Author pre-print of forthcoming work in Chen, M., Szafir, D. A.; Borgo, R.; Padilla, L.; Edwards, D. J.; Fisher, B. (Eds.): Visualization Psychology. Springer. 2022.

Design Cognition in Data Visualization

Paul Parsons

Abstract In this chapter I introduce the field of design cognition and its relevance to data visualization. I outline two historically dominant paradigms of design cognition. The first, promoted by Herbert Simon in the 1970s, is the rational problem solving paradigm which is based on information processing psychology and problem solving theory. The second, promoted by Donald Schön in the 1980s, is the reflective practice paradigm which is based on constructivist philosophy and situated views of cognition. I describe some of their strengths and weakness, and some attempts to reconcile their differences. Underlying philosophical issues pertaining to cognition and epistemology are briefly discussed. I then examine implications of these two paradigms for four data visualization topics: defining, automating, modeling, and teaching data visualization design. In discussing these topics, possible avenues of future research are proposed.

1 Introduction

How do designers formulate and solve design problems? What kinds of cognitive processes do they rely on while doing so? These are the types of questions asked by researchers studying design cognition. Rather than highlighting the methods, tools, or outcomes of designers, studies in design cognition investigate how and why designers think in the ways they do while designing. Design cognition has been studied across a wide variety of domains, including engineering [3], architecture [28], computer science [6], instructional design [46], and graphic design [62]. Across these disciplines, many aspects of cognition in design have been investigated, including, among others, episodic memory [26], fixation [49], chunking [32], bias [13], abductive reasoning [8], analogical reasoning [65], metacognitive monitoring and control [4], and recall [10].

Paul Parsons
Purdue University, West Lafayette, IN, USA, e-mail: parsonsp@purdue.edu

Much of the research on design cognition is concerned with how designers navigate the complexity and uncertainty of real-world design situations [5, 59, 61]. A number of core strategies have been identified through empirical investigation, including conjecture-based problem formulation, problem-solution co-evolution, analogical reasoning, mental simulation, and fixated solution generation [4, 18]. Many of the cognitive processes that are relevant for studying the use of visualizations are also important for understanding their design. For instance, studies have shown that designers rely on chunking to ideate effectively [32], employ abductive reasoning during concept selection [14], are influenced by color in ways that bias their thinking while sketching [13], and struggle with fixation while generating ideas [9]. Given the considerable evidence of these issues impacting design across numerous disciplines, it is likely that they are also important for understanding visualization design.

In terms of methods for investigating design cognition, previous research has heavily relied on "protocol studies" to elicit cognitive processes [12, 18]. This method, which is already well-known to visualization researchers doing human-subjects studies, involves asking designers to 'think-aloud' while doing a design activity. These studies generate verbal protocols that can be transcribed and analyzed with the goal of uncovering aspects of thinking and reasoning. This kind of approach can be taken with individual designers who work alone on design problems, or with teams of designers working together. Team-based protocols can be used to elicit socio-cognitive facets of collaborative design cognition [2].

Design cognition can also be studied in both controlled settings, such as a lab or workshop, and in less controlