Design Activity Framework for Visualization Design

Sean McKenna, Dominika Mazur, James Agutter, and Miriah Meyer

Abstract— An important aspect in visualization design is the connection between what a designer does and the decisions the designer makes. Existing design process models, however, do not explicitly link back to models for visualization design decisions. We bridge this gap by introducing the *design activity framework*, a process model that explicitly connects to the nested model, a well-known visualization design decision model. The framework includes four overlapping activities that characterize the design process, with each activity explicating outcomes related to the nested model. Additionally, we describe and characterize a list of exemplar methods and how they overlap among these activities. The design activity framework is the result of reflective discussions from a collaboration on a visualization redesign project, the details of which we describe to ground the framework in a real-world design process. Lastly, from this redesign project we provide several research outcomes in the domain of cybersecurity, including an extended data abstraction and rich opportunities for future visualization research.

Index Terms—Design, frameworks, process, cybersecurity, nested model, decisions, models, evaluation, visualization

1 Introduction

As the field of visualization matures, theories and models that capture the *how* of visualization design have become more prevalent, from evaluation strategies [11, 29, 34, 49, 55] to the design process itself [32, 37, 54, 55, 60, 66]. While these theories and models for the design process largely address the *how* of visualization design, they fail to explicitly describe the connections of those actions back to the *why* of visualization design decisions — these design decisions are described instead by separate design decision models [44, 45]. Furthermore, visualization design is known to be messy, iterative, and complex [16, 32, 54, 60, 61, 66], characteristics that are not fully described in existing visualization process models.

In particular, we encountered the insufficiencies of current visualization process models while working on a project with a multidisciplinary design team consisting of 2 visualization experts, 2 designers, and 1 psychologist. Our team was tackling 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 [54], 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 methods that are useful at any given point? What types of outcomes should I be working towards along the way? How do I know my outcomes 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, or a lack of implementable and immediately usable guidance that helps a visualization practitioner explicitly navigate a real-world visualization design process.

On the other hand, the two designers in the group were accustomed to working with an iterative and open design process, and to exploring a broad range of methods for generating and evaluating outcomes throughout. Consequently, the two designers brought different experiences and a unique set of insights and tools to our collaborative visualization design process. Although other visualization researchers stress that design and creativity research methods can play an effective role in visualization design [23, 37, 61], design research also tends to more explicitly emphasize and highlight the complex nature of the design process [2, 9], as well as put an emphasis on constraints for design [48, 52]. It is not clear, however, how this design research explicitly addresses or captures well-established visualization design decisions, namely those described by the levels of the nested model [45]. As a team, we struggled to reconcile the visualization design decisions we wanted to make with these various creative process models.

By reflecting on our own design process as a team, we identified a need for a process framework that balances the flexibility and actionability of models from the design community with the explicit outcomes and decisions necessary for visualization design. To meet this goal, we propose the *design activity framework* to explicitly bridge the gap between the activities that visualization designers do with the visualization decisions they make. This framework is composed of four overlapping activities: *understand*, *ideate*, *make*, and *deploy*. Each activity has a specific motivation to help place the visualization designer within the framework, as well as defined, tangible outcomes that relate to the levels of the nested model. Visualization designers produce outcomes in each activity using both *generative* methods as well as *evaluative* ones — the breadth and formality of these methods can, and should, be dictated by the real-world constraints of a project. We developed the design activity framework to overcome shortcomings in existing visualization design process models [16, 23, 32, 37, 54, 60, 66] and to incorporate ideas from a broad range of models in HCI [47, 63, 68] and design [1, 6, 10, 14, 27, 30, 36, 38, 56, 59, 65, 67–69].

The primary contribution of this work is the new design activity framework for providing actionable guidance throughout the visualization design process. This framework makes explicit the link between the design process and visualization design decisions. In addition, we provide two secondary contributions: first, an extensive list of exemplar methods for use throughout the design activity framework, drawing on both well-known methods from the visualization community as well as many less common methods found in the design literature; and second, we use the context of our redesign project to illustrate how the details of a real-world design process are captured by the framework. Lastly, we highlight a tertiary contribution from our cybersecurity redesign project, where the framework enabled us to produce research insights at both the domain characterization and abstraction levels of the nested model.

We first introduce related design decision and process models in Section 2, and then frame our real-world visualization redesign project in Section 3. In Section 4 we present the primary contribution of this paper, the design activity framework, using our redesign project as an illustrative example. Next we highlight a secondary contribution of exemplar methods in Section 5. Finally, in Section 6, we provide our tertiary contribution in the field of cybersecurity including an extension of an existing data abstraction and several key opportunities for future visualization research, followed by a general discussion of the design activity framework in Section 7.

- *S. McKenna and M. Meyer are with the School of Computing, University of Utah, Salt Lake City, UT, 84112. E-mail: [sean,miriah]@cs.utah.edu*
- *D. Mazur is with the Department of Psychology, University of Utah, Salt Lake City, UT, 84112. E-mail: dominika.mazur@gmail.com*
- *J. Agutter is with the College of Architecture and Planning, University of Utah, Salt Lake City, UT, 84112. E-mail: agutterja@arch.utah.edu*

2 RELATED WORK

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 [19]. As such, there exist a variety of different theoretical models for visualization design and even more that have been adapted and used by visualization designers. In this work we focus on two kinds of models for visualization design: decision models and process models [44]. **Decision models** capture the *what* and *why* of design by characterizing the rationale behind the decisions that a designer makes. **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 a visualization designer to verify and validate the design decisions they make along each step of the design process. This is highlighted by Schön's reflection-in-action concept, which emphasizes that the process of doing and thinking are complementary to each other [53]; thus, the design process and its many design decisions are intricately interconnected.

2.1 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 versus solution-driven, along with specific operationalization of that mindset, or how decisions get made by an individual or a team, such as autocratic versus autonomic [12]. Similarly, Tang *et al.* characterize design decisions into three groups: planning, problem space, and solution space decisions, in order to better realize the effect decisions have on design [58]. Through studying the process of expert designers, Wu *et al.* identify three classes of design strategies: forward working, problem switching, and backward working strategies [69]. Furthermore, several researchers have broken down decision making into different kinds of high-level design judgments: *e.g.* appearance, compositional, navigational, *etc.* [46, 68]. While these many models have a utility to analyze and compare general decisions and strategies for design, these models 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 [45] is the de facto design decision model. 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. A recent extension to the model, called the nested blocks and guidelines model [44], provides a more fine-grained characterization of individual design decisions as blocks at each level, with guidelines describing the relationships between blocks. 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 needs to make along the way [44].

2.2 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. In this regard, we consider design as a challenge which combines and mixes both engineering and creative design processes [27, 38, 61, 68], 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 towards a single solution [38]. On the other hand, a **creative design process** begins with more gradual problem scoping, and the process has many overlapping activities where many different possibilities are explored before choosing a single solution [38]. As recognized by researchers in the design [27], HCI [38,68], and visualization [61] communities, the combination and balanced mixture of these two types of process models is useful for characterizing the design process.

Visualization-specific design process models describe unique aspects for designing and evaluating visualization systems; however, they largely do not connect back to visualization design decisions and do not explicitly incorporate aspects of a creative design process. The seminal research method of multidimensional longitudinal case studies [55] 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, respectively — to perform user-centered design [32,60,66], but they are solely engineering design process models. The design process model used by both Lloyd *et al.* [37] and Goodwin *et al.* [23] is drawn from an international standard on human-centered design, ISO13407, which has recently been updated, ISO9241-210 [16]. This standard's model describes different design activities as a cycle, emphasizing an engineering approach. Goodwin *et al.* accompany this engineering process model with specific methods for eliciting creativity from end users [23], a step towards including aspects of a creative design process. Vande Moere and Purchase further characterize the role of design in visualization, arguing for a balanced approach that mixes both creative and engineering aspects [61]. While the visualization community is beginning to embrace aspects of a creative design process, none of these process models explicitly link back to visualization design decisions.

The model closest to the design activity framework is the nine-stage framework for conducting design studies [54], which 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 outcomes should be expected at each step is not described. 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. The nine-stage framework, while the first model of its kind to provide guidance for conducting design studies, does not give actionable advice for knowing what stage a designer is in, what kinds of methods to employ, or the specific outcomes and decisions a designer should make, particularly in the middle four design stages. The design activity framework is largely inspired by the nine-stage framework, in particular to provide actionable guidance not currently available within this process model.

3 VISUALIZATION REDESIGN PROJECT

The motivation behind the design activity framework stems from our experience of working as a multidisciplinary visualization design team on a redesign project. This seven-month long 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, relying 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 mock-ups for how to redesign the visualizations 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 that 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 available for implementing our ideas by software developers, confidentiality issues surrounding cybersecurity data, and the engineering realities of working within a large software system.

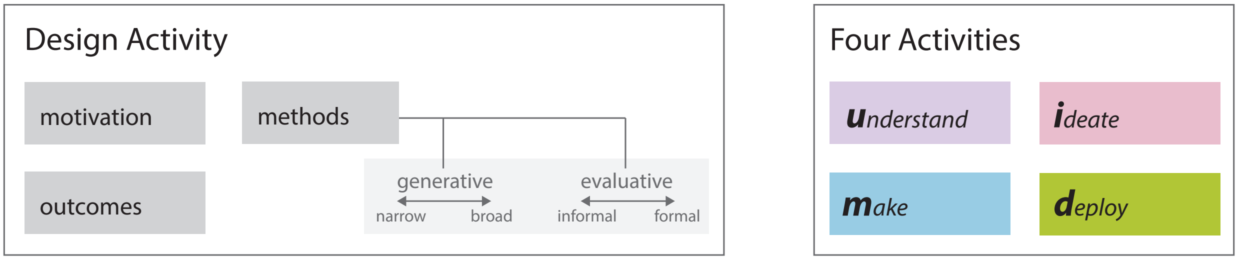

Fig. 1. We present the framework's breakdown of a *design activity*: motivation, outcomes, and methods, where methods can be generative or evaluative. Additionally, we provide four overlapping, multilinear activities which compose the design activity framework.

For our redesign process, we first took a step back from the existing tool to characterize and understand the visualization challenges in the field of cybersecurity analysis. Significant research exists on the types of tasks and data that cybersecurity analysts work with [15, 18, 21], which we analyzed to develop a series of design requirements to motivate and guide our redesign. We found that several of these insights regarding the domain characterization offer rich opportunities for visualization that are not currently met by existing tools, as discussed in detail in Section 6. Based on our analysis of the literature and interviews with analysts, we brainstormed ideas for improving RVS and prototyped several visualization redesigns. Finally, the company's RVS development team selected a handful of our recommended changes, implemented these within the tool, and validated their effectiveness with several cybersecurity analysts. We discuss the details of this redesign process, including the methods used and the specific outcomes of our design activities, in Section 4.4.

Throughout the course of this project we were careful to establish a connection between our motivations, actions, and visualization design decisions, which we believe aided us in efficiently and effectively working together towards a validated visualization redesign. At the conclusion of the project our design team conducted a series of reflective discussions about this process, resulting in the design activity framework for visualization design which we discuss next.

4 Design Activity Framework

In this section we present the *design activity framework*, a flexible structure meant to guide a designer through the real-world, multilinear, and iterative process of developing a visualization for a specific problem or application domain. We envision the framework as a lens that designers can 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 [45] to explicitly link the actions a visualization designer takes 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 present the idea of a *design activity* in Section 4.1, which forms the basis of the new framework. From there we describe, in Section 4.2, the four activities contained in the framework: *understand*, *ideate*, *make*, and *deploy*. For each activity, we articulate the motivation, outcomes, and the explicit link back to the levels of the nested model. To further ground the framework, we discuss each activity in Section 4.4 within the context of our cybersecurity redesign project, providing an example of how the framework aids in guiding and summarizing a real-world visualization project.

4.1 A Design Activity

At the core of this framework is the concept of an **activity**, a group of actions a designer takes to work towards a specific outcome, or set of outcomes. Many creative process models tend to avoid breaking a process into sequential steps, stages, or phases, but, rather, they use the term activities [1, 10, 47, 56, 67, 69], which are not necessarily linear, and they are often overlapping. Each activity is composed of several key components: a *motivation*; clear, tangible *outcomes* related to design decisions; and a collection of *methods*. We pictorially represent the design activity framework in Figure 1.

The **motivation** of an activity is the specific purpose behind the methods and actions that are performed within that activity. For example, is my motivation to brainstorm new ideas to solve a specific problem? Or, is it to test the efficacy of an aspect of my chosen visualization for a specific task? By matching a real-world motivation to those specified for each activity in the framework a visualization designer can place themselves within a specific design activity, which helps in choosing appropriate methods and identifying outcomes.

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

We highlight two distinct kinds of methods used in each design activity: *generative* versus *evaluative*. Generative methods are largely meant to be divergent and create many outcomes, such as methods for brainstorming [27, 41] or increasing creativity [23, 40]. Evaluative methods, on the other hand, are convergent and filter outcomes, such as methods that elicit feedback from domain experts [4, 34] or user studies [8, 35]. This distinction between generation and evaluation is common within the design community [6, 12, 24, 62, 62]. Interestingly, some methods can be both generative and evaluative, such as observation and interviewing. In the design activity framework we consider generative and evaluative methods as vital components of each activity, unlike process models which capture evaluation as a single, unique stage in the design process [16, 23, 32, 37, 60, 66].

The design activity framework further characterizes the methods based on two spectrums. First, generative methods can be used *narrowly to broadly*. For example, a 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 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.

4.2 The Four Activities

We have identified four overlapping, critical activities for designing visualizations for real-world problems and applications: understand, ideate, make, and deploy. As shown in Figure 2, 3 of the 4 activities map to several levels of the nested model, implying that a specific design activity can be used to make different types of visualization design decisions. Conversely, a designer focusing on just one type of design decision will often move through different activities; thus, the culmination of a complete visualization could involve moving through this

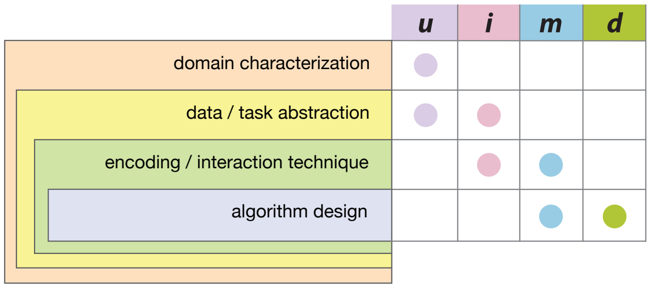

Fig. 2. Here, we illustrate the overlap of the design activity framework with respect to the levels of the nested model [45]. It is important to note that all three inner levels of the nested model each exist across two activities in the framework; thus, a visualization designer must think carefully about which levels of the nested model any process outcome corresponds with.

framework in a complex, iterative, and *multilinear* fashion. By multilinear we mean that a process combines forward, linear movement with cyclic, backwards, and parallel movements.

Next, we articulate the unique motivations and outcomes for each of the four activities in the design activity framework. We present a list of exemplar methods in Table 1, and a more extensive list in the Supplemental Materials, with each method characterized by which activities it is effective for, and whether it can be used for generation, evaluation, or both — we provide a detailed discussion of these methods in Section 5.

4.2.1 Understand

The first activity in the framework is to **understand** the problem domain and target users. The motivation for this activity is: *to gather, observe, and research available information to find the needs of the user.* The outcomes of this activity are commonly referred to as design requirements [16, 23, 30, 37, 43, 51]. We specifically characterize requirements into one of three classes: *opportunities*, *constraints*, and *considerations*. **Opportunities** encompass the data and task abstraction outcomes that have a potential to impact the work and field of the target users. They may also include higher level themes discovered through the domain characterization, such as workflow inefficiencies. **Constraints** are rigid limitations from the project itself that the visualization designer must work with, such as tight deadlines or display limitations. **Considerations**, however, are a looser, more flexible form of constraints that a designer should strive to consider, such as the importance of aesthetics or usability. Together, these three classes of outcomes for the understand activity play a crucial role in all following activities, and they often get reconsidered, adjusted, and prioritized throughout the design process.

Outcomes for the understand activity fall into the outer two levels of the nested model, the domain characterization and abstraction levels. These outcomes 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 — this is referred to as situation blocks in the nested blocks and guidelines model [44]. Furthermore, the outcomes also include contextual information about the project itself, such as real-world project considerations *i.e.* time, budget, expertise, *etc.* Outcomes 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.

4.2.2 Ideate

The second activity in the framework is the **ideate** activity, which has the motivation: *to generate good ideas for supporting the understand outcomes*. The outcome of the ideate activity is a set of ideas that are most often externalized in a variety of forms, from sketches to wireframes to even low-fidelity prototypes. It is important to note that the act of externalizing an idea onto some medium often results in the generation of additional ideas as they become more concrete [19].

These 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 support the *understand* outcomes. At the technique level the ideas 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 in the next section. Thus, the ideate activity supports very broad exploration of the high-level design space for supporting a specific problem, leaving more detailed design decisions to later activities. Ideation is commonly considered as a separate activity in the design community [6, 10, 36, 54, 63, 65, 67], and this separation highlights the different kinds of design decisions made within the visualization design process.

4.2.3 Make

The **make** activity is the third activity in the framework. This activity's motivation is: *to concretize ideas into tangible prototypes.* The outcome from the make activity is a set of prototypes, where prototypes are "approximations of a product along some dimensions of interest" [25]. These prototypes test aspects of design decisions made at the inner two levels of the nested model, the technique and algorithm levels. These prototypes explicitly explore the design decisions related to actualizing a specific visualization or interaction technique. While low-fidelity prototypes can exist in the ideate activity, prototypes for the make activity are usually 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. This activity is not just implementing a given design; rather, the activity, including development or coding, also involves critical visualization design decisions [61].

We note that most engineering design process models couple the ideate and make activities together — we believe that these two activities have related, but different, motivations and hence outcomes for visualization design, making their separation important for a careful consideration of all types of visualization design decisions. 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.

4.2.4 Deploy

The fourth activity in the framework is the **deploy** activity, with the motivation: *to bring a prototype into effective action in a real-world setting in order to support the target users' work and goals.* The overall outcome of this activity is a usable visualization system. This activity and its methods are largely dominated by those from software engineering, with the focus of supporting target users utilizing the tool. Thus, the outcomes of the deploy stage touch on decisions made at the algorithm level of the nested model, as well as other decisions that are not necessarily about the visualization design itself, such as integration with existing software, databases, *etc.* This activity is the ultimate goal of problem-driven visualization design since it supports real-world users in their own work environments.

4.3 Flow of the Framework

In our experience, a visualization design process never seems to progress cleanly through a set of designated stages; this fact motivated our synthesis of the design activity framework, which can be pieced together in many different ways to best suit the needs of a project. This aligns with creative process models from the design community

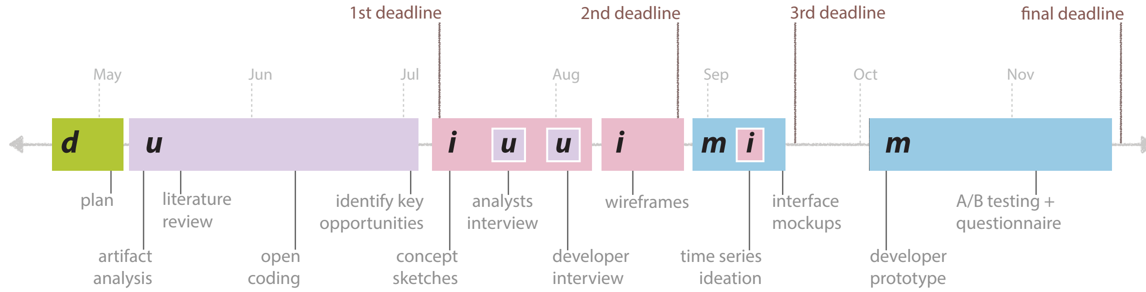

Fig. 3. This timeline provides a concise overview of our redesign project. Key design activities are located in the middle, highlighting both backwards movement and activities nesting within each other when team members worked separately. Our redesign project contained several key time constraints, or deadlines, listed at the top. Towards the bottom, we highlight numerous methods and outcomes utilized throughout our project.

which already emphasize that design is messy, iterative, and multilinear [13, 33, 59]. These creative process models advocate that there is no one right way in which to engage in the activities of a framework.

There are two basic principles for the design activity framework when it comes to the flow of the design process. First, the activities are ordered when moving forward: understand, ideate, make, deploy. While a project can start with any activity, as with our cybersecurity redesign project which started from a tool in the deploy activity, forward movement must happen in an ordered fashion, even if the methods used are very narrow and informal. Backward movement, however, can move to any previous 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 as two activities that occur concurrently. Taken together, these two movement principles support both iteration and multilinearity.

Other visualization process models are also characterized in similar ways, supporting ordered forward movement with iteration [16, 32, 54, 60, 66], and others that suggest an overlap between stages, such as the nine-stage framework [54] and the international standard for human-centred design activities [16]. These models, however, are often represented linearly or cyclically and imply the need to start at the beginning of the process, making it difficult, for example, to capture the process of a visualization redesign project.

To further illustrate the two movement principles we provide a timeline for our redesign project in Figure 3, where activities are represented as colored boxes — we discuss our process in more detail in Section 4.4 This timeline shows the flow of the project through multiple activities, including nested activities and both forward and backward movement. The timeline is annotated with many of the methods we used and the outcomes we developed in our redesign process.

Other researchers have shown the feasibility and usefulness of a design process timeline as an effective way to communicate a design process [43, 64], to foster collaboration [5], and to highlight some aspects of the multilinear nature of a design process [1, 68]. Communication of the design process is important for not only understanding and evaluating the visualization research process itself, but also to support replicability of problem-driven work. Visualization models like the nested model [45] are now widely used to communicate design decisions made over the course of a visualization process, and we advocate for the design activity framework to structure communication of a visualization design process in a similar way.

4.4 Redesign Project: an Example

To provide context and ground the design activity framework in a real-world example, here we will walk through our redesign project and describe our design process using the framework. This section focuses on each activity of our redesign, presented in rough, chronological order, and focusing on the methods we used and the outcomes we achieved. We will refer to methods listed in Table 1 using a prefix, such as the method of controlled experiments (*M-28*). We note that this description is a simplification of our process for the purposes of illustration — Figure 3 presents further details.

4.4.1 Deploy — Redesign Project

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 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 outcome, 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 walkthrough 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. We have not yet returned to the deploy activity as the RVS company is still involved with major redesigns of the tool.

4.4.2 Understand — Redesign Project

The field of cybersecurity analysis has many types of users, from those within companies that 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 [15, 18, 21] across these different facets — 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 forty articles from several 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 as the best representative samples with the highest impact for forming our domain characterization.

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. As individuals we tagged information broadly, and we then adjusted these tags 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 outcomes, which pointed to a number of unmet needs and opportunities for visualization research. Some of these outcomes included *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.

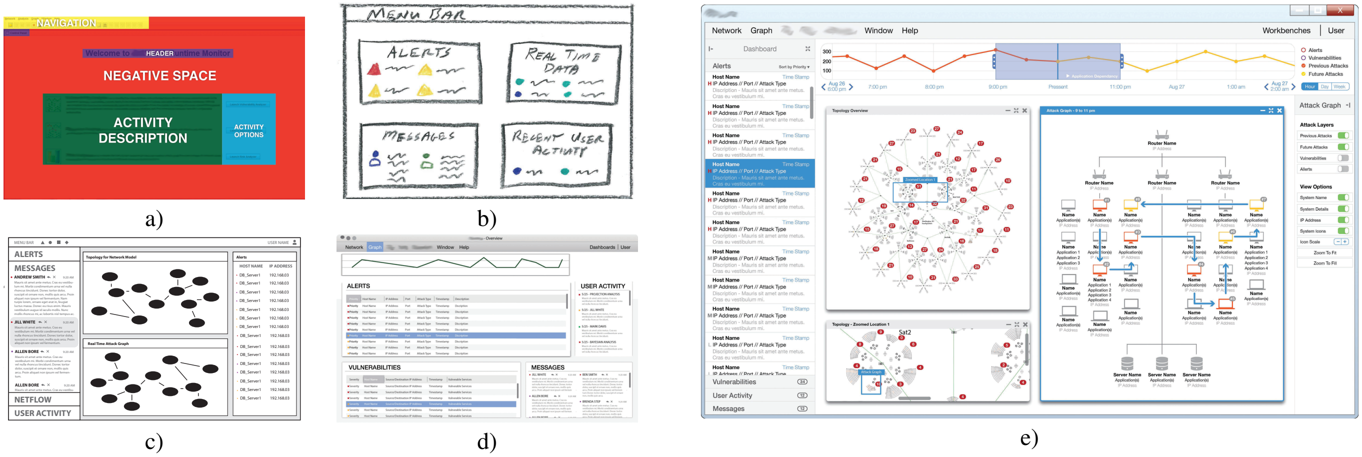

Fig. 4. We provide an overview of the outcomes for our redesign project, starting from our a) software analysis which resulted in b) initial concept sketches and c) wireframes. As we focused on more of the details, we moved into the make activity with d) laying out interface components and e) designing a fully-detailed revised interface. These outcomes are provided at full-resolution in the Supplemental Materials.

