some of our own projects that have begun with a tool already deployed to end users, but forward movement must happen in an ordered fashion, even if the design methods used are very narrow and informal. Backward movement, however, can move to any previous design activity. The second principle is that activities can be nested or conducted in parallel, meaning that forward or backward movement to a different activity can happen within an activity, such as revisiting an understanding while brainstorming new ideas, or two activities that occur concurrently across a visualization design team. Taken together, these two movement principles support both iteration and multilinearity. We illustrate these principles in Fig. 3.5.

Many visualization process models are also characterized in similar ways, supporting ordered forward movement with iteration [8], [10]–[13]; and researchers in visualization and design suggest an overlap between stages, such as the nine-stage framework [8] and the international standard for human-centered design activities [12]. However, researchers often represent their design models

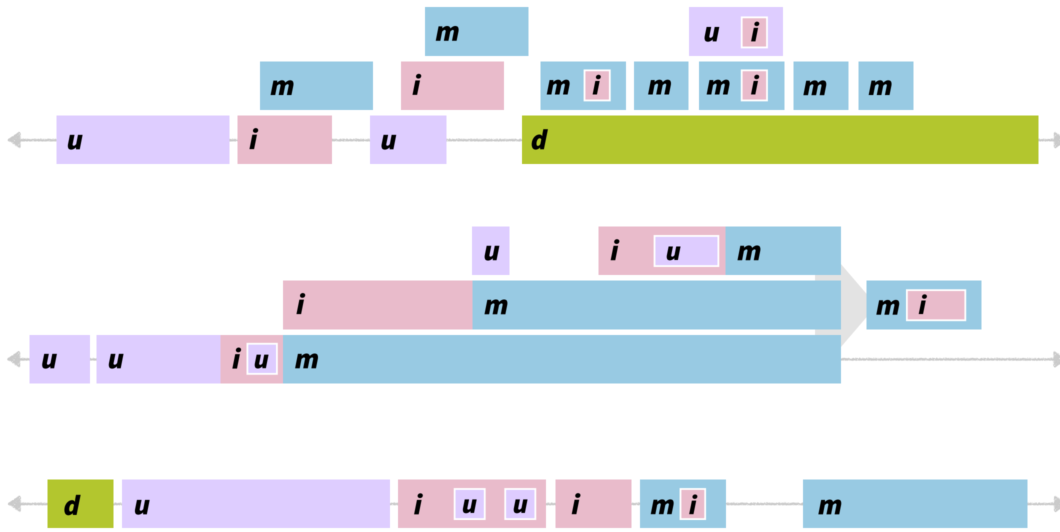


Fig. 3.6. Examples of real-world design timelines. To identify the movement principles, we summarized several design studies into concise, communicative design process timelines that capture the flexibility of backward movement plus parallel and nested activities. Each colored box corresponds to a design activity. At the top, this timeline represents the Variant View design study process [89]. In the middle, we illustrate the design process for Shotviewer [21]. At the bottom, we present our redesign project's process [1], detailed in Chapter 4.

linearly [8] or cyclically [12] and thus imply the need to start at the beginning of the process, making it difficult, for example, to capture the process of a more complex visualization redesign project. We want to ensure flexibility of the design activity framework to maximize its utility for a wide variety of visualization design projects.

To establish and showcase these two movement principles, we provide several design process timelines from different visualization projects and design studies [1], [21], [89] in Fig. 3.6. In these timelines, design activities are represented with color boxes as in Fig. 3.3. We created the top-most timeline in Fig. 3.6 from the Variant View design study using the researchers' description of their design process [89]. We constructed the middle timeline with the creator of Shotviewer [21] to capture the parallel nature of that design study. We include a detailed discussion and annotation of the bottom process timeline [1] in Chapter 4. These timelines show the flow of a project across multiple activities, from nested and parallel activities to backward movement.

Other researchers have shown the feasibility and usefulness of a design process timeline as an effective way to communicate a design process [86], [90], foster collaboration [91], and highlight some aspects of the multilinear nature of a design process [38], [62]. Communication of the design process is important not only for understanding and evaluating the visualization research process itself, but also for supporting replicability of problem-driven work. For novices, this communication can be vital as it can improve team communication of concepts and facilitate faster onboarding across skillsets. Visualization models such as the nested model [17] are now widely used to communicate design decisions made over the course of a visualization process, and we advocate for the design activity framework as a way to structure the communication and reporting of a visualization design process in a similar way.

Another unique aspect of the design activity framework is its merging of the creative arts with engineering approaches. The design timeline is one example of this concept, with complex and flexible aspects, such as the nested and parallel activity support, coupled inside a linear timeline to show project progression in an engineering mindset. By enabling this fluid coupling of design activities, the design activity framework supports many complex paths of the design process such as those in the creative arts. However, the timelines also encourage a linear focus on producing artifacts and completing methods in order to meet deadlines, which are engineering aspects. This dynamic and dual nature of the design activity framework provides additional benefits to visualization designers with a broad range of expertise and experience, since they can practice and experiment with either of these aspects they are less familiar with to gain experience and grow their skill set.

3.9 Discovering Design Methods

To help visualization designers tackle a real-world design project, we present a list of exemplar methods that designers can use throughout the design activity framework. This list contains methods commonly found in the visualization literature, as well as many more that come from the design, human-computer interaction, software engineering, sociology, and anthropology literature. We present a list of 33 methods in Table 3.1 and a more extensive list of 100 methods in Appendix A. We shortened the list by picking those that were mentioned within the framework and redesign project we present in the next chapter, along with both commonly used and potentially novel or interesting methods for visualization design.

We informally coded and characterized each method by the activities of the design activity framework in which it can be used — *understand* (***u***), *ideate* (***i***), *make* (***m***), and *deploy* (***d***). It is important to note that many methods can and often are used in different design activities. We also categorized the methods as being generative (***g***), evaluative (***e***), or both in nature. Several methods, e.g., graffiti walls (*M-43*), interviewing (*M-51*), and observation (*M-58*), have more complex characterizations than presented in this table; please see Appendix A for a more complete and detailed characterization. Additionally, we marked methods as appearing within the visualization literature (***v***). Finally, each method includes a definition and reference to aid visualization designers in bringing these methods into practice. As the design activity framework targets designers performing problem-driven visualization work, it is worth noting that many of the listed methods involve collaboration with domain experts, such as bull's-eye diagramming (*M-12*), contextual inquiry (*M-27*), paper prototyping (*M-61*), and speed dating (*M-80*).

The list is by no means a complete compendium of methods for visualization design, but rather a step toward understanding the large space of actions a designer can take throughout the visualization design process. Our goal in creating this list of methods is twofold: first, the table serves

Table 3.1. Numerous exemplar methods within the framework. We coded methods into each design activity: *understand* (***u***), *ideate* (***i***), *make* (***m***), and *deploy* (***d***). We tagged generative (***g***) and evaluative (***e***) methods, along with those commonly reported within the visualization community (***v***). We present definitions to assist designers employing methods. Appendix A contains a more extensive list of 100 methods.

#	method	*u*	*i*	*m*	*d*	*g*	*e*	*v*	definition
1	A/B testing			•	•		•	•	"compare two versions of the same design to see which one performs …better" [25]
8	artifact analysis	•				•	•	•	"systematic examination of the material, aesthetic, and interactive qualities of objects" [25]
12	bull's-eye diagramming	•	•				•		"gather a set of data (e.g., issues, features, etc.)…. plot the data on the target [diagram], and set priorities" [92]
13	buy a feature	•	•	•			•		"express trade-off decisions…. ask [participants] to purchase features …. encourage them to [justify decisions]" [92]
16	coding	•	•	•		•	•	•	"break data apart and identify concepts to stand for the data [open coding], [but] also have to put it back together again by relating those concepts [axial coding]" [93]
23	concept sketching		•			•		•	"convert ideas into concrete forms that are easier to understand, discuss, evaluate, and communicate" [24]
26	constraint removal	•	•			•		•	"barriers [are] transformed into a positive resource through which to create new ideas" [47]
27	contextual inquiry	•				•	•	•	"go where the customer works, observe the customer as he or she works, and talk to the customer" [80]
28	controlled experiment	•		•	•	•	•	•	"help us to answer questions and identify casual relationships" [82] & "widely used approach to evaluating interfaces and styles of interaction, and to understanding cognition in the context of interactions with systems" [81]
29	creative matrix		•			•			"[spark] new ideas at the intersections of distinct categories…. encourage the teams to fill every cell of the grid" [92]
35	example exposure		•	•		•		•	"excite ideas by exposing the subject to a solution for the same problem" [94]
38	field notes (diary, journal)	•				•	•		"four types of field notes: jottings, the diary, the log, and the notes" & "keep a note pad with you at all times and make field jottings on the spot" & "a diary chronicles how you feel and how you perceive your relations with others" [95]
42	frame of reference shifting		•			•			"change how objectives and requirements are being viewed, perceived, and interpreted" [94]
43	graffiti walls	•	•	•	•	•	•		"open canvas on which participants can freely offer their written or visual comments, directly in the context of use" [25]
44	heuristic evaluation			•	•		•	•	"assess an interface against a set of agreed-upon best practices, or usability 'rules of thumb"' [25]
51	interviewing	•	•	•	•	•	•	•	"direct contact with participants, [collect] personal accounts of experience, opinions, attitudes, and perceptions" [25]
53	literature review	•				•	•	•	"distill information from published sources, capturing the essence of previous research" [25]
54	love/breakup letters	•	•	•	•	•	•		"personal letter written to a product…[to reveal] profound insights about what people value and expect" [25]
58	observation	•	•	•	•	•	•	•	"attentive looking and systematic recording of phenomena: including people, artifacts, environments, events, behaviors and interactions" [25]
61	paper prototyping		•	•		•		•	"create a paper-based simulation of an interface to test interaction with a user" [96]
62	parallel prototyping		•	•		•		•	"creating multiple alternatives in parallel may encourage people to more effectively discover unseen constraints and opportunities, enumerate more diverse solutions, and obtain more authentic and diverse feedback" [97]
63	personas	•					•		"consolidate archetypal descriptions of user behavior patterns into representative profiles, to humanize design" [25]
67	prototyping		•	•		•		•	"tangible creation of artifacts at various levels of resolution, for development and testing of ideas within design" [25]
69	questionnaire	•	•	•	•	•	•	•	"survey instruments designed for collecting self-report information from people about their characteristics, thoughts, feelings, perceptions, behaviors, or attitudes, typically in written form" [25]
72	role-playing	•	•		•	•	•		"acting the role of the user in realistic scenarios can …highlight challenges, presenting opportunities" [25]
73	rose-thorn-bud	•	•	•			•		"identifying things as positive, negative, or having potential" & tag outcomes as rose, thorn, or bud, accordingly [92]
75	sample data				•	•		•	"provide real data and tasks …. illustrating [tools] with convincing examples using real data" [27]
80	speed dating		•	•			•		"compare multiple design concepts in quick succession" [25]
81	stakeholder feedback	•	•	•	•		•	•	"letting [experts] "play" with the system and / or observe typical system features" [28]
87	technology probe	•	•	•		•	•	•	"simple, flexible, and adaptable technologies with three …goals: …understanding the needs and desires of users, …field-testing the technology, and …inspiring users and researchers to think about new technologies" [98]
89	thought experiment	•	•	•			•		"think about research questions as if it were possible to test them in true experiments" [95]
97	weighted matrix	•	•	•			•		"[rank] design opportunities against key success criteria" & "identify and prioritize…opportunities" [25]
98	wireframing		•	•		•		•	"schematic diagramming: an outline of the structure and essential components of a system" [92]

as additional guidance for real-world, actionable usage of the design activity framework by finding potential methods within a specific design activity; and second, the table contains many methods that are not commonly, if at all, found in the visualization literature, and therefore provides new design methods that more experienced visualization designers could utilize to potentially enhance their design process. For example, Goodwin et al. introduce several novel creativity techniques for visualization design such as generating ideas using the method of constraint removal (*M-26*) [47].

We can use the framing of a design activity to help visualization designers find effective design methods. We define effectiveness here as a reflection in two parts: short-term and long-term effectiveness. For short-term effectiveness, a design method must successfully achieve the desired visualization artifact for the design activity — we argue that this completed artifact is one way to validate a design method. The long-term effectiveness of a method can be established when the method is used within the development of a deployed visualization tool: one that is evaluated with, and given to, real end users. Thus, visualization designers can determine if a design method was effective within their visualization project by reflecting on these two questions:

1. Did you achieve your desired visualization artifacts?
2. Did you successfully deploy a visualization tool to users as a result of this method?

Furthermore, we acknowledge that this methods table provides another benefit to using the design activity framework in real-world projects. Specifically, tagging methods as generative or evaluative is useful for visualization designers in practice. Primarily generative design methods may be commonly utilized in creative art and design for producing many artifacts. Conversely, evaluative methods may be more common in engineering approaches to winnow visualization artifacts into a smaller set. Thus, the design activity framework flexibly supports visualization designers picking and choosing appropriate design methods that can merge these creative and engineering design aspects. Along these lines, visualization designers can push themselves to use more generative or more evaluative methods, or new design methods in general, as they grow and enhance their abilities for and expertise at creating and comparing visualization tools and systems.

3.10 Summary

In this chapter, we have introduced the design activity framework as a new process model for visualization design. This model identifies four activities for visualization design: *understand*, *ideate*, *make*, and *deploy*. Within each of these activities, we specify succinct definitions, motivations, and expected visualization artifacts to be achieved. Furthermore, visualization designers can justifiably link design decisions they make with the visualization artifacts they produce through the levels of the nested model. These decisions highlight potential validation or evaluative methods to perform, and we provide a table of user-centered design methods for further discoverability of new methods to utilize. We also showcase the flexibility of the design activity framework with generalized flow principles illustrated in several design timelines for tracking and documenting the visualization design process. Together, these contributions improve upon existing design process models due to the design activity framework's overall increased achievability, flexibility, justifiability, discoverability, and actionability.

The design activity framework is specific to visualization in a variety of ways. First, this process model was formulated through reflection on many visualization design projects, so we cannot claim that the stages, components, artifacts, and methods will generalize to other kinds of design. The four design activities, while they have generalized names, do have descriptive motivations that are specific to the creation of a visualization tool or system. Similarly, the design artifacts are defined as being specific to visualization, such as task abstractions or visualization encodings. Some of the general principles of the design activity framework may be useful to other kinds of design, but we have used and validated the framework only for visualization design in a series of case studies and in an external validation with novice visualization designers.

Unlike previous design models for visualization, the design activity framework more clearly lays out multiple elements of a design stage: its motivation, visualization artifacts, and design methods. Additionally, the connection to design decisions, through the nested model, is not included in any previous visualization design process model. The design activity framework does have many parallels with the nine-stage framework, and the four design activities could be considered an extension upon the four core design phases of Sedlmair et al. [8]. However, the design activity framework is based on the concept of a design activity with clear, descriptive definitions and numerous possible design methods to employ, and it supports flexible and dynamic timelines. We also carefully constructed this framework to incorporate and merge the creative art and design aspects with an engineering approach to provide a more comprehensive snapshot of the visualization design process. For all these reasons, the design activity framework is a novel and useful design process model for the visualization community.

4

Case Study: Redesigning a System

We initially formulated the design activity framework based on our reflections on a previous project for which I had started as a novice visualization designer. For this project, we worked as a multidisciplinary team with a cybersecurity firm to tackle a redesign of that firm's visualization system [1]. We focused our redesign to create a series of sketches, wireframes, and mockups that visually communicated our visualization and interface ideas to the company's development team in order to improve their tool. Although the company's development budget and time limited the final changes to the system, we found that a reflection on this project yielded useful insights that we used in order to create the design activity framework. In particular, the visualization design artifacts and methods we utilized with other designers played a significant role in the success of this project, and we discuss one user-centered design method, qualitative coding of user research papers, in detail. We reflected on our team's overall design process by incorporating and summarizing the methods and artifacts into a design timeline. Previous models, such as the nested model and nine-stage framework, did not describe these artifacts, our decisions, and our design process fully, but we were able to more effectively capture, characterize, and explain multiple aspects of our visualization design process using the design activity framework.

4.1 Project Overview

Our multidisciplinary design team consisted of two designers, one psychologist, a visualization expert, and myself, a visualization novice. On this team, only one of the five members had used the nine-stage framework previously. We tackled the challenge of redesigning an existing visualization tool in the area of cybersecurity. As our team attempted to adopt the nine-stage framework for conducting design studies [8], we struggled to answer questions such as: If I'm not starting from the beginning, where exactly am I in the design process? What are the range of design methods that are useful at any given point? What types of visualization artifacts should I be working toward along the way? How do I know my artifacts are good, or even just good enough, when balanced against real-world constraints? We believe that these questions point to a lack of *actionability* in current visualization process models, namely the nine-stage framework for design studies. In other words, this model lacked implementable and immediately usable guidance that could help visualization designers explicitly navigate a real-world visualization design process.

This seven-month project focused on improving the usability and effectiveness of an existing, robust visualization system (RVS) for cybersecurity analysis. Analysts working with cybersecurity data focus on maintaining the security of computer networks. They rely on data about how a network is functioning, known network attack patterns, and a broad range of external sources of knowledge. Specifically, our team was tasked with providing ideas and mockups for how to redesign the visualizations and interface within RVS — the implementation of these redesigns within RVS was handled by developers at the company that developed and maintains RVS.

Over the course of our redesign project, we worked with developers, researchers, and managers at the RVS company; several Department of Defense intrusion analysts who use RVS; and several cybersecurity analysts at the University of Utah. This redesign project included several real-world constraints for our design team, namely a strict time frame for producing redesign ideas, limited funding for implementing our ideas by software developers, confidentiality issues surrounding cybersecurity data, and the engineering realities of working within a large software system. The nine-stage framework and also the nested model were unable to clearly capture and track these design constraints and the complex, collaborative process stemming from multiple team coordination, both of which were essential to our project.

When we started this project, our design team was stuck because the nine-stage framework was tailored to design study work requiring close collaboration with domain experts, which we did not have in this project. This constraint significantly changed the possible methods we could perform and artifacts we could obtain. After we finished the project, we researched other design process models for additional guidance, and our reflections of these models for the visualization design process resulted in the creation of the design activity framework. This new data visualization process model generalizes to visualization design, not just design studies, and the guidance and terminology better match the collaborative activities we underwent. For this case study, we will describe our redesign project and design process using the design activity framework. We will focus on a single design activity at a time. We present this discussion in rough, chronological order. In this discussion, we incorporate the methods we utilized and the visualization artifacts that were achieved. Each method number we reference stems from the collection of example methods in Table 3.1 and the full list of 100 methods included in Appendix A, such as the method of controlled experiments (*M-28*).

4.2 Deploy Activity

Since our redesign project focused on analyzing an existing visualization system, the RVS, we started our design process in the *deploy* activity. Rather than test RVS and simply clean up usability and aesthetic issues, however, our design team was tasked with thinking of the broader task of cybersecurity analysis, the needs of users within that workflow, and the role of visualization for exploring computer network data. Ultimately, the RVS company was interested in incorporating new visualization components into their tool.

Although *deploy* is commonly the final activity for a completed, successful visualization system, evaluating a deployed system may reboot the entire design process to any earlier design activity in order to extend, edit, or even redesign the system. In our redesign project, we started in the *deploy* activity with the existing RVS tool as the given *deploy* visualization artifact, forming a constraint within our project. We received a copy of RVS in order to understand what needs it currently addressed and what constraints it already contained. We used a walk-through tutorial and sample dataset (*M-75*) built by the RVS company to explore the features and efficacy of the tool.

Our analysis of RVS revealed that it was necessary for us to take a step back to the *understand* activity so that we could better discern the needs of cybersecurity analysts. Since the final visualization system is managed by the RVS company, we did not revisit the *deploy* activity ourselves, but we do know that the changes that the company made to the tool were eventually pulled in as part of their visualization product. However, the scope of our redesign visualization project only started at this activity and took a step back to envision other possible encodings and techniques to improve the existing visualization system.

4.3 Understand Activity

The field of cybersecurity analysis has many types of users, from those within companies who maintain their own networks, to the military, which maintains and monitors traffic across a global network grid. A number of cognitive scientists have spent significant time observing and interviewing cybersecurity analysts [99]–[101] across these different networks. We used the published work from these experts to form our base understanding of the field as we had limited access to cybersecurity experts ourselves. First, we conducted an extensive and broad literature review (*M-53*) across a series of 40 articles from three key domains: cybersecurity visualization, situational awareness, and cognitive task analysis. From this review, we informally evaluated the articles based on their relevance and descriptive quality, isolating three of the articles [99]–[101] as the best representative samples with the highest impact for forming our domain characterization for the visualization of cybersecurity data. We discuss this method in more detail and reflect on its use later in Section 4.7.

Next, for these three articles, each member of our team did an informal open coding of the papers (*M-16*) to pull out salient themes. We each tagged information broadly and then adjusted these tags

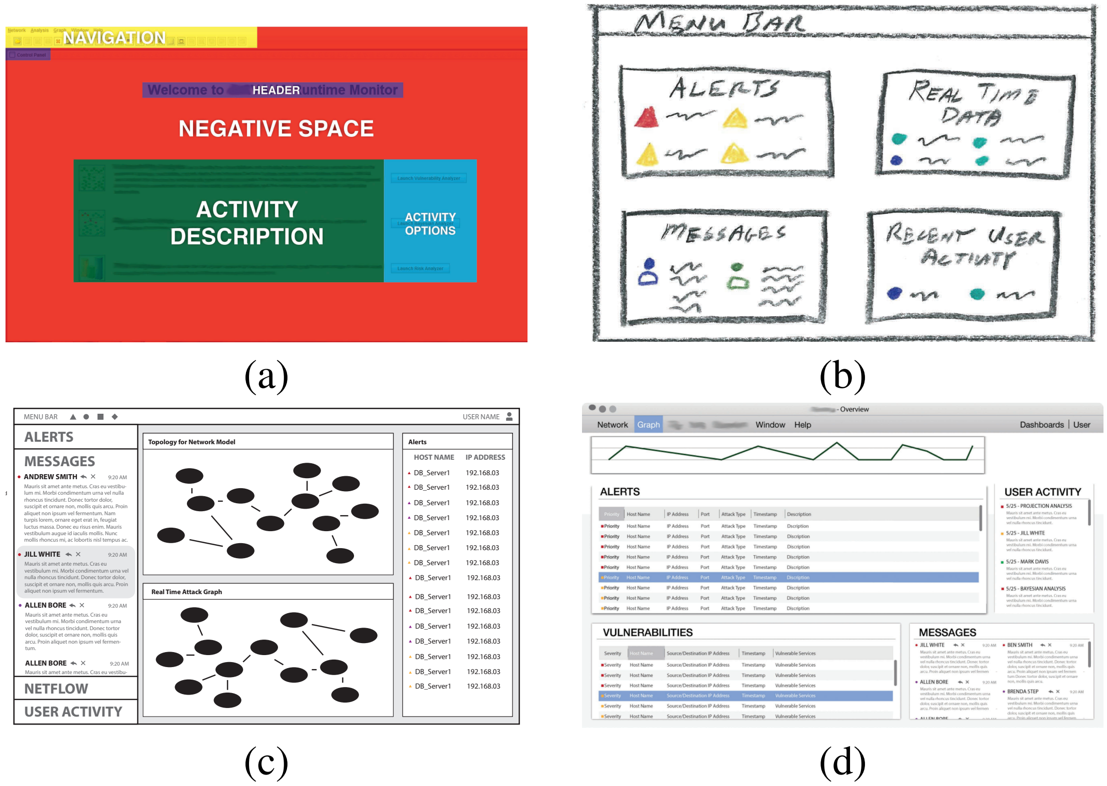


Fig. 4.1. Visualization artifacts for the redesign project. This overview of the project illustrates our team's (a) software analysis, which resulted in (b) initial concept sketches and (c) wireframes. With a greater focus on the details, we moved into the *make* activity and began (d) laying out interface components.

as a team over a series of meetings to organize and consolidate the key insights we pulled from the papers. These insights formed our initial set of visualization artifacts, which pointed to a number of unmet needs and opportunities for visualization research. Some of these artifacts included design *opportunities*, such as supporting provenance-based tasks, increasing the scalability of visualizations to real-world datasets, preserving data context as it is filtered across many different visualizations, and optimizing the representations of temporal data.

We revisited the RVS system with these design opportunities in mind. Since we were working with an existing, deployed version of the RVS software, we performed a broad artifact analysis (*M-8*) on the current software architecture, illustrated in Fig. 4.1(a). By examining the workflow supported by RVS, we identified which opportunities the tool already supported and which aspects of the tool could be improved, and then evaluated these against our initial list of design opportunities. These findings were combined with our project-specific *constraints* and *considerations*, which included four months of the visualization team's time, one month of a developer's time, and existing visual conventions in the field of cybersecurity such as highlighting critical alerts in red.

Lastly, we conducted a series of semistructured interviews with different stakeholders to identify needs and aspirations (*M-51*): a developer who works on RVS and several cybersecurity analysts and managers at the University of Utah. Based on this feedback, we met as a design team and informally evaluated and filtered the list of design opportunities by reaching a group consensus on those we felt best addressed the unmet needs of our target users, balanced against the strengths and weaknesses of RVS and taking into account the real-world constraints and considerations of the project. The final thematic design opportunities for our visualization redesign were 1) interface usability, 2) workflow improvements, 3) tool desirability, and 4) temporal data representation. We also developed a more low-level list of all visualization artifacts, such as a detailed data and task abstraction.

4.4 Ideate Activity

After our design team had identified the specific design opportunities, constraints, and considerations for our visualization redesign, we were ready to come up with ideas to meet user needs. The *ideate* activity took several months as we sketched out a series of possible ideas for modifying the current design of RVS. First, each member of our team began to develop separate concept sketches (*M-23*) tackling a specific design opportunity, as illustrated in Fig. 4.1(b). We chose this first method based on the experience of the designers in our group as they were used to sketching out possible concepts. We then came together as a team to review these sketches and evaluate them based on which ones possessed the most potential for impacting a redesign of RVS. This evaluation process was very informal; we met as a visualization team and discussed some of the pros and cons for each concept, ultimately coming to a group consensus. These meetings were conducted as informal design critiques. We also shared a subset of these idealized sketches with the researchers and managers at the RVS company in order to further validate, filter, and confirm the different design concepts.

The ideas and concept sketches relied on two key data abstractions that we identified: computer networks and time series data. For example, one of our ideas for the visualization of a computer network is a simplification of the nodes into subgroups and supporting details-on-demand in order to allow the visualization to scale to a larger dataset. Scalability can be a later concern in the *deploy* activity, but we found that perhaps revisiting the data abstraction could help simplify and improve the resulting visualizations so they could handle and show more data at once. For the time series data, we explored ideas for derived data, such as network alerts or general traffic and activity. For each data type, we explored various encodings and interaction techniques that would scale to different levels of the data; this scaling is critical due to the quantity and spread of real-world cybersecurity data.

The concept sketches proved to be useful in exploring different ideas, but we wanted to explore some of these ideas in more depth and detail. Thus, we synthesized the paper concept sketches into very low-fidelity paper prototypes (*M-61*) that highlighted interactions inside the tool. These ideas were eventually finalized into more concrete wireframes (*M-98*), shown in Fig. 4.1(c), to mimic the look and feel of a real tool. Again, we evaluated these wireframes very informally, internally as a visualization team and showed our ideas to different members of the RVS company, to check that our *ideate* visualization artifacts (sketches, paper prototypes, and wireframes) were on track for meeting the analysts' needs. Due to the main constraint of time within the project, we were unable to evaluate these wireframes more formally with the cybersecurity analysts.

4.5 Make Activity

The *make* activity was conducted in small part by our visualization team and also in part by the RVS development team. As a design team, we generated a number of digital mockups; several of these were detailed wireframes (*M-98*) that focused on the layout of different visualizations and interaction mechanisms, as shown in Fig. 4.1(d).

In addition, we also mocked up more detailed prototypes (*M-67*) that showed how the different visualizations would link together through user interactions. These prototypes synthesized all our design ideas into a revised interface, as illustrated in Fig. 4.2. The purpose behind this method was to envision what RVS *could* be even though a complete software implementation was beyond what RVS developers could produce given our constraint of time. We considered real-world datasets and user workflows when creating and formulating both the digital mockups and detailed wireframes, which are the visualization artifacts we created in the *make* activity. Even though some designers may consider these artifacts as ideas and not prototypes, our visualization team had finalized making decisions at the abstraction level and focused not on new ideas but on encodings and interaction techniques in these artifacts, so we argue that this goes beyond the *ideate* activity and resulted in visualization prototypes that could be tested with users.

After we finalized these detailed and revised mockups, the RVS development team focused on implementing these concepts within the existing software. We note that the distinction here between the visualization team and development team is somewhat unique to our redesign project; most often in visualization design these two groups of people overlap or work closely in cohort. As a result of this implementation process, the development team created a software prototype (*M-67*) that they evaluated with several network security analysts who work with RVS. The RVS company sought a quick and easy approach to minimize the time needed by network security analysts to participate; thus, this evaluation consisted of an A/B testing method (*M-1*) coupled with a questionnaire (*M-69*). This evaluation received positive feedback over the previous version of RVS, which we took as a validation of the design ideas that had became concretized within the final visualization artifact: a new prototype of RVS. Although the company behind RVS likely continued implementing changes and deploying aspects to their visualization system, we were not involved with this design and development process.

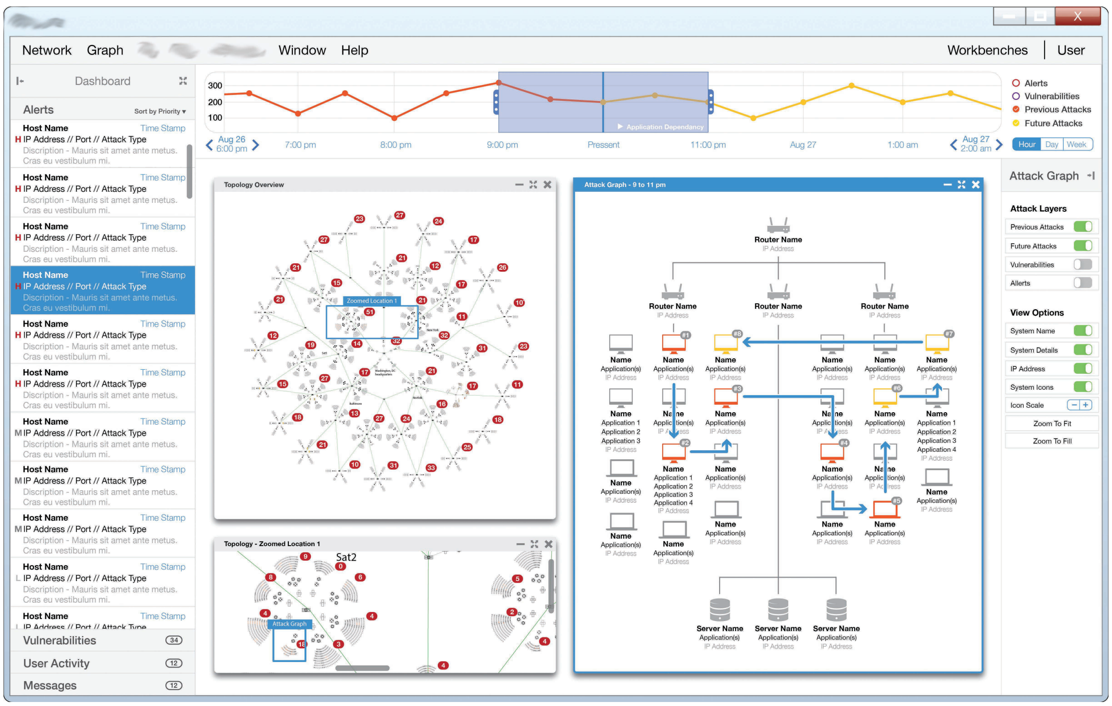


Fig. 4.2. Final visualization redesign artifact. Our team created this artifact as part of the *make* activity for the redesign project: the design of a fully detailed revised digital mockup interface. This artifact contains significant changes for the final visualization system from a new timeline visualization to a more comprehensive overview and detail view of the network alerts.

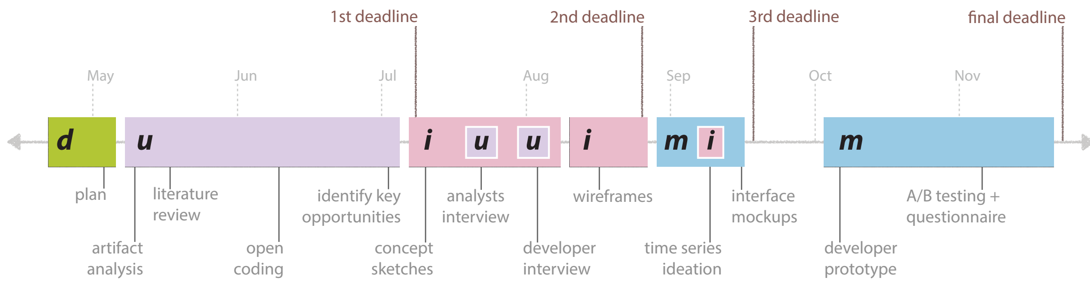


Fig. 4.3. Redesign project timeline. This timeline provides a concise overview of our redesign project. Key design activities are located in the middle, highlighting both backward movement and activities nesting within each other when team members worked separately. We encountered several key time constraints, or deadlines, listed at the top. Toward the bottom, we also highlight numerous design methods and visualization artifacts utilized.

4.6 Project Timeline

We provide a timeline for our redesign project in Fig. 4.3, where design activities are represented in the timeline as described in the previous chapter. This visualization project's timeline shows the flow over time as we conducted multiple design activities, including nested activities and both forward and backward movement. The timeline is annotated with many of the design methods we used and some of the visualization artifacts we created and validated during our redesign. Characterizing our design process and creating this timeline were possible using the language and structure of the design activity framework, which previous visualization design models did not support or outline.

We show this timeline in more detail in Fig. 4.3 than previously shown in the bottom of Fig. 3.6. The difference of these two timelines shows the flexibility of the design activity framework to capture both the high-level view of a design project and the pattern or flow of design activities as well as the low-level detail of specific visualization artifacts and design methods utilized. In the next section, we explore in detail the user-centered design method of qualitative coding for three papers, which was our team's first primary focus for the project. During this method, we uncovered codes and design opportunities that could be useful for other visualization designers in this space, and we reflect on this method as a substitute for users in the visualization design process.

These design timelines could be further enhanced by highlighting levels of the nested model or details of visualization design decisions made throughout the process. We found that keeping track of these activities can help visualization designers meet deadlines by focusing on specific design activities and visualization artifacts as well, which is why we highlight these deadlines explicitly in the timeline. These visualization design timelines can help the communication of a design process internally with a design team during the process or externally after the fact with research colleagues or managers to provide a concise overview of the results and work conducted.

4.7 Qualitative Coding Design Method

To promote the discovery and adaptation of new design methods for novel problems, we focus on our team's use and the impact of a user-centered design method: **qualitative coding**. As we explained in the previous chapter, there is a benefit to exploring and validating the use of novel and unique design methods in the context of real-world visualization projects. The *qualitative coding* method played a key role in the *understand* activity of this project. For this design method, we first discuss our motivation behind why we utilized this approach in the context of our design process. Then, we highlight the visualization artifacts achieved, followed by results and implications of what we learned and a discussion of the method's efficiency, effectiveness, and limitations. Lastly, we present recommendations for using this method in the context of cybersecurity visualization design.

When tasked with redesigning a large cybersecurity tool, our design team had limited access to end users. Despite the fact that a fully deployed tool already existed, we were taking a step back to find users' needs in the first design activity: *understand*. Our motivation in this activity was to better understand the needs and design opportunities for network security analysts to redesign the firm's tool. But how do we identify user needs without direct access to end users? Many researchers have studied users in this domain from a variety of perspectives, particularly with cognitive task analyses. For this project, we built on this rich existing body of knowledge through qualitative coding of three cognitive task analyses.

We took inspiration from the social sciences [93] to help structure our analysis by performing an open coding on three key cognitive task analysis (CTA) papers from the field. Qualitative researchers often use coding as a method to organize, structure, and consolidate information into a structured framework. Open coding is a subset of qualitative coding that focuses on the original content to form the codes the researcher makes, as opposed to axial coding, which incorporates existing categories to tag onto the source material [93]. This method has been utilized by visualization researchers to perform various post hoc analyses [28], [30], [102], [103], but we had not seen this method used in the *understand* activity to pinpoint user needs for cybersecurity.

After two weeks of extensive literature review, four members of our design team identified and performed a deep reading on these three CTA papers [99]–[101], pulling out key quotes, paraphrases, and models. Each piece of data corresponds to rows of our coding table, and we met several times over a month to better organize, iterate on, and consistently tag this information across all three papers. These meetings and iterative coding process were crucial to allow the design team to come to an agreement on our final codes. After a month of open coding the three papers, we consolidated the data in a final meeting.

category	sub-category	evidence	author	pages
communities	attackers	"... increasingly sophisticated technical and social attacks from organized criminal operations"	D'Amico	19
data	external	"information published on hacker websites"	D'Amico	29
data	processed	"incident report, intrusion set, problem set from other organizations, information about the source and or sponsor of attack" & "incident reports are [often] textual documents"	D'Amico	35
data	raw	"network packet traffic, netflow data or host-based log data"	D'Amico	25
design guidelines	tutorial	"tutorial on how to get started; not just the user's manual .... certification process so people can become certified"	Erbacher	212
design guidelines	uncertainty visualization	"visualization should have a weight based on the accuracy of info" & "force-directed graphs where trust is the primary spring force"	Erbacher	210,212
other	metaphor	"Cyber security is essentially a human-on-human adversarial game played out by automated avatars. "	Fink	46
phases	situational awareness	"During the first stage, a CND analyst acquires data about the monitored environment, which is typical of the perceptual stage of situation awareness."	D'Amico	32
responsibilities	communication	"importance of analyst communication in the data transformation"	D'Amico	30
roles	managers	"most were active analysts; a few were managers"	D'Amico	23
roles	network analyst	"computer network defense (CND) analysts"	D'Amico	19
workflows	investigate	"If a vulnerability scan returned a suspect IP address, he would then have to go through several different tools in different windows to get information about the IP, such as the host name, its location in the network or building, its OS version and update status, its owner, and the owner's phone number."	Fink	49

Fig. 4.4. Sample of qualitative codes. We generated these codes over a month of consuming three cognitive task analysis research papers, and iterated over the codes until the final version. We established these codes based on a series of evidence, both quotes and paraphrases, from the source papers. Some example categories include the cybersecurity data, general design guidelines, phases, roles, responsibilities, tools, and workflows. As a result of this design method, our team pinpointed a series of user needs to consider for redesigning a cybersecurity tool.

4.7.1 Visualization Artifacts

We present a sample visualization artifact resulting from our coding method in Fig. 4.4. Each piece of information is organized across one or more papers and into a hierarchy of categories. At the top-most level, we identified categories such as data, design guidelines, phases, roles, responsibilities, tasks, terminology, tools, and workflows. Additionally, we tagged information with subcategories on a finer scale. A more complete table of the data can be found on the project's website.*

Focusing on the data from these three CTAs enabled us to identify user needs without the user, as we had limited access to cybersecurity analysts. Over the course of a few weeks, our design team synthesized the codes into a set of distinguishable design opportunities, such as provenance, scalability, usability, desirability, data type handling, and a data hierarchy continuity. We used our knowledge from the qualitative coding method to prioritize this list and distinguish opportunities with the most potential to impact cybersecurity analysts. This method produced our final thematic design opportunities for improvements to the existing tool: usability, workflow improvements, desirability, and temporal data representation.

* http://mckennapsean.com/projects/vizsec-design-methods/

4.7.2 Results and Implications

After identifying key design opportunities, our design team iterated a series of ideas for the company to improve their tool. We sketched out and detailed a more usable welcome screen, added a widget for sharing messages among analysts, highlighted recent user activity to promote sharing, visually clarified distinctions between vulnerabilities and alerts, and created a new overview timeline visualization to coordinate all views. A software developer incorporated these changes, and the updated tool was tested with Department of Defense analysts using an A/B evaluation method. The result of this evaluation was that the redesigned tool was more usable and effective than the previous design.

Lastly, the qualitative coding method enabled us to identify extensions to a well-known data hierarchy model for cybersecurity situational awareness [99] — we present this extension in Fig. 4.5. The original data model describes how analysts process, filter, sort, and select datasets, as datasets transfer from raw data into situational awareness. Our extensions highlights the data feedback loop, clearly shows the outputs from this feedback loop, and provides identification of tasks for filtering the data across levels.

4.7.3 Discussion of the Method

The qualitative coding method was efficient as compared to more complex methods, such as a multiple-analyst cognitive task analysis; we conducted the qualitative coding in under two months. As for the effectiveness of this method, we were able to transfer our user needs into a set of concrete design opportunities to produce the desired outcome: understanding of user needs without direct access to users. These design opportunities led to the final redesign of a deployed tool that analysts found more usable and effective than before. The complete table of our coding results can be utilized by others to identify, categorize, and prioritize different user needs in future cybersecurity design projects. A limitation to this work is that it focuses on the details for only three papers; these results may be extended by coding additional papers from this field. Furthermore, one caveat to this approach is that published research may not reflect all the nuances of an operational environment. Thus, this method should not simply be used to replace access to real users.

4.7.4 Recommendations

- Start your coding method on a few papers to develop an initial set of codes; select papers from appropriate venues: e.g., in the field of cybersecurity visualization consider venues such as VizSec, VIS, CHI, HFES, Behavior & Information Technology, Computers & Security, FIRST, HST, AMCIS, SAM, CyCon, FloCon, CogSIMA, DHS CATCH, HCI HAS, CTS SECOTS.
- On the first pass, highlight and tag key pieces of information; we suggest starting with the categories we identified for cybersecurity visualization.

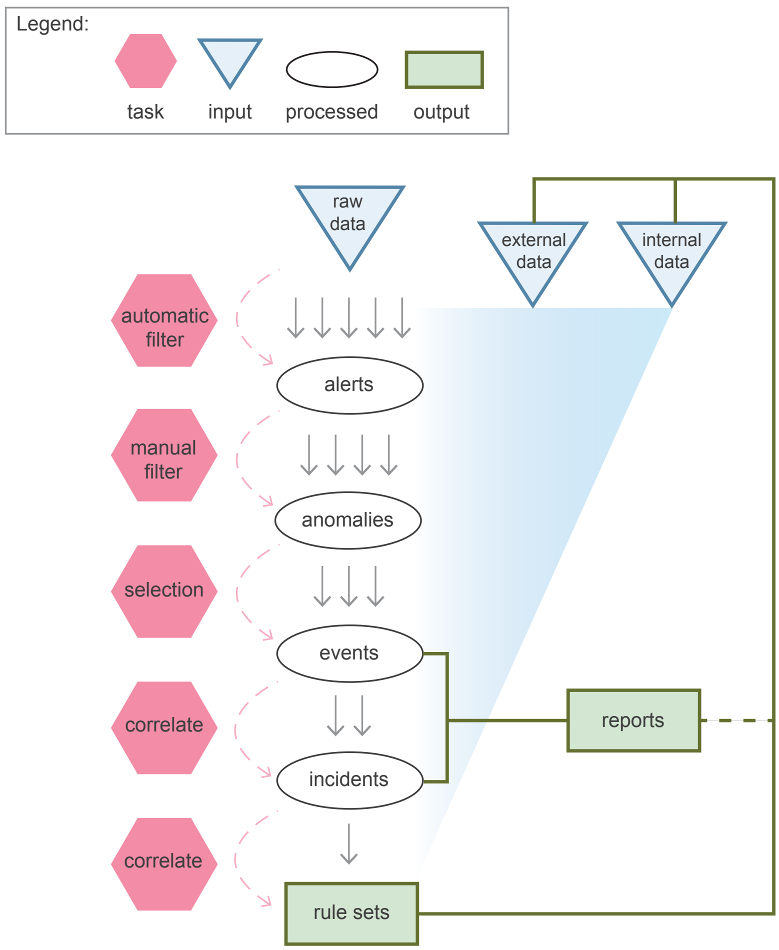


Fig. 4.5. Extended data hierarchy model. We created an extension to the data hierarchy model presented by D'Amico et al. [99], highlighting how various results feed back to raw data, while also pinpointing several key tasks. We established this extension as part of the qualitative coding method, which we used to motivate the redesign of a software tool. This high-level model enabled our team to consider and target specific stages in our redesign.

- Limit the time and scope on your first pass of coding; spend more time to meet as a team and agree on codes.
- Once you reach a consensus on codes, expand to more papers and divide up the work, allowing some overlap in coverage for consistency.

4.8 Summary

In this chapter, we discussed a visualization redesign project, in which our team faced unique design challenges that we were unable to explicitly capture using the existing nine-stage framework for design studies. To show the achievability and actionability of the design activity framework, we walked through the different design activities our team navigated through and the associated visualization artifacts, such as a software analysis and digital mockup interface. Next, we showed how the design activity framework timelines can succinctly communicate and describe a visualization project in its entirety. We concluded by performing a deep dive into a user-centered design method, qualitative coding, to illustrate the importance of discovering new design methods in order to adapt to a project's constraints and achieve success.

5
Case Study: Cybersecurity Dashboard

In this chapter, we discuss another case study for visualizing cybersecurity data, specifically, building a cybersecurity dashboard to support visual communication across multiple types of cyber users [2], [4]. One of the challenges in the domain of cybersecurity is limited access to end users. We found certain design methods to be effective with this design constraint in designing data visualizations, such as personas and data sketches [2]. Through a collaboration with cybersecurity researchers and users, we utilized the design activity framework to structure a design process that studied how cyber information is communicated among network analysts and managers. We then created a tool to aid that communication, BubbleNet [4]. Using the design activity framework as a backbone for this project, we balanced human-centered design methods with an informal agile development process to produce a useful and effective dashboard for domain users. We were also able to build upon and repurpose the design opportunities we had identified with existing users from our redesign project into this dashboard project [4]. We examined the role of two user-centered design methods, user personas and data sketches, that helped us find appropriate visual encodings for the dashboard. Furthermore, we conducted a formal usability study with a standardized quantitative questionnaire in order to validate the usability of the final dashboard tool. By reflecting on this design process, we have found that visualization design artifacts played a crucial role in communicating and reporting on the design process, and we have observed how the design activity framework can help successfully shape a real-world visualization design study.

5.1 Design Study Motivation

Over the past 10 years, roughly two *billion* pieces of digitized personal information have been lost or stolen, largely by hackers [104]. Several noteworthy breaches include the Sony Pictures' discovery that over 100 terabytes of data ranging from films to employee information to sensitive business documents were copied off their networks; the publication of names, addresses, phone numbers, and emails by hackers with administrative access to the United States' largest bank, JP Morgan Chase; and the leaking of sensitive personal information of T-Mobile customers from a breach within the Experian credit agency, everything from names to social security and passport numbers.

Such hacks are becoming increasingly prevalent and sophisticated, making the maintenance of a safe and secure computer network challenging, yet critical. Maintaining security on these computer networks is tricky, particularly due to the scale of the data as well as the constantly evolving nature of cybersecurity attacks [100], [105]. Often, these attacks require a human interpretation in order to uncover, stop, and recover from them [99]. Network analysts struggle with a very data-intensive task for which it is easy to make mistakes, errors, and miscalculations [100]. Visualization is one way for analysts to both explore and present this large data space, but analysts have been known to be hesitant about trusting visualizations for their own workflows [101].

In this chapter, we describe a design study focusing on the domain of cybersecurity. In this design study, we worked with two dozen cybersecurity experts over the span of two years with the goal of improving how analysts discover and present interesting anomalies and patterns within computer network data. To the best of our knowledge, this is the first end-to-end design study within this domain. Conducting the design study presented an interesting set of design constraints: limited access to the analysts and data, multiple types of end-users, and deployment limitations. Some of these constraints go against guidelines for conducting design studies from the nine-stage framework, such as arguments for an up-front winnowing of users and collection of data [8]. Addressing these issues, however, allowed us to validate a number of other guidelines for incorporating user-centered design methods into a cybersecurity project [2], as well as for making use of a variety of discourse channels [106]. By reflecting on the use of these channels and design methods, we found explicit connections of this design study to the design activity framework based on the project's steps, design methods, and visualization artifacts.

The primary contribution of this design study is the design, evaluation, and deployment of an interactive dashboard, BubbleNet, for visualizing patterns in cybersecurity data. BubbleNet is designed to not only support the discovery of patterns, but also facilitate presentation of these patterns to various stakeholders. We discuss a problem characterization for this domain, along with a data and task abstraction. A secondary contribution of this work is a detailed discussion of the design process, including use of several different user-centered design methods [2], as well as an application of the channels of discourse strategy [106].

In the first part of this chapter, we discuss related work for cybersecurity visualization and then describe the data and task abstraction. Next, we examine a methodical design process for the unique design constraints we encountered along with a detailed discussion of two specific design methods useful for data visualization design. We then evaluate the BubbleNet dashboard both with a usability study and then through deployment to real users. Lastly, we reflect on implications from what we have learned while building the dashboard as applied to both the cybersecurity and visualization communities.

5.2 Related Work in Cybersecurity

The tasks of discovery and presentation are open challenges in terms of visualization for cybersecurity. Many visualization tools and techniques are designed to fit the data, not the users [103]. Furthermore, visualization and cybersecurity research is largely evaluated with use-cases involving toy datasets and researchers, not practitioners in the field [103]. In addition, very few tools have considered how to present cyber information to stakeholders with less technical experience and knowledge, such as IT personnel or network managers. Large organizations often have analysts working together in teams and with a variety of other individuals, such as their managers, in order to convey priorities and matters of importance to those in leadership roles who make decisions [2], [107].

Numerous cybersecurity researchers have adapted existing visualizations for data in this domain, but very little of this work has tested the usability or utility for network analysts. Different researchers have plotted cybersecurity data on bar and scatterplots [70], [108], [109]. Other researchers have explored using a heatmap or matrix to encode various attributes and hierarchies within the data [68], [69], [108], [110]. Parallel coordinates have also been utilized by several researchers to visualize multiple dimensions of data [111]–[113]. Goodall and Sowul went beyond a single parallel coordinates view with other details-on-demand visualizations such as charts and maps into a simple dashboard [114]. We noted that there is potential for us to create a dashboard that combines and links multiple visualizations together and then evaluate its usability and utility with end users.

Visualization research has sought out novel visual representations tailored to cybersecurity data. Network graph layouts have been adapted and focused within this domain [115]–[117]. Map-like visualizations of the entire Internet seek to preserve the spatial location of similar types of computers across multiple datasets [118]. Aggregating sliding slices of time is discussed by Fischer and Keim in order to support the workflow of network analysts dealing with large quantities of data [119]. These techniques could be useful, but most of them have not been evaluated with respect to their usability or effectiveness for network analysts with real data.

A number of cybersecurity researchers have studied the usability and effectiveness of their tools, but there is no common evaluation framework to utilize [103]. Researchers have developed custom surveys [69], [120]–[122], which make comparison difficult and may not account for response bias [123]. Leschke and Nicholas evaluated a tool with a standardized usability survey [124] and others have performed formal user studies [125], [126], but none discuss deployment. Landstorfer et al. designed a visualization in a user-centered design process but garnered only initial user feedback [68]. Hao et al. worked with analysts to showcase the utility of web-based visualization dashboards for network security but did not employ the users' own data [70]. Although visualization researchers have worked with users, we have found no end-to-end design study in this space, from abstraction to deployment.

5.3 Problem Characterization and Abstraction

The outer two levels of the nested model focus on the definition of a specific problem domain and the types of data and tasks that users perform at an abstract level to map to other visualization challenges [17]. To characterize the domain problems and opportunities, we analyzed real cybersecurity datasets, talked with a variety of researchers and end users, and surveyed a wide array of related research into cybersecurity users. These activities largely fall into the *understand* activity, to better understand user problems and challenges, but there are also some aspects of the *ideate* activity, as researchers analyzed data and created initial prototypes to explore possible variations of the data schema, such as with data sketches. Many of the abstractions we present here consist of high-level *understand* artifacts, but some of the terminology and realizations were adapted over time, even after we built the dashboard and documented our research process to determine more concise and cohesive definitions for this domain and the datasets.

Most domain research in cybersecurity focuses solely on data analysis, but the task of presentation is a vital one for network analysts, as information must often be conveyed to other people for decisions to be made [2]. Often, this information to convey and decisions to be made surround a problem or an incident [99]. One analyst we spoke with summarized why presentation is challenging: *"Pictures are great when going up to management because you have 60 seconds to make your case"* (analyst #4, or A4). Numerous cybersecurity incidents can result in negative outcomes, such as information disclosure, theft, and denial of service [127].

Cybersecurity includes a variety of data types such as logs of computer functionality, but network security is a subset that focuses on multiple computer interactions with a base unit of a **network record**. A *network record* is metadata associated with the communication between two computers. The metadata can include a variety of information, such as time, location, priority, category, and various other attributes, collected from the details of the data such as the timestamp and IP address. The variety of network security datasets includes raw packet capture, net flow, intrusion detection systems, and firewall logs. Each dataset corresponds roughly with network records, but the key differences are the associated attributes or metadata.

The basic unit of network security analysis is a **pattern**, a collection of network records that represent some recurring or abnormal behavior, which can be benign or malicious. One way to create patterns is to summarize or aggregate records in different ways such as those coming from a specific computer, general location, or subsets of time. Benign patterns represent typical, authorized network records, such as typical outgoing web traffic along port 80. However, patterns can be malicious, such as a network scan from a single external computer in order to find vulnerabilities or disrupt an organization's network. These malicious patterns can be a collection of many network records such as a network scan or even a single record where a hacker exfiltrates a sensitive document.

Pattern recognition and finding anomalies is a crucial aspect for data science and machine learning in particular. Several researchers have adopted machine learning techniques for cybersecurity [126] and also for finding anomalies in social media analysis [128], [129]. These researchers discuss the rich and deep applications of machine learning for each domain. Due to the large scales of data in cybersecurity, these techniques can and often are utilized to find subsets of potentially interesting network records to visualize, but humans are still frequently required to analyze these results and are a critical component of this triage process [126].

Another way to formulate patterns is to consider different aggregations of network records, such as time and location. Many cybersecurity visualizations have been developed for showing hierarchical time-varying aspects of the data [111], [119]. From working with users, we found that aggregation to a larger scale by *hours* and *days* is both useful and interesting. Network security datasets are commonly aggregated by IP address, and these datasets can be visualized in many ways, from IP grids to Internet maps [68], [69], [108]–[111], [114], [118]. Aggregation of computers can also occur by their location of an IP address, through such databases as MaxMind GeoLite2 [130], used by other visualization tools, such as EMBER [131]. We found that geolocation is the simplest and most intuitive way to present cyber information to different users. Although not ideal, location can enable users to formulate patterns that correspond to geopolitical entities such as *countries*. For visualizing anomalies, it is also useful to compute statistical information such as averages.

For this design study, the task focus was on the discovery and presentation of cybersecurity patterns. Presentation of patterns requires simple and easily understood visualizations for consumption by users who are not domain experts. Discovery of patterns is an important part of network security analysis, encompassing tasks identified by previous researchers such as perception, detection, and monitoring [99]. Two analysts equate discovering these patterns to finding a needle in a haystack, and the importance of aggregation is illustrated by this analyst's insight on our aggregation choices of hour, day, and country: *"We would have never have seen that [pattern] any other way, maybe if we even had [data] formatted a different way that pattern would have never emerged"* (A1). Finding patterns can be particularly challenging since cyber attackers are dynamic and constantly change their methods. For both discovery and presentation, some important tasks include the ability to *identify* interesting patterns as well as *compare* patterns to find differences. For example, an interesting pattern could be activity at a certain hour of the day or a specific attribute between two countries.

5.4 Design Process

As we conducted this design study, we focused on validating the utility of the design activity framework to capture the design methods we used, the resulting visualization artifacts, and our design rationale. As a result of using the design activity framework, we created a dashboard for visualizing

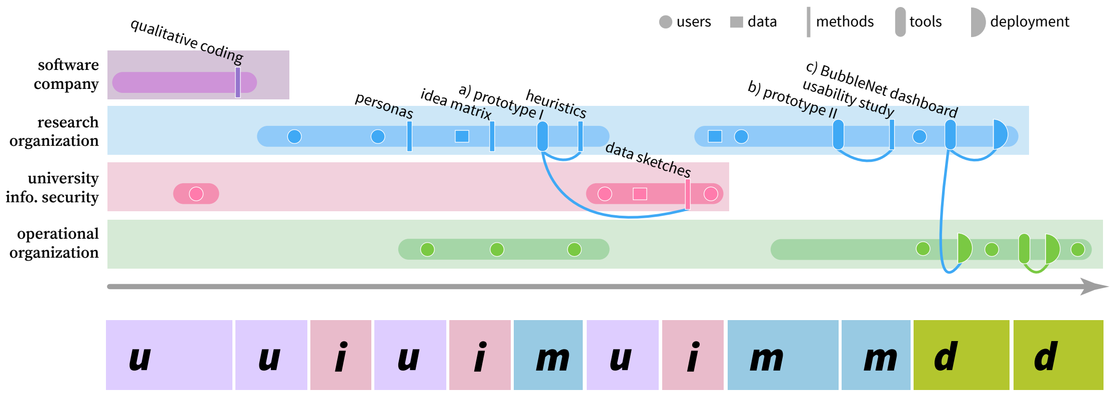


Fig. 5.1. Overview of our design process. Four distinct channels played a role in BubbleNet's design. The first (top-most) channel was previous work, and the second and fourth channels involved users in two distinct settings, both research and operational. The third channel involved a network analyst from the University of Utah. Each channel incorporated different sets of users and data, but the final design and deployment occurred due to the interaction of artifacts and user feedback across all channels. We highlight the various design activities conducted below the main timeline.

cybersecurity patterns. To present these patterns, we emphasized users beyond the network analyst. As such, it was necessary to incorporate these other users, their needs, and workflows into the design process in order to create the final BubbleNet dashboard. This design process highlighted key insights into the connection, similarities, and differences of user-centered design and a design study. These insights make this work unique compared to past user-centered design papers for cybersecurity. In particular, we found that the framing of the design activity framework succinctly captured, described, and guided our design process more effectively than the nine-stage framework process for design studies.

While employing the design activity framework, we reflected on our design process and modeled it in the form of the Wood et al. discourse channels, which are *"complex relationship between producers and consumers of a visualization"* [106]. In this work, we utilized four distinct discourse channels: a software company, a research organization, university information security, and an operational organization. These different discourse channels interacted together in unique ways that led to successful visualization artifacts as a result.

We present an overview of our design process in Fig. 5.1. Each row and color corresponds to a different discourse channel. Each channel has different users, data sources, and design methods that were employed. The primary visualization artifacts of this process are the prototypes and tools, with other figures in this chapter showing each. We created two dashboard prototypes during this process, as shown in Figs. 5.2, 5.3. The final BubbleNet dashboard is presented later in this chapter, and this dashboard is linked to deployments in two discourse channels.

The design activity framework timelines enabled us to reflect on how these projects interacted together, through artifacts and transference of design rationale. At the bottom of Fig. 5.1, we highlight the different activities of the design activity framework, from the perspective of the primary

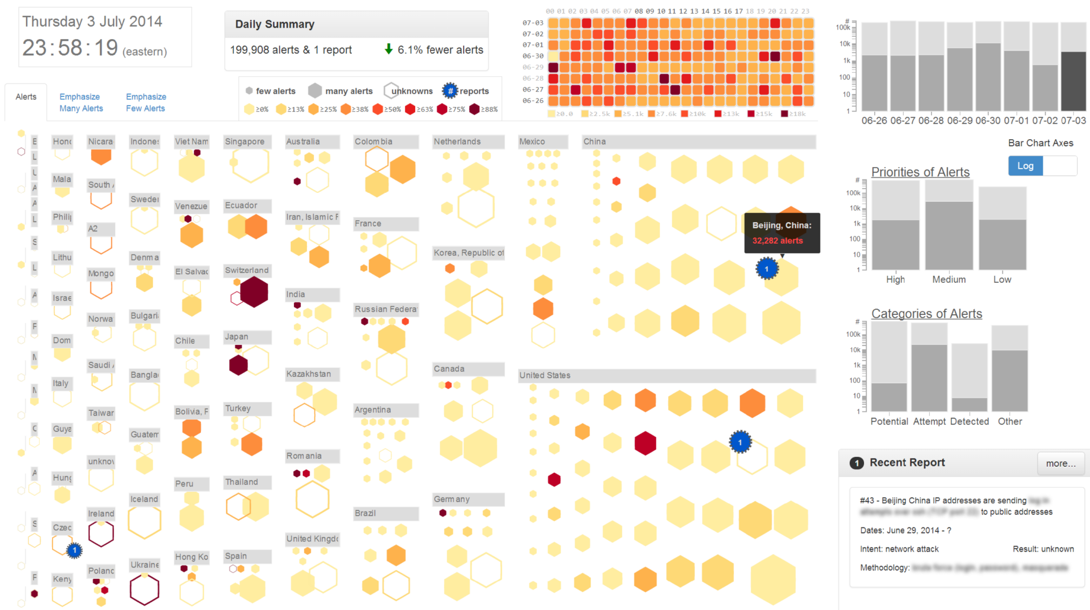


Fig. 5.2. First dashboard prototype. We created this initial prototype after using the personas design method, targeting our first design for analysts and managers. The primary encoding of this prototype is a treemap of countries around the world, sized and colored by the number of alerts or network records. The individual hexagons show a hierarchy of the treemap with individual cities of a country. The dashboard also incorporated report datasets, overlaying critical information onto the treemap.

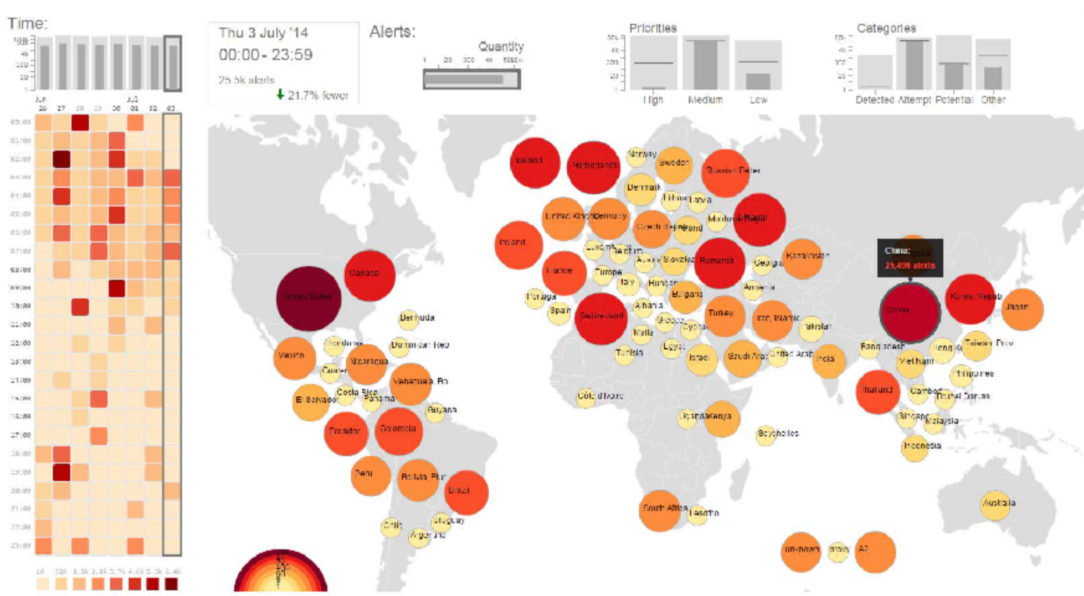


Fig. 5.3. Second dashboard prototype. We implemented this design after conducting the data sketches method to redesign the dashboard and simplify the encoding of geopositioned data, removing the hierarchy of a treemap and instead using a map. The heatmap seen in the previous prototype was rearranged to align with a temporal bar chart and to give it more screenspace and link the two views.

visualization designer across the various discourse channels, design methods, and resulting visualization artifacts. What is unique is how a previous design process, presented in Chapter 4, fed into this design study and the unique impact that the previous visualization artifacts had on the final BubbleNet dashboard visualization system.

A previous domain analysis informed this design study: a qualitative coding of cognitive task analysis papers [2]. In order to establish specific user needs, we performed a series of contextual semistructured interviews at a research organization. As a result, we identified four key user personas for designing the dashboard [2], which we discuss in detail later in this chapter. By evaluating project constraints using the personas, we further focused the project on two specific user personas: network analysts and managers. By using the timelines inspired from the design activity framework, we were able to actively connect these different discourse or project channels together, and the visualization artifacts identified through different design methods, such as qualitative codes and personas, focused the design of a cybersecurity dashboard on the task of communication for specific types of users.

After selecting this subset of users, we adapted user needs from a previous project [1] and prioritized these needs for each user persona. Examples of these needs or *user requirements* include scaling to real-world data on a single screen, preserving data context, emphasizing temporal representations of patterns, designing visualizations for presenting to others, and keeping the tool both intuitive and easy to use. Next, we sketched two dozen visualization encoding ideas to weigh them against each need. As a result, we scored each idea by combining these priorities and weights, resulting in several key ideas with the most potential. We created the first prototype from these ideas, shown in Fig. 5.2. This prototype contained a treemap of network records, organized by city and country.

We evaluated this prototype using Nielsen's usability heuristics and Gestalt principles. Specifically, we investigated the different views and interactions with respect to the usability heuristics, marking both the successes and areas to improve. We checked the visualization encoding for any violations of the Gestalt principles, for example grouping in the treemap satisfied the proximity principle, but the lack of proximity between two time-based visualizations (heatmap and temporal bar chart) failed to satisfy the continuation principle since days were encoded on two malaligned axes. This evaluation method highlighted low-level changes to fix, but we desired to evaluate the dashboard at a higher level first: the data abstraction and visualization encodings to see if a treemap was really most appropriate for communication of cyber data.

To perform this evaluation, we turned to the data sketches method [9]. Through existing tools and techniques, we showed data sketches [2] to a collaborating network analyst to gather feedback on different encodings. This feedback discouraged us from using a treemap since it took significant time to present and explain the encodings to an analyst. Furthermore, implementing the spatial treemap algorithm [132] uncovered trade-offs between the spatial location (topology) and aspect ratio of each element (squarified). In other words, spatially relevant treemaps were more challenging to read and to compare size. We discuss further details on the data sketches design method later in this chapter. However, the feedback received on the data sketches validated our initial data abstraction of location-based aggregation since abstractions such as network graphs were too complex for a simple summary view, whereas location-based views required little to no explanation. Thus, we iteratively developed a location-based encoding that is simpler and more intuitive for a larger variety of users, shown in Fig. 5.3.

In Section 6.3, we illustrate further details of our design process by placing visualization artifacts, such as concept sketches, into design activity framework worksheets. Next, a usability study was performed on this second prototype to evaluate its usability, resulting in the final BubbleNet dashboard, which we discuss later in this chapter. BubbleNet was deployed in a research environment, but significant changes were necessary to create the final tool for deployment into an operational environment. These aspects of evaluation and deployment will be discussed further.

As previously discussed, a number of user-centered design methods have been utilized in the cybersecurity visualization literature, such as interviews, observations, usability testing, focus groups, and workshops. A few methods were used in the context of a larger design process, but none of these methods were validated in the context of contribution to a completed, deployed visualization tool. Many other user-centered design methods have yet to be demonstrated for cybersecurity visualization design. We explored two design methods in detail for designing a visualization: *personas* and *data sketches*. These design methods played instrumental roles in the *understand* and *ideate* activities. Next, we discuss the two methods in detail, along with our motivation to place the method in the context of the larger design process. Then we highlight the visualization artifacts achieved, followed by results and implications of what we learned and a discussion of the methods' efficiency, effectiveness, and limitations. We then summarize each method by presenting recommendations for use in cybersecurity visualization design.

5.5 Personas Design Method

The personas design method was utilized as a way to identify potential users for a cybersecurity dashboard for communication of cyber information. We began this design study with a broad, and fuzzy, goal, requiring us to take a step back and identify the needs of the users; again, we started in the *understand* design activity. But who were the real users for a dashboard? With the task of communication, we surmised that more than one type of user was meant to utilize the dashboard. We could not find much research discussing users beyond network analysts, so our motivation was to uncover information on a range of users for cybersecurity to help form the design opportunities for this project. This motivation is an ideal fit for the personas design method.

The personas method is often utilized within the user experience, design, and HCI communities [25], [133]–[136]. Personas are *"documents meant to foster communication within a design team as archetypes of users, their behaviors, and their knowledge"* [25]. Within the cybersecurity domain, Stoll et al. describe a specific methodology for using personas, highlighting their benefits for cybersecurity visualization design [66]. Here, we further this work in three ways. First, we describe how personas benefit the communication within a design team. Second, we add visual elements to our personas to promote fast visual comparison of multiple user profiles and highlight interactions between personas. Third, we tailor our personas to the field of cybersecurity by incorporating key aspects of cyber situational awareness.

We developed the personas based on a dozen semistructured interviews conducted over six weeks with various stakeholders: network analysts, managers, researchers embedded in cyber operations, and various other cybersecurity and business-focused users. Reflecting on the data gathered during these interviews and existing literature, we produced personas for four types of users: analyst, manager, director, and CEO. Once we identified the four types of users for our project, we narrowed the project's focus to specifically design our dashboard for only two of the personas: analysts and managers. By isolating these two types of users, we were able to keep our focus consistent throughout the rest of the design process; from development to evaluation, these two user archetypes became the key motivation to justify and balance all our decisions as a design team.

5.5.1 Visualization Artifacts

We present the resulting personas from our design study in Fig. 5.4. The four personas are a cyber analyst, a network operations center (NOC) manager, a director of information technology (IT), and a chief executive officer (CEO). For each persona, we pinpointed the goal or domain-specific task for each archetypal user and visually illustrated the user's cyber knowledge and situational awareness (SA) focus. We also considered the range or window of temporal data that each user requested, illustrating how to represent visualization-specific needs within a persona. Next, we highlighted each user's key cyber SA questions, pulling from an existing question taxonomy as a basis [137]. Lastly, we identified the general flow of both decisions (downward) and information (upwards) of these personas to characterize interactions taking place between them.

5.5.2 Results and Implications

Personas played a critical role in helping us decide which users and needs to target in our design process. Narrowing the focus of our dashboard project early was crucial due to the time constraints of our project. We decided that the dashboard should not be too high level for only CEOs or just another tool for analysts. We targeted our dashboard to both cyber analysts and managers by combining features for analysts to quickly explore the data with standard visualizations for managers to easily comprehend the details of the data; see Fig. 5.2 for the first prototype of our design using these two personas. Furthermore, the narrowed design focus uncovered several key user needs for our project. By brainstorming off these needs, we were able to ideate upon various dashboard designs and compare how they worked for different users based on the personas we created. Some examples of these needs for dashboard designs include intuitive and easy-to-use, clear communica-

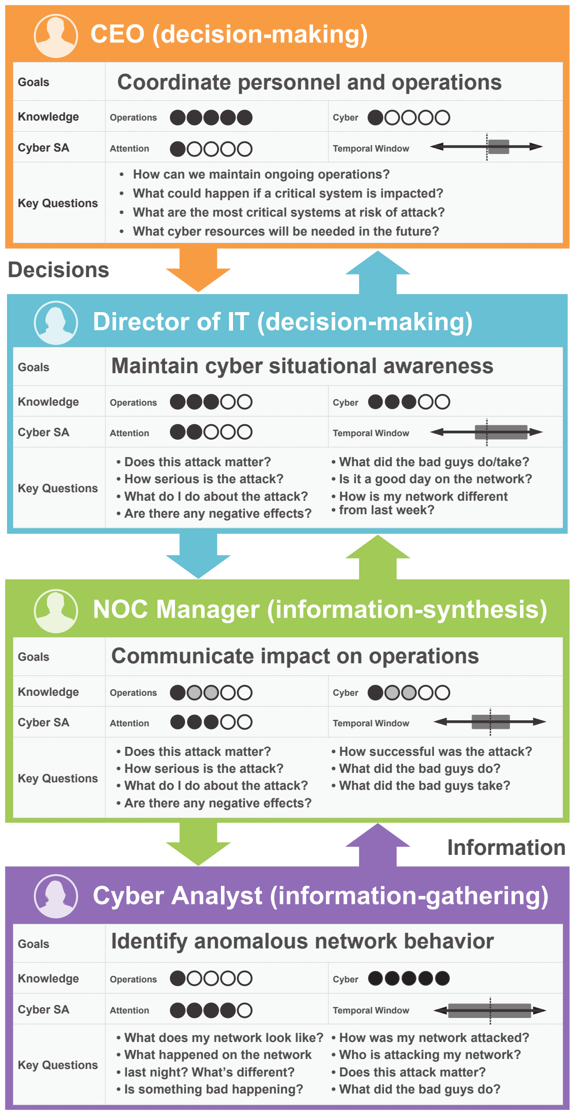


Fig. 5.4. Cybersecurity personas for visual communication. We identified four visual personas for cybersecurity visualization, showing the role decisions and information play across all users. The personas method was particularly effective at narrowing our design focus and facilitating consistent communication as a design team.

tion and presentation, provision of details-on-demand, simplified aggregation of data, adaptability, and promotion of collaboration between users. The personas continued to aid our design team in both communicating and evaluating the dashboard, up until its deployment.

5.5.3 Discussion of the Method

The personas presented in Fig. 5.4 can be used as a starting point or tailored by others in future visualization design projects for cybersecurity. Furthermore, these personas can be modified for different project motivations and user needs; it is common for personas to alter and become more refined over time [135]. The personas design method took less than three months, including the interview process, and resulted in the design of a deployed dashboard. Thus, the personas method can be both efficient and effective for cybersecurity visualization design. Additionally, the personas method can be data-driven, where personas are built and evaluated against data directly captured from users [136].

5.5.4 Recommendations

- Use personas to target the right users for a design or to evaluate a design with your users in mind.
- Talk with real users to build personas; if you cannot, use existing research or qualitative coding of the literature.
- Pinpoint user goals, knowledge, behaviors, and activities, focusing on both similarities and differences across users.
- Incorporate visual encodings when appropriate to enable easier and faster comparison across personas.
- Use and adapt personas over time; keep them as a living document to fuel multiple design projects.

5.6 Data Sketches Design Method

As originally pioneered, data sketches allow a designer to *"quickly and flexibly produce transient and uncertain visual representations of domain data by scavenging existing applications for functionality that allow data, interactions, and functionality to be combined"* [9]. In other words, a data sketch is a visualization developed using available software tools. We incorporated data sketches into our design of the cybersecurity dashboard during our *understand* and *ideate* design activities in order to establish a more complete data and task abstraction for the communication of cyber information. Our motivation was to better understand an analyst's needs, and to ideate further on the potential design options; we also sought recommendations for cybersecurity dashboard design. We asked a network security analyst at the University of Utah to provide real-world data for the data sketches, and followed-up with this analyst to get feedback on the sketches.

We obtained a network flow dataset from our collaborator containing over 2.3 million network flows, which captured over 0.4 TB throughput on the university's network. This dataset captured a five-minute snapshot of the network traffic. In developing data sketches of this flow dataset, our focus was not on the scale or optimization of the data, but how to best represent the data. The question we wished to answer was this: With this raw dataset and a network security analyst user, what views are appropriate, or inappropriate, to use in a dashboard?

We spent a month sketching with this dataset. We utilized Python to simplify, aggregate, and parse the data in various ways, and used Tableau, Gephi, and D3.js to produce a variety of visualizations. Even with these powerful visualization tools, it was still challenging to explore this relatively small cybersecurity dataset. To supplement our own sketches, we also included images from existing literature of less common and more complex visual representations that made use of real-world cybersecurity data [110], [116], [119].

5.6.1 Visualization Artifacts

We present an overview of the 20 data sketches we produced in Fig. 5.5; please see the project website* for a full-page version of each data sketch. We categorized each data sketch into four high-level groupings — network graphs, maps, aggregated charts, and time — which helped guide our discussion with our network analyst. We performed a free-form, informal evaluation session with our analyst for three hours to see which visual representations were easily understood and potentially most useful. These data sketches can be repurposed in future projects for further brainstorming.

5.6.2 Results and Implications

We showed each data sketch to our analyst; here we summarize the analyst's feedback for each kind of data sketch.

- *Network Graphs:* The analyst was unconvinced that the graphs could show meaningful insights at scale with each node representing a single IP address. Furthermore, the layout algorithm confused the analyst since it positioned each IP address at a location that was not meaningful to the analyst.
- *Maps:* In contrast to the network graph sketches, the map representations garnered positive feedback from the analyst, in particular the cartograms due to their novelty.
- *Aggregated Charts:* These charts concerned the analyst because the finest level of detail was not available. We also included one data sketch to show a 3D data chart, which seemed to entice

*http://mckennapsean.com/projects/vizsec-design-methods/

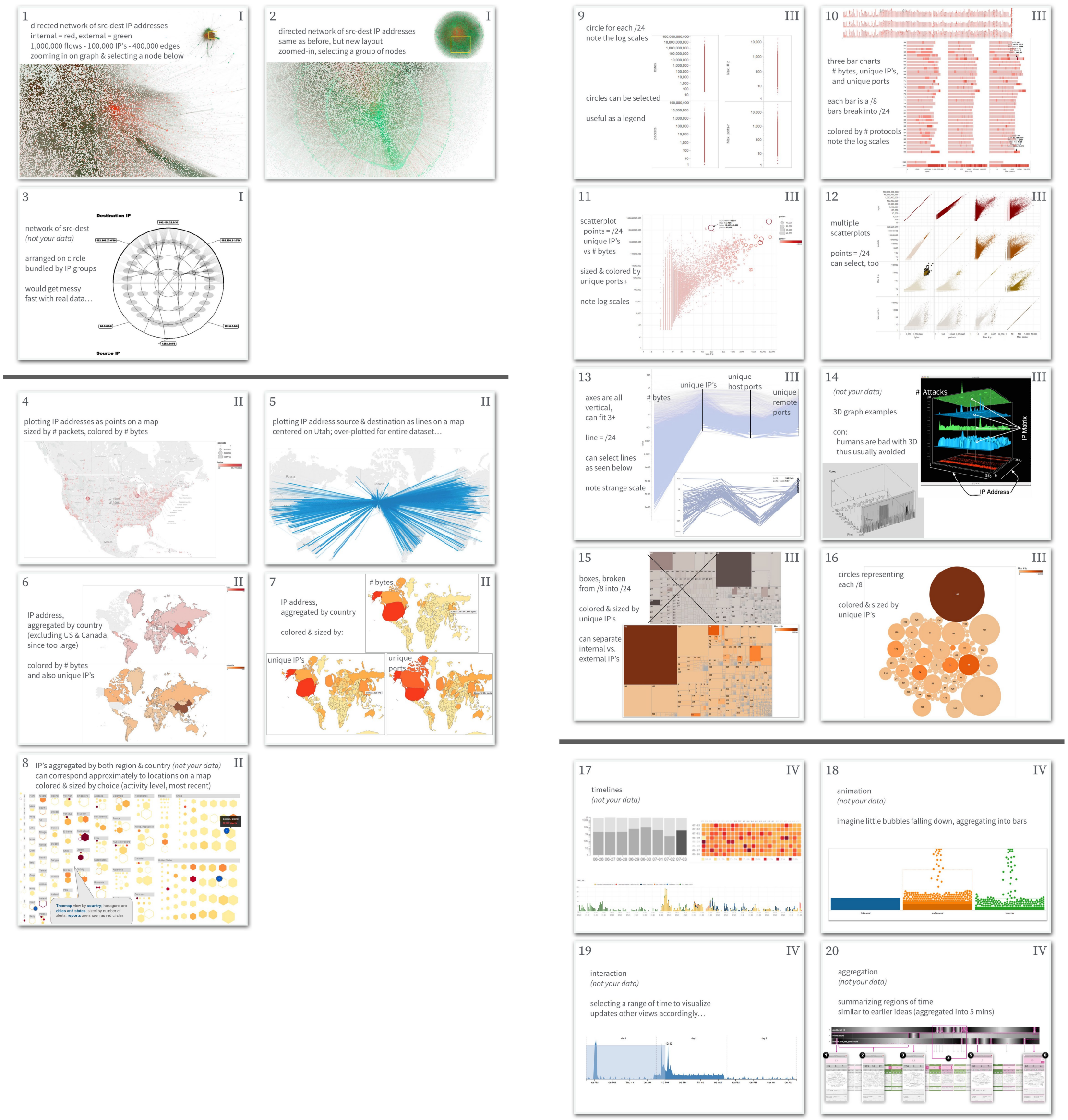


Fig. 5.5. Overview of the 20 data sketches. We evaluated these data sketches with a cybersecurity analyst; this feedback was critical to our redesign of a cybersecurity dashboard in Fig. 5.3. We categorized each sketch into four groups: network graphs, maps, aggregated charts, and time. We pulled several data sketches from existing literature [110], [116], [119].

the analyst despite our continued warnings about the usability challenges of 3D for cybersecurity visualization [69]. More unique kinds of visualization, such as parallel coordinates and treemaps, confused the analyst on first glance and required further explanation. After explanation, the analyst commented that parallel coordinates seemed promising for exploring multidimensional data, whereas the treemaps, which showed the IP address hierarchy, seemed less useful.

- *Time:* These sketches were discussed in less detail; however, the analyst stated that the timestamp was one of the least important data fields to him.

After reviewing the analyst's feedback, we synthesized several considerations for cybersecurity dashboard design:

- Complex 3D graphics and interactions can be perceptually misleading and distract from the visual representation.
- Certain visual encodings, such as parallel coordinates and treemaps, may require significant explanation and should generally not be used in a dashboard.
- Precise details on the time scale may not be immediately vital.
- Summary views for communication can use aggregation.
- Aggregation of data should be immediately obvious.
- A map-based view could aid the discovery of patterns.

With these considerations in mind, we revisited our initial dashboard design and performed another iteration on the *ideate* and *make* design activities to produce the final dashboard design shown in Fig. 5.3. The major change we made in the final design is the type of encoding, using a map view with aggregation over time. This change was, in part, driven by the results of the data sketches method, which showed the potential of aggregation and map-based views for discovering and communicating cyber data.

5.6.3 Discussion of the Method

We found that data sketches were very time efficient; the entire process took about two months to set up, perform, evaluate, and analyze. Furthermore, these data sketches were effective in our design process for producing a set of recommendations for dashboard design, and for pinpointing certain representations of the data as promising. Furthermore, this method provided some key insights for our redesign of the dashboard, which is currently deployed to users. These data sketches and the feedback we received can be used by others to inspire and evaluate their own visualization design projects for cybersecurity.

Our approach had several limitations. First, we took several of the sketches from images in the literature, and thus they were not based on our collaborator's data. Unfortunately, many of the tools in visualization papers, particularly for cybersecurity, tend not to be publicly available or provide a consistent data format for others to easily and readily use the tools for such an exercise. This meant we either had to not include these more unique and interesting visualizations in our set, or compromise by showing alternative data; we opted for the latter and included a brief description of the data being used for each encoding. The second limitation was that we received feedback on the data sketches from only one analyst. Although additional analyst feedback would be preferable, the feedback we did receive was helpful for allowing us to cull potential design ideas and focus on a smaller subset of ideas quickly.

5.6.4 Recommendations

- Incorporate real data whenever possible; if you cannot, use realistic datasets like the VAST challenge datasets.
- Repurpose the tools you know, and experiment with new ones (e.g., Python, Tableau, Gephi, D3.js, Processing, Excel, Spotfire, Arcsight, Splunk).
- Utilize real-data examples of visualization tools if a tool is unavailable or requires excessive time to input your data.
- Explore both interaction and animation in your data sketches.
- During evaluations, provide users with tasks or prompts if your goals require focusing the user feedback.
- Users may provide initial positive feedback on sketches because they are novel; consider re-evaluating at a later time.
- Introducing many data sketches at once can overload users; consider introducing sketches in multiple sessions.

5.7 BubbleNet Dashboard

As a result of the various design methods and visualization design process, we created BubbleNet, a novel dashboard to visualize and communicate patterns in cybersecurity datasets. In this section, we present the encodings and design justifications behind each view of the BubbleNet dashboard, shown in Fig. 5.6. In BubbleNet, each view supports interactive selection of elements. This selection pivots the data in all other views on the fly to the given selection, which supports identifying interesting patterns and comparing them as well.

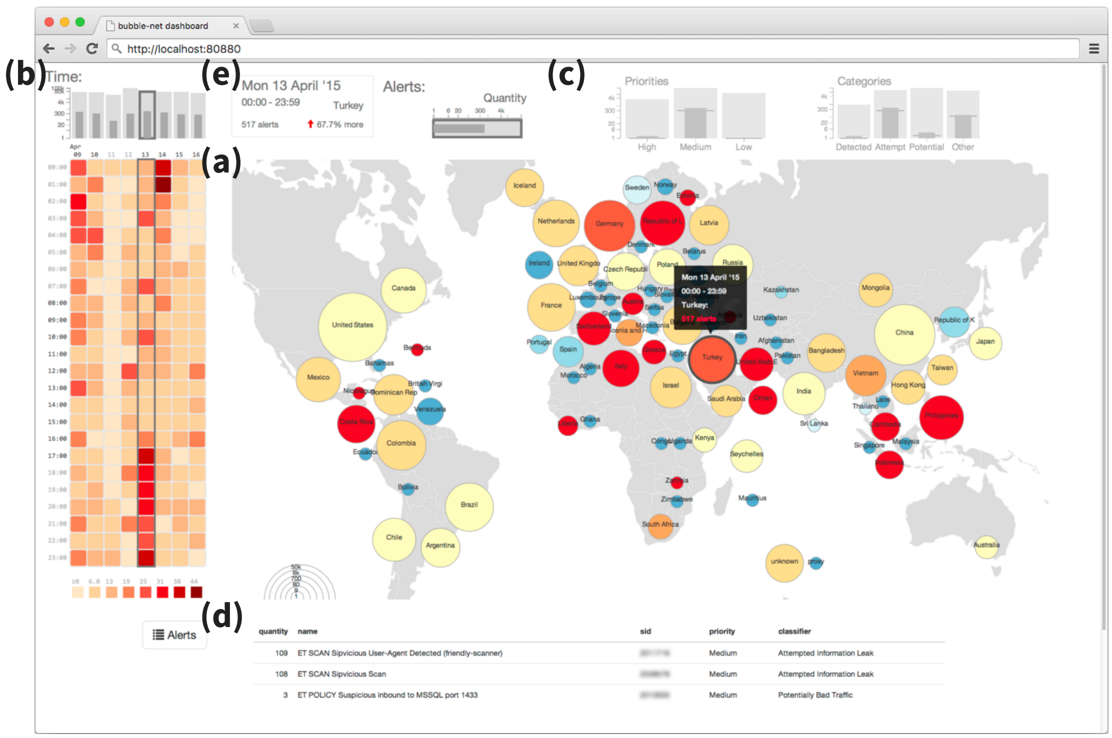


Fig. 5.6. BubbleNet dashboard. This dashboard consists of multiple, interactive views, labeled by their corresponding encodings: (a) location map based on a Dorling cartogram, (b) temporal chart and heatmap, (c) attribute bullet bar charts, (d) record details table, and (e) selection overview.

5.7.1 Location View

BubbleNet's primary view is a location-based map view shown in Fig. 5.6(a). This encoding is a Dorling-like cartogram [138] that animates circles to preserve spatial location. The implementation here is a simplification of the Dorling cartogram algorithm [139]. Each circle represents an aggregation of network records by country, and the Dorling-like cartogram is similar to a force-directed layout, initialized by the country centroids. Each circle is encoded in size by the quantity of records, and deviations from an average are encoded using color where red is more records than average and blue is less. Size is encoded on a log scale due to both the importance of visualizing a single record as well as the large range of record values, up to hundreds of thousands.

After gathering feedback on the initial treemap prototype, we learned that the details of the location (e.g., city) were less important and less amenable to visualization in a single view. As discussed previously, there are also caveats to utilizing a treemap algorithm since there are trade-offs between location and the squarified nature of the treemap. Furthermore, treemaps were not desired by us as designers due to the aesthetic requirements of whitespace, since they are space-filling, unlike a map that has more whitespace. This is why the first prototype used hexagons instead of rectangles in the treemap in order to provide more whitespace between elements, but we switched to circles since they are simpler and pack effectively on a map that utilizes whitespace more aesthetically, in our opinion.

Originally, the dashboard dual-encoded color and size to the number of records as in Fig. 5.3, but the usability study introduced in the next section obtained requests from users to show change visually on the map. We could not geolocate some records via MaxMind [130], so they were placed on an empty portion of the map to save space. Interactions with various other views in the dashboard result in an animation of the force-directed layout algorithm, and these animated transitions did not appear to distract or annoy users but did captivate them. This animation enabled a more consistent map view for users, unlike the treemaps, which resulted in more significant changes of size and location due to trade-offs of the underlying algorithms.

5.7.2 Temporal View

Two views in Fig. 5.6(b) encode time: a bar chart of network records per day with a common horizontal axis of days that aligns with a temporal heatmap beneath it where its vertical axis is by hour. The bar chart provides a quick overview of each day, and the heatmap provides details by the hour to support quick pattern discovery. It would be possible to derive similar encodings for different aggregates of time. The heatmap limits the number of days to a week in order to avoid data overload and reduce color perception issues by keeping the heatmap squares larger. The bar chart and heatmap views are arranged along a common axis due to early user feedback and the heuristics evaluation, which resulted in moving, enlarging, and linking these two encodings to create an effective temporal pattern filter.

5.7.3 Attributes View

The BubbleNet dashboard also includes bar charts and bullet charts for different attributes of the data, e.g., the priority and category for each network record, shown in Fig. 5.6(c). Bullet charts are inspired by Stephen Few's bullet graphs for dashboards [140]. Bullet graphs encode a value, a qualitative ranking, an average, and a projection into a single element, but a *bullet chart* is simplified: an inner bullet represents a subset of the full bar. In other words, the entire world's value is represented as a lighter bar, and the value of a selected country is the smaller, darker bullet inside it, as in Fig. 5.6(c). Furthermore, the bullet chart similarly encodes the average for an individual country using a thin, dark line.

Bullet charts enable showing a subset of a larger value, i.e., a country's value with respect to the world's amount. Unlike bullet graphs, bullet charts show a quantitative subset, and this subset enables quick comparison through interaction. As with previous scales, we incorporated a log scale for these bar charts. We considered alternative encodings of the data across all views, such as orders of magnitude markers [141], but these encodings required significant explanation and collided with encoding subsets. A log scale helps to visually show both extremely large and extremely small values at the cost of comparing values precisely, but interaction supports comparing precise values using text.

5.7.4 Records View

A details-on-demand table view in Fig. 5.6(d) provides a summary of the different records in any selection. This summary includes the quantity, user-friendly name, ID or type of record, and the detailed attribute information. These details enable analysts to understand what is happening in any selected aggregate of network records in the dashboard. As such, we created this table and dataset by request of all analysts during the usability study. Inclusion of network record details is critical to this discovery of patterns. In our evaluation, analysts told us that they were able to not only discover patterns using BubbleNet, but also envision using this dashboard to present what they found.

5.7.5 Selection and Interaction

Interaction is a crucial component of most elements on the BubbleNet dashboard. Most interactions involve a selection that specifies some pattern, which updates the selection window in Fig. 5.6(e) with details such as the date, time, country, number of records, and the deviation from average. Furthermore, a visual summary of the pattern's total records is shown in a horizontal bullet chart. For example, selecting four countries results in very different patterns in the heatmap, as shown in Fig. 5.7. We provide a video showcasing all possible interactions on the project website.[†]

[†] http://mckennapsean.com/projects/bubble-net/

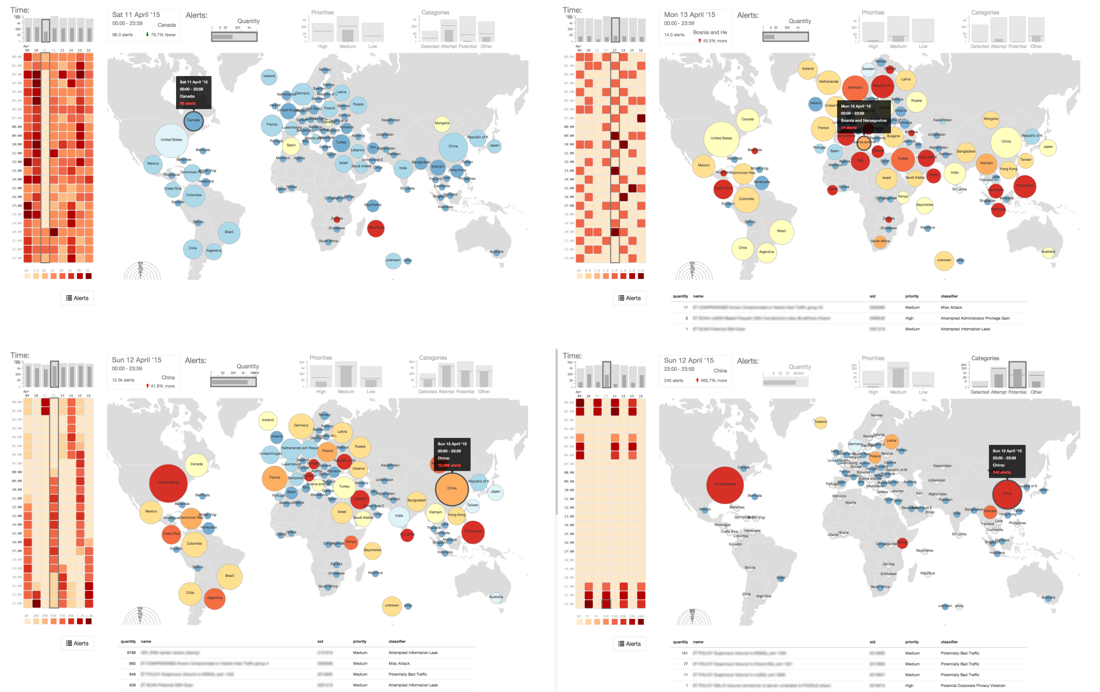


Fig. 5.7. Visible patterns in the dashboard. Most elements of the BubbleNet dashboard are interactive and update all other views accordingly. For example, selecting four countries shows significantly different patterns in the hourly heatmap for each of these views.

All interactions with the dashboard require one click or fewer, meaning a user can hover over any element for an updated view of the patterns in BubbleNet. This hover over affects all other views, and BubbleNet also provides a pop-up of this selection as in Fig. 5.6(a). By clicking on any element, that selection becomes locked in place and updates the selection window in Fig. 5.6(e). Otherwise, when a user hovers off an item, its previous selection is reset.

By default, the initial pattern is the most recent day and the entire world. As such, the bullet charts in Fig. 5.6(c) look like regular bar charts until a country is selected to show this country as a subset of the world's pattern. Through feedback from users, we found that reducing clicks for selection was desired in a dashboard setting and enabled fast comparison of two selections, by selecting one element and hovering on and off another element. We also added keyboard interactions to more easily navigate selections through time and to reset back to the default pattern.

One can compare the interaction of each view with our tasks, discussed previously. For location, temporal, and attribute views, all the elements were interactive, e.g., hovering or clicking on a country, day, hour, or type of attribute. These selections supported pivoting data to identify and compare patterns. The records table view supports identification and comparison of patterns but not pivoting since analysts often use their own tools for this purpose.

5.7.6 Implementation

We created the BubbleNet dashboard presented in Fig. 5.6 using D3.js for all visualization components. Each interaction filters a different portion of the same dataset loaded in the web browser. These datasets are prepared via a set of back-end Python scripts that aggregate network security datasets into summaries by day, broken apart by location and by hour with statistics precomputed on the data. Lastly, these daily summaries are combined in Python to produce JSON files for the web dashboard, so incorporating datasets that update in real-time is possible but currently requires a refresh of the page.

The visualizations shown here, in the video, and in the usability study all showcase real data from a large organizational network, capturing a summary of a month's worth of data or about a million records. In particular, the dataset shown is from an intrusion detection system, which automatically flags important network records as alerts for network analysts. These alerts can be generated by predefined rules, which is most often the case, or by more sophisticated machine learning techniques. The BubbleNet dashboard is designed in such a way to support visualization of any dataset that can be broken into network records and geolocated, so it works best when analyzing traffic over the Internet. When it comes to scalability, the dashboard maintains interactivity with millions of records due to aggregation done on the backend.

5.8 Evaluation and Deployment

Evaluation is undoubtedly an important aspect to designing tools for users, for cybersecurity [103] but more broadly as well. First, we discuss the evaluation methodology of a usability study. This study is a combination of formative and summative evaluation since we prioritized key issues on a high-fidelity prototype but user needs were also uncovered. The results of this study highlight the usability of BubbleNet, and the BubbleNet dashboard in Fig. 5.6 was thus deployed in a research environment. However, this study also highlighted missing elements of utility from the BubbleNet dashboard, so a final design iteration in the *make* activity was required to address these elements before we could *deploy* the tool in an operational environment.

5.8.1 Evaluation Methodology

To improve upon the second prototype from Fig. 5.6, we performed a usability study with network analysts and managers from both research and operational organizations using real-time, real-world data from an organizational network. The intent of this study was to improve the design and see if the prototype met the needs of both analysts and managers. Nine cybersecurity professionals participated in the study: five analysts, four managers. Each participant took part in a one-hour long think-aloud session, conducted by one moderator with an observer taking notes, both of whom participated in this research. Each session contained a scripted walk-through of the prototype, several prescribed tasks to complete, open-ended questions about how users would use the prototype, and distribution of a system usability scale [123].

To examine data from the think-aloud session, I analyzed the notes taken by the observer using a qualitative coding methodology [93]. I conducted this coding, through an open tagging of two users' comments and consolidating tags to all other user comments. Furthermore, the system usability scale is a standardized survey technique [123] used to evaluate the prototype's usability, and other researchers have utilized such a survey [124]. This usability survey has been used to evaluate the usability of systems for 30 years with its set of 10 standardized statements rated on a Likert scale; it works well with a small group of users [142]. By combining this survey with a qualitative coding methodology, we sought to increase the analytical rigor of evaluating our prototype to determine if it was ready to be deployed to users.

5.8.2 Evaluation Results

After coding the participants' comments, we formed the following categories of tags: desired task, that task's intended target in the dashboard, and its artifact. Example tasks include to present, filter, or identify with any of the views presented in the BubbleNet dashboard, and example artifacts include successes, struggles, and failures along with other tags such as feature suggestions. These tags provided a unique view on the qualitative data, and we prioritized and implemented a list of features for BubbleNet in Fig. 5.6. The added features include details-on-demand records view, better selection feedback, new map color encoding, and keyboard interactions. This analysis process gathered the key successes of the BubbleNet dashboard:

- Temporal pattern detection was simple and easy using the heatmap: *"I keep getting drawn to the heatmap and these darker areas, because they certainly stand out"* (A4) and *"[The heatmap] helps find those temporal patterns"* (A1)
- Users expressed that the dashboard's utility was for discovering patterns and trends in the data: *"The majority of what we are looking for is patterns and this just makes patterns which is faster"* (A4)
- One-click-or-fewer interactions worked very well: *"It's very responsive and dynamic; the fact that it changes as I narrow [in] is the best"* (manager #2, or M2)
- Most interactions occurred with the bullet charts and heatmap: *"I could write a splunk query to do this, but this is easier"* (A5)
- No expressed dislike for animation in the map view: *"[The] best part is the instant visual gratification"* (A4)

Furthermore, this analysis derived a set of design considerations for future cybersecurity dashboards, discussed in the next section. With the first few participants of the study, we discovered

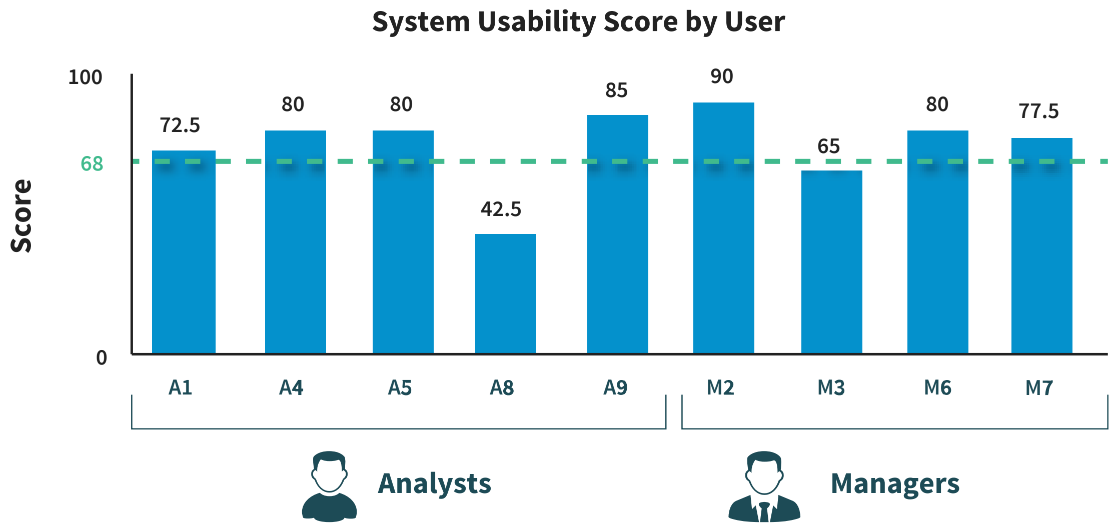


Fig. 5.8. Dashboard usability scores as rated by analysts and managers. These are the final scores of a system usability survey of nine users, both network analysts and managers. The average score of the dashboard is 75, above the average usability score of 68 [142].

a common usability issue since the bullet charts had two different bars to click on. Along with visual bugs, we fixed these issues right away to focus feedback on less obvious issues. Quantitatively, these design alterations could raise concern, but, since the changes were motivated by and reduced user frustrations, we hypothesize that the quantitative results from the usability survey would have improved if we had re-run the study with these fixed usability issues.

The prototype gave users novel insights into their dataset. For example, one participant found a pattern in a particular country and told us that he *"never would have got[ten] there by looking at the alerts in text format"* (A1). This same analyst told us that he could imagine this dashboard being used with other kinds of datasets as well: *"pretty much everything: flow data, [firewall logs], [proxy logs], anything"* (A1). This statement helped confirm that the abstraction was at just the right level since the dashboard could adapt to so many cybersecurity datasets.