We revisited the RVS system with these 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 Figure 4a. By examining the workflow supported by RVS we identified which opportunities the tool already supported, which aspects of the tool could be improved, and evaluated these against our initial list of opportunities. These findings were combined with our project-specific *constraints* and *considerations*, which included: four months of the design team's time; one month of a developer's time; and existing visual conventions in the field such as highlighting critical alerts in red.

Lastly, we conducted a series of semi-structured interviews with different stakeholders to identify needs and aspirations (*M-51*), where these stakeholders included a developer that works on RVS and several cybersecurity analysts at the University of Utah. Based on this feedback, we met as a design team and informally evaluated and filtered the list of opportunities by reaching a group consensus on those we felt best met 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 redesign were: 1) usability; 2) workflow improvements; 3) desirability; and 4) temporal data representation. We also developed a more low-level list of all outcomes, which included a detailed data and task abstraction.

4.4.3 Ideate — Redesign Project

After our design team had identified the specific design opportunities, constraints, and considerations for our redesign, we were ready to come up with ideas. This activity took up several months worth of time 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 opportunity, as illustrated in Figure 4b — 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 design 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 sub-groups and supporting details-on-demand in order to allow the visualization to scale to larger dataset. 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 very critical due to the quantity and spread of real-world cybersecurity data.

While the concept sketches proved to be useful in exploring different ideas, 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 Figure 4c, to mimic the look and feel of a real tool. Again, we evaluated these wireframes very informally, internally as a design team and with different members of the RVS company, to check that our redesigns 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 analysts. These sketches and wireframes formed the outcomes of our ideate activity.

4.4.4 Make — Redesign Project

The make activity was conducted in part by our design 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 Figure 4d. 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 of our design ideas into an idealized, revised interface, as illustrated in Figure 4e. The purpose behind this method was to envision what RVS *could* be even though the software implementation was beyond the scope of what the developers could achieve given the constraint of one-month of their time.

After we finalized these detailed and revised mockups, the RVS development team focused on implementing these concepts into the existing software. We note that the distinction here between the design team and development team is somewhat unique to our redesign project — most often in visualization design these two groups of people are the same. As a result of this implementation process, the development team created a software prototype (*M-67*), which 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 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 outcome: a new prototype of RVS.

5 Exemplar Methods

As a secondary contribution to this work, we present a list of exemplar methods that can be used 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 shortened list of 40 methods in Table 1, and a more extensive list of 100 methods in the Supplemental Materials. We shortened to these 40 methods by picking those which were mentioned within the framework and redesign project, along with both commonly used and potentially novel or interesting methods for visualization design.

Each method is characterized as to which activities of the design activity framework 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. The methods are also categorized as being either generative (***g***), evaluative (***e***), or both in nature. There are several methods, *e.g.* graffiti walls (*M-43*), interviewing (*M-51*), and observation (*M-58*), which have more complex characterizations than presented in this table; please see the Supplemental Materials for a more complete and detailed characterization. Some methods are also marked 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 problem-driven visualization work, it is worth noting that many of the listed methods involve 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 towards understanding the large space of actions a designer can take throughout the design process. Our goal in creating this list of methods is two-fold: first, the list serves 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 list contains many methods that are not commonly, if at all, found in the visualization literature, therefore providing new methods to potentially enhance the visualization 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*) [23].

6 Results of the Redesign Project

A tertiary contribution of this work are several *understand* outcomes which consist of new considerations and rich opportunities for cybersecurity visualization design — these types of domain characterization and abstraction contributions have been advocated by others [44, 45]. These outcomes were identified through both the open coding and interviews, as explained in Section 4.4.2. The new design considerations we introduce are a new data abstraction and a vital feedback loop for cybersecurity analysis. These considerations stem from the original data hierarchy model presented in a cognitive task analysis by D'Amico and Whitley, which illustrates how analysts process, filter, sort, and select important information out of the data and transform that raw data into situational awareness for cybersecurity [15] (see Figure 1 in their paper). In their work, analysts start with raw data, or network packets, and filter data from alerts to events and eventually to **rule sets**, or high-level descriptions of a collection of multiple incidents. Based on our research and interviews, however, there are several other key types of data that analysts use: internal or "in-house" data such as firewall rules or incident reports; external data such as hacker websites and mailing lists; and processed output data, *i.e.* incident reports. Additionally, rule sets often become detection signatures to automatically filter raw data, creating a feedback loop in the analysts' workflow. This idea of a feedback loop is not new; we saw them in several different task diagrams for cybersecurity analysts [15, 18]. Ideally, incident reports and other processed data by analysts would be

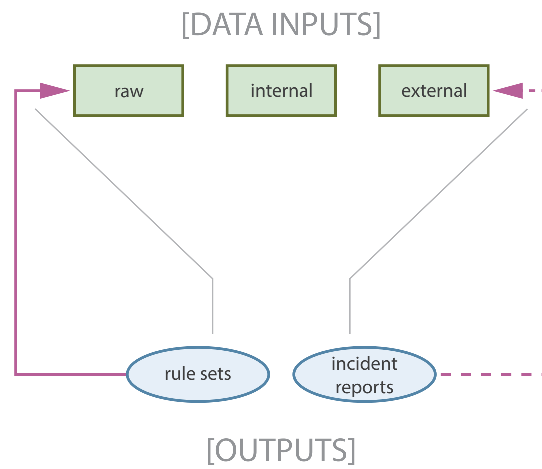

Fig. 5. We present a simplified extension of a data hierarchy model for how cybersecurity analysts transform raw data into cybersecurity situational awareness [15]. This simplified model focuses on a novel data abstraction: new data types such as internal data which is kept "in-house", external data which may be shared or obtained from websites, social media, phone calls, *etc.*, as well as processed data like incident reports. There exists a data feedback loop from rule sets back to raw data, and, ideally, such a loop would exist for incident reports, but this is an active research problem in this domain.

incorporated into this feedback loop, but currently this is difficult to do since incident reports are disparate and not easily searchable [15]. We present a new data abstraction for the data hierarchy model in Figure 5, highlighting the additional data types as well as an idealized feedback loop from incident reports to internal data, and external data if shared. This model established several key considerations for our redesign process, and we suspect it could similarly aid future cybersecurity visualization tools to better address the needs of analysts.