We present the quantitative results of the usability survey in Fig. 5.8. The system usability scale provides a standard set of questions where an average system would receive a score of 68 out of 100 [142], and the usability of our prototype was found to be above average: 74.7. Each individual question can be broken into a set of characteristics [142], and by doing so we found that the BubbleNet dashboard scored high on learnability and ease of use. By analyzing the results of analysts versus managers, we found no significant differences. However, network managers rated BubbleNet as less complex, less cumbersome, and easier to learn. We did have one outlier (A8), who was two standard deviations lower than the average, which lowered the final score due to the relatively small sample size. We hypothesize that this user simply rates things more strictly since this user still achieved tasks successfully and had similar concerns as other analysts.

5.8.3 Deployment

After the usability study, further development led to the final BubbleNet dashboard. Then, we deployed BubbleNet to users with real-time data in a research network operations center. However, we developed and deployed BubbleNet with only a single data source and a short time range, so it was arguable how useful its design could be for other users. This problem is coupled with the fact that the usability survey scored lower on a question that arguably could be interpreted with respect to its utility: *"I think that I would like to use this system frequently."*

To gauge its operational utility, we further demonstrated the BubbleNet dashboard with multiple relevant datasets to different analysts at three cyber operations centers. Analysts and managers provided qualitative feedback via comments, both during the demonstration as a group, and in private conversations afterward. These demonstrations, feedback, and design iterations took place in the fourth design channel of Fig. 5.1. In summary, this feedback highlighted the simplicity of the flat map, the conjunction of small multiples with interaction, and a critical area for improvement with respect to scaling to multiple data sources.

This feedback from operational analysts led to the final design iteration and deployed operational tool. To incorporate multiple data sources, significant trade-offs existed between displaying all data and the tight integration required for linked small multiples as presented in BubbleNet. As such, this final tool utilizes the assembly-canvas metaphor [143], similar to Tableau's dashboards where a custom visualization dashboard is built on the fly. The flat map serves as the background for any geospatial data. A left-most palette lists the available data sources. When selecting data sources that are not geospatial, a floating visualization palette is placed on the screen for the user to select a different visualization for the data. These palettes support customization of numerous visualizations: e.g., treemap, node-link diagrams, sunburst charts, and timelines, and this customized dashboard can be saved and shared.

After implementing this final tool, end users have expressed an interest in adopting it for daily use. Next steps for the project include a formal, summative end user evaluation. While developing this final design, we identified several design considerations for future development, such as establishing consistent visual encodings across varied datasets and connecting these visualizations through interaction. Although not in the scope of this project, these considerations remain important for continuing operational deployment.

5.9 Summary

We uncovered a set of implications for dashboard visualization of cybersecurity data that others can use. First, analysts sought details of the data whereas managers wanted the broader impact of an incident on the larger network. Second, there are many different ways to aggregate and provide details of the underlying data, so it is imperative to use and adapt multiple cyber visualizations to different needs over time. Third, we discovered that a map for cyber data is not completely useless. Users are able to situate themselves and pivot data to find novel insights, and a map is one way to scaffold a visualization into other visual representations and encodings [144]. Fourth, fast hover-over interactions are very appropriate to reduce the number of required clicks to pivot visualizations using animation and provide quick details on demand.

In summary, we found that the design activity framework more comprehensively captured the design process of this project and the multiple discourse channel interactions, in which the previous nine-stage framework failed to connect visualization artifacts and design decisions across projects. However, the BubbleNet dashboard is not the end of research or development into cybersecurity dashboards. The use of a map does not work for all data, and there is more work needed to find more effective encodings such as broader impact of cybersecurity incidents. Nevertheless, the design process of BubbleNet shows how other design studies can work with collaborators and users beyond just data analysts. When working with these other types of users, it becomes more important to balance and prioritize appropriate sets of user needs to design, develop, and deploy effective, domain-specific visualization tools.

6
Design Activity Worksheets

For visualization pedagogy, an important but challenging notion to teach is design, including showing students how to make and also characterize their decisions when designing and evaluating a visualization encoding, user interaction, or data visualization system. Our introduction of a new framework for visualization design in Chapter 3 codifies some of the high-level steps of the process for visualization designers, but the four activities lack a breakdown or example of concrete steps to facilitate novices utilizing this framework to walk through their own real-world design process. To externally validate the framework and provide such a framing, we created new teaching materials for the design activity framework, such as a visualization design worksheet for each design activity, a lecture on the design process showcased within a real-world project, and resources for learning how to design and sketch visualizations [6].

These design activity worksheets for visualization novices present a high-level summary of each activity with more actionable, guided steps for students to walk through the process of designing their own visualization system. Furthermore, we validated the effectiveness and use of these worksheets and the overall framework in the context of a graduate-level visualization course taught at the University of Utah [6]. For this external evaluation, we surveyed the class and worked with 13 students who voluntarily utilized these design worksheets for their cumulative projects. We conducted a series of interviews to garner additional open-ended feedback and suggestions that highlight the strengths and limitations of these worksheets as teaching materials. In this chapter, we present four worksheets, one for each design activity, with five concrete steps and guidance on each sheet, and we externally evaluated the effectiveness of the design activity framework and these worksheets in a pedagogical setting.

6.1 Increasing Actionability of the Framework

To teach design in data visualization, educators combine many foundational components, from user interface principles [34] to data and encoding taxonomies [33]. Additional pedagogical materials for the field focus on visual or perceptual principles [32], [35], [145] as a basis for creating and judging data visualizations. Educators may also apply these principles and techniques in the classroom through the use of design critiques or a cumulative project. These visualization projects could be guided by several textbooks that expound upon different design processes [34], [35] or design decision models [33] to help novice visualization designers learn how to effectively and methodically build and validate visualization systems.

Many of the pedagogical approaches to the visualization design process, however, are theoretical in nature. From our own combined teaching experiences, we have witnessed students struggle to incorporate these theory-based design concepts into their practical, hands-on projects. As such, we believe there is an opportunity for new approaches to teaching the next generation of visualization designers, equipping them with not just theoretical knowledge but also the practical steps for building better systems and tools.

In Chapter 3, we described one such theoretical model of the design process with four actionable design activities: *understand*, *ideate*, *make*, and *deploy*. Each activity includes a goal, target artifact or outcomes, and a plethora of design methods to choose from, each of which is described in such a way to make the process model more actionable. We found, however, that the theoretical framing of the model restricted and limited its use and actionability in the classroom or class project settings. To address these limitations, we crafted a design worksheet for each design activity with steps to assist students walking through the visualization design process for the first time. Furthermore, we sought to validate the use of this framework and the worksheets in cumulative projects for a graduate-level visualization course.

6.2 Related Work in Pedagogy for Design

For the past few decades, pedagogy for data visualization and human-computer interaction has begun to shift from academic or theoretical foundations toward including skills for design, critique, and critical analysis [71]–[73], [146]. Educators have come to realize that they must rapidly adapt their teaching methods to the growing body of diverse students [147], [148], from undergraduates across disciplines to graduate students in standard courses, flipped classrooms [73], [147], and online environments [74]. A recent approach among educators is to employ active learning [72], [74], [75], where techniques and methods are used to encourage deeper analysis and synthesis as opposed to just passively observing a lecture [74]. For example, a common approach observed in most classroom settings is practical data visualization exercises, to give students opportunities to critically analyze a data visualization or existing visualization tool and work with their peers to analyze the outcomes

[73], [149].

When it comes to data visualization design, the core concepts of active learning can help overcome some of the challenges faced by educators when teaching concepts surrounding design thinking [73], from considering broad divergent visualization ideas [16], to evaluating based on visualization principles, and leveraging existing designs to create something new. An effective pedagogical methodology is the use of design studios incorporated into the classroom setting [73], [146], [150], inspired by its use in fields such as architecture, design, and art. For visualization, educators often incorporate these studios as workshops or practical exercises and also through real-world projects for students to learn about design outside the classroom [149]. For example, VizItCards [73] was created to help novices practice, and it is used to reinforce visualization concepts during workshops. Human-computer interaction educators have noted that computer science students in a design studio tend to focus more on idea refinement rather than broad idea generation and innovation [151]. Other active learning approaches include the use of design workshops [73], [76], [77], rich discussions [152], [153], and design games [75].

Within the data visualization pedagogy, guidance for how to design data visualizations, both generating and evaluating visualization artifacts, is missing clear steps for novices. When teaching data visualization design, educators often incorporate user interface principles [34], teach taxonomies of data and encoding [33], illustrate ideal visual principles [32], [145], explain perceptual principles [35], and generally empower students with the ability to evaluate, criticize, and judge data visualizations. These principles and concepts often get applied in courses through design critiques or encased within a cumulative project [154]. These cumulative projects are an alternative to in-class design studios, where students must acquire their own datasets, come up with ideas to visualize data for different tasks, and build an interactive, multiview visualization system to support these tasks in the data. By providing novices with realistic, hands-on experience, students can reapply these skills in their own future projects. However, novices may struggle to conduct their own design process, perhaps referring to textbooks that include their own design process methodologies [34], [35] or research papers that detail the design process or design decisions [8], [33], but often these models are high level, terminology heavy, less actionable, and theoretical in nature. For novices, it is often useful to have a clear set of guidelines or instructions to walk through this process for the first time. However, no such step-by-step guidance currently exists for the data visualization design process.

Educators have worked on concretizing steps for the ideation process. Specifically, the five design-sheet methodology [16] utilizes worksheets to structure and guide visualization students through the ideation process. This approach by Roberts et al. encourages engineering students to think divergently and creatively and sketch out ideas on paper when first designing a visualization. Their approach begins with brainstorming, followed by three unique designs and a realization sheet for the final tool. The authors evaluated this teaching methodology with master's-level students in information visualization. Over several years, 53 students completed these design sheets over the span of two hours to come up with different ways to visualize their own chosen dataset, after which they received feedback and additional time to finalize their designs [16]. After grading the sheets, the researchers compared these grades with the students' project and final course grades, as well as an anonymous survey, to gather feedback on the entire course including the design sheets. Generally, they found that the design sheets aided students in planning and organizing a design section for the project write-up, and it also encouraged students to think divergently, which is a skill not often taught in computer science. However, in a workshop at the 2016 IEEE VIS Conference that used these worksheets, we experienced a limitation by using this approach too soon: many steps must occur first, such as data collection, identifying the challenge, focusing on a target user, and finding tasks. Roberts et al. elude to this limitation with different preparation steps [16], but these steps can be nontrivial and tricky for novices. Thus, it would be beneficial to establish and evaluate more worksheets beyond just ideation for data visualization design pedagogy.

Educators also face several unique challenges, such as visualization preconceptions, visual literacy, classroom time limits, and increasing class sizes. First, students may have preconceived notions about data visualization and principles that can affect the learning process and how they evaluate a given data visualization [72], [155]. Furthermore, the concept of visualization literacy can be traced back to how we introduce, teach, and incorporate visualizations throughout schooling, from elementary school [156] and beyond. In college-level courses, a key challenge can be tight deadlines and schedules [73], [147], with limited course time for practical exercises, design studios, and cumulative projects, which often last only a month or two and may include noncolocated learners [74]. As course enrollments grow, it is important to think of how to scale visualization design feedback and evaluation; one recommended approach is to utilize peers to help scale this process and provide students with more practice on visualization evaluation and critique in order to apply principles and concepts taught throughout a course [72], [152], [157]. Additionally, concepts that are heavily theoretical or mathematical in nature may be more challenging to teach to novices. Instead, it may be ideal to simplify terminology and focus on simpler, comprehensive aspects of the theory [26] so that students can more readily grasp, apply, and build upon these theoretical constructs. In our own experience, we have observed the challenges of teaching theoretical design aspects such as the nested model [17] to visualization novices. By simplifying these aspects into a more step-driven process, students may be able to apply and learn these visualization design concepts on their own more effectively.

6.3 Worksheets for the Design Activity Framework

To create the visualization design worksheets, we first reflected on the design process and decisions illustrated by the design activity framework [1] across several of our own projects. Here, we discuss our process behind creating these teaching materials and provide examples of their use. The worksheets and teaching materials are located on a public-facing website* for their dissemination and use by others, and we encourage feedback and improvements to these teaching materials by other visualization educators over time.

Inspired by the five design-sheet methodology [16], we wanted to integrate the visualization design process in its entirety into visualization design worksheets to enhance the teaching of an otherwise theoretical design process to new students. Our first goal was to create a worksheet for each of the visualization design activities: *understand*, *ideate*, *make*, and *deploy*. To do so, we reflected on our combined research and design experiences across each visualization design activity, and then broke down these activities into a series of tangible and generalized steps (see Table 6.1). These steps are akin to design methods that can be generative or evaluative in nature. We created several introductory and template worksheets to help guide students through filling out each design activity worksheet, and we include these materials in Appendix B.

We introduce the first design worksheet for the *understand* activity in Fig. 6.1. At the top of each sheet, we describe the desired goal and resulting visualization artifact or outcome for the activity. Each sheet can be numbered in the top right for keeping track of order and for planning and retrospection. For each worksheet's five steps, we included additional helper text to help students find the answer and complete each design worksheet. Please find the details for each of the design activity worksheets in Figs. 6.1, 6.2, 6.3, and 6.4. We included warnings about when to jump back to previous boxes or worksheets, and icons to illustrate the expected type of answer for each box: a list, a sketch, or a table. Lastly, the bottom contains a list of icons pointing to the potential next activity of the visualization design process. These decisions about which activity to perform next are tricky and require reflection on the part of visualization designers to verify that their current visualization artifacts are sufficient and that they succeed in addressing the desired problem or challenge in enough detail. This is why many visualization design processes may be iterative and complex to perform.

To create the worksheets, we combined the design activity framework definitions with related worksheets used by a colleague running design studios in our university's architecture and design department, resulting in the five steps for each visualization design activity, shown in Table 6.1. Knowing that engineering students could benefit from focusing on creating many types of visualization artifacts, we utilized four of the steps for generation. For example, the *ideate* sheet used three sketches as in the five-design sheet methodology. We targeted each step's instructions for a single visualization project based on our own experience building visualization systems. After our first iteration, we presented the worksheets to our visualization research group and received a series of recommendations for improvements, including to place more of a focus on the users earlier in the process and to simplify complex, theoretical terminology. For example, the terminology of the nested model [17], [18] was included in the original design, but it was determined that the

* https://design-worksheets.github.io/

Table 6.1. Five steps for each design activity. We break down each visualization design activity into five concrete steps. The first four steps of the process are generative, to establish design requirements, encoding and interaction sketches, visualization prototypes, or visualization systems. The fifth step is always evaluative, to compare different visualization artifacts in order to justify design decisions and record that reasoning down for later use. We shared these five steps with novice visualization designers, students, using design worksheets as a template, as in Figs. 6.1, 6.2, 6.3, 6.4.

understand	***ideate***	***make***	***deploy***
identify the challenge & users	select a design requirement	set an achievable goal	pinpoint a target audience
find questions & tasks	sketch first idea	plan encodings & layouts	fix usability concerns
check with users or explore data	sketch another idea	plan support for interactions	improve points of integration
brainstorm design requirements	sketch final idea	sketching additional views	refine the aesthetics
compare & rank design requirements	compare & relate your ideas	build the prototype & check-in	consider a method to evaluate

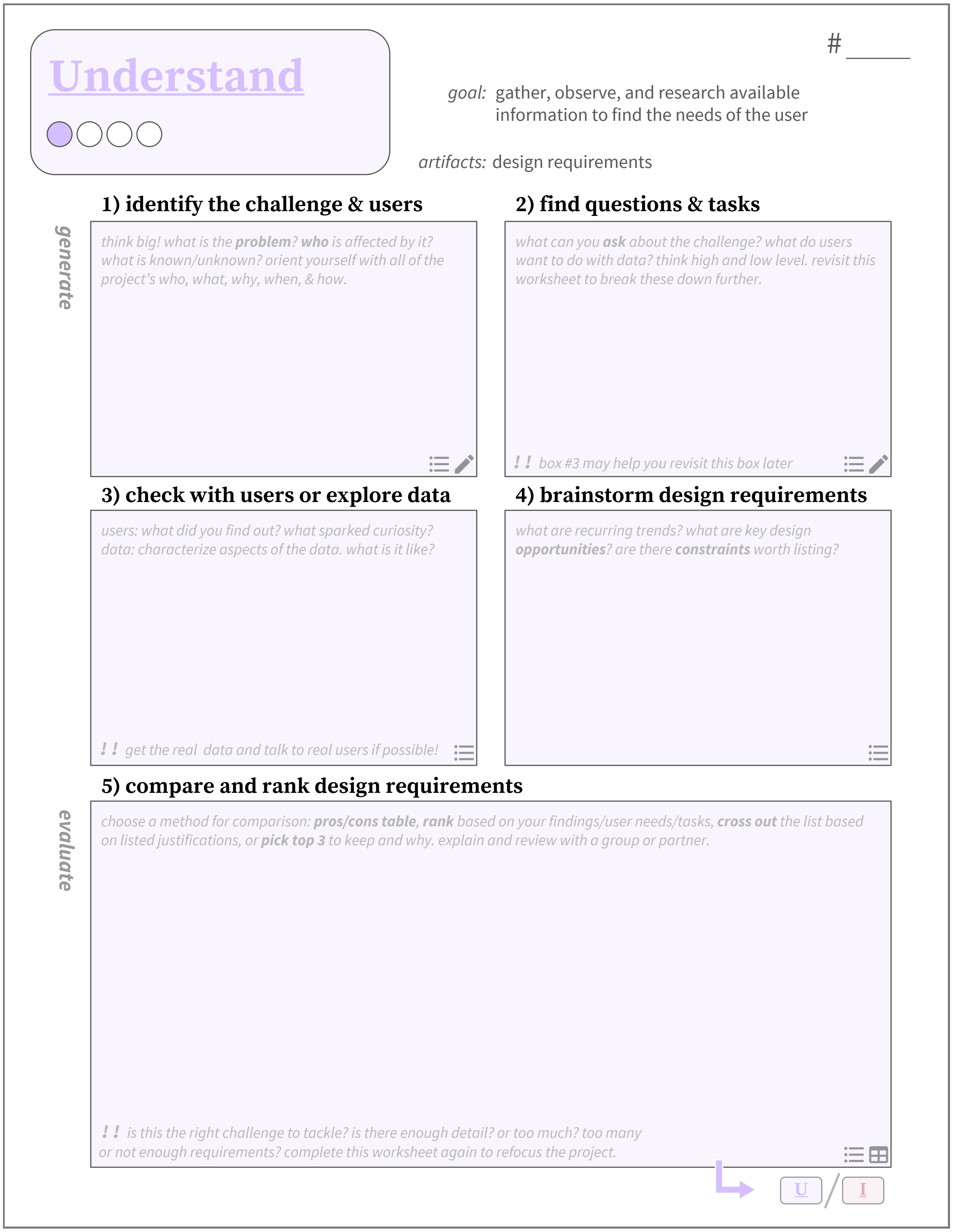
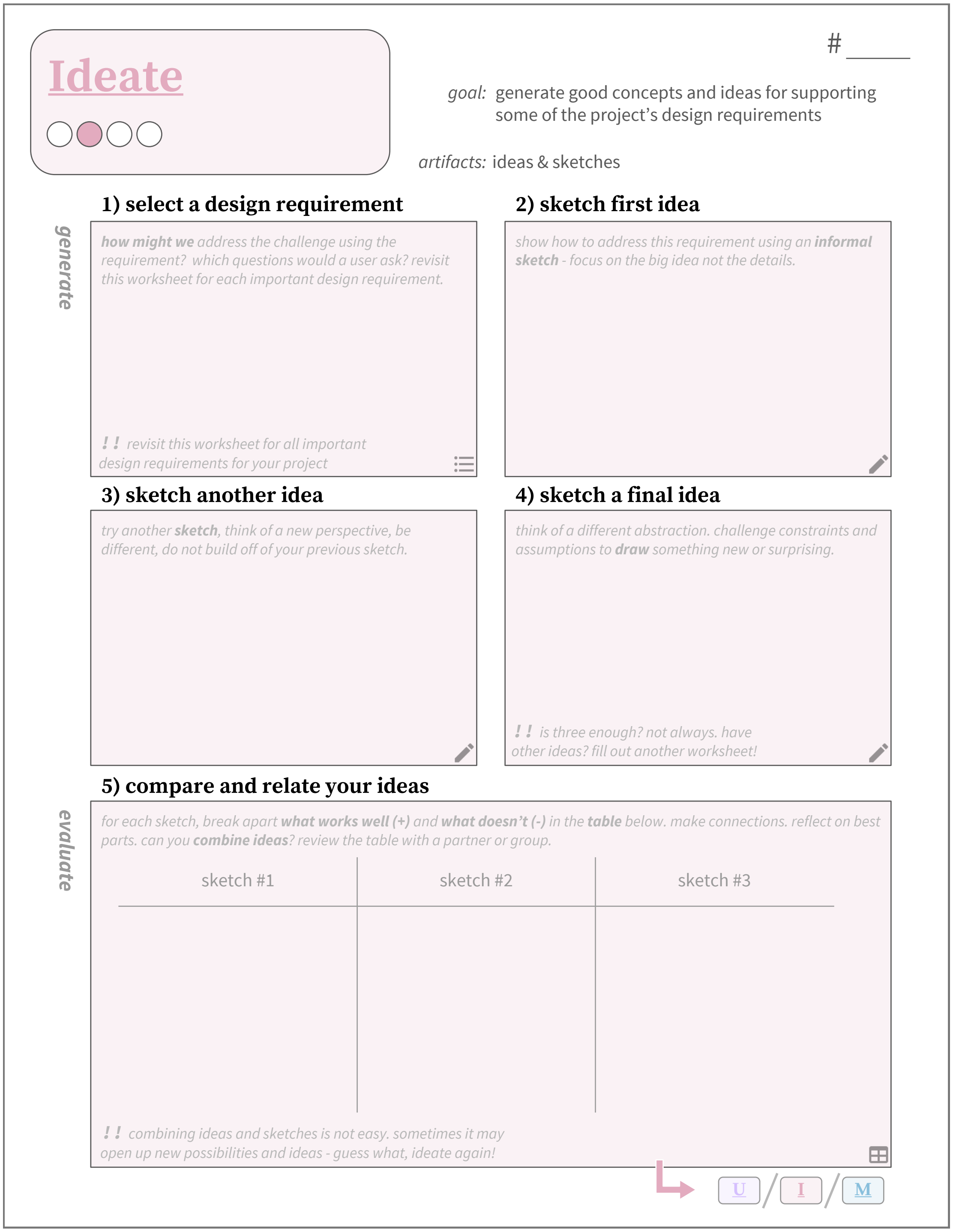


Fig. 6.1. Worksheet for the *understand* activity. We tailored this worksheet to help students identify their problem, users, data, and requirements for a data visualization system.



Ideate

______

goal: generate good concepts and ideas for supporting some of the project's design requirements

artifacts: ideas & sketches

generate

1) select a design requirement

__how might we__ address the challenge using the requirement? which questions would a user ask? revisit this worksheet for each important design requirement.

!! revisit this worksheet for all important design requirements for your project

2) sketch first idea

show how to address this requirement using an __informal sketch__ - focus on the big idea not the details.

3) sketch another idea

try another __sketch__, think of a new perspective, be different, do not build off of your previous sketch.

4) sketch a final idea

think of a different abstraction. challenge constraints and assumptions to __draw__ something new or surprising.

!! is three enough? not always. have other ideas? fill out another worksheet!

evaluate

5) compare and relate your ideas

for each sketch, break apart __what works well (+)__ and __what doesn't (-)__ in the __table__ below. make connections. reflect on best parts. can you __combine ideas__? review the table with a partner or group.

sketch #1	sketch #2	sketch #3

!! combining ideas and sketches is not easy. sometimes it may open up new possibilities and ideas - guess what, ideate again!

U / I / M

Fig. 6.2. Worksheet for the *ideate* activity. In this activity, novice visualization designers must target a specific design opportunity or requirement in order to draw and compare three sketches.

Make

goal: concretize ideas into tangible prototypes which are approximations of a product in some aspects

artifacts: prototypes

#_____

generate

1) set an achievable goal

what should the prototype **achieve**? what are the specific **criteria for success**? break a larger goal into parts with clearer feature sets.

! ! break a goal apart into multiple and create a worksheet for each sub-goal

2) plan encodings & layouts

what are good visualization **encodings** or **layouts** for which data? use the ideas you just came up with, and remember to justify for users and their tasks.

3) plan support for interactions

what can the user do? what is required given the chosen encodings? **justify** your design decisions.

4) sketching additional views

what other parts of the data must be seen? brainstorm how to show this data in the tool.

! ! if you are thinking up new ideas to visualize, go back to the Ideate activity!

evaluate

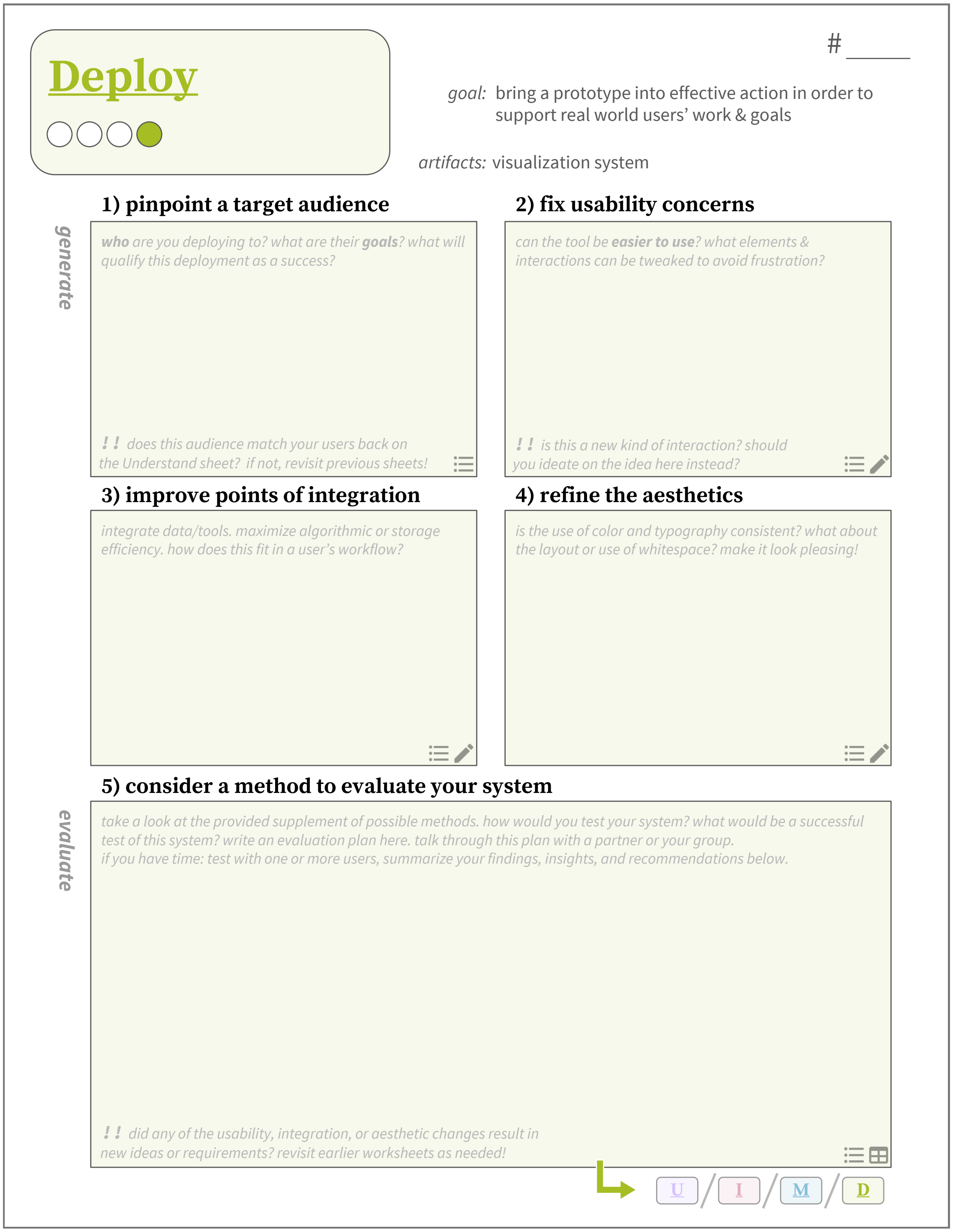
5) build the prototype and check-in

are your **goals met** by the prototype? test with users if possible. are design decisions properly justified? do any need to be revisited? were any new constraints or limitations discovered? write down your progress and additional justifications below. review this progress and the prototype with a partner or your group.

! ! did the prototype meet its goal/s? measure its success. make sure you have addressed the design requirement. does the prototype try to do too much?

U / I / M / D

Fig. 6.3. Worksheet for the *make* activity. Visualization novices use this worksheet to create a prototype system, by planning out their encodings, layouts, and interactions, along with coding it all together.



Deploy

\# ______

goal: bring a prototype into effective action in order to support real world users' work & goals

artifacts: visualization system

generate

1) pinpoint a target audience

__who__ are you deploying to? what are their __goals__? what will qualify this deployment as a success?

! ! does this audience match your users back on the Understand sheet? if not, revisit previous sheets!

2) fix usability concerns

can the tool be __easier to use__? what elements & interactions can be tweaked to avoid frustration?

! ! is this a new kind of interaction? should you ideate on the idea here instead?

3) improve points of integration

integrate data/tools. maximize algorithmic or storage efficiency. how does this fit in a user's workflow?

4) refine the aesthetics

is the use of color and typography consistent? what about the layout or use of whitespace? make it look pleasing!

evaluate

5) consider a method to evaluate your system

take a look at the provided supplement of possible methods. how would you test your system? what would be a successful test of this system? write an evaluation plan here. talk through this plan with a partner or your group.
if you have time: test with one or more users, summarize your findings, insights, and recommendations below.

! ! did any of the usability, integration, or aesthetic changes result in new ideas or requirements? revisit earlier worksheets as needed!

U / I / M / D

Fig. 6.4. Worksheet for the *deploy* activity. In the final activity, visualization designers seek to make prototypes more useful and evolve them into a polished visualization system. As part of this process, deployment must get the system into the hands of the appropriate audience, and the system must address key points of integration, speed up necessary processes, and improve the tool's aesthetics.

worksheets were less novice friendly due to this terminology, so we simplified the worksheets and recommendations or hints for each step. Still, we focused on capturing visualization artifacts and the evaluative decisions that get made by visualization designers. We also walked through one of our projects using the worksheets to identify further elements to add: more helper text, warning icons, expected results for each step, and a label at the top for attachments.

Before introducing design worksheets to students, we needed to form a basis of understanding, both in terminology and contextualized as a real-world visualization example. We created an 80-minute lecture on visualization design, which teaches both the design activity framework and the nested model for visualization design decisions [17]. This model was used to help categorize design decisions that were contextualized within a visualization design project, a cyber-security visualization dashboard [4]. By utilizing this real-world visualization project, we were able to explain the theory with actual, tangible concepts. We include a copy of our lecture materials and this example on the project website.[†]

To help teach a real-world design process, we mocked up design activity worksheets for the BubbleNet dashboard project [4]. These design worksheets, shown in Fig. 6.5, served as an example that we taught to students in our visualization design lecture. By illustrating how students could utilize the worksheets with a real-world example, we hoped that the worksheets would seem more tangible and actionable for novice visualization designers. By walking back through our design study process, we incorporated various visualization artifacts as linked sketches and printouts to match each with their respective design activities, in approximate temporal order. Illustrated in Fig. 6.5, these worksheets match the order of design activities presented in the project timeline, Fig. 5.1. Despite this design study being complex and iterative, both the design activity framework and design worksheets are able to succinctly capture and convey distinct and crucial aspects of the visualization design process.

6.4 Evaluating the Worksheets

In order to externally evaluate the framework and visualization design worksheets, we employed them in a classroom setting. We gave a lecture on visualization design to all 66 students in our university's graduate-level visualization course. The lecture was followed by an in-class exercise that had students analyze and redesign an existing visualization using the first two worksheets: *understand* and *ideate*. As part of the course, students formed groups to complete a cumulative project: to design and build a web-based interactive visualization system. I recruited and mentored 13 volunteers from the course to complete the design worksheets for each of their six group projects. One student was not part of the original volunteers, but due to complications with her project she reached out to the teaching staff for further help and guidance for visualization design within the context of her

[†] https://design-worksheets.github.io/

Fig. 6.5. Design activity worksheet example. We showed students how to use the design worksheets with linked sketches, summarizing our design process to create the BubbleNet dashboard [4]. This real-world project showcases how to utilize the worksheets and highlights how to capture a complex, iterative design process. A detailed copy of each example worksheet and associated sketches is included online with the worksheet materials.

project. For details on the project expectations, deadlines, and grading criteria, please see the project website.

To evaluate the efficacy of the design activity framework and worksheets in supporting the visualization design process of a cumulative, final project, we conducted a full-course survey, a focused survey for students who used the worksheets, and interviews with student participants to elicit in-depth worksheet feedback and clarify necessary details. Additionally, the mentor met weekly with each visualization group to provide feedback on their design process and on the worksheets. These meetings provided a basis for obtaining in-person observations, in addition to the feedback acquired anonymously through the surveys and detailed interviews. The questions and prompts used for the surveys and interviews are included in Appendix C.

To gather anonymous feedback and assess the utility of the framework and design worksheets, we sent an online survey to the students at the end of the visualization course. Specifically, we asked questions about students' comfort level with visualization design before and after taking the course along with which factors taught them how to design visualizations: lectures, in-class exercises, design worksheets, and the cumulative project. Additionally, we sent an anonymous, online survey to those who used the design worksheets. In this survey, we asked which worksheets worked well and which ones did not, and why, along with 10 questions about the usefulness of the worksheets. To avoid positivity bias, these questions varied between positive and negative wording.

After the student projects were completed, 11 students, at least one from each project, participated in a semistructured interview to provide feedback on the visualization design worksheets. This feedback serves as an external validation of the framework and worksheets for data visualization design. We audio recorded each interview to more efficiently take notes and transcribed participant responses to ensure accuracy and correctness. With each interviewee, we explained the goals of the study and acquired consent to utilize feedback and quotes for publication. The interview questions focused on digging deeper into the survey findings. We asked open-ended questions to gather suggestions for improvement and for the next iteration of the visualization design worksheets. At the beginning of the interview, we asked students to briefly describe the steps of the visualization design process in their own words in order to informally test recall of the high-level concepts of the framework and worksheets.

6.5 Evaluation Results

For the full-class survey, we received 25 responses. Twenty-three students showed an improvement in their comfort level for visualization design, on average 2 out of 5 points higher by the end of the course. Students ranked these improvements based on where they learned how to design, which was primarily through the lectures, projects, and class exercises. The design worksheets received a significantly larger portion of neutral responses for helping students learn, possibly because only some students used them in their projects. We compared the ratio of agreement to disagreement of these materials helping students learn. The design worksheets were on the level of other methods utilized in the course: design worksheets (13:1), lectures (23:1), exercises (20:2), and projects (18:2).

For the survey sent to the students who used the visualization design worksheets, we received a total of seven responses. Overall, *ideate* (six students) and *understand* (four students) were selected as the most helpful design activity worksheets for their projects. Students stated that the *ideate* worksheet helped them critique their own designs, and *understand* helped jumpstart a visualization project. As stated by students, *ideate "is the most clear worksheet"* and *"critique of one's own design was most helpful"* and that both *understand* and *ideate* worksheets *"helped to get the project off the ground."* On the flip side, the *deploy* (four students) worksheet was selected as the least helpful because students often did not have sufficient time to focus on this activity, as they stated, *"We were out of time"* and *"Once the projects were off the ground, [deploy] didn't seem too important."* Student feedback highlighted a benefit for the specific steps in an activity to organize and record their design process, explaining that for the worksheets: *"It seemed that the amount of text, guidance, and time taken achieved a proper balance."* Additional feedback highlighted some drawbacks to the worksheets, such as vague terminology or phrasing, creative limitations, and not enough structure. To uncover more information, we conducted interviews as a follow-up.

During the follow-up interviews, we asked students to describe the design process in their own words, and all captured the process with descriptions of the various design activities. Specifically, four students correctly recalled the names of each visualization design activity, but four other students were fuzzy on the *deploy* activity—possibly since most groups were not involved in this activity. As with the survey, all students found the *understand* and *ideate* worksheets the most useful since they forced them to consider different tasks, users, and ideas, whereas *deploy* was often not reached in the course of the project. Students noted that the worksheets provided a structured way to organize and compare notes about different visualization design artifacts. Three students stated that the worksheet example visualization project was helpful in illustrating how to use the design worksheets. Nine students followed their own design process informed or exactly prescribed by the design activity worksheets. One group that conducted their own visualization design process acknowledged that their design process, although different, still adhered to the steps provided in each visualization design activity. Another student recognized the flexibility of the visualization design worksheets: *"If I had a different project, I would use each box in different ways depending on the context"* (participant #8, or P8).

All students agreed that evaluation was a necessary and important step for visualization design in order to pinpoint flaws in their understanding of the problem, users, tasks, interactions, and encodings. One group discovered that their visualization project was better suited to a subset of users, and another group realized that a particular encoding resulted in points overplotting from feedback during an advisor meeting. All students agreed that design worksheets helped them document their visualization design process for their final project report. These design worksheets served as a *"snapshot in time"* (P1) and were sufficiently detailed to explain their design process for the report. Eight students described an iterative process that occurred, although informal and not written on any of their own design worksheets. When digging into this process, students pinpointed that the worksheets helped organize their thoughts. Furthermore, the activities helped guide them as novice designers, such as one student who used the visualization design worksheets for the first time later in the course of the project and stated that *"When I used [the] worksheets, it kept me focused on what I was doing and trying to get more ideas or more [encodings]"* (P8).

An intriguing finding was that four students employed the worksheets in surprisingly creative ways. For example, one student loaded the *ideate* worksheet in PDF form on her tablet and zoomed in to sketch various aspects of her visualization design, allowing her to expand and use more space for the visualization sketches. Also, another detail-oriented student transferred the design worksheets into textual form, listing all of the steps and hints, so that he could brainstorm and add detail to the problem and requirements over time, as a living document. For detail-oriented novice designers, having this additional space and flexibility helped harness their own creative ability. Four students expressed frustration with the paper design worksheets because they preferred another format, whether digital, larger paper, or the ability to structure their notes how they wish. As one student put it, *"I think the concepts are very helpful in the worksheets .... [but] for a free form thinker ... if you box it in then it is sort of restricting your creativity, as it tells you how much you have to fit into where"* (P9). Students suggested improvements and other feedback, which we explore next.

6.6 Improving the Worksheets

To address restrictions on creativity mentioned by that last student, a key improvement recommended by five students was to convert the visualization design worksheets into a checklist for each step, the same as the steps shown in Table 6.1. Based on the interviews, we recommend two formats for guiding the visualization design process: a checklist and worksheets. The worksheets did provide structure, *"It's like a checklist to make sure everything is covered"* (P11), but they also limited free-form thinkers: *"If you have a lot of things on your mind, you won't fit everything in the box anyways so the boxes are actually wasting paper"* (P6). Some visualization designers recommend paper for sketching [16], but others in the design community argue digital sketching can have functional benefits, such as shapes, undo, layers, duplication, and manipulation of details through zooming [43], which two students utilized and felt was vital to their visualization design process. Another recommendation was to transform the worksheets into an app: *"a clickable, interactive worksheet, where you click on this [and] it will connect you with the other worksheet and have a screenshot"* (P8).

Students also suggested adding more worksheets to the materials. Six students felt that *"those activities frame the process well"* (P2). However, two students brought up a crucial aspect of evaluation and feedback: that it might be worthwhile to devote a whole worksheet to these concepts, otherwise *"If you have it on the other worksheets, [evaluation] doesn't seem to have as much value"* (P10). A key challenge for a visualization design project can be finding the right dataset. Four students requested a visualization design worksheet to help probe into and explore the dataset or datasets that a group may want to visualize. By providing guidance, steps, and questions on aspects of the dataset, potential issues with visualizing the dataset could be avoided later, which is exactly what happened with two student projects. Lastly, three students requested a visualization design worksheet on how to structure the code of a visualization system, particularly in the case of one group with no computer science background. Such a resource would help students brainstorm on how to structure classes in their code, especially for building data visualization systems. Specific guidelines for particular languages, such as designing visualizations for the web in Javascript, could be useful even for more experienced student programmers.

Furthermore, some minor tweaks can be made to improve the visualization design worksheets. Three students noted that the gray, helper text on the worksheets confused them at a times, so a low-level editing pass and clarification could help the visualization design worksheets. One student even suggested fleshing out that text into more of a template but then providing blank boxes on separate design worksheets for each activity to be filled out with less clutter overall. Five students noted that having another example visualization project using the worksheets, such as a good student project, would help steer students toward knowing what to put onto the design worksheets and define more clear expectations. We also received recommendations to use a date-field rather than a blank number-field to encourage students to simply organize their group worksheets over time as the numbers were not often used and harder to coordinate among group members. We also asked students about the visual result and warning icons, and the consensus was that most students did not realize what these were for so their use may be superfluous or should be made more clear. Additionally, providing weekly advisor feedback was crucial for improving students' confidence in visualization design: *"because we got to meet with [an advisor] then we had time carved out [for] doing the worksheets"* (P5).

6.7 Summary

In this chapter, we introduced an external validation of the design activity framework using four new worksheets to guide and teach students through the visualization design process. For each of the four design activities, we identified five steps, four generative and one evaluative with additional tips and hints, to help guide visualization novices through the design process. By deploying these worksheets for use in a visualization classroom and cumulative project, we observed how students utilized and interacted with the worksheets. We conducted a qualitative evaluation of the worksheets using semistructured interviews, and we found that the *understand* and *ideate* worksheets were the most beneficial and appreciated by the students we worked with. Furthermore, students suggested adding more materials and worksheets, such as incorporating VizItCards [73] or new worksheets for exploring data and outlining programming advice for good code. As an initial, external validation of the worksheets and design activity framework, we consider this evaluation a success for actionably guiding students through a visualization design project.

7

Reflections on Other Types of Research

The design activity framework can apply to and provide insight into types of research beyond design studies. This is one of the benefits of the framework, since it can generalize to other types of research. In this chapter, we explore two kinds of applied research projects: technique-driven [3] and evaluation [5] research. After exploring each project, we reflect on the project's outcomes, research contributions, and implications using the design activity framework. From these reflections, we have noted that the design activity framework can capture steps of other research processes and the visualization artifacts that are generated and evaluated.

In practice, visualization design can have implications for many types of visualization research and be a research activity in itself, such as multiple approaches to conduct research through design [158], [159] or the application of action design research to the process of creating visualization systems [48]. For example, the design study methodology model includes research aspects, such as reflection and writing publications [8]. Action design research would similarly emphasize such reflection and learning throughout the visualization design process [48]. Members of the visualization community need to consider other types of research, beyond design studies or application-driven work. Munzner identified a series of publication types for the field of data visualization — technique, design study, systems, evaluation, and model [160] — which is similar to different kinds of research contributions, such as in the field of human-computer interaction [161]. In this chapter, we discuss two applied projects that contain primary contributions in technique and evaluation research.

For technique-driven research, we worked with a statistician and biology collaborators to develop a new method for encoding projected correlation data coupled with an interactive system to explore these correlation projections [3]. We reflect on the application of this technique to our biology collaborators' challenges, to explore large datasets for which current visualization methods for correlation did not scale appropriately. As a result of analyzing this project through the design activity framework, we pinpoint some of the design study pitfalls we fell into that previous visualization design models did not capture. By illustrating how the design activity framework describes technique-driven work, we show that design activities work as steps for general visualization design.

For evaluation research, we conducted an exploration and evaluation of a design space for visual narrative story flow, coupled with initial studies into a few types of flow, such as steppers and scrollers [5]. In this work, the design activity framework can be used to frame the steps of experimental design and identify study limitations from a lack of generalizability. The design activity framework timelines and worksheets further emphasize documenting and recording the visualization design process, which are important for reproducibility in evaluation research. Overall, we found that the design activity framework can support thinking about other types of research beyond just design studies, particularly research with an applied focus.

7.1 A Technique-Driven Project

In this section, we provide details for a technique-driven research project, the s-CorrPlot [3]. We first explain the motivation for this visualization technique, and then we provide the high-level mathematical derivation of the visual encoding as well as the interactive component of the technique. To conclude this project's description, we include the application of this technique to an applied domain, visualization for biology datasets. We reflected on this project using the design activity framework in order to identify pitfalls and clarify the reasons why this project strayed from being a problem-driven design study. We discovered that the design activity framework can apply to steps of a technique-driven design process and enable us to identify further pitfalls for design study research. Lastly, we found, through this reflection, that visualization ideas in technique-driven work can span across more levels of the nested model, such as a new algorithm.

7.1.1 Visualizing Correlation

Many data analysis applications use the degree of correlation between variables as a key measure of interdependence. The most common techniques for exploratory analysis of pairwise correlation in multivariate datasets, such as scatterplot matrices and clustered heatmaps, however, do not scale well to large datasets, either computationally or visually. We present here a new visualization technique that is capable of encoding pairwise correlation between hundreds of thousands variables, called the s-CorrPlot [3]. The s-CorrPlot encodes correlation spatially between variables as points on scatterplot using the geometric structure underlying Pearson's correlation. Furthermore, we extend the s-CorrPlot with interactive techniques that enable animation of the scatterplot to new projections of the correlation space, as illustrated in the companion video on our project website.* We pro-

*http://mckennapsean.com/projects/s-corrplot/

vide the s-CorrPlot technique and tool as an open-source R-package that we gave to our biology collaborators studying correlation of gene expression.

7.1.2 Statistical Correlation

The s-CorrPlot is based on a geometrical interpretation of correlation [162], [163] where datasets are represented with a **variable** as a vector in $\mathbb{R}^n$, where n is the number of **observations** per variable. In this interpretation, Pearson's correlation is the cosine of the angle between the mean centered variables. Thus, correlation can be spatially represented as p points on a $(n-2)$-sphere. In statistical language, the points on this sphere are termed standardized variables.

Pearson's correlation coefficient, $\hat{r}$, for any two variables $\mathbf{x} = \{x_1, \ldots, x_n\}$ and $\mathbf{y} = \{y_1, \ldots, y_n\}$ can be written as

$$\hat{r}(\mathbf{x}, \mathbf{y}) = \frac{\tilde{\mathbf{x}} \cdot \tilde{\mathbf{y}}}{||\tilde{\mathbf{x}}|| \; ||\tilde{\mathbf{y}}||} \tag{7.1}$$

which highlights the geometrical nature of the correlation coefficient [3].

In this geometric interpretation, the standardization of a variable can be viewed as a projection onto the **correlation sphere** — a specific, $(n-2)$-sphere embedded in $\mathbb{R}^n$. To standardize a variable, first, the mean of the variable is subtracted from each observation, and second, the variable is scaled to unit length. For any two variables, their correlation is now directly encoded through the relative positions of their **standardized variables**. For any two standardized variables close to each other on the sphere, their dot product, and thus their correlation coefficient, is close to 1, and, for those that lie on opposite sides, it will be -1.

7.1.3 The s-CorrPlot Encoding

The s-CorrPlot represents each variable as a point on a scatterplot as a novel way to encode and read correlation. The scatterplot results from an orthogonal projection of the standardized variables on the multidimensional correlation sphere. We can project the standardized variables that lie on the correlation sphere onto a plane through the origin. After this projection step, the variables can be displayed as points on a scatterplot, the **s-CorrPlot**.

Based on Equation 7.1, the correlation coefficient for two variables is equal to the dot product between their vectors, such as for $\mathbf{p}$ and $\mathbf{x}$ as illustrated in Fig. 7.1(a) and reflected in Fig. 7.1(b). The vertical grid lines in the s-CorrPlot, as in Fig. 7.1(c), specify values of equal correlation to $\mathbf{p}$ for any location in the scatterplot. Thus, since $\mathbf{x}_U = U\mathbf{x}$, it follows that, for any vector $\mathbf{x}$, the correlation to $\mathbf{p}$ is directly encoded in the first component of the vector $\mathbf{x}_U$. In fact, any vector that projects onto the line V_U, shown in Fig. 7.1(b), has the same first component value, and thus the same correlation to $\mathbf{p}$. Moving V along the vector $\mathbf{p}$ produces grid lines as shown in Fig. 7.1(c). For further details about the mathematical derivation of the s-CorrPlot, including the uncertainty bounds and density estimation technique, please see the original publication [3].

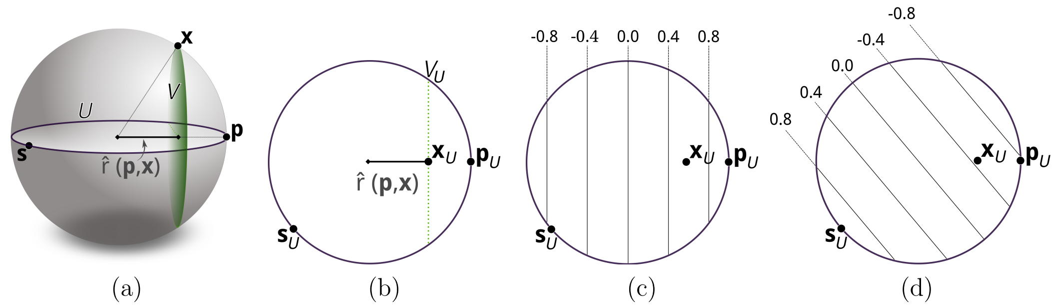


Fig. 7.1. Three variables (**p**, **s**, and **x**) with four observations each projected onto the s-CorrPlot. We can illustrate our variables as standardized vectors on (a) the correlation sphere, directly in 3D. The correlation coefficient between any two variables is the dot product between their standardized vectors, such as with **p** and **x**. With these two standardized variables, a $(n-2)$-flat V is defined. The s-CorrPlot is defined by the projection plane U, containing both **p** and **s**. Projection onto U results in (b) the s-CorrPlot, preserving correlation coefficients to both **p** and **s**. In the s-CorrPlot, V projects to a vertical line V_U of equal correlation to **p**. As such, these vertical lines can be generalized (c) as grid lines along U, denoting sets of equidistant correlation values to **p**. Similarly, (d) grid lines to **s** can be shown.

7.1.4 Interactive Exploration

This spatial encoding of correlation affords several advantages. For example, the technique can encode categorical information using color, and the projection is computed in linear time at interactive frame rates. We also designed the s-CorrPlot technique to incorporate both interaction and animation, unlike previous static correlation encodings [164]–[166]. In doing so, we illustrate how the s-CorrPlot can be paired with multidimensional exploration techniques, in the spirit of existing systems that employ user-driven exploration [167]–[169]. To easily understand the interactive exploration aspects, we advise watching the companion video available online.†

The s-CorrPlot employs several simple aspects of user-driven exploration to help examine the space of possible projections. These interactions increase the effectiveness of the underlying spatial encoding of the s-CorrPlot. Users drive the exploration of the multidimensional correlation sphere by selecting the variables **p** and **s** of interest. After selecting a new variable, the s-CorrPlot is reoriented through a continuous animation of a rotation between the current projection and a newly selected one, by interpolating across the vectors chosen for the projection. In addition, we orient the viewer by projecting the primary vector to a fixed location on the far right of the s-CorrPlot and draw the gridlines vertically with respect to this primary vector in order to preserve the spatial encoding throughout the animation. This animation results in seeing structures, such as clusters of correlated variables, moving together (or apart) in 3D; perceptually, this is known as seeing *"shape from motion"* [170].

† `http://mckennapsean.com/projects/s-corrplot/`

7.1.5 Employing the s-CorrPlot in Biology

While creating the s-CorrPlot visualization technique, we worked in tandem with biology collaborators to customize and tailor aspects of the data visualization tool for their problem: exploring correlation of gene expression datasets [3]. Biologists often analyze the correlation of **gene expression** — how much a gene is turned on or off in a cell — across datasets to gain insights into gene functions and to infer novel relationships between genes [171]. This analysis seeks to answer questions pertaining to the relationship of correlation between genes, especially how these relationships change over time, across species, or in the presence of disease.

We worked with a biologist at the University of Utah who is tackling similar questions by studying genes that work together in the brain in order to uncover genetic influences on brain function, behavior, and disease. Using high-throughput sequencing, he measures the expression level of genes in specific brain regions, even to the detail of expression of **exons**, which are subparts of genes. He takes these measurements in different strains of mice, which form the observations in his dataset. The genes and exons are the variables he wants to correlate and study.

His typical study involves several dozen observations, and approximately 10,000 to 100,000 variables, where each variable is a measurement of expression from a gene or exon in a brain region with observations across different subjects. The state-of-the-art approach for studying the correlation of gene expression is weighted gene co-expression network analysis (WGCNA) [172]–[174]. WGCNA uses the correlation or similarity of genes to construct a weighted network among all genes, and this network forms gene modules based on topological overlap. However, WGCNA was designed to support only 10,000 to 20,000 genes, so it does not scale to the size of datasets that our collaborator struggles to analyze.

At first, our collaborator explored 38,365 genes in two regions of the brain, with 22 observations, using the s-CorrPlot, shown at the top of Fig. 7.2. Since each gene can exist in either brain region, the plot shows a combined total of 76,730 variables. The gene expression levels measured in brain region 1 are shown in red, and those in brain region 2 are shown in blue. Our collaborator first looked at only brain region 1 (red), orienting the s-CorrPlot using the first principal component for these variables — he noted that no strong clusters emerged. He then did the same for only brain region 2 (blue), and saw a significant grouping of correlated and anticorrelated points, shown in the dashed ovals. Overlaying the two brain regions confirmed interesting differences across the correlation of all genes between these two regions. The biologist anticipated that the differences in the correlation structure of the data reflect differences in the cell types and mechanisms that regulate gene expression and the function of the two brain regions.

Using a different dataset, our collaborator visualized the expression levels of 120,000 exons across the same two brain regions, as shown in the bottom half of Fig. 7.2. This particular dataset contains 60,000 exons in each brain region, for a total of 120,000 variables, with each variable containing 37 observations. This is the first analysis of correlation at the exon level that our collaborator is aware

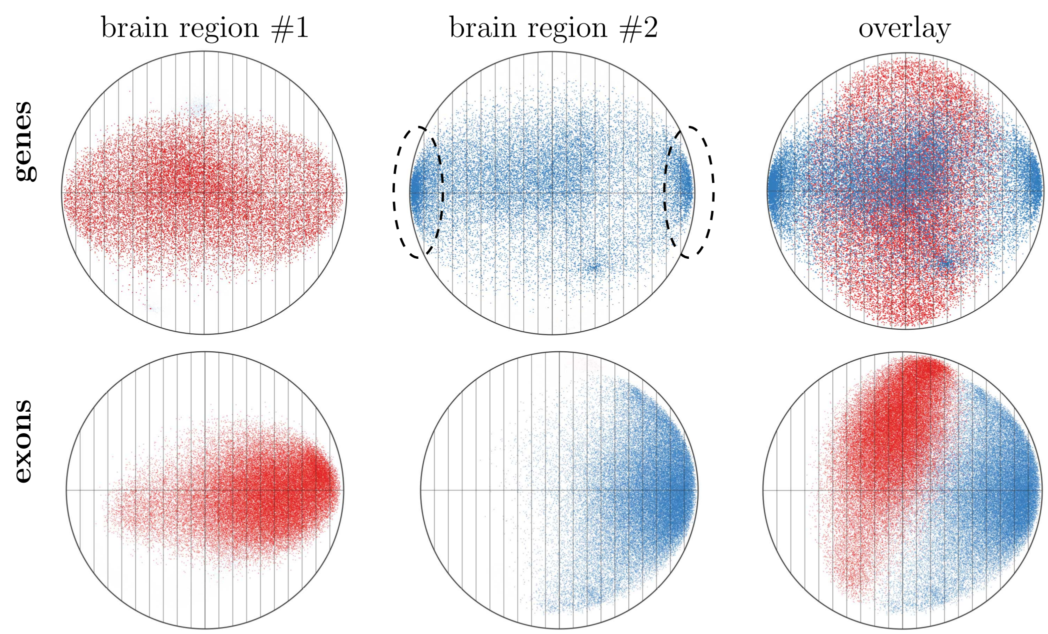


Fig. 7.2. Two biological datasets visualized in the s-CorrPlot. Each dataset contains 76,730 (genes) and 120,000 (exons) variables, with 22 and 37 observations, respectively. For each dataset, genes and exons have been colored according to two brain regions in which the expression levels were measured, resulting in separate and combined overlay visualizations. The s-CorrPlot highlights different patterns of correlation in each of these brain regions due to the gene and exon expression patterns varying on a global scale, i.e., the blue and red regions of these plots show different patterns, distributions, and clusters across the two regions, which indicates potentially significant differences in their biological processes.

of, perhaps due in part to the inability of existing tools to handle these large datasets. With the s-CorrPlot, our collaborator was able to interactively explore the many exons and deduce that there are also region-specific patterns at the exon level. He noted that the patterns in the exon dataset are significantly different from the data at the gene-level, indicating that differences in these brain regions could be described at a smaller scale than genes.

Taken as a whole, the differences in the patterns between the two regions of the brain are completely unknown and unexplored in our collaborator's field. These observations have prompted him to design follow-up computational studies and wet-lab experiments, fueled by hypotheses, which are formed by his use of the s-CorrPlot for correlation analysis. Based on Fig. 7.2, he commented: *"This is revealing new brain-region specific patterns in the data that we were completely unaware of. It offers the potential for deriving entirely new hypotheses about the functional relationships between genes in different brain regions that we can test experimentally."*

7.1.6 Applying the Design Activity Framework

Although we created the s-CorrPlot technique for biology collaborators, this project was not a typical visualization design study. In fact, this project's primary contribution was a novel technique, generalizable and driven by an algorithm stemming from statistical theory. When evaluating this tool, we focused not on the problem domain or use-case in detail but rather if the technique provided analytical insight through a case study with our collaborators. This validation is in line with technique-driven research [28] but not with a design study, which involves validation of contributions consisting of a problem characterization, abstraction, and tool that solves a domain expert's problem [8]. When reflecting on our process with the design activity framework, I realized that we never fully characterized or tried to solve a more complete problem. When using this new tool, our collaborators would find interesting patterns but then only had more questions and needed to run further lab studies to learn more. We simply focused heavily on the novelty within the specific tool rather than a design study.

Design studies can be faced with threats to their validity at multiple levels, such as one at the problem-characterization level: *"wrong problem"* [17]. This threat is characterized as *"target users do not in fact have these problems"* [17], but the s-CorrPlot tool did, in fact, solve a problem faced by our collaborators, since they were unable to visualize their full dataset with existing techniques. Using the design activity framework, we were able to realize that instead what this project faced was an "incomplete problem" threat, where we focused too heavily on only one of our collaborators' dataset types. By not taking a step back for the bigger picture, we left our collaborators with more questions than answers using this tool, and this design study could have been improved by revisiting the *understand* activity and fleshing out the situation blocks [18] more completely. Nevertheless, this project was still a success as technique-driven research. We can characterize the contributions of the s-CorrPlot using the levels of the nested model, touching on the encoding, interaction, and algorithm levels. The nested model is useful when determining which forms of validation are necessary to evaluate a visualization system, and the types of evaluation we employed on the s-CorrPlot match these levels: complexity analysis, interactive framerates with large datasets, qualitative image analysis, and case studies with our collaborators.

By turning to the design activity framework, we formed insights about the design process and application of the s-CorrPlot technique to design study work. For the s-CorrPlot project, I joined the team after the development of an initial prototype; the nine-stage framework [8] would classify this as the "implement" step or the *make* design activity. To improve the tool to meet our collaborators' needs, I visited their lab weekly to observe the types of research problems they faced, better understand the problem domain, and see how they utilized the prototype, which is like going back to the "discover" stage or *understand* activity. While following the nine-stage framework, we had an incomplete problem focus and fell into several problems in the "design" stage [8]. However, when looking to the *understand* worksheet, I was able to pinpoint several missing factors from our design process: thinking about users' large-scale problems or challenges, different types of datasets to solve this, and tasks to perform on the data. Our design process for this project had focused on talking with and observing users, and the prototype that we had deployed early on impacted the focus of these interactions significantly. Rather than focus on the bigger picture and problems faced by our collaborators, we had iterated on a prototype visualization tool to solve a single problem. Thus, our reflections on the design activity framework contribute new pitfalls for design studies, adding to previous ones [8]:

- PF-33. not communicating information across teammates on the problem characterization and abstraction (discover / *understand*)
- PF-34. failing to identify the broad problem: solving small, specific problems may not be useful or impactful enough for domain experts (discover / *understand*)
- PF-35. not tailoring the system for domain experts but focusing on designing novel solutions (discover / *understand*)
- PF-36. deploying a prototype too early, thus limiting the problem and design focus — instead aim to develop technology probes [98] to shape design requirements, abstractions, and ideas (*deploy*)

The design activity framework, as a process model, is able to capture multiple aspects of the s-CorrPlot project. When designing this tool, there were distinct, iterative phases of the prototype. For example, the scatterplot encoding is one such visualization artifact, which we validated in terms of its ability to scale across many variables and dimensions, and to overlay multiple datasets. A concept separate from the tool is the notion of interaction, with selection of and animation between projections; this concept was not a novel contribution on its own but added further value to the technique and s-CorrPlot tool. Most of these artifacts stem from the *ideate* activity, but they were realized in a prototype system through the *make* activity. We repeatedly deployed this system to our collaborators as well. The visualization artifacts of this technique-driven process correlate to the levels and contributions designated by the nested model, and we note that a modification of the design activity framework is required to allow ideas from technique-driven research to map to algorithmic-level decisions [17]. We have illustrated here how the design activity framework can uniquely apply to general visualization design, such as the technique-driven project employed here. In particular, the framework seems to work with research that has this applied focus.

7.2 An Evaluation Project

In this project, we focused on several evaluation methods that enabled us to generate guidelines for the applied area of creating data visualization stories [5]. First, we preface this research with an overview and motivations for the project, to study the visual narrative flow of data visualization stories. By providing examples of two types of flow, scrollers and steppers, we show how different these story experiences are to readers. Next, we characterize this design space for visual narrative flow into seven flow-factors. We conducted a crowdsourced study in order to measure and compare engagement for different narrative flows. By reflecting on this project using the design activity framework, we provide an initial validation that steps for experimental design can map onto steps of the design process, where visualization artifacts can capture evaluation guidelines and increase the reproducibility through documentation of an experimental design. We also discuss how the framework enabled us to reason about the evaluation methods' limitations of generalizability.

7.2.1 Project Motivation

Data-driven stories that tightly integrate visualizations have become a popular communication device in a variety of fields [175], which has led the visualization research community to investigate the design factors that practitioners employ to craft narratives, from visual and interactive techniques [175], [176] to specific genres [177]–[179]. Specific knowledge on these factors is growing, but we still have little knowledge of which are predominant for, and how they may be combined to create, effective ***visual narrative flows*** [5], which combine a reader's input with story components and congruent visual feedback that tell the story matching the author's intent and voice.

An ongoing informal debate on visual narrative flow centers around the effects of allowing readers to navigate through data-driven stories using either a click/tap input or a scroll input. We refer to this debate as the *stepper vs. scroller debate* (illustrated in Fig. 7.3). Clicking to step through a story is like a slideshow, while scrolling is akin to panning up and down a long document. Practitioners from *The New York Times* recently advocated for scrolling because their readers tend not to fully consume stories that are delivered with steppers [180], [181]. Others advocate for steppers, as they point out several potential issues pertaining to the use of scrolling (e.g., "scrolljacking") [182].

(a) scroller narrative flow (b) stepper narrative flow

Fig. 7.3. Two examples of visual narrative flow. These two examples highlight differences within a data visualization story. The scroller, by Yee and Chu, walks through (a) a story to teach a basic concept of machine learning [183], where scrolling not only moves down the page but also moves visualizations and continuously controls their linked animated transitions. We transformed this scroller into (b) a stepper narrative flow that uses buttons for navigating the story across the story text with timed animated transitions.

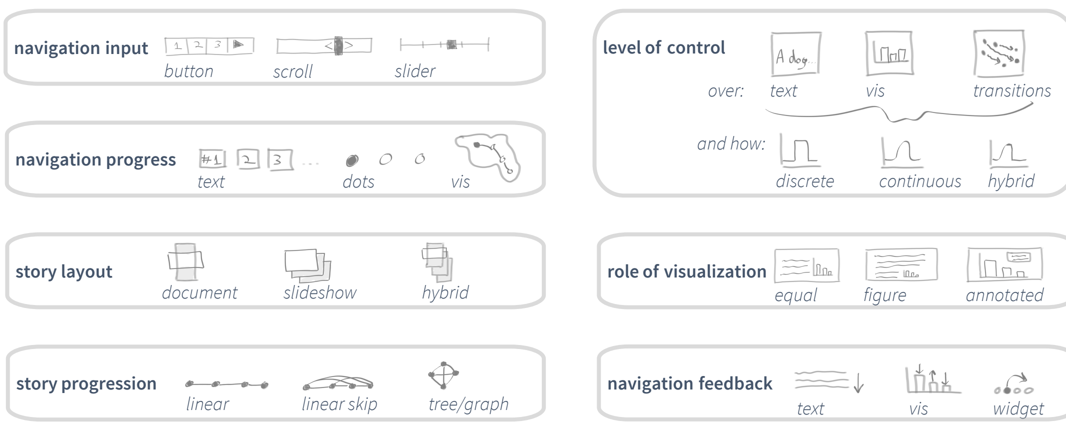


Fig. 7.4. Seven factors for visual narrative flow. We identified these flow-factors by analyzing and coding a corpus of 80 visual data stories. Each flow-factor contains multiple properties, which are not necessarily mutually exclusive, and hybrids can and often do occur in data visualization stories created by authors.

7.2.2 Visual Narrative Flow

Here we introduce seven factors that contribute to visual narrative flow along with illustrations of the various properties for each flow-factor as in Fig. 7.4. These factors were identified through a series of individual consumption sessions, group discussions, and informal coding of a corpus containing 80 stories [5], referred to with *S-#*. This design space framework captures aspects of flow such as a reader's input, connection of story components, and the visual feedback. Previous work by Segel and Heer focused on high-level story components (e.g., animation, progress bars) along with story genres and narrative approaches [175]. However, these flow-factors build upon their work by breaking down these properties to characterize and explore a broader range of visual data-driven stories than otherwise initially possible. For further details on each of these factors, please see the original publication [5].

The expressivity of the design space can be evaluated by looking at the model's descriptive and generative power [7]. One type of visual narrative flow is the **stepper**: linear skip progression with button or swipe input, discrete control over elements, slideshow layout, and a progress widget. Conversely, **scrollers** commonly have linear progression, continuous control over elements, document layout, and no progress widget. There are also different kinds of scrollers, some of which use discrete control to trigger animations (e.g., *S-2*, *S-3*, *S-5*) whereas others do so continuously based on the scroll position (e.g., *S-1*, *S-13*, *S-18*). Thus, the design space characterizes differences between these discrete and continuous scrollers that the community previously called one category, demonstrating the framework's descriptive power.

7.2.3 Crowdsourced Study on Engagement

To study how different flow-factors of visual narrative flow affect readers' engagement, we conducted a large-scale crowdsourced study with 240 participants using Amazon Mechanical Turk. We selected four conditions to study in detail. We identified two baseline conditions, one with only text and another with only static visuals, to first see if there is a measurable benefit to scrolling stories with visualization or with animation. Lastly, we included a stepper narrative flow to explore measuring the difference in engagement we witnessed in our exploratory studies. Specifically, our conditions were:

- *text*: a text-only story (baseline 1)
- *visual*: text paired with static visual images (baseline 2)
- *stepper*: text paired with visualizations and animated transitions via a stepper
- *scroller*: text paired with visualizations and animated transitions via a continuous scroller

For the two baseline conditions, we hypothesized that the inclusion of visualizations (***H1***) and animation (***H2***) would increase the visual appeal, attention, novelty, and felt involvement (all attributes of engagement) for readers. Furthermore, from our observations of readers, we hypothesized that transitions of dynamic data are more engaging using continuous control than discrete one (***H3***). In other words, readers from previous studies expressed that continuous scrolling was more of a gimmick until they experienced the final story chapter, which uses continuous scrolling to show the time steps of an algorithm. Thus, our hypotheses were:

- ***H1***: Visualizations contribute to make the data-driven story more engaging.
- ***H2***: Animated transitions contribute to make the data-driven story more engaging.