Based on our literature review, interviews, and new data abstraction we identify several open opportunities for visualization in the field of cybersecurity: provenance; data type handling; and data hierarchy continuity. For provenance, visualization researchers could focus on providing tools for analysts to track and document their findings while using a visualization tool to explore their data, ultimately for the purpose of automatically generating reports and sharing their analysis process [15, 18]. The opportunity of data type handling would seek to visualize a broader variety of data types, such as a variety of external data coming from websites, social media, images, *etc.* [21]. Lastly, related to the data hierarchy model, the continuity opportunity points to finding ways to scale visualizations to the current, massive datasets while retaining the ability to go back to the raw data [15, 18, 21].

7 Discussion

Throughout our redesign project we worked closely as a design team composed of designers, a psychologist, and visualization experts. Our different perspectives and experience led to a richer and more informed design process. When working together as a team 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 [31]. This allowed us to synthesize the ideas and perspectives on the design process from several different fields into the design activity framework.

As a process model, the main 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 like the nine-stage framework [54]. Specifically, the design activity framework also includes motivations, outcomes, and explicit ties

Table 1. This table presents several exemplar methods and where they fit within the framework; we provide a more extensive list of 100 methods in the Supplemental Materials. We coded each method into one or more of the four design activities: understand (***u***), ideate (***i***), make (***m***), and deploy (***d***). Additionally, we tagged whether each method was largely generative (***g***) or evaluative (***e***), or both. Lastly, we tagged the methods we have seen commonly reported within the visualization community (***v***) and also present succinct definitions for each method.

#	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" [42]
7	appearance modeling		•	•	•	•			"refined model of a new idea that emphasizes visual styling" [50]
8	artifact analysis	•				•	•	•	"systematic examination of the material, aesthetic, and interactive qualities of objects" [42]
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" [50]
13	buy a feature	•	•	•			•		"express trade-off decisions. . . . ask [participants] to purchase features . . . . encourage them to [justify decisions]" [50]
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]" [57]
22	concept map		•				•		"sense-making tool that connects a large number of ideas, objects, and events as they relate to a certain domain" [42]
23	concept sketching		•			•		•	"convert ideas into concrete forms that are easier to understand, discuss, evaluate, and communicate" [33]
26	constraint removal	•	•			•		•	"barriers [are] transformed into a positive resource through which to create new ideas" [23]
27	contextual inquiry	•				•	•	•	"go where the customer works, observe the customer as he or she works, and talk to the customer" [4]
28	controlled experiment	•		•	•	•	•	•	"help us to answer questions and identify casual relationships" [35] & "widely used approach to evaluating interfaces and styles of interaction, and to understanding cognition in the context of interactions with systems" [8]
29	creative matrix		•			•			"[spark] new ideas at the intersections of distinct categories. . . . encourage the teams to fill every cell of the grid" [50]
35	example exposure		•	•		•		•	"excite ideas by exposing the subject to a solution for the same problem" [26]
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" [3]
42	frame of reference shifting		•			•			"change how objectives and requirements are being viewed, perceived, and interpreted" [26]
43	grafitti walls	•	•	•	•	•	•		"open canvas on which participants can freely offer their written or visual comments, directly in the context of use" [42]
44	heuristic evaluation			•	•		•	•	"assess an interface against a set of agreed-upon best practices, or usability 'rules of thumb"' [42]
48	importance/difficulty matrix	•	•	•			•		"plotting items by relative importance and difficulty . . . look for related groupings, and set priorities" [50]
49	incubation		•			•			"add programmed delay to allow sub-conscious processing to take place" [26]
51	interviewing	•	•	•	•	•	•	•	"direct contact with participants, [collect] personal accounts of experience, opinions, attitudes, and perceptions" [42]
53	literature review	•				•	•	•	"distill information from published sources, capturing the essence of previous research" [42]
54	love/breakup letters	•	•	•	•	•	•		"personal letter written to a product. . . [to reveal] profound insights about what people value and expect" [42]
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 [42]
58	observation	•	•	•	•	•	•	•	"attentive looking and systematic recording of phenomena: including people, artifacts, environments, events, behaviors and interactions" [42]
61	paper prototyping		•	•		•		•	"create a paper-based simulation of an interface to test interaction with a user" [39]
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" [17]
63	personas	•					•		"consolidate archetypal descriptions of user behavior patterns into representative profiles, to humanize design" [42]
67	prototyping		•	•		•		•	"tangible creation of artifacts at various levels of resolution, for development and testing of ideas within design" [42]
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" [42]
72	role-playing	•	•		•	•	•		"acting the role of the user in realistic scenarios can . . . highlight challenges, presenting opportunities" [42]
73	rose-thorn-bud	•	•	•			•		"identifying things as positive, negative, or having potential" & tag outcomes as rose, thorn, or bud, accordingly [50]
75	sample data				•	•		•	"provide real data and tasks . . . . illustrating [tools] with convincing examples using real data" [49]
80	speed dating		•	•			•		"compare multiple design concepts in quick succession" [42]
81	stakeholder feedback	•	•	•	•		•	•	"letting [experts] "play" with the system and / or observe typical system features" [34]
85	suspended judgement		•			•			"postpone premature decisions or dismissing an idea" & "generate as many ideas as possible" [26]
86	task analysis	•		•	•	•	•	•	"breaks down the constituent elements of a users work flow, including actions and interactions" [42]
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" [28]
89	thought experiment	•	•	•			•		"think about research questions as if it were possible to test them in true experiments" [3]
97	weighted matrix	•	•	•			•		"[rank] design opportunities against key success criteria" & "identify and prioritize. . . opportunities" [42]
98	wireframing		•	•		•		•	"schematic diagramming: an outline of the structure and essential components of a system" [50]

to the nested model in order to help guide a designer through the visualization design process. The motivation enables a designer to determine which activity they are currently performing, which then allows the designer to identify potential methods, clarify outcomes, and place decisions within 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 — this extension presents interesting, future work.

The framework supports a large amount of flexibility by enabling and emphasizing a workflow that includes both the nesting of activities and activities occurring in parallel. As shown in Figure 3, the design activity framework can represent a process where many activities are pieced together in different ways according to the motivation of the project at any given time. We feel that 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 [14], nor do they even agree that any such model could capture the "black box" of design [19]. Furthermore, considering design as a wicked problem [7,20,22,68] it can be challenging to know where to go next, when to stop, and what makes an effective design [7]. These challenges exist for many design process models, including the design activity framework, pointing to the opportunity for further investigation.

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 Figure 2. We were able to identify several corner cases where outcomes of a process could begin to overlap into an additional level of the nested model. Furthermore, the framework does not include a planning activity which is present in other process models [16,54]. While 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 [54] 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. With respect to the design activity framework, further research could extend the framework such as more finely defining or breaking apart specific activities, adding new activities, or making the connection to a different design decision model.