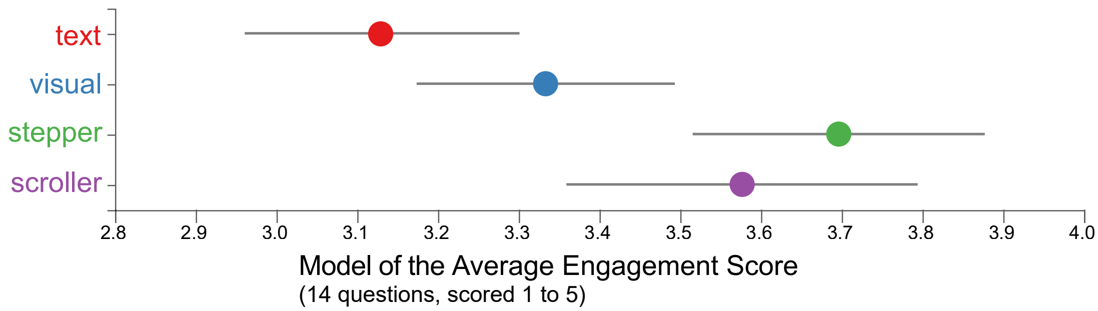


Fig. 7.5. Average engagement score across conditions. These scores stem from a mixed-effects model that represents the average engagement score and 95% confidence interval of all 14 questions for 240 participants across the four conditions. The model shows increased engagement when using visuals and especially when using animated transitions, but the effects of the other visual narrative flows, stepper and scroller, were not significantly different.

- ***H3***: Pairing dynamic transitions with continuous control contributes to make the data-driven story more engaging.

We broke apart the machine learning story [183] into two chapters, and each participant went through each chapter and then filled out a survey at the end. For measuring engagement, we adapted a validated questionnaire from O'Brien and Toms [184] containing 14 questions on reader-perceived engagement across attributes such as usability, attention, aesthetics, and novelty. We performed a linear mixed effects analysis using R [185] and lme4 [186] to study the relationship between different types of narrative flows and reader-perceived engagement (all 14 questions). We obtained the p-values reported here through likelihood ratio tests of the full effects model to one without the effect of different visual narrative flows. Our project website includes all materials and conditions used for the study.[‡]

Fig. 7.5 shows the results of the model, which contains the average engagement score for all 14 questions. According to the model, the different conditions tested for this story affected the engagement score significantly ($p < .001$, $\lambda^2(3) = 30.71$), supporting ***H1***. In other words, readers ranked the engagement of stories with visualizations higher than the first baseline, text-only story. It is important to note that these effects are small, since they are averages of the 14 questions on engagement. Certain questions had a stronger effect across the visual narrative flows, such as visualizations scoring higher on an aesthetics question: *"This reading experience appealed to my visual senses."* We provide an engagement model for each question and condition on our project website.

The animated transition conditions scored significantly higher on engagement than static visualizations ($p < .001$, $\lambda^2(2) = 18.04$), supporting ***H2*** and suggesting that animated transitions increase the reader-perceived engagement. We tested interaction effects of the engagement per chapter of the story but found none to be significant, and so they were left out of the model. Additionally, we found a significant effect on the chapter of the story ($p = .017$, $\lambda^2(1) = 5.72$), where the sec-

[‡] https://narrative-flow.github.io/

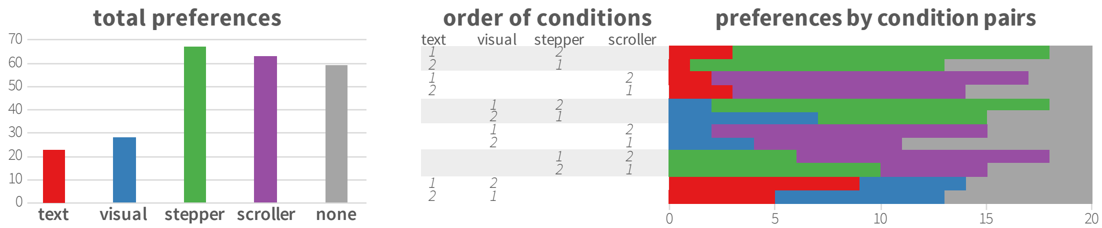


Fig. 7.6. Participant preferences across conditions. We recorded readers' preferences between their two selected conditions ($N = 240$). Each pair included 40 participants and was balanced based on which condition was first or second. On the left, preference totals across all conditions emphasize that participants largely preferred conditions with visualizations and animation (stepper, scroller); otherwise they had no preference.

ond chapter received, on average, a higher engagement score (0.10). In other words, the animated conditions scored even higher in engagement for the second chapter of the story, which contained dynamic transitions. A question on novelty scored higher for flows with animations: *"The reading experience was different from a typical online reading experience."* Subsequently, animated transitions, such as navigation feedback, showed a measurable benefit for reader-perceived engagement.

Regarding the stepper versus scroller debate, we did not find a significant difference in engagement via our questionnaire, failing to support ***H3***. Whereas steppers scored higher for engagement on average, the difference over scrollers was not significant in the model. Thus, we are unable to conclude if continuous control over dynamic transitions, via scrolling, improves engagement measured in the second chapter of the story. We note that the difference did vary by reader preference and only for certain questions, such as those regarding usability. Although our findings do not support ***H3***, a carefully controlled user study may be able to investigate and measure this effect.

Furthermore, we conducted an analysis on the preferred conditions selected by the participants, shown in Fig. 7.6. Note that each participant saw two conditions, the order shown in the table. They ranked which of the two they preferred for the story, or possibly none. We found that the stepper and scroller were largely preferred over other conditions by almost twice as many participants. However, a large portion of the participants overall did not have a preference between the two conditions they experienced. Lastly, Fig. 7.6 shows split in preference across participants for both stepper and scroller. By inspecting the detailed breakdown of preferences, more participants preferred animated transitions and scrollers for the second chapter of the story.

7.2.4 Applying the Design Activity Framework

This project aimed to support the creation of a tool to help story authors write, develop, and combine visualizations and animated transitions with text. Although this project was also applied research, a major focus of this work was to evaluate the qualitative story reading experience and compare engagement of different visual narrative flows. As such, a process model for design studies is not an apt fit to describe the steps we went through. However, the design activity framework is able to generally describe some of these stages, to create visualization artifacts, such as study hypotheses and story "prototypes." One stage of this project was qualitatively coding a corpus of many visual data stories, and this method was quite similar to other times we employed this approach in the *understand* activity. As before, the design requirements we identified were critical to shaping the project, and here they are the seven flow-factors we identified that motivated how we conducted user evaluation. Then, we employed user studies that resulted in guidelines for future tools. The design activity framework is able to capture and describe the results of these evaluation methods to facilitate other visualization practitioners designing and evaluating tools and techniques in this space.

By connecting the design process to design decisions, we were able to reflect on how the study's guidelines fit within the overall design process of these evaluation methods. By mapping to the nested model, the design activity framework emphasized that the evaluation and user studies for this project correspond to decisions made at the encoding and interaction levels. For example, the different types of visual feedback, interaction techniques, and levels of control all correspond to decisions we had to make when designing a story on one or both of these levels. With this mapping to decisions, we identified a limitation to the project. Specifically, the crowdsourced study investigated reader-reported engagement and preference to compare visual narrative flows. As a result of this study, we established guidelines to utilize animation and visuals to increase engagement, implications for the *make* and *deploy* activities. While the design activity framework can describe these guidelines succinctly, the nested model extension [18] elucidates this limitation, since the stories could be viewed as stacks of blocks, and the user study guidelines were shown to be effective for only a single story rather than generalized across many different stories.

On the other hand, we noted that we could reflect on how the design activity framework can be utilized to model and describe this experimental design process. As before, the experimental designer must first approach a given problem to identify potential tasks, datasets to utilize, and which users to test, and then establish hypotheses for the research problem. For example, this study explored aspects of visual narrative flow for a single dataset and focused on reading and comprehension tasks to measure engagement using a questionnaire. Many of these components correlate with actions and visualization artifacts of the *understand* activity. A similar design process can be followed to create visualization artifacts used as materials in a study, such as tools, techniques, systems, and instantiations of a given encoding or interaction. This process will involve testing for usability and feasibility, such as through the use of pilots, similar to prototypes, in the *make* activity. The process may not perfectly overlap, but it shows that the design activity framework can map to this style of formative or evaluative work in ways that the nine-stage framework cannot. Additionally, the lack of generalizability of the study presented here could be uncovered by reflecting on assumptions and artifacts generated, such as the single story, one dataset, and subset of visualizations, which is fixed throughout all of the conditions.

Lastly, the design activity framework worksheets emphasize the importance of documenting and recording the design process, which overlaps with the importance of reproducibility in experimental design. By recording visualization artifacts and design decisions, a visualization designer can justify why specific tasks, encodings, or interactions are selected or modified. In a design study, this notion of reproducibility is not the goal, and Sedlmair et al. promote the goal of transferability instead [8] due to the subjective nature of field work and based on methodologies of ethnography and action research. However, in other types of design work, such as for experimental evaluation, it is of vital importance to include materials and decisions to enable effective reproduction of the evaluation in order to validate, or invalidate, its results. Providing enough of the materials and procedures used in an experimental design allows others to build off of, refute, or refine the guidelines established as a result of an evaluation approach. By capturing more design decisions and visualization artifacts throughout the process, the design activity framework could increase the reproducibility of a project's evaluation. This record can even provide benefit for application or problem-driven work for others with similar challenges who want to build on the requirements, ideas, prototypes, and systems that have been generated and shared with the research community.

8

Discussion and Future Work

The design activity framework lays the groundwork for further research and models that connect creative and engineering processes [14], [36]–[38] for visualization design. Through its design activity motivations, visualization artifacts, design worksheets, timelines, design methods, and connection to design decisions, the design activity framework is more achievable, flexible, justifiable, discoverable, and actionable than existing models in the field. By building on these existing models [8], [17], [18], the design activity framework retains the application and benefits from these existing models while also extending its scope to other kinds of situations and visualization projects, particularly for applied visualization design. However, the design activity framework is by no means complete. Future projects may reveal new aspects such as crucial activities for visualization design or different kinds of visualization artifacts. The design activity framework can and should evolve over time to adapt to more kinds of visualization design projects to keep its use valid and current with how visualization systems are designed for people.

In this chapter, we first explore the necessary scope, limitations, and avenues for future work based on the design activity framework (Section 8.1). Although the design activity framework can apply to other types of research beyond application-driven design study projects, more examination and validation are required to investigate this notion in further detail. Next, we discuss implications from the first case study, a cybersecurity redesign project (Section 8.2). Following this discussion, we focus on insights gained in our second case study for cybersecurity on the topics of winnowing users, casting roles for collaborators, and utilizing multiple discourse channels for visualization design (Section 8.3). The use of user-centered design methods impacted both case studies positively, capturing visualization artifacts across design activities (Section 8.4). More work still needs to be done to continue the validation of this framework, and we explore the limitations of the evaluation techniques used for the design activity worksheets in the classroom (Section 8.5). Next, we relate these design processes to those used in development and software engineering, within the context of building data visualization systems. Specifically, agile methodologies are a popular choice for software engineering processes that have begun to combine engineering and creative approaches (Section 8.6).

8.1 Scoping the Design Activity Framework

The design activity framework is one approach to capture the steps of the visualization design process, and we have argued that one of its benefits over existing models is not only its increased actionability in a wide array of projects but also its comprehensibility for visualization designers and other collaborators. By extending the core design phases of the nine-stage framework for visualization design studies [8], the design activity framework inherits many of the connections to existing design study projects, while also considering the broader applications of visualization design. Another key component is its connection to the *what* and *why* of visualization design: design decisions. By highlighting connections to the nested model [17], the design activity framework supports visualization designers through the act of carefully and methodically identifying appropriate methods for validating and evaluating visualization artifacts in a design activity.

The framework promotes increased design process flexibility by enabling and emphasizing a workflow that includes both the nesting of activities and activities occurring in parallel. As shown in Fig. 3.6, the design activity framework can represent a process in which many activities are pieced together in different ways according to the motivation of the project at any given time. We believe this flexibility enables the framework to more completely capture the true nature of multilinear, real-world visualization design in ways that previous visualization process models and their representations do not.

In addressing the design process more generally, the design community does not have a consensus on any particular process model [60], nor do members of this community even agree that any such model could capture the "black box" of design [39]. Furthermore, considering design as a "wicked problem" [38], [158], [187], [188], it can be challenging to know where to go next, when to stop, and what makes an effective design [187]. These challenges exist for many design process models, including the design activity framework, pointing to opportunities for further investigation.

As a process model, the primary goal of the design activity framework is to guide visualization designers through a design process. We believe that the framework will be useful to those with a broad range of expertise. The actionability of the framework stems from the inclusion of more than just activities and methods, as is done in other models such as the nine-stage framework [8]. Specifically, the design activity framework also includes motivations, visualization artifacts, and explicit ties to the nested model in order to help guide a designer through the visualization design process. The motivation enables a visualization designer to determine which activity is currently being performed, which then allows the designer to identify potential methods, clarify artifacts, and make decisions with respect to the nested model. Although the design activity framework targets problem-driven visualization work, we could not identify a concrete reason why it could not be useful for technique-driven work as well; we reflected on previous projects in order to explore this extension.

As illustrated in Chapter 7, the design activity framework has the ability to encompass and represent other types of research beyond only design studies, such as technique-driven or evaluative research projects. These projects are just the beginning of formalizing adaptations of the design activity framework for these styles of research. For example, custom worksheets for research activities within these approaches, with varying artifacts and suggested methods, could be created, utilized, and validated. It may be necessary to reshape the design activity framework and the concepts on which it is based for these different research modes. This is an interesting avenue for future work to explore and shed further light on the connection of research activities, such as reflection, learning, and writing [8], [48], and where they fit in the broader context of visualization design.

Visualization design, from experiments to systems to encoding or interaction techniques, plays a role in visualization research beyond just that of design studies. Researchers may often still work in teams and receive benefit from common terminologies for the research activities they perform. However, it is also unclear if the design activity framework is the appropriate model for all these kinds of work, so new models or extensions may be necessary. We recognize that problem-driven research encompasses many types of projects, however, and the design activity framework may be a useful lens for a variety of visualization design projects. Moreover, future work should explore these facets and consider how to best teach, guide, and support visualization novices in all aspects of visualization design.

It is also important to adapt this framework and process beyond academia and pedagogy, to better understand how visualization designers work in industry and on product teams to build visualization tools and systems. It may be necessary to adopt more applicable or succinct terminology for such applications, and these modifications could benefit the design activity framework as a process model for a whole variety of interdisciplinary visualization projects engaging members such as visual designers, data visualization experts, and software engineers.

The design activity framework has several limitations, the first of which is that the framework's connections to the nested model may not always be as clean as those shown in Fig. 3.4. We were able to identify several corner cases where visualization artifacts could begin to overlap with an additional level of the nested model. Furthermore, the framework does not include a planning activity, which is present in other process models [8], [12]. Although important for design, we feel that planning is unique and complementary to the design activity framework. For example, the *precondition stages* of the nine-stage framework [8] could be combined with the design activity framework to serve as the planning activity. Lastly, we believe that there is still much to understand and articulate about the design process for visualization. We consider all design models to be a work in progress, and the design activity framework is by no means excluded. Further research could extend the framework, including more finely defining or breaking apart specific activities, adding new activities, or making the connection to a different design decision model.

Another limitation to this framework is the methodology by which it was created. We created the design activity framework based on our own experiences and after an extensive literature review. However, design in the real-world may not always coincide with these experiences, largely from academia. It is important to recognize this limitation, as visualization design may vary in other disciplines, such as in industry. Furthermore, the design activity framework has numerous visualization specific components, but certain aspects are also generalized for broader design. To evaluate the efficacy, utility, and application of this model, it would be important to validate in other domains with other types of designers in a wide variety of situations, design methods, and target artifacts.

To this purpose, it would be useful to evaluate the design activity framework further. Since the framework was largely established through our own experiences, another useful external evaluation of the framework would involve other visualization designers, or even general user interface or experience designers. This evaluation could take place through a variety of means: interviews, design workshops, group discussions, or card sorting, to name a few. Similarly, the design worksheets and concepts could be applied in new design projects, to measure the time taken, methods explored, ideas generated, or quality of the final artifacts, in comparison to a team's previous design projects. These evaluations could also support extending the model to other kinds of design. By working with other designers, the design activities, worksheets, and concepts of this visualization process model could be compared and evaluated by other real-world designers.

The design activity framework bridges creative design with an engineering process model, but there is plenty of future work to pursue and explore along these lines. For instance, we reviewed a handful of design models which use a grid to represent design activities, but many more creative design process models exist that may utilize other paradigms or concepts. On the other hand, there are also a number of engineering process models which utilize cyclical design concepts which we did not utilize. These concept variations could be adapted into the design activity framework or expanded into their own process model. It is important to consider these different styles of thinking and working, as a single model may not work for all types of visualization designers or all kinds of situations.

A number of open questions remain for future work. For example, we established, evaluated, and reflected using this framework for problem-driven methodologies, and it would be useful to rigorously, but cautiously, validate the use of the framework for other methodologies, such as algorithmic-driven work. Furthermore, in the list of exemplar methods we include novel methods for visualization design, but the utility and effectiveness of these methods for designing a visualization system have yet to be tested and verified. Most visualization process models have not yet addressed a series of challenging questions: Where should I go next in the process? What method is the best for my situation? When do I know my design is effective enough? We believe these future directions provide rich opportunities for models to further explore the visualization design process.

8.2 Working in Design Teams

Throughout our redesign project in Chapter 4, we worked closely as a design team composed of designers, a psychologist, and visualization practitioners. Our different perspectives and experience led to a richer and more informed design process. When working together, we found that having common terms and definitions for design was critical in promoting effective and efficient communication among all members — as such, we spent significant time and effort learning from each other to better understand, and speak in, each other's domain languages [189]. This effort allowed us to synthesize the ideas and perspectives on the design process from several different fields in the design activity framework. Moreover, we coined the framework's terms to help future visualization design teams with a common set of definitions and terminology that can be used for communicating specific aspects as well as an overview of their design process, as in the form of design timelines.

Working in design teams can provide its own set of challenges and obstacles. For example, this project involved a separate developer role that had limited communication with the design team, so the decisions and changes to the tool and code were made by the company. By reflecting on how industry and product teams conduct similar processes, e.g., with agile methodologies [190], [191], our approach is clearly very different. In these teams, the designers will often work hand in hand with the developers to identify potential areas for improvement and features for a sprint or development cycle. This redesign project was a special case, since we were invited to partake in the project specifically for our outside experience and knowledge, but integration of design and development teams can increase communication and realization of ideas and potentially address more user needs than if these teams work separately and with little communication. Having teams work together or comprehend the design activities and visualization artifacts more completely could increase the level of communication across teams and have a higher rate of impact when utilizing a common framework to discuss and present aspects of visualization design.

8.3 Casting, Collaborators, and Channels

Upon reflecting on the BubbleNet design study in Chapter 5, we realized that winnowing and casting of user roles[8] occurred later in the user-centered design process highlighted in Fig. 5.1. Unlike a typical design study, our set of domain experts were unable to give dedicated, recurring time to the project. By reviewing previous detailed cognitive studies of users and through interviews, personas were crafted to identify different potential users [2]. As a result, users were winnowed into two types, analysts and managers. This approach was motivated by domain constraints: limited access to users and data. Furthermore, the design activity framework highlights where the winnowing and casting of user roles occurred when revisiting personas in the *understand* activity, and we incorporated these roles when evaluating the dashboard in *make* with a usability study. The design process figure, when incorporating these multiple channels, succinctly shows where and how users were involved with different generative or evaluative methods as well as deployments.

The task of presentation influenced the unique design process of this project. Presentation inherently involves two or more parties, so it could involve users beyond a data analyst. In a design study methodology, Sedlmair et al. describe several different collaborator roles, such as front-line analysts and gatekeepers [8]. Simon et al. identified alternative collaborator roles, such as liaisons [192] who bridge visualization research to complex domains. Although we worked with several liaisons, the user personas identified four kinds of users, only one of whom, the network analyst, is a domain expert in cybersecurity. Other users, such as network managers, have some domain knowledge, but another domain was clearly at work here: an organizational domain. Large organizations need to disseminate information up a chain of command in order for decisions to be made and passed down [2]. With multiple domains and types of users, this work challenges the role of a single domain expert as the optimum collaborator. It is important to identify these different user roles and design tools that adapt to their needs, and the design activity framework, especially the *understand* worksheet, pushes these concepts.

Lastly, working in the cybersecurity domain benefited from the multiple discourse channel approach [106], as highlighted in Fig. 5.1. By reflecting on our design process, this multiple channel approach is particularly beneficial with the unique design constraints we faced: limited access to users and data, multiple types of users, and balancing trade-offs to deploy tools. The design of BubbleNet occurred within the second channel at a research organization, but this design would not have been as successful without the design methods and knowledge gained from the other channels. For example, the third channel represents a collaboration with a university network analyst, which enabled us to validate abstractions of network security data and critically changed BubbleNet's location view. By working at an operational organization in the fourth channel, BubbleNet's design influenced and inspired new encodings to be implemented by a team of developers, leading to operational tool deployments. As discussed in Chapter 5, deploying a tool is a complex process that involves further design trade-offs, but the visualization community needs to discuss these trade-offs in order to get tools in the hands of users.

8.4 User-Centered Design Methods

In our case studies, we demonstrated how user-centered design methods can be both efficient and effective for visualization design. Specifically, we highlighted the projects' visualization artifacts, guiding motivations, and final results of each design method: qualitative coding, personas, and data sketches. When performing these methods for data visualization, we noted that the motivations and visualization artifacts aligned well with the activities of the design activity framework, both *understand* and *ideate*. These methods can also be used in other activities, i.e., qualitative coding can be used as an evaluative method paired with other techniques in any activity, and data sketches could also be utilized in the *make* activity when parallel prototypes are being built and tested using this approach.

User-centered design methods can also help a designer establish user needs, uncover design opportunities, and evaluate ideas. These types of design methods can be particularly useful in the early stages of *understand* and *ideate* for the visualization design process. The three design methods discussed in both case studies can involve any number of users. We encourage future visualization design projects to broaden the methodologies, methods, and techniques at their disposal in order to more completely explore the design space for data visualization in a given domain. Ultimately, embracing user-centered design method will help us as a community be more efficient at building effective visualization tools across domains, users, data, and tasks.

8.5 Design Worksheet Evaluation

When using the design activity worksheets in the classroom, we observed the process by which students perform visualization design with these worksheets, guiding them through actionable steps and facilitating effective visualization design discussions both within a group and with their mentor. As students highlighted: *"In having that methodically prescribed … you break down the process into those clear steps, … it is an intuitive flow," "This was really good guidance for us … well categorized for the beginner," "It was my first time doing something at this scale, and I didn't know where to start. It was nice to have steps along the way,"* and *"We considered more options than we would have,"* which demonstrated the benefit of generating ideas. Despite the many improvements that can be made, we see the use of these design worksheets as a success for teaching the visualization design process to students in their cumulative projects.

This work explored utilizing the worksheets in a visualization course for the first time, and plenty of work lies ahead in their use as a teaching methodology for visualization design. For example, one avenue of future work is to continue improving and using the worksheets each year in the visualization course. Additionally, more methods for evaluating the usefulness of the materials could be investigated, such as performing a grading analysis between students' worksheet grades and other grades throughout the course [16]. The design worksheets could also be integrated more tightly with course content through the use of design workshops or other assignments. By implementing some of the requested changes to the worksheets, their pedagogical value could increase by enabling students to more effectively find and assess datasets, work more effectively in a team, or better outline and develop code. Another core aspect to design is its iterative nature, and more study could be conducted on iteration and its use, benefits, and limitations in visualization design projects. Lastly, performance measurements of these worksheets over time could help assess how valid and useful the design activity worksheets are for more types of students conducting visualization design.

Additional future work involves increasing the accessibility of the design worksheets as pedagogical material. The existing materials have been deployed online, but the resources, worksheets, and timelines are presented only as static materials or examples. By building these materials into an interactive system, team members could more effectively plan and coordinate their work on a visualization project. For example, the checklist or worksheets could provide questions or hints on demand when someone fills in their answers into a system, and this system may even dynamically interact and suggest new generative or evaluative methods based on students' input. On the other hand, this system has the potential to provide automatic generation of project timelines using the design activity framework, to support actionability of using this framework to judge where designers are and where they should be. This meta-view of a project could help guide students by asking additional questions on their status and success, to suggest activities to pursue next. These accessible, interactive visualization design process systems could help train visualization designers and possibly scaffold this process with new design methods as students learn and grow.

One limitation of the evaluation for the design worksheets is that the evaluator served also as the project mentor. Due to limitations of time and availability throughout the project and to perform the evaluation, this overlapping of roles was necessary, but we acknowledge that more robust evaluation would involve additional mentors or evaluators. Another future evaluation method is to interview visual designers on product teams in industry to assess with heuristics, or gather feedback on, the utility, usefulness, and impact of the design worksheets. One challenge is separating out the role of the mentor; without a mentor, would student groups have been as successful or confident going through their visualization design process? We determined that this interference was critical to maximize the potential knowledge we could obtain from utilizing the design worksheets. By observing and clarifying aspects on the worksheets, we learned a great deal about both the benefits and limitations of these design worksheets for a visualization project, which spurred numerous insights about improvements to be made on future iterations of the worksheets. Without these connections to each student project, the qualitative evaluation at the end of the course would have been significantly limited, and any confusing aspects would have further hindered the teaching impact of the worksheets. Educators recognize the importance of providing students with solutions along the way for the steps of a project to avoid pitfalls at early stages that cause later failures [77], i.e., do not punish students for failing to understand an earlier assignment in a later one. Nevertheless, we acknowledge that further evaluation may necessitate a separation of these roles to reduce potential bias and impact.

Incorporating visualization design worksheets and exercises in the classroom presents many challenges. For example, some students may struggle or feel intimidated by being asked to sketch, especially given a time restriction. Unfortunately, many classrooms have designated schedules and limited time slots, so the workshops or exercises must be completed in the time allotted, but many visualization exercises take longer than expected to execute [147]. Another element to consider is the role of tangibility in sketching visualization designs [76], [157]. Incorporation of these tangible visualization principles could be used within the design activity worksheets. As classroom sizes increase, it is important to reconsider how design critiques, feedback, and mentorship are conducted [74], [193], [194]. For example, with online learners in the hundreds or thousands, peer review, critique, and feedback can help mitigate some of these challenges [74], while also pushing students to practice the learned concepts in a structured fashion. As data visualization design changes, so too must the exercises, workshops, and materials we utilize to teach these concepts.

These visualization design worksheets are one step toward building more effective teaching tools for data visualization and design, but much work lies ahead. One clear area for future work involves materials for design inspiration: brainstorm visualization encodings, abstractions, and tasks. Initial work shared by He and Adar in VizItCards [73] is a step in this direction, and we encourage the community to continue this line of work. The one student in our study who used VizItCards would have liked to see the cards generalized for other visualization challenges. Furthermore, the visualization design process, steps, and guidance can always be improved to be more descriptive, more clear, sufficiently succinct, and encompass other design methods and methodologies. Other common methods for teaching are design studios [73] and exercises [195], and it would be worthwhile to adapt design worksheets for these settings. Lastly, scaffolding these design materials from visualization novices to experts could provide support for the visualization design worksheets to grow and expand for more advanced, creative, and flexible use.

8.6 Software Engineering and Agile Development

A field related to the design of data visualization tools is software engineering or how people develop tools and systems. The field of software engineering is about how programmers design, implement, test, and document the creation of software [196]. These concepts overlap with visualization design when designers have to develop something into a system or tool, and as such the methods and recommendations may overlap across fields. The body of software engineering research is vast, with core components such as how to generate requirements (necessary properties for something to solve real-world problems); general design principles such as managing data and events to function or object-oriented code; and testing, management, maintenance, and associated economics for software engineering [190]. In other words, software engineering involves many aspects from how to write good, effective code to managing teams; business practices; and working with clients or customers. All these concepts and associated methods can often relate to the field of visualization design, and, in turn, software engineering can incorporate design methods [197]. However, the majority of this dissertation focuses on the design and theoretical visualization components rather than on those from software engineering bodies of knowledge.

Agile development started in the 1990s with a series of methods aimed at revolutionizing and adapting software engineering processes with lightweight alternatives that quickly adapt to user needs in a changing workplace [190]. In 2001, 17 software developers, including Jeff Sutherland and Alistair Cockburn, sat down at the Snowbird Mountain Resort in Utah to discuss publication of their thoughts and insights as the *Manifesto for Agile Software Development* [191]. Traditional software engineering process models were very much incremental before agile approaches, and agile approaches are in principle about promoting customer satisfaction, adapting to clients' changing requirements, working software releases with the best technology on the order of weeks, forming co-located and cross-functional teams, implementing test-driven development, and reflecting on how to make the product more effective [191]. Various realizations of agile approaches have emerged over the years, from Scrum to Rapid Application Development (RAD) and eXtreme Programming (XP) [190], which all utilize a variety of software engineering methods such as pair programming, daily stand-ups, scrum boards, and sprint planning exercises. A key cornerstone of these approaches is how success is measured in terms of how well the software works [191]. Straying from more complex methods that focus on incremental changes, agile methodologies are designed to be iterative in nature, with teams adapting requirements to the software over time [190].

These agile approaches cross aspects of design for an engineering process with creative processes, but their focus is largely on software development, and they are not tailored to visualization specifically. For example, the notion of a sprint is at the core of the agile Scrum approach, where a sprint manages and timeboxes activities and development work into a concrete time frame for tackling backlog items, daily updates, and progress reviews [190]. This set deadline provides a clear goal for developing a working prototype to test with clients or customers. An example is the Google Ventures sprint, a five-day process that breaks apart the software engineering and design process into five big steps: mapping out the problem space and finding a focus, sketching out ideas, comparing and evaluating the ideas to find the best, building the software or other prototype, and testing with the clients or customers [198]. The goal of this sprint is to determine if an idea or potential product has a measurable impact on the clients or customers. Such an impact can shave months off building a fully working tool or system. By preparing for sprints with tools such as whiteboards, timers, and paper along with electronic device limitations, open schedules, and an appropriate team of experts, a facilitator, and a decider, a sprint process can focus a team to be very productive in a short amount of time [198]. As a result, this agile sprint method combines software engineering and development with project management [190] and business ideas [198]. Designers of visualization software systems, tools, and techniques can incorporate aspects of these agile approaches, such as sprints. Although the focus of this dissertation is less on general principles of software engineering or agile development, many aspects and methods from these methodologies can be applied and can fit into the context of the design activity framework for visualization design.

9
Conclusion

The main contribution of this dissertation is a design activity framework for visualization design [1]. By building on existing visualization design process and decisions models, the design activity framework seeks to support both a creative and engineering visualization design process, one with both iterative and linear aspects at its core. By prescribing four design activities, *understand*, *ideate*, *make*, and *deploy*, the framework allows visualization designers to check their progress and current activity with clear definitions, motivations, and expected results or visualization artifacts, such as requirements, ideas, prototypes, and visualization systems. These activities map to levels of the nested model to help designers choose appropriate validation approaches, and the framework identifies a series of potentially useful and impactful methods for generating and evaluating visualization artifacts. Lastly, the framework's activities can be represented in a concise timeline format, to support design team communication and retrospection.

The role of the design activity framework is illustrated through two case studies for cybersecurity. The first case study explored a redesign project of a company's visualization tool [1]. The design activity framework was established based on our reflections of the various activities we performed as a design team, utilizing user-centered design methods such as qualitative coding of research papers on user studies [2]. The next case study examined the creation of a visualization dashboard called BubbleNet [4]. As a result of this design study, we created a dashboard tailored for visualizing geographic and temporal patterns in security-based alert datasets, which we evaluated through a series of methods including a usability study. Like the previous case study, user-centered design methods advanced the project through personas to winnow and target specific users and data sketches to compare and contrast possible visual encodings with a cybersecurity analyst [2]. The framework guided the visualization design activities of this second case study from the ground up, spanning multiple discourse channels.

In order to externally evaluate the framework and increase its actionability, we investigated teaching novices how to conduct a visualization design process. To this end, we created design worksheets for students to use in a cumulative project for a graduate visualization course [6]. To more effectively teach the visualization design process, we created teaching materials for the course: a lecture on visualization design, an introduction to the design activity framework, resources on sketching, and four design activity worksheets. We tailored each worksheet to the four activities for visualization design, including step-by-step instructions, guidance, tips, and hints to guide students through their first visualization project. We worked with 13 students throughout the course, helping them with their projects and teamwork, and we externally evaluated the worksheets using group observations, online surveys, and interviews. Overall, the design worksheets were a successful first attempt, and students highlighted a number of improvements for the future.

By reflecting on other types of research projects we have conducted, we explored how visualization design and the design activity framework relate to other kinds of research, both technique-driven [3] and evaluative work [5]. In particular, we identified new types of pitfalls for design studies and observed how the steps of the visualization design process mapped to both technique-driven prototypes and experimental design research. Over time, it is important to build upon these design models and continue to validate their usefulness and effectiveness for all kinds of visualization designers, from students in the classroom to expert practitioners in industry. By working with more visualization designers over time, new models can expand upon this work to capture new types of visualization research and design projects. As the field of data visualization matures, it is crucial we modify and adapt our design models in order to better prescribe, describe, encompass, analyze, and communicate the visualization design process.

A
Table of Design Methods

To aid in the search for new design methods, we present an extended table of design methods in Table A.1. This table is an extension of Table 3.1 from Chapter 3. For Table A.1, we mapped 100 design methods into the design activity framework [1]. Each design method is marked for which design activities it may be utilized, e.g., *understand* (***u***), *ideate* (***i***), *make* (***m***), or *deploy* (***d***). Within an activity, we coded these methods based on its main purpose, to be generative (***g***) or evaluative (***e***). We noted when generating this table that methods may play differing roles in one activity to the next. Additionally, we tagged the methods we have seen commonly reported within the visualization community (***v***). Lastly, each method includes a succinct definition and source for visualization designers to research and learn more about a desired design method.

Table A.1. 100 exemplar design methods.