8 Conclusions and Future Work

We present a novel visualization design process model, the design activity framework, which begins to address the messy, iterative, and multilinear process of real-world visualization design. The framework explicitly connects the actions a visualization designer takes with the visualization design decisions that are made by directly connecting each design activity with the corresponding levels of the nested model [45]. In addition to the framework, we provide a series of exemplar methods within that framework that visualization designers can utilize, including well-known methods from the visualization literature as well as methods from other communities which are less common. Furthermore, we provide several key opportunities and considerations for the design of future tools in the domain of cybersecurity visualization. The overall goal of this work is to help guide visualization designers through the design process, encourage visualization designers to consider new design methods for generation or evaluation, and to assist the compact communication of a design process, as in Figure 3. We consider all design models to be a work-in-progress, and the design activity framework is by no means excluded. Further validating this framework against additional visualization design projects remains an interesting avenue for future work.

There are a number of additional open questions for future work. For example, we established this framework from a problem-driven methodology, and it would be useful to rigorously, but cautiously, validate the use of the framework for a technique-driven approach. 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. There are also a series of challenges yet to be addressed by most visualization process models: 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 to further explore the role of design for visualization.

Acknowledgments

The authors wish to thank Michael Sedlmair, Mike Kirby, Alex Bigelow, Ethan Kerzner, Nina McCurdy, Sam Quinan, and Kris Zygmunt for their feedback on this work. We thank Matthew Parkin for his involvement in the redesign project, producing detailed designs shown in Figure 4. This work is sponsered in part by the Air Force Research Laboratory and the DARPA XDATA program, and by the U.S. Army Research Office under a prime contract issued to Intelligent Automation, Inc. The content of the information does not necessarily reflect the position or the policy of the Government or Intelligent Automation, Inc., and no official endorsement should be inferred.

References

[1] C. Atman, K. Deibel, and J. Borgford-Parnell. The process of engineering design: A comparison of three representations. *International Conference on Engineering Design*, 2009.

[2] A. Baker and A. van der Hoek. Ideas, subjects, and cycles as lenses for understanding the software design process. *Design Studies*, 31(6):590–613, Nov. 2010.

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

[4] H. Beyer and K. Holtzblatt. *Contextual Design: Defining Customer-Centered Systems*. Elsevier, 1997.

[5] M. Bohøj, N. G. Borchorst, N. O. Bouvin, S. Bødker, and P.-O. Zander. Timeline collaboration. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 523. ACM, Apr. 2010.

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

[7] R. Buchanan. Wicked problems in design thinking. *Design Issues*, 1992.

[8] P. Cairns and A. Cox. *Research Methods for Human-Computer Interaction*. Cambridge University Press, 2008.

[9] J. Cao, Y. Riche, S. Wiedenbeck, M. Burnett, and V. Grigoreanu. End-user mashup programming: through the design lens. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 1009. ACM, Apr. 2010.

[10] M. Cardella, C. Atman, J. Turns, and R. Adams. Students with differing design processes as freshmen: Case studies on change. *International Journal of Engineering Education*, 24(2):246, 2008.

[11] S. Carpendale. Evaluating information visualizations. *Information Visualization*, 2008.

[12] H. Christiaans and R. A. Almendra. Accessing decision-making in software design. *Design Studies*, 31(6):641–662, Nov. 2010.

[13] M. da Gandra, M. P. Hea, M. van Neck, and M. Hea. (In)forming the Information Design Student. *Parsons Journal for Information Mapping*, 2013.

[14] M. da Gandra and M. van Neck. *InformForm, Information Design: in theory, an informed practice*. 2012.

[15] A. D'Amico and K. Whitley. The real work of computer network defense analysts. *Visualization for Cyber Security*, pages 19–37, 2008.

[16] I. DIS. ISO 9241-210: 2009. Ergonomics of human system interaction-Part 210: Human-centred design for interactive systems (formerly known as 13407). *International Organization for Standardization (ISO). Switzerland*, 2010.

[17] 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. *ACM Transactions on Computer-Human Interaction*, 17(4):1–24, Dec. 2010.

[18] R. F. Erbacher, D. A. Frincke, P. Chung Wong, S. Moody, and G. Fink. A multi-phase network situational awareness cognitive task analysis. *Information Visualization*, 9(3):204–219, Jan. 2010.

[19] D. Fallman. Design-oriented human-computer interaction. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, 2003.

[20] R. Farrell and C. Hooker. Design, science and wicked problems. *Design Studies*, 34(6):681–705, Nov. 2013.

[21] G. A. Fink, C. L. North, A. Endert, and S. J. Rose. Visualizing cyber security: Usable workspaces. In *Visualization for Cyber Security*. IEEE, Oct. 2009.

[22] W. Gaver. What should we expect from research through design? In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 937. ACM, May 2012.

[23] 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 Transactions on Visualization and Computer Graphics*, 19(12):2516–2525, Aug. 2013.

[24] B. Hanington. Methods in the making: A perspective on the state of human research in design. *Design Issues*, 19(4):9–18, 2003.

[25] B. Hartmann, S. Klemmer, and M. Bernstein. Reflective physical prototyping through integrated design, test, and analysis. In *Proceedings of the ACM Symposium on User Interface Software and Technology*, pages 299–308. ACM, 2006.

[26] N. Hernandez, J. Shah, and S. Smith. Understanding design ideation mechanisms through multilevel aligned empirical studies. *Design Studies*, 31(4):382–410, 2010.

[27] T. Howard, S. Culley, and E. Dekoninck. Describing the creative design process by the integration of engineering design and cognitive psychology literature. *Design Studies*, 29(2):160–180, Mar. 2008.

[28] H. Hutchinson, W. Mackay, B. Westerlund, B. B. Bederson, A. Druin, C. Plaisant, M. Beaudouin-Lafon, S. Conversy, H. Evans, and H. Hansen. Technology probes: inspiring design for and with families. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 17–24. ACM, 2003.

[29] T. Isenberg, P. Isenberg, J. Chen, M. Sedlmair, and T. Moller. A systematic review on the practice of evaluating visualization. *IEEE Transactions on Visualization and Computer Graphics*, 19(12):2818–2827, 2013.

[30] S. Jones, P. Lynch, N. Maiden, and S. Lindstaedt. Use and influence of creative ideas and requirements for a work-integrated learning system. In *IEEE International Requirements Engineering*, pages 289–294. IEEE, 2008.