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
1	A/B testing						●		●	●	"compare two versions of the same design to see which one performs statistically better against a predetermined goal" [25]
2	activity map	●	●								"structuring activities of stakeholders and showing how they relate to one another.... take a list of activities gathered during research and see how they are grouped based on their relationships" [24]
3	AEIOU framework	●	●								"organizational framework reminding the researcher to attend to, document, and code information under a guiding taxonomy of Activities, Environments, Interactions, Objects, and Users" [25]
4	affinity diagramming		●		●		●				"process used to externalize and meaningfully cluster observations and insights from research, keeping design teams grounded in data as they design" [25]
5	algorithmic performance	●	●				●		●	●	"quantitatively study the performance or quality of visualization algorithms.... common examples include measurements of rendering speed or memory performance" [30]
6	analogical reasoning	●		●						●	"cognitive strategy in which previous knowledge is accessed and transferred to fit the current requirements of a novel situation" [199]
7	appearance modeling			●		●		●			"refined model of a new idea that emphasizes visual styling" [92]
8	artifact analysis	●	●							●	"systematic examination of the material, aesthetic, and interactive qualities of objects contributes to an understanding of their physical, social, and cultural contexts" [25]
9	automated logging	●	●				●		●	●	"captures the users' patterns of activity. simple reports - such as on the frequency of each error message, menu-item selection, dialog-box appearance, help invocation, form-field usage, or web-page access.... can also capture performance data for alternative designs" [34]

... *continued on the next page*

Table A.1 – continued

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
10	behavioral prototype				●						"simulating situations of user activity to understand user behaviors and build early concepts.... through observation and conversation, user behaviors help the team further build on the concepts" [24]
11	beta releases							●		●	"before software is released, it is sometimes given ...to a larger set of representative users. these users report problems with the product .... often uncontrolled" [190]
12	bull's-eye diagramming	●	●		●						"ranking items in order of importance using a target diagram.... gather a set of data (e.g., issues, features, etc.).... plot the data on the target, and set priorities" [92]
13	buy a feature	●	●		●		●				"game in which people use artificial money to express trade-off decisions.... ask [participants] to purchase features within the budget.... encourage them to articulate their deliberations" [92]
14	card sorting	●	●		●					●	"participatory design technique that you can use to explore how participants group items into categories and relate concepts to one another" [25]
15	case study	●	●		●		●		●	●	"research strategy involving in-depth investigation of single events or instances in context, using multiple sources of research evidence" & "focuses on gaining detailed, intensive knowledge about a single instance or a set of related instances" [25]
16	coding	●	●		●		●			●	"break data apart and identify concepts to stand for the data [open coding], [but] also have to put it back together again by relating those concepts [axial coding]" [93]
17	cognitive map	●	●								"reveal how people think about a problem space, and visualize how they process and make sense of their experience.... most effective when used to structure complex problems and to inform decision making" [25]

... continued on the next page

Table A.1 – continued

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
18	cognitive task analysis	●	●						●	●	"study of cognition in real-world contexts and professional practice at work" [200]
19	cognitive walk-through	●					●		●	●	"usability inspection method that evaluates a system's relative ease-of-use in situations where preparatory instruction, coaching, or training of the system is unlikely to occur" [25]
20	collage	●									"allows participants to visually express their thoughts, feelings, desires, and other aspects of their life that are difficult to articulate using traditional means" [25]
21	competitive testing	●	●						●	●	"process of conducting research to evaluate the usability and learnability of your competitors' products…. focuses on end-user behavior as they attempt to accomplish tasks" [25]
22	concept map				●						"visual framework that allows designers to absorb new concepts into an existing understanding of a domain so that new meaning can be made" & "sense-making tool that connects a large number of ideas, objects, and events as they relate to a certain domain" [25]
23	concept sketching			●						●	"convert ideas into concrete forms that are easier to understand, discuss, evaluate, and communicate than abstract ideas that are described in words" & "about making abstract ideas concrete" [24]
24	concept sorting				●						"disciplined effort to go through a collection of concepts, rationally organize them, and categorize them into groups" [24]
25	consistency inspection	●					●		●	●	"verify consistency across a family of interfaces, checking for consistency of terminology, color, layout, input and output formats, and so on" [34]
26	constraint removal	●		●						●	"barriers [are] transformed into a positive resource through which to create new ideas" [47]

… continued on the next page

Table A.1 – continued

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
27	contextual inquiry	●	●							●	"go where the customer works, observe the customer as he or she works, and talk to the customer about the work" [80]
28	controlled experiment	●	●				●		●	●	"help us to answer questions and identify casual relationships" [82] & "widely used approach to evaluating interfaces and styles of interaction, and to understanding cognition in the context of interactions with systems.... question they most commonly answer can be framed as: does making a change to the value of variable X have a significant effect on the value of variable Y?" [81]
29	creative matrix			●							"format for sparking new ideas at the intersections of distinct categories.... ideate at intersections of the grid.... encourage the teams to fill every cell of the grid" [92]
30	creative toolkits	●	●	●	●	●	●				"collections of physical elements conveniently organized for participatory modeling, visualization, or creative play by users, to inform and inspire design and business teams" & "foster innovation through creativity" [25]
31	debugging							●		●	"activity to find and fix bugs (faults) in the source code (or design) .... purpose of debugging is to find out why a program doesn't work or produces a wrong result or output" [190]
32	diagramming					●					"can effectively clarify structural relationships, describe processes, show how value flows through the system, show how the system evolves over time, map interactions between components, or work with other similar aspects of the system" & "process of translating your ideas into diagrams helps reduce ambiguity" [24]

... continued on the next page

Table A.1 – continued

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
33	documentation							●		●	"online help, manuals, and tutorials …to provide training, reference, and reminders about specific features and syntax" [34] & "document relevant facts, significant risks and tradeoffs, and warnings of undesirable or dangerous consequences from use or misuse of software" & "for external stakeholders …provide information needed to determine if the software is likely to meet the …users' needs" [190]
34	ergonomics evaluation	●	●				●		●	●	"assessment of tools, equipment, devices, workstations, workplaces, or environments, to optimize the fit, safety, and comfort of use by people" & "five criteria: size, strength, reach, clearance, & posture" [25]
35	example exposure			●		●				●	"excite ideas by exposing the subject to a solution for the same problem" [94]
36	excursion	●		●						●	"participants remove themselves from a task, take a mental or physical journey to seek images or stimuli and then bring these back to make connections with the task" [47]
37	experience prototyping				●	●	●				"fosters active participation to encounter a live experience with products, systems, services, or spaces" [25]
38	field notes (diary, journal)	●	●								"four types of field notes: jottings, the diary, the log, and the notes" & "keep a note pad with you at all times and make field jottings on the spot" & "a diary chronicles how you feel and how you perceive your relations with others around you" & "a log is a running account of how you plan to spend your time, how you actually spend your time, and how much money you spent" & "three kinds of notes: notes on method and technique; ethnographic, or descriptive notes; and the notes that discuss issues or provide an analysis of social situations" [95]

… continued on the next page

Table A.1 – continued

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
39	five W's	●	●							●	"popular concept for information gathering in journalistic reporting …. captures all aspects of a story or incidence: who, when, what, where, and why" [201], [202]
40	focus group	●	●				●		●	●	"small group of well-chosen people…guided by a skilled moderator…[to] provide deep insight into themes, patterns, and trends" [25]
41	foresight scenario			●		●					"considering hypothetical futures based on emergent trends and then formulating alternative solutions designed to meet those possible situations" [24]
42	frame of reference shifting			●							"change how objectives and requirements are being viewed, perceived, and interpreted" [94]
43	graffiti walls	●	●	●			●		●		"open canvas on which participants can freely offer their written or visual comments about an environment or system, directly in the context of use" [25]
44	heuristic evaluation						●		●	●	"informal usability inspection method that asks evaluators to assess an interface against a set of agreed-upon best practices, or usability 'rules of thumb"' [25]
45	idea evaluation				●						"evaluating ideas with regard to four dimensions - novelty, workability, relevance, and specificity" & "novelty: nobody has expressed it before" & "workability: does not violate known constraints or …easily implemented" & "relevance: satisfies the goals set by the problem solver" & "specificity: worked out in detail" [203]
46	ideation game			●		●					"engaging stakeholders in game-like activities to generate concepts" & "game-building and game-playing mindsets allow participants to cut through barriers of creativity and think more openly" [24]

… continued on the next page

Table A.1 – continued

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
47	image quality analysis	●	●				●		●	●	"classical form of qualitative result inspection…the qualitative discussion of images produced by a (rendering) algorithm…. common to show and assess visually that quality goals had been met" [30]
48	importance/difficulty matrix		●		●		●				"a quad chart for plotting items by relative importance and difficulty …make a poster showing a large quad chart, label horizontal axis Importance, label vertical axis Difficulty …plot items horizontally by relative importance, plot items vertically by relative difficulty …look for related groupings, and set priorities" [92]
49	incubation			●							"add programmed delay to allow sub-conscious processing to take place" [94]
50	interactive tutorial							●		●	"uses the electronic medium to teach the novice user by showing simulations of the working system, by displaying attractive animations, and by engaging the user in interactive sessions"[34] & "[present] the work-product to the other participants …. [take] the role of explaining and showing the material to participants" [190]
51	interviewing	●	●		●		●		●	●	"fundamental research method for direct contact with participants, to collect firsthand personal accounts of experience, opinions, attitudes, and perceptions" & unstructured vs. guided vs. structured [25]
52	key performance indicators						●		●		"critical success factors for your product or service" & "quantifiable, widely accepted business goals" & "reflect the activities of real people" [25]
53	literature review	●	●							●	"distill information from published sources, capturing the essence of previous research or projects as they might inform the current project" & "collect and synthesize research on a given topic" [25]

… continued on the next page

Table A.1 – continued

#	method	u g	u e	i g	i e	m g	m e	d g	d e	v	definition
54	love/breakup letters	•	•		•		•		•		"personal letter written to a product…[to reveal] profound insights about what people value and expect from the objects in their everyday lives" [25]
55	measuring users (eye tracking)	•	•				•		•	•	"understanding what people do, how they do it, and how they react…. participants in research studies can be important data sources…. eye-tracking tools that tell us where people are looking on a screen…. skin response or cardiovascular monitors can provide insight into a user's level of arousal or frustration" [82]
56	mindmapping			•							"visual thinking tool that can help generate ideas and develop concepts when the relationships among many pieces of related information are unclear" & also: graphic organizer, brainstorming web, tree diagram, flow diagram [25]
57	morphological synthesis			•							"organizing concepts under user-centered categories and combining concepts to form solutions…a solution is a set of concepts that work together as a complete system" [24]
58	observation	•	•		•		•		•	•	"attentive looking and systematic recording of phenomena: including people, artifacts, environments, events, behaviors and interactions" [25] & e.g. participant vs. fly-on-the-wall, axis from obtrusive to unobtrusive like in the field of ethnography [82]
59	online forum							•	•		"permit posting of open messages and questions" & also known as: mailing lists, bulletin boards, etc. [34]
60	online suggestions								•		"allow users to send messages to the maintainers or designers…. encourages some users to make productive comments" [34]
61	paper prototyping			•		•				•	"create a paper-based simulation of an interface to test interaction with a user" [96]

… continued on the next page

Table A.1 – continued

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
62	parallel prototyping			●		●				●	"creating multiple alternatives in parallel may encourage people to more effectively discover unseen constraints and opportunities, enumerate more diverse solutions, and obtain more authentic and diverse feedback from potential users" & "[this method] yields better results, more divergent ideas, and [designers] react more positively to critique" [97]
63	personas		●								"consolidate archetypal descriptions of user behavior patterns into representative profiles, to humanize design focus, test scenarios, and aid design communication" [25]
64	photo studies	●									"invite the participant to photo-document aspects of his or her life and interactions, providing the designer with visual, self-reported insights into user behaviors and priorities" [25]
65	pilot testing								●	●	"placing offerings in the marketplace to learn how they perform and how users experience them…. method for testing innovation solutions by placing them in contexts where they function as real offerings" [24]
66	POEMS framework	●	●								"observational research framework used to make sense of the elements present in a context…. five elements are: People, Objects, Environments, Messages, and Services" [24]
67	prototyping			●		●				●	"tangible creation of artifacts at various levels of resolution, for development and testing of ideas within design teams and with clients and users" [25]
68	provocative stimuli			●		●				●	"trigger new ideas by exposing the subject to related and unrelated pointers, pictures, sounds" [94]

… continued on the next page

Table A.1 – continued

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
69	questionnaire	●	●		●		●		●	●	"survey instruments designed for collecting self-report information from people about their characteristics, thoughts, feelings, perceptions, behaviors, or attitudes, typically in written form" [25]
70	reflection		●		●					●	"[ask participants] what they knew…that they hadn't known at the outset" [47]
71	roadmap					●		●		●	"plan for implementing solutions…. helps explore how solutions are to be built up, with short-term initiatives as a foundation on which long-term solutions are based" & "prioritizing the order of implementation" [24]
72	role-playing	●	●	●	●	●	●	●	●		"acting the role of the user in realistic scenarios can forge a deep sense of empathy and highlight challenges, presenting opportunities that can be met by design" [25]
73	rose-thorn-bud		●		●		●				"technique for identifying things as positive, negative, or having potential" & tag outcomes as rose, thorn, or bud, accordingly [92]
74	round robin			●	●						"activity in which ideas evolve as they are passed from person to person" [92]
75	sample data							●		●	"create benchmark datasets…provide real data and tasks …. illustrating [tools] with convincing examples using real data" [27]
76	semantic differential	●	●		●		●		●		"linguistic tool designed to measure people's attitudes toward a topic, event, object, or activity, so that its deeper connotative meaning can be ascertained" [25]
77	simulation	●					●				"deep approximations of human or environmental conditions, designed to forge an immersive, empathic sense of real-life user experiences" [25]

… continued on the next page

Table A.1 – continued

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
78	social mapping	●	●								"a visual representation of relationships between objects and spaces …. maps reflect people's beliefs about the spaces and objects around them: how they define those spaces, how they categorize them, and what they feel about them" [204]
79	spatial mapping	●	●								"a visual representation of relationships between people …. maps reflect people's beliefs about the spaces and objects around them: how they define those spaces, how they categorize them, and what they feel about them" [204]
80	speed dating				●		●				"compare multiple design concepts in quick succession" & "exposing people to future design ideas via storyboards and simulated environments before any expensive technical prototypes are built" [25]
81	stakeholder feedback		●		●		●		●	●	"demoing the visualization to a group of people, often and preferably domain experts, letting them "play" with the system and / or observe typical system features as shown by the representatives" [28]
82	stakeholder map	●	●								"visually consolidate and communicate the key constituents of a design project" [25]
83	statistical analysis	●	●		●		●		●	●	"many critical decisions need to be made when analyzing data, such as the type of statistical method to be used, the confidence threshold, as well as the interpretation of the significance test results" [82]
84	storyboarding			●		●				●	"visually capture the important social, environmental, and technical factors that shape the context of how, where, and why people engage with products" & "build empathy for end users" [25]
85	suspended judgment			●							"postpone premature decisions or dismissing an idea" & "generate as many ideas as possible" [94]

… continued on the next page

Table A.1 – continued

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
86	task analysis	●	●				●		●	●	"breaks down the constituent elements of a user's work flow, including actions and interactions, system response, and environmental context" & can be conducted on a tool or a human [25]
87	technology probe	●	●	●	●	●	●			●	"simple, flexible, and adaptable technologies with three interdisciplinary goals: the social science goal of understanding the needs and desires of users in a real-world setting, the engineering goal of field-testing the technology, and the design goal of inspiring users and researchers to think about new technologies" [98]
88	think-aloud protocol	●					●		●	●	"asks people to articulate what they are thinking, doing, or feeling as they complete a set of tasks that align with their realistic day-to-day goals" [25]
89	thought experiment		●		●		●				"think about research questions as if it were possible to test them in true experiments…. what would the experiment look like?" [95]
90	usability report	●	●				●		●	●	"focuses on people and their tasks, and seeks empirical evidence about how to improve the usability of an interface" [25]
91	usability testing	●	●				●		●	●	"carried out by observing how participants perform a set of predefined tasks…. take notes of interesting observed behaviors, remarks voiced by the participant, and major problems in interaction" [28]
92	user journey map	●	●								"breaks down a users' journey into component parts to gain insights into problems that may be present or opportunities for innovations…. activities are shown as nodes" [24]
93	video ethnography	●									"capture peoples' activities and what happens in a situation as video that can be analyzed for recognizing behavioral patterns and insights" & "similar to photo ethnography" [24]

… continued on the next page

Table A.1 – continued

#	method	*u*		*i*		*m*		*d*		*v*	definition
		g	*e*	*g*	*e*	*g*	*e*	*g*	*e*		
94	video scenario				●		●				"short movie showing the attributes of a new concept in use…. identify a new concept to represent…. record video or take still photos of each scene" [92]
95	visual metrics	●	●				●		●	●	"automatic procedures which compare one solution to another…. based on the definition of one or more image quality measures that capture the effectiveness of the visual output according to a desired property of the visualization" [28]
96	voting		●		●		●		●	●	"a quick poll of collaborators to reveal preferences and opinions" [92]
97	weighted matrix		●		●		●				"matrix ranks potential design opportunities against key success criteria" & "help identify and prioritize the most promising opportunities" [25]
98	wireframing			●		●				●	"schematic diagramming: an outline of the structure and essential components of a system" [92]
99	wishful thinking	●		●						●	"[participants are] asked to think about aspirations for [their domain]…. what would you like to know? what would you like to be able to do? what would you like to see?" [47]
100	Wizard of Oz						●			●	"participants are led to believe they are interacting with a working prototype of a system, but in reality, a researcher is acting as a proxy for the system from behind the scenes" [25]

B
Design Worksheet Materials

We provide additional worksheets and materials for the design activity framework. These resources were utilized in conjunction with the incorporation of these materials in a visualization graduate course, as discussed in Chapter 6. First, we include two introductory worksheets used to teach the framework. In Fig. B.1, we introduce the high-level components of the design activity framework with some examples. Next we provide sketching tips and resources for the design worksheets, as seen in Fig. B.2. The four design activity worksheets are shown in Chapter 6. Lastly, we provide a front worksheet template in Fig. B.3 to guide students through the different components of the design activity framework worksheets.

Design Activity Framework

What **artifacts** can we create?

- design requirements
- ideas & sketches
- prototypes
- visualization systems

How do we get **artifacts**?

Writing on worksheets, sketching, or building with code. Artifacts can be generated or evaluated using **methods**.

What do we do with **artifacts**?

Build ideas to address real needs. Combine them. Find novel ways to solve problems. Record to track a project's evolution. Revisit for inspiration. Evaluate them.

What is a **design activity**?

Actions taken in order to achieve a set of **artifacts**.
4 activities: *Understand, Ideate, Make, & Deploy.*

The **design worksheets** provide guided *methods* for obtaining *artifacts*. Artifacts should flow from activity to activity, so do refer back to them later on as each artifact is used. You can print out program screenshots if that will help you design.

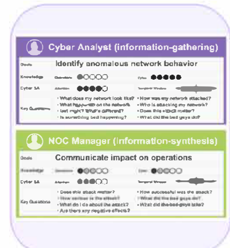
Feel free to work on worksheets individually but come back and fill out one as a group. Label each with a unique number at the top. This number is important for using additional sheets for space. Expected results for each box are shown as icons at the bottom.

Always double-check the first box on the sheet! For example, in the *Understand* activity, have you captured the right challenge, with enough detail? Watch out for ***! !*** warnings which provide cautionary tips on when to revisit earlier worksheets. You can continue to any activity listed at the bottom of the worksheet.

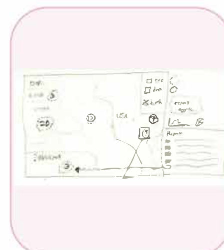
These worksheets provide sample methods to guide your design process, but feel free to explore alternative methods for generating and evaluating artifacts: http://bit.ly/2edEswv

example artifacts

understand



ideate



make

deploy

Fig. B.1. Introductory worksheet for the design activity framework worksheets.

Sketching User Interfaces

How do I **sketch**?

Get the creative juices flowing and start by sketching known solutions or ideas. Then think of new and different ways to solve the challenge. Sketch with crayons, sharpies, or colored pencils. While artistic drawing is a practiced skill, anyone can illustrate an idea on paper!

Where do I sketch? What if I need **more space**?

Sketch on the provided worksheets or on separate paper. For separate sheets, try large-format, dot, or graph paper.

When more space is needed (especially during *Ideate*), tag additional papers with the activity symbol, worksheet number, and box number: e.g. **[I-3-2]** for the *Ideate* activity, your third worksheet, and box 2) on that sheet.

What other sketching **resources** would you recommend?

Tips & Tricks

- http://bit.ly/2dapM3Z
- http://bit.ly/2e0sRRM
- http://bit.ly/1SMkacn
- http://bit.ly/2dQyI0b
- http://bit.ly/2dnH8cj

Sketching Paper

- http://paperkit.net/dottedpaper
- http://sneakpeekit.com/

Wireframes, Mock-ups, and Interactive Screens

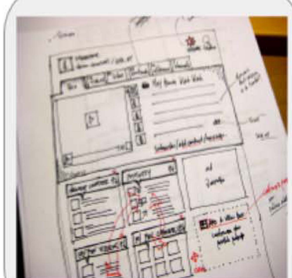
- https://balsamiq.com/
- https://moqups.com/
- https://www.invisionapp.com/

Additional Resources

- http://bit.ly/1BL4m3s

example sketches

original concept sketch for Twitter, as a simplified, live status update
http://bit.ly/1EvqTgX



an idea for a customizeable Vimeo profile page
http://bit.ly/2edxLL6

Fig. B.2. Introductory worksheet on sketching and related resources.

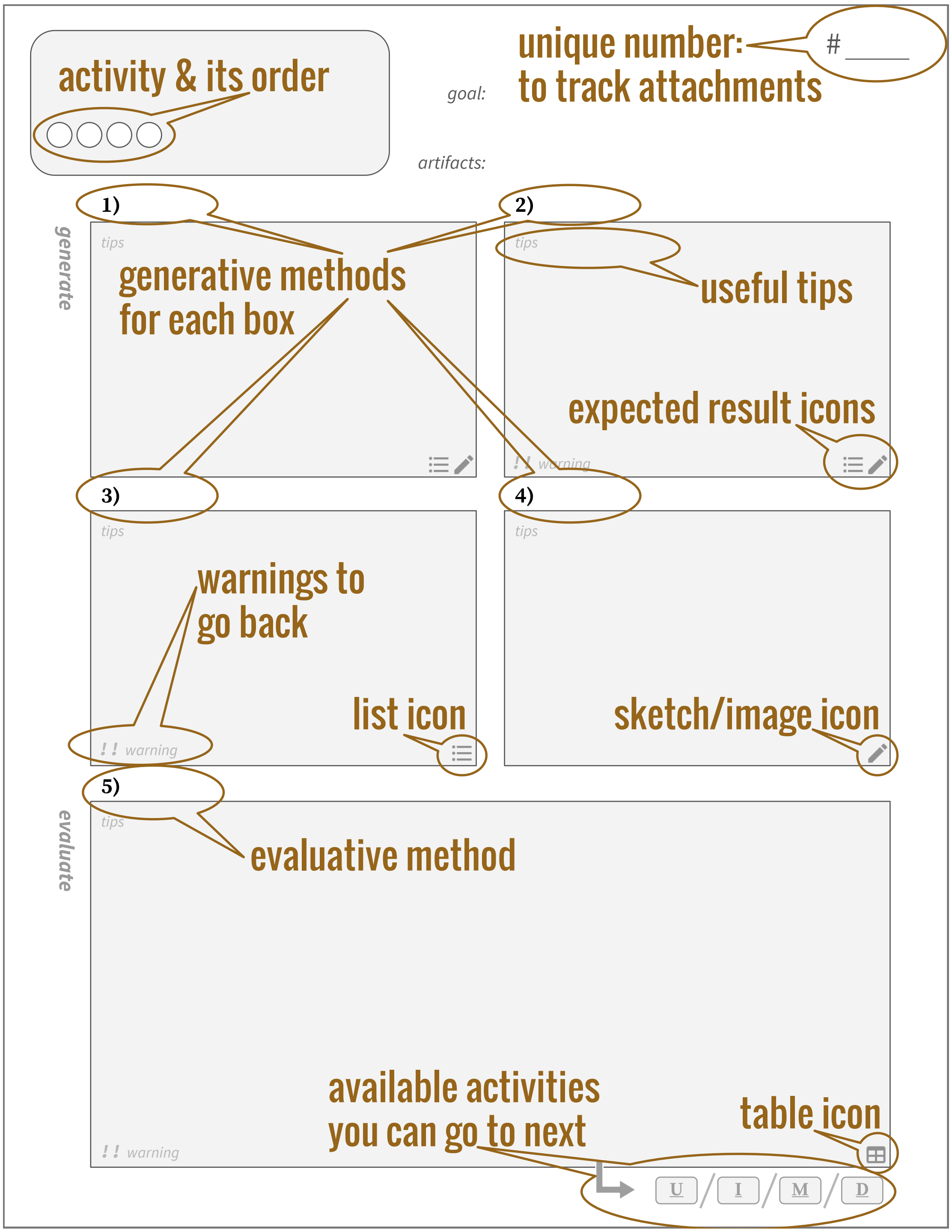


Fig. B.3. Template for the different components of the design activity framework worksheets.

C

Worksheet Evaluation Materials

In Chapter 6, we discussed the methodology behind our process for evaluating the design activity framework worksheets. We utilized these worksheets for students undergoing a cumulative project for the course, and we surveyed students at the end of the course on their experiences and opinion of the materials. We provided the prompts and questions in the two online questionnaires, one of which was sent to the entire class and another to only 13 volunteers who utilized the worksheets as part of their project. For the survey questions, responses were in one of the following formats: five-point Likert scale (denoted with L), numbered scale (1–5, denoted N), free-form text response (denoted T), or a multiple select choice of design activities (denoted UIMD). Lastly, we conducted semistructured interviews with 11 of the student volunteers; the interview guide with base questions is included. These materials serve the purpose of adding reproducibility to the evaluation methodology and, further, enabling external validation of this work.

C.1 Course Web Survey

Please complete the following questions honestly based on your experience in this course.

1. How comfortable were you with visualization design before this class? (N)
2. How comfortable are you with visualization design after this class? (N)
3. The lectures in this course helped me learn how to design visualizations. (L)
4. The exercises in class helped me learn how to design visualizations. (L)
5. The provided design worksheets helped me learn how to design visualizations. (L)
6. The final project helped me learn how to design visualizations. (L)

C.2 Project Web Survey

During your final project, we utilized several design activity worksheets: *understand*, *ideate*, *make*, & *deploy*. Please provide honest feedback on your use and opinion of these teaching materials.

1. Which worksheets were the most helpful? Did a particular worksheet help you generate and evaluate artifacts? Did any worksheet build up your confidence in the design process? (UIMD)
2. Why is that? (T)
3. Which worksheets were the least helpful? Did a particular worksheet hinder the generation and evaluation of artifacts? Did any worksheet fail to build up your confidence in the design process? (UIMD)
4. Why is that? (T)
5. My group was able to generate and evaluate artifacts with the help of the design worksheets. (L)
6. Filling out the design worksheets took valuable time away from my final project. (L)
7. It was possible to fully complete each worksheet using the materials given to me during the course. (L)
8. The design worksheets were used to record artifacts or decisions I made, after they were already made. (L)
9. I was able to better structure my design process with the help of the design worksheets. (L)
10. There was not enough structure or guidance provided on the worksheets in order to complete them. (L)
11. I was able to figure out where to go next in the design process with the help of the design worksheets. (L)
12. There was too much information or text included on the blank design worksheets. (L)
13. I was able to capture design decisions easily and succinctly using the design worksheets. (L)
14. The system developed for my final project would have been just as effective if we had not used the design worksheets. (L)
15. What are some suggested changes or improvements for the worksheets? (T)
16. Please include any additional comments on the design worksheets. (T)

C.3 Interview Guide

Thanks for agreeing to meet and discuss your experiences this semester. We are looking for sincere, honest feedback of the design worksheets we utilized. This includes elements on the worksheets, when they were introduced & taught, when we utilized them, how we utilized them, and anything else you think would have helped you learn more and practice being a better visualization designer.

Are you OK with recording this interview? It is for me to review later without scribbling notes.

1. The worksheets were created for teaching the design process. Can you describe the steps of the design process, briefly in your own words?
2. What were the most useful aspects of the worksheets for your project? Least useful?
3. How did you fill out the worksheets? Was it done as a group or individually? When was it most comfortable for you to fill these out?
4. Was it helpful to have an evaluation step? Think of sketches, ideas. Why or why not?
5. Were the design worksheets helpful for documenting your design process? Are the worksheets a useful tool for helping you write a process book? e.g. think of what you did during the *ideate* and *make* parts of your design process.
6. Did the worksheets help guide you through the design process? For example, where to go next, revisiting sheets, etc. Why or why not?
7. Are you encouraged to try out new design methods after using the sheets? Did you?
8. Are there any additional worksheets that could have been helpful for your project? Did you feel any steps were missing from the worksheets?
9. Did the worksheets limit your choices or fail to capture the complexity of how you design a visualization? Please explain.
10. Would you say that the design worksheets helped build your confidence in design? In what ways could it have helped you learn more & built up your confidence more effectively?
11. Do you have any additional comments, questions, or feedback?

Thank you again for participating in this interview. This work is part of my PhD dissertation, to create teaching materials for visualization design. I greatly appreciate advising your final group and seeing how these worksheets helped shape your final project. It was a wonderful experience getting to see your group interact together and create _____. I look forward to seeing your work in the future, and feel free to reach out to me if you have any lingering questions about visualization design, these worksheets, or anything else.

We hope to report on our findings using these worksheets in a classroom setting as a short paper at a visualization conference. Are you comfortable with quotes being taken anonymously from this interview to be included in that publication? It is totally fine to say no.

Before we conclude, do you have any other questions for me? Thanks again for your time!

References

[1] S. McKenna, D. Mazur, J. Agutter, and M. Meyer, "Design activity framework for visualization design," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2191–2200, 2014.

[2] S. McKenna, D. Staheli, and M. Meyer, "Unlocking user-centered design methods for building cyber security visualizations," in *Proc. Int. Symp. on Vis. for Cyber Security (VizSec)*, 2015, pp. 1–8.

[3] S. McKenna, M. Meyer, C. Gregg, and S. Gerber, "s-CorrPlot: An interactive scatterplot for exploring correlation," *J. Computational Graphical Statist.*, vol. 25, no. 2, pp. 445–463, 2016.

[4] S. McKenna, D. Staheli, C. Fulcher, and M. Meyer, "BubbleNet: A cyber security dashboard for visualizing patterns," *Comput. Graph. Forum (EuroVis)*, vol. 35, no. 3, pp. 281–290, 2016.

[5] S. McKenna, N. Henry Riche, B. Lee, J. Boyd, and M. Meyer, "Visual narrative flow: Exploring factors shaping data visualization story reading experiences," *Comput. Graph. Forum (EuroVis)*, vol. (to appear), 2017.

[6] S. McKenna, A. Lex, and M. Meyer, "Worksheets for guiding novices through the visualization design process," (to be submitted to *Pedagogy Data Vis., IEEE VIS Workshop*), 2017.

[7] M. Beaudouin-Lafon, "Designing interaction, not interfaces," in *Proc. Int. Work. Conf. Advanced Visual Interfaces (AVI)*, ACM, 2004, pp. 15–22.

[8] M. Sedlmair, M. Meyer, and T. Munzner, "Design study methodology: Reflections from the trenches and the stacks," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 12, pp. 2431–2440, 2012.

[9] D. Lloyd and J. Dykes, "Human-centered approaches in geovisualization design: Investigating multiple methods through long-term case study," *IEEE Trans. Vis. Comput. Graphics*, vol. 17, no. 12, pp. 2498–2507, 2011.

[10] I. Wassink, O. Kulyk, and B. van Dijk, "Applying a user-centered approach to interactive visualisation design," in *Trends in Interactive Visualization*, R. Liere, T. Adriaansen, and E. Zudilova-Seinstra, Eds., London, UK: Springer, 2009, pp. 175–199.

[11] L. C. Koh, A. Slingsby, J. Dykes, and T. S. Kam, "Developing and applying a user-centered model for the design and implementation of information visualization tools," in *Proc. Int. Conf. Inform. Vis.*, IEEE, 2011, pp. 90–95.

[12] DIS and ISO, "Ergonomics of human system interaction–Part 210: Human-centred design for interactive systems," *Int. Org. Standardization*, 2010.

[13] M. Tory and T. Möller, "Human factors in visualization research," *IEEE Trans. Vis. Comput. Graphics*, vol. 10, no. 1, pp. 72–84, 2004.

[14] A. Vande Moere and H. Purchase, "On the role of design in information visualization," *Inform. Vis.*, vol. 10, no. 4, pp. 356–371, 2011.

[15] A. Bigelow, S. Drucker, D. Fisher, and M. Meyer, "Reflections on how designers design with data," in *Proc. Int. Work. Conf. Advanced Visual Interfaces (AVI)*, ACM, 2014, pp. 17–24.

[16] J. C. Roberts, C. Headleand, and P. D. Ritsos, "Sketching designs using the five design-sheet methodology," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 419–428, 2016.

[17] T. Munzner, "A nested model for visualization design and validation," *IEEE Trans. Vis. Comput. Graphics*, vol. 15, no. 6, pp. 921–928, 2009.

[18] M. Meyer, M. Sedlmair, P. S. Quinan, and T. Munzner, "The nested blocks and guidelines model," *Inform. Vis.*, vol. 14, no. 3, pp. 234–249, 2015.

[19] M. Brehmer, S. Ingram, J. Stray, and T. Munzner, "Overview: The design, adoption, and analysis of a visual document mining tool for investigative journalists," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2271–2280, 2014.

[20] N. McCurdy, J. Lein, K. Coles, and M. Meyer, "Poemage: Visualizing the sonic topology of a poem," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 439–448, 2016.

[21] E. Kerzner, L. A. Butler, C. Hansen, and M. Meyer, "A shot at visual vulnerability analysis," *Comput. Graph. Forum (EuroVis)*, vol. 34, no. 3, pp. 391–400, 2015.

[22] P. Quinan and M. Meyer, "Visually comparing weather features in forecasts," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 389–398, 2016.

[23] W. S. Cleveland and R. McGill, "Graphical perception: Theory, experimentation, and application to the development of graphical methods," *J. Amer. Statistical Assoc.*, vol. 79, no. 387, pp. 531–554, 1984.

[24] V. Kumar, *101 Design Methods: A Structured Approach for Driving Innovation in Your Organization*. Hoboken, NJ: John Wiley & Sons, 2012.

[25] B. Martin and B. M. Hanington, *Universal Methods of Design: 100 Ways to Research Complex Problems, Develop Innovative Ideas, and Design Effective Solutions*. Beverly, MA: Rockport, 2012.

[26] G. Kindlmann, "Algebraic visualization design for pedagogy," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[27] C. Plaisant, "The challenge of information visualization evaluation," in *Proc. Int. Work. Conf. Advanced Visual Interfaces (AVI)*, ACM, 2004, pp. 109–116.