[31] R. M. Kirby and M. Meyer. Visualization collaborations: What works and why. *IEEE Comput. Graph. Appl.*, 33(6):82–88, Nov. 2013.

[32] 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. *Information Visualization*, pages 90–95, 2011.

[33] V. Kumar. *101 Design Methods: A Structured Approach for Driving Innovation in Your Organization*. 2012.

[34] H. Lam, E. Bertini, P. Isenberg, C. Plaisant, and S. Carpendale. Empirical studies in information visualization: Seven scenarios. *IEEE Transactions on Visualization and Computer Graphics*, 18(9):1520–1536, Nov. 2011.

[35] D. J. Lazar, D. J. H. Feng, and D. H. Hochheiser. *Research Methods in Human-Computer Interaction*. John Wiley & Sons, 2010.

[36] W. Lidwell and K. Holden. *Universal Principles of Design*. Rockport Publishers, 2010.

[37] D. Lloyd and J. Dykes. Human-centered approaches in geovisualization design: Investigating multiple methods through long-term case study. *IEEE Transactions on Visualization and Computer Graphics*, 17(12):2498–2507, 2011.

[38] J. Löwgren. Applying design methodology to software development. In *Proceedings of the Conference on Designing Interactive Systems*, pages 87–95. ACM, Aug. 1995.

[39] M. Maguire. Methods to support human-centred design. *International Journal of Human-Computer Studies*, 55(4):587–634, 2001.

[40] N. Maiden, S. Manning, S. Robertson, and J. Greenwood. Integrating creativity workshops into structured requirements processes. In *Proceedings of the Conference on Designing Interactive Systems*, pages 113–122. ACM, 2004.

[41] N. Maiden, C. Ncube, and S. Robertson. Can requirements be creative? experiences with an enhanced air space management system. *International Conference on Software Engineering*, pages 632–641, 2007.

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

[43] P. McLachlan, T. Munzner, E. Koutsofios, and S. North. LiveRAC: Interactive Visual Exploration of System Management Time-Series Data. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 1483. ACM, Apr. 2008.

[44] M. Meyer, M. Sedlmair, P. S. Quinan, and T. Munzner. The nested blocks and guidelines model. *Information Visualization*, 2013.

[45] T. Munzner. A nested model for visualization design and validation. *IEEE Transactions on Visualization and Computer Graphics*, 15(6):921–8, 2009.

[46] H. G. Nelson and E. Stolterman. *The Design Way: Intentional Change in an Unpredictable World : Foundations and Fundamentals of Design Competence*. Educational Technology, 2003.

[47] J. S. Olson and T. P. Moran. Mapping the method muddle: Guidance in using methods for user interface design. In M. Rudisill, C. Lewis, P. G. Polson, and T. D. McKay, editors, *Human-Computer Interface Design: Success Stories, Emerging Methods, and Real-World Context*, pages 269–300. Morgan Kaufmann Publishers Inc., June 1996.

[48] B. Onarheim and S. Wiltschnig. Opening and constraining: constraints and their role in creative processes. In *Proceedings of the DESIRE Network Conference on Creativity and Innovation in Design*, pages 83–89. Desire Network, Aug. 2010.

[49] C. Plaisant. The challenge of information visualization evaluation. In *Proceedings of the Conference on Advanced Visual Interfaces*, pages 109–116. ACM, May 2004.

[50] H. B. Review. Vision Statement: A Taxonomy of Innovation. `http://hbr.org/2014/01/a-taxonomy-of-innovation/ar/1`, 2014. Accessed: 2014-02-20.

[51] W. W. Royce. Managing the development of large software systems. In *Proceedings of IEEE WESCON*, volume 26. Los Angeles, 1970.

[52] J. C. Savage, C. J. Moore, J. C. Miles, and C. Miles. The interaction of time and cost constraints on the design process. *Design Studies*, 19(2):217–233, Apr. 1998.

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

[54] M. Sedlmair, M. Meyer, and T. Munzner. Design study methodology: Reflections from the trenches and the stacks. *IEEE Transactions on Visualization and Computer Graphics*, 18(12):2431–2440, 2012.

[55] B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proceedings of the AVI workshop on Beyond Time and Errors Novel Evaluation Methods for Information Visualization (BELIV)*, pages 1–7. ACM, 2006.

[56] C. M. Snider, S. J. Culley, and E. A. Dekoninck. Analysing creative behaviour in the later stage design process. *Design Studies*, 34(5):543–574, Sept. 2013.

[57] A. Strauss and J. Corbin. *Basics of Qualitative Research: Grounded Theory Procedures and Techniques*. 1990.

[58] A. Tang, A. Aleti, J. Burge, and H. van Vliet. What makes software design effective? *Design Studies*, 31(6):614–640, Nov. 2010.

[59] R. Teal. Developing a (non-linear) practice of design thinking. *International Journal of Art & Design Education*, 29(3):294–302, Oct. 2010.

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

[61] A. Vande Moere and H. Purchase. On the role of design in information visualization. *Information Visualization*, 10(4):356–371, Sept. 2011.

[62] R. H. von Alan, S. T. March, J. Park, and S. Ram. Design science in information systems research. *Management Information Systems Quarterly*, 28(1):75–105, 2004.

[63] K. Vredenburg, J. Mao, P. Smith, and T. Carey. A survey of user-centered design practice. *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, pages 471–478, 2002.

[64] D. B. Walz, J. J. Elam, and B. Curtis. Inside a software design team: knowledge acquisition, sharing, and integration. *Communications of the ACM*, 36(10):63–77, Oct. 1993.

[65] C. Ware. *Visual Thinking: for Design*. Morgan Kaufmann, 2010.

[66] I. Wassink, O. Kulyk, and B. van Dijk. Applying a user-centered approach to interactive visualisation design. In *Trends in Interactive Visualization*, pages 175–199. Springer, 2009.

[67] A. Wodehouse, M. Ing, and W. Ion. The integration of information & ideas: Creating linkages through a novel concept design method. *Parsons Journal for Information Mapping*, 2010.

[68] T. V. Wolf, J. A. Rode, J. Sussman, and W. A. Kellogg. Dispelling "design" as the black art of CHI. In *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, page 521. ACM, Apr. 2006.

[69] J.-C. Wu, C.-C. Chen, and H.-C. Chen. Comparison of designer's design thinking modes in digital and traditional sketches. *Design & Technology Education*, 17(3):37–48, Nov. 2012.