[28] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale, "Empirical studies in information visualization: Seven scenarios," *IEEE Trans. Vis. Comput. Graphics*, vol. 18, no. 9, pp. 1520–1536, 2012.

[29] S. Carpendale, "Evaluating information visualizations," in *Information Visualization*, A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, Eds., Berlin, DEU: Springer, 2008, pp. 19–45.

[30] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair, and T. Moller, "A systematic review on the practice of evaluating visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2818–2827, 2013.

[31] 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 Inform. Vis. (BELIV)*, ACM, 2006, pp. 1–7.

[32] A. Cairo, *The Functional Art: An Introduction to Information Graphics and Visualization*. Thousand Oaks, CA: New Riders, 2012.

[33] T. Munzner, *Visualization Analysis and Design*. Boca Raton, FL: CRC Press, 2014.

[34] B. Shneiderman and C. Plaisant, *Designing the User Interface: Strategies for Effective Human-Computer Interaction*. Boston, MA: Addison-Wesley, 2009.

[35] C. Ware, *Visual Thinking: For Design*. San Francisco, CA: Morgan Kaufmann, 2008.

[36] J. Löwgren, "Applying design methodology to software development," in *Proc. Conf. on Designing Interactive Syst.: Processes, Practices, Methods, and Techn.*, ACM, 1995, pp. 87–95.

[37] T. Howard, S. Culley, and E. Dekoninck, "Describing the creative design process by the integration of engineering design and cognitive psychology literature," *Des. Stud.*, vol. 29, no. 2, pp. 160–180, 2008.

[38] T. V. Wolf, J. A. Rode, J. Sussman, and W. A. Kellogg, "Dispelling "design" as the black art of CHI," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2006, pp. 521–530.

[39] D. Fallman, "Design-oriented human-computer interaction," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2003, pp. 225–232.

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

[41] H. Christiaans and R. A. Almendra, "Accessing decision-making in software design," *Des. Stud.*, vol. 31, no. 6, pp. 641–662, 2010.

[42] A. Tang, A. Aleti, J. Burge, and H. van Vliet, "What makes software design effective?" *Des. Stud.*, vol. 31, no. 6, pp. 614–640, 2010.

[43] J.-C. Wu, C.-C. Chen, and H.-C. Chen, "Comparison of designer's design thinking modes in digital and traditional sketches," *Des. and Technol. Edu.*, vol. 17, no. 3, pp. 37–48, 2012.

[44] H. G. Nelson and E. Stolterman, *The Design Way: Intentional Change in an Unpredictable World*. Cambridge, MA: MIT Press, 2012.

[45] S. Pugh, *Total Design: Integrated Methods for Successful Product Engineering*. Boston, MA: Addison-Wesley, 1991.

[46] B. Buxton, *Sketching User Experiences: Getting the Design Right and the Right Design*. San Francisco, CA: Morgan Kaufmann, 2010.

[47] S. Goodwin, J. Dykes, S. Jones, I. Dillingham, G. Dove, A. Duffy, A. Kachkaev, A. Slingsby, and J. Wood, "Creative user-centered visualization design for energy analysts and modelers," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2516–2525, 2013.

[48] N. McCurdy, J. Dykes, and M. Meyer, "Action design research and visualization design," in *Proc. Workshop on Beyond Time and Errors on Novel Evaluation Methods for Vis. (BELIV)*, ACM, 2016, pp. 10–18.

[49] A. Baker and A. van der Hoek, "Ideas, subjects, and cycles as lenses for understanding the software design process," *Des. Stud.*, vol. 31, no. 6, pp. 590–613, 2010.

[50] J. Cao, Y. Riche, S. Wiedenbeck, M. Burnett, and V. Grigoreanu, "End-user mashup programming: Through the design lens," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2010, pp. 1009–1018.

[51] B. Onarheim and S. Wiltschnig, "Opening and constraining: Constraints and their role in creative processes," in *Proc. Conf. on Creativity and Innovation in Des.*, Desire Network, 2010, pp. 83–89.

[52] J. C. Savage, C. J. Moore, J. C. Miles, and C. Miles, "The interaction of time and cost constraints on the design process," *Des. Stud.*, vol. 19, no. 2, pp. 217–233, 1998.

[53] K. Vredenburg, J.-Y. Mao, P. W. Smith, and T. Carey, "A survey of user-centered design practice," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2002, pp. 471–478.

[54] J. S. Olson and T. P. Moran, "Mapping the method muddle: Guidance in using methods for user interface design," in *Proc. Workshop on Human-Computer Interface Design: Success Stories, Emerging Methods, and Real-World Context*, M. Rudisill, C. Lewis, P. G. Polson, and T. D. McKay, Eds., San Francisco, CA: Morgan Kaufmann, 1995, pp. 269–300.

[55] M. Cardella, C. Atman, J. Turns, and R. Adams, "Students with differing design processes as freshmen: Case studies on change," *Int. J. Eng. Edu.*, vol. 24, no. 2, pp. 246–259, 2008.

[56] S. Jones, P. Lynch, N. Maiden, and S. Lindstaedt, "Use and influence of creative ideas and requirements for a work-integrated learning system," in *Int. Requirements Eng. Conf.*, IEEE, 2008, pp. 289–294.

[57] T. Brown, *Change by Design: How Design Thinking Transforms Organizations and Inspires Innovation*. New York, NY: Harper Business, 2009.

[58] W. Lidwell and K. Holden, *Universal Principles of Design*. Beverly, MA: Rockport, 2010.

[59] A. Wodehouse and W. Ion, "The integration of information and ideas: Creating linkages through a novel concept design method," *Parsons J. Inform. Mapping*, vol. 2, no. 2, pp. 1–7, 2010.

[60] M. da Gandra, M. van Neck, et al., *InformForm, Information Design: In Theory, an Informed Practice*. London, UK: Mwm Creative, 2012.

[61] R. Teal, "Developing a (non-linear) practice of design thinking," *Int. J. Art and Des. Edu.*, vol. 29, no. 3, pp. 294–302, 2010.

[62] C. Atman, K. Deibel, and J. Borgford-Parnell, "The process of engineering design: A comparison of three representations," in *Int. Conf. Eng. Des.*, 2009.

[63] C. M. Snider, S. J. Culley, and E. A. Dekoninck, "Analysing creative behaviour in the later stage design process," *Des. Stud.*, vol. 34, no. 5, pp. 543–574, 2013.

[64] J. R. Goodall, A. A. Ozok, W. G. Lutters, P. Rheingans, and A. Komlodi, "A user-centered approach to visualizing network traffic for intrusion detection," in *Extended Abstracts on Human Factors in Computing Syst. (CHI-EA)*, ACM, 2005, pp. 1403–1406.

[65] R. F. Erbacher, "Visualization design for immediate high-level situational assessment," in *Proc. Int. Symp. on Vis. for Cyber Security (VizSec)*, ACM, 2012, pp. 17–24.

[66] J. Stoll, D. McColgin, M. Gregory, V. Crow, and W. K. Edwards, "Adapting personas for use in security visualization design," in *Proc. Int. Workshop on Vis. for Cyber Security (VizSec)*, Springer, 2007, pp. 39–52.

[67] C. L. Paul, R. Rohrer, and B. Nebesh, "A "design first" approach to visualization innovation," *Comput. Graph. Appl.*, vol. 35, no. 1, pp. 12–18, 2015.

[68] J. Landstorfer, I. Herrmann, J. E. Stange, M. Dörk, and R. Wettach, "Weaving a carpet from log entries: A network security visualization built with co-creation," in *Proc. Conf. on Visual Analytics Sci. and Technol. (VAST)*, IEEE, 2014, pp. 73–82.

[69] A. Komlodi, P. Rheingans, U. Ayachit, J. R. Goodall, and A. Joshi, "A user-centered look at glyph-based security visualization," in *Proc. Int. Workshop on Vis. for Cyber Security (VizSec)*, IEEE, 2005, pp. 21–28.

[70] L. Hao, C. G. Healey, and S. E. Hutchinson, "Flexible web visualization for alert-based network security analytics," in *Proc. Int. Symp. on Vis. for Cyber Security (VizSec)*, ACM, 2013, pp. 33–40.

[71] H. Rushmeier, J. Dykes, J. Dill, and P. Yoon, "Revisiting the need for formal education in visualization," *Comput. Graph. Appl.*, vol. 27, no. 6, pp. 12–16, 2007.

[72] M. A. Hearst, "Active learning assignments for student acquisition of design principles," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[73] S. He and E. Adar, "VizItCards: A card-based toolkit for infovis design education," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 561–570, 2017.

[74] J. Beyer, H. Strobelt, M. Oppermann, L. Deslauriers, and H. Pfister, "Teaching visualization for large and diverse classes on campus and online," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[75] A. Godwin, "Let's play: Design games and other strategies for introducing visualization through active learning," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[76] S. Huron, S. Carpendale, J. Boy, and J. D. Fekete, "Using VisKit: A manual for running a constructive visualization workshop," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[77] A. Zoss, "Challenges and solutions for short-form data visualization instruction," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[78] N. Maiden, C. Ncube, and S. Robertson, "Can requirements be creative? Experiences with an enhanced air space management system," in *Proc. Int. Conf. Software Eng.*, IEEE, 2007, pp. 632–641.

[79] N. Maiden, S. Manning, S. Robertson, and J. Greenwood, "Integrating creativity workshops into structured requirements processes," in *Proc. Conf. on Designing Interactive Syst.: Processes, Practices, Methods, and Techn.*, ACM, 2004, pp. 113–122.

[80] H. Beyer and K. Holtzblatt, *Contextual Design: Defining Customer-Centered Systems*. San Francisco, CA: Morgan Kaufmann, 1997.

[81] P. Cairns and A. Cox, *Research Methods for Human-Computer Interaction*. New York, NY: Cambridge University Press, 2008.

[82] J. Lazar, J. H. Feng, and H. Hochheiser, *Research Methods in Human-Computer Interaction*. Hoboken, NJ: John Wiley & Sons, 2010.

[83] R. H. von Alan, S. T. March, J. Park, and S. Ram, "Design science in information systems research," *Manage. Inform. Syst. (MIS) Quart.*, vol. 28, no. 1, pp. 75–105, 2004.

[84] B. Hanington, "Methods in the making: A perspective on the state of human research in design," *Des. Issues*, vol. 19, no. 4, pp. 9–18, 2003.

[85] W. W. Royce, "Managing the development of large software systems: Concepts and techniques," in *Proc. Int. Conf. Software Eng.*, IEEE, 1970, pp. 328–338.

[86] P. McLachlan, T. Munzner, E. Koutsofios, and S. North, "LiveRAC: Interactive visual exploration of system management time-series data," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2008, pp. 1483–1492.

[87] B. Hartmann, S. R. Klemmer, M. Bernstein, L. Abdulla, B. Burr, A. Robinson-Mosher, and J. Gee, "Reflective physical prototyping through integrated design, test, and analysis," in *Proc. Symp. on User Interface Software and Technol. (UIST)*, ACM, 2006, pp. 299–308.

[88] M. da Gandra, M. P. Hea, M. van Neck, and M. Hea, "(in)forming the information design student," *Parsons J. Inform. Mapping*, vol. 5, no. 3, pp. 1–7, 2013.

[89] J. A. Ferstay, C. B. Nielsen, and T. Munzner, "Variant View: Visualizing sequence variants in their gene context," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2546–2555, 2013.

[90] D. B. Walz, J. J. Elam, and B. Curtis, "Inside a software design team: Knowledge acquisition, sharing, and integration," *Commun. ACM*, vol. 36, no. 10, pp. 63–77, 1993.

[91] M. Bohøj, N. G. Borchorst, N. O. Bouvin, S. Bødker, and P.-O. Zander, "Timeline collaboration," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2010, pp. 523–532.

[92] Harvard Business Review, *Vision statement: A taxonomy of innovation*, http://hbr.org/2014/01/a-taxonomy-of-innovation/ar/1, 2014. (visited on 02/20/2014).

[93] A. Strauss and J. Corbin, *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. Thousand Oaks, CA: SAGE, 1990.

[94] N. V. Hernandez, J. J. Shah, and S. M. Smith, "Understanding design ideation mechanisms through multilevel aligned empirical studies," *Des. Stud.*, vol. 31, no. 4, pp. 382–410, 2010.

[95] H. R. Bernard, *Research Methods in Anthropology: Qualitative and Quantitative Approaches*. Lanham, MD: Rowman Altamira, 2011.

[96] M. Maguire, "Methods to support human-centred design," *Int. J. Human-Comput. Stud.*, vol. 55, no. 4, pp. 587–634, 2001.

[97] S. P. Dow, A. Glassco, J. Kass, M. Schwarz, D. L. Schwartz, and S. R. Klemmer, "Parallel prototyping leads to better design results, more divergence, and increased self-efficacy," *Trans. Comput.-Human Interaction*, vol. 17, no. 4, 18:1–18:24, 2010.

[98] H. Hutchinson, W. Mackay, B. Westerlund, B. B. Bederson, A. Druin, C. Plaisant, M. Beaudouin-Lafon, S. Conversy, H. Evans, H. Hansen, N. Roussel, and B. Eiderbäck, "Technology probes: Inspiring design for and with families," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2003, pp. 17–24.

[99] A. D'Amico and K. Whitley, "The real work of computer network defense analysts," in *Proc. Int. Workshop on Vis. for Cyber Security (VizSec)*, Springer, 2008, pp. 19–37.

[100] R. F. Erbacher, D. A. Frincke, P. Chung Wong, S. Moody, and G. Fink, "A multiphase network situational awareness cognitive task analysis," *Inform. Vis.*, vol. 9, no. 3, pp. 204–219, 2010.

[101] G. A. Fink, C. L. North, A. Endert, and S. J. Rose, "Visualizing cyber security: Usable workspaces," in *Proc. Int. Workshop on Vis. for Cyber Security (VizSec)*, IEEE, 2009, pp. 45–56.

[102] M. Sedlmair, C. Heinzl, S. Bruckner, H. Piringer, and T. Möller, "Visual parameter space analysis: A conceptual framework," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2161–2170, 2014.

[103] D. Staheli, T. Yu, R. J. Crouser, D. O. Gwynn, S. McKenna, and L. Harrison, "Visualization evaluation for cyber security: Trends and future directions," in *Proc. Int. Symp. on Vis. for Cyber Security (VizSec)*, 2014, pp. 49–56.

[104] O. Kashan, *Information is beautiful: World's biggest data breaches*, http://informationisbeautiful.net/visualizations/worlds-biggest-data-breaches-hacks/, 2015. (visited on 10/14/2015).

[105] D. M. Best, A. Endert, and D. Kidwell, "7 key challenges for visualization in cyber network defense," in *Proc. Int. Symp. on Vis. for Cyber Security (VizSec)*, New York, NY: ACM, 2014, pp. 33–40.

[106] J. Wood, R. Beecham, and J. Dykes, "Moving beyond sequential design: Reflections on a rich multi-channel approach to data visualization," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2171–2180, 2014.

[107] R. F. Erbacher, D. A. Frincke, P. C. Wong, S. Moody, and G. Fink, "Cognitive task analysis of network analysts and managers for network situational awareness," in *Proc. Conf. on Vis. and Data Anal.*, vol. 7530, SPIE, 2010.

[108] P. Ren, Y. Gao, Z. Li, Y. Chen, and B. Watson, "IDGraphs: Intrusion detection and analysis using histographs," in *Proc. Int. Workshop on Vis. for Cyber Security (VizSec)*, IEEE, 2005, pp. 39–46.

[109] K. Lakkaraju, W. Yurcik, and A. J. Lee, "NVisionIP: Netflow visualizations of system state for security situational awareness," in *Proc. Workshop on Vis. and Data Mining for Comput. Security (VizSec)*, ACM, 2004, pp. 65–72.

[110] H. Koike, K. Ohno, and K. Koizumi, "Visualizing cyber attacks using IP matrix," in *Proc. Int. Workshop on Vis. for Cyber Security (VizSec)*, IEEE, 2005, pp. 91–98.

[111] R. Erbacher, K. Christensen, and A. Sundberg, "Designing visualization capabilities for IDS challenges," in *Proc. Int. Workshop on Vis. for Cyber Security (VizSec)*, IEEE, 2005, pp. 121–127.

[112] X. Yin, W. Yurcik, M. Treaster, Y. Li, and K. Lakkaraju, "VisFlowConnect: Netflow visualizations of link relationships for security situational awareness," in *Proc. Workshop on Vis. and Data Mining for Comput. Security (VizSec)*, ACM, 2004, pp. 26–34.

[113] H. Choi, H. Lee, and H. Kim, "Fast detection and visualization of network attacks on parallel coordinates," *Comput. Security*, vol. 28, no. 5, pp. 276–288, 2009.

[114] J. R. Goodall and M. Sowul, "VIAssist: Visual analytics for cyber defense," in *Proc. Conf. on Technol. for Homeland Security (HST)*, IEEE, 2009, pp. 143–150.

[115] S. Foresti, J. Agutter, Y. Livnat, S. Moon, and R. Erbacher, "Visual correlation of network alerts," *Comput. Graph. Appl.*, vol. 26, no. 2, pp. 48–59, 2006.

[116] T. Taylor, D. Paterson, J. Glanfield, C. Gates, S. Brooks, and J. McHugh, "FloVis: Flow visualization system," in *Cybersecurity Appl. Technol. Conf. for Homeland Security (CATCH)*, IEEE, 2009, pp. 186–198.

[117] C. Paul, R. Rohrer, P. Sponaugle, J. Huston, and B. Nebesh, "CyberSAVI: A cyber situation awareness visual interface for mission-level network situation awareness," in *Proc. Int. Symp. on Vis. for Cyber Security (VizSec)*, 2013.

[118] J. J. Fowler, T. Johnson, P. Simonetto, M. Schneider, C. Acedo, S. Kobourov, and L. Lazos, "IMap: Visualizing network activity over Internet maps," in *Proc. Int. Symp. on Vis. for Cyber Security (VizSec)*, ACM, 2014, pp. 80–87.

[119] F. Fischer and D. A. Keim, "NStreamAware: Real-time visual analytics for data streams to enhance situational awareness," in *Proc. Int. Symp. on Vis. for Cyber Security (VizSec)*, ACM, 2014, pp. 65–72.

[120] S. Musa and D. J. Parish, "Using time series 3D AlertGraph and false alert classification to analyse snort alerts," in *Proc. Int. Workshop on Vis. for Cyber Security (VizSec)*, Springer, 2008, pp. 169–180.

[121] G. Fink, P. Muessig, and C. North, "Visual correlation of host processes and network traffic," in *Proc. Int. Workshop on Vis. for Cyber Security (VizSec)*, IEEE, 2005, pp. 11–19.

[122] R. Blue, C. Dunne, A. Fuchs, K. King, and A. Schulman, "Visualizing real-time network resource usage," in *Proc. Int. Workshop on Vis. for Cyber Security (VizSec)*, Springer, 2008, pp. 119–135.

[123] J. Brooke, "SUS — a quick and dirty usability scale," *Usability Evaluation in Industry*, vol. 189, no. 194, pp. 4–7, 1996.

[124] T. R. Leschke and C. Nicholas, "Change-link 2.0: A digital forensic tool for visualizing changes to shadow volume data," in *Proc. Int. Symp. on Vis. for Cyber Security (VizSec)*, ACM, 2013, pp. 17–24.

[125] J. Rasmussen, K. Ehrlich, S. Ross, S. Kirk, D. Gruen, and J. Patterson, "Nimble cybersecurity incident management through visualization and defensible recommendations," in *Proc. Int. Symp. on Vis. for Cyber Security (VizSec)*, ACM, 2010, pp. 102–113.

[126] S. Amershi, B. Lee, A. Kapoor, R. Mahajan, and B. Christian, "CueT: Human-guided fast and accurate network alarm triage," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2011, pp. 157–166.

[127] J. D. Howard and T. A. Longstaff, "A common language for computer security incidents," Sandia National Labs, Tech. Rep., 1998.

[128] J. Zhao, N. Cao, Z. Wen, Y. Song, Y.-R. Lin, and C. Collins, "#FluxFlow: Visual analysis of anomalous information spreading on social media," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 1773–1782, 2014.

[129] N. Cao, C. Shi, S. Lin, J. Lu, Y.-R. Lin, and C.-Y. Lin, "TargetVue: Visual analysis of anomalous user behaviors in online communication systems," *IEEE Trans. Vis. Comput. Graphics*, vol. 22, no. 1, pp. 280–289, 2016.

[130] MaxMind, *GeoLite2 free databases*, http://dev.maxmind.com/geoip/geoip2/geolite2/, 2015. (visited on 10/19/2015).

[131] T. Yu, R. Lippmann, J. Riordan, and S. Boyer, "EMBER: A global perspective on extreme malicious behavior," in *Proc. Int. Symp. on Vis. for Cyber Security (VizSec)*, ACM, 2010, pp. 1–12.

[132] J. Wood and J. Dykes, "Spatially ordered treemaps," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1348–1355, 2008.

[133] J. Pruitt and J. Grudin, "Personas: Practice and theory," in *Proc. Conf. on Designing for User Experiences (DUX)*, ACM, 2003, pp. 1–15.

[134] S. Faily and I. Flechais, "Persona cases: A technique for grounding personas," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2011, pp. 2267–2270.

[135] Y.-N. Chang, Y.-k. Lim, and E. Stolterman, "Personas: From theory to practices," in *Proc. Nordic Conf. on Human-Comput. Interaction: Building Bridges (NordiCHI)*, ACM, 2008, pp. 439–442.

[136] J. McGinn and N. Kotamraju, "Data-driven persona development," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2008, pp. 1521–1524.

[137] C. L. Paul and K. Whitley, "A taxonomy of cyber awareness questions for the user-centered design of cyber situation awareness," in *Proc. Int. Conf. on Human Aspects of Inform. Security, Privacy, and Trust (HAS)*, Springer, 2013, pp. 145–154.

[138] M. Bostock, *Pseudo-Dorling cartogram*, http://bl.ocks.org/mbostock/4055892, 2015. (visited on 10/20/2015).

[139] D. Dorling, *Area Cartograms: Their Use and Creation*, ser. Concepts and Techn. in Modern Geography. London, UK: Environmental Publications, 1996.

[140] S. Few, "Bullet graph design specification," Perceptual Edge, Tech. Rep., 2010.

[141] R. Bargo, J. Dearden, and M. W. Jones, "Order of magnitude markers: An empirical study on large magnitude number detection," *IEEE Trans. Vis. Comput. Graphics*, vol. 20, no. 12, pp. 2261–2270, 2014.

[142] J. Sauro, *Measuring usability with the System Usability Scale*, `http://measuringu.com/sus`, 2011. (visited on 02/02/2015).

[143] M. Ogrinz, *Mashup Patterns: Designs and Examples for the Modern Enterprise*. Boston, MA: Addison-Wesley, 2009.

[144] G. E. Marai, "Visual scaffolding in integrated spatial and nonspatial analysis," in *Proc. EuroVis Workshop on Visual Analytics (EuroVA)*, The Eurographics Assoc., 2015.

[145] E. R. Tufte, *The Visual Display of Quantitative Information*. Cheshire, CT: Graphics Press, 1986.

[146] Y. J. Reimer and S. A. Douglas, "Teaching HCI design with the studio approach," *Comput. Sci. Edu.*, vol. 13, no. 3, pp. 191–205, 2003.

[147] P. Rheingans, "Minor adventures in flipped classrooms, team-based learning, and other pedagogical buzzwords," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[148] G. Domik, "A data visualization course at the university of paderborn," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[149] A. Kerren, J. T. Stasko, and J. Dykes, "Teaching information visualization," in *Information Visualization: Human-Centered Issues and Perspectives*, A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, Eds., Berlin, DEU: Springer, 2008, pp. 65–91.

[150] S. Greenberg, "Embedding a design studio course in a conventional computer science program," in *Creativity and HCI: From Experience to Design in Education*, Springer, 2009, pp. 23–41.

[151] K. Cennamo, S. A. Douglas, M. Vernon, C. Brandt, B. Scott, Y. Reimer, and M. McGrath, "Promoting creativity in the computer science design studio," in *Proc. Tech. Symp. Comput. Sci. Edu. (SIGCSE)*, ACM, 2011, pp. 649–654.

[152] A. Johnson, "Teaching data visualization in evl's cyber-commons classroom," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[153] B. Craft, R.-m. Emerson, and T. J. Scott, "Using pedagogic design patterns for teaching and learning information visualization," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[154] M. Eggermont, C. Perin, B. Aseniero, and R. Fallah, "Leveraging biological inspiration in an information visualization class," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[155] E. Mustafaraj, "Students' prior knowledge of data visualization," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[156] B. Alper, N. Henry Riche, F. Chevalier, J. Boy, and M. Sezgin, "C'est la vis: Visualization literacy at elementary school," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2017.

[157] W. Willett and S. Huron, "A constructive classroom exercise for teaching infovis," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[158] W. Gaver, "What should we expect from research through design?" In *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2012, pp. 937–946.

[159] Ž. Obrenović, "Design-based research: What we learn when we engage in design of interactive systems," *Interactions*, vol. 18, no. 5, pp. 56–59, 2011.

[160] T. Munzner, "Process and pitfalls in writing information visualization research papers," in *Information Visualization*, A. Kerren, J. T. Stasko, J.-D. Fekete, and C. North, Eds., Berlin, DEU: Springer, 2008, pp. 134–153.

[161] J. O. Wobbrock and J. A. Kientz, "Research contributions in human-computer interaction," *Interactions*, vol. 23, no. 3, pp. 38–44, 2016.

[162] A. P. Dempster, *Elements of Continuous Multivariate Analysis*. Boston, MA: Addison-Wesley, 1969.

[163] J. L. Rodgers, W. A. Nicewander, and L. Toothaker, "Linearly independent, orthogonal, and uncorrelated variables," *The Amer. Statistician*, vol. 38, no. 2, pp. 133–134, 1984.

[164] L. Corsten and K. Gabriel, "Graphical exploration in comparing variance matrices," *Biometrics*, vol. 32, no. 4, pp. 851–863, 1976.

[165] M. W. Trosset, "Visualizing correlation," *J. Computational Graphical Statist.*, vol. 14, no. 1, pp. 1–19, 2005.

[166] B. Falissard, "Focused principal component analysis: Looking at a correlation matrix with a particular interest in a given variable," *J. Computational Graphical Statist.*, vol. 8, no. 4, pp. 906–912, 1999.

[167] D. F. Swayne, D. Cook, and A. Buja, "XGobi: Interactive dynamic data visualization in the X Window System," *J. Computational Graphical Statist.*, vol. 7, no. 1, pp. 113–130, 1998.

[168] D. F. Swayne, D. T. Lang, A. Buja, and D. Cook, "GGobi: Evolving from XGobi into an extensible framework for interactive data visualization," *Computational Statist. & Data Anal.*, vol. 43, no. 4, pp. 423–444, 2003.

[169] N. Elmqvist, P. Dragicevic, and J.-D. Fekete, "Rolling the dice: Multidimensional visual exploration using scatterplot matrix navigation," *IEEE Trans. Vis. Comput. Graphics*, vol. 14, no. 6, pp. 1141–1148, 2008.

[170] S. Ullman, *The Interpretation of Visual Motion*. Cambridge, MA: MIT Press, 1979.

[171] J. Seo and B. Shneiderman, "Interactively exploring hierarchical clustering results [gene identification]," *Comput.*, vol. 35, no. 7, pp. 80–86, 2002.

[172] P. Langfelder and S. Horvath, "WGCNA: An R package for weighted correlation network analysis," *Bioinformatics*, vol. 9, no. 1, pp. 559–571, 2008.

[173] M. C. Oldham, G. Konopka, K. Iwamoto, P. Langfelder, T. Kato, S. Horvath, and D. H. Geschwind, "Functional organization of the transcriptome in human brain," *Nature Neuroscience*, vol. 11, no. 11, pp. 1271–1282, 2008.

[174] K. D. Winden, M. C. Oldham, K. Mirnics, P. J. Ebert, C. H. Swan, P. Levitt, J. L. Rubenstein, S. Horvath, and D. H. Geschwind, "The organization of the transcriptional network in specific neuronal classes," *Molecular Syst. Biology*, vol. 5, no. 1, pp. 291–308, 2009.

[175] E. Segel and J. Heer, "Narrative visualization: Telling stories with data," *IEEE Trans. Vis. Comput. Graphics*, vol. 16, no. 6, pp. 1139–1148, 2010.

[176] C. D. Stolper, B. Lee, N. Henry Riche, and J. Stasko, "Emerging and recurring data-driven storytelling techniques: Analysis of a curated collection of recent stories," Microsoft Research, Tech. Rep. MSR-TR-2016-14, 2016.

[177] F. Amini, N. Henry Riche, B. Lee, C. Hurter, and P. Irani, "Understanding data videos: Looking at narrative visualization through the cinematography lens," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2015, pp. 1459–1468.

[178] F. Amini, N. Henry Riche, B. Lee, A. Monroy-Hernandez, and P. Irani, "Authoring data-driven videos with DataClips," *IEEE Trans. Vis. Comput. Graphics*, vol. 23, no. 1, pp. 501–510, 2017.

[179] B. Bach, N. Kerracher, K. W. Hall, S. Carpendale, J. Kennedy, and N. Henry Riche, "Telling stories about dynamic networks with graph comics," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2016, pp. 3670–3682.

[180] A. Tse, *Why we are doing fewer interactives*, https://github.com/archietse/malofiej-2016/blob/master/tse-malofiej-2016-slides.pdf, 2016. (visited on 11/28/2016).

[181] G. Aisch, *Information+ conference: Data visualization and the news*, https://vimeo.com/182590214, 2016. (visited on 03/22/2017).

[182] R. Kosara, *The scrollytelling scourge*, https://eagereyes.org/blog/2016/the-scrollytelling-scourge, 2016. (visited on 11/28/2016).

[183] S. Yee and T. Chu, *A visual introduction to machine learning*, http://www.r2d3.us/visual-intro-to-machine-learning-part-1/, 2015. (visited on 11/28/2016).

[184] H. L. O'Brien and E. G. Toms, "The development and evaluation of a survey to measure user engagement," *J. Amer. Soc. Inform. Sci. Technol.*, vol. 61, no. 1, pp. 50–69, 2010.

[185] R Development Core Team, *R: A Language and Environment for Statistical Computing*, R Foundation for Statistical Computing, Vienna, AUT, 2008.

[186] D. Bates, M. Mächler, B. Bolker, and S. Walker, "Fitting linear mixed-effects models using lme4," *J. Statistical Software*, vol. 67, no. 1, pp. 1–48, 2015.

[187] R. Buchanan, "Wicked problems in design thinking," *Des. Issues*, vol. 8, no. 2, pp. 5–21, 1992.

[188] R. Farrell and C. Hooker, "Design, science and wicked problems," *Des. Stud.*, vol. 34, no. 6, pp. 681–705, 2013.

[189] R. M. Kirby and M. Meyer, "Visualization collaborations: What works and why," *Comput. Graph. Appl.*, vol. 33, no. 6, pp. 82–88, 2013.

[190] A. Abran, P. Bourque, R. Dupuis, and J. W. Moore, *Guide to the Software Engineering Body of Knowledge (SWEBOK)*. Hoboken, NJ: IEEE Press, 2001.

[191] K. Beck, M. Beedle, A. van Bennekum, A. Cockburn, W. Cunningham, M. Fowler, J. Grenning, J. Highsmith, A. Hunt, R. Jeffries, et al., "Manifesto for agile software development," 2001. (visited on 05/18/2017).

[192] S. Simon, S. Mittelstädt, D. A. Keim, and M. Sedlmair, "Bridging the gap of domain and visualization experts with a liaison," in *Eurographics Conf. on Vis. (EuroVis) - Short Papers*, The Eurographics Assoc., 2015.

[193] T. von Landesberger, F. Brodkorb, P. Schneider, and K. Ballweg, "Tool for teaching visualization techniques: Learning and homework assignments for multivariate data visualization," in *Pedagogy Data Vis., IEEE VIS Workshop*, 2016.

[194] M. A. Hearst, P. Laskowski, and L. Silva, "Evaluating information visualization via the interplay of heuristic evaluation and question-based scoring," in *Proc. Conf. Human Factors in Computing Syst. (CHI)*, ACM, 2016, pp. 5028–5033.

[195] E. Bertini, *Teaching | Information Visualization*, `http://enrico.bertini.io/teaching/`, 2017. (visited on 01/19/2017).

[196] ISO, IEC, and IEEE, "Systems and software engineering – Vocabulary," *Int. Org. Standardization*, 2010.

[197] K. Peffers, T. Tuunanen, M. A. Rothenberger, and S. Chatterjee, "A design science research methodology for information systems research," *J. Manage. Inform. Syst. (JMIS)*, vol. 24, no. 3, pp. 45–77, 2007.

[198] J. Knapp, J. Zeratsky, and B. Kowitz, *Sprint: How to Solve Big Problems and Test New Ideas in Just Five Days*. New York, NY: Simon & Schuster, 2016.

[199] M. Gonçalves, C. Cardoso, and P. Badke-Schaub, "What inspires designers? preferences on inspirational approaches during idea generation," *Des. Stud.*, vol. 35, no. 1, pp. 29–53, 2014.

[200] B. Crandall, G. A. Klein, and R. R. Hoffman, *Working Minds: A Practitioner's Guide to Cognitive Task Analysis*. Cambridge, MA: MIT Press, 2006.

[201] Z. Zhang, B. Wang, F. Ahmed, I. Ramakrishnan, R. Zhao, A. Viccellio, and K. Mueller, "The five Ws for information visualization with application to healthcare informatics," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 11, pp. 1895–1910, 2013.

[202] H.-J. Schulz, T. Nocke, M. Heitzler, and H. Schumann, "A design space of visualization tasks," *IEEE Trans. Vis. Comput. Graphics*, vol. 19, no. 12, pp. 2366–2375, 2013.

[203] D. L. Dean, J. M. Hender, T. L. Rodgers, and E. L. Santanen, "Identifying quality, novel, and creative ideas: Constructs and scales for idea evaluation," *J. Assoc. Inform. Syst.*, vol. 7, no. 10, pp. 646–698, 2006.

[204] E. Goodman, M. Kuniavsky, and A. Moed, *Observing the User Experience: A Practitioner's Guide to User Research*. San Francisco, CA: Morgan Kaufmann, 2012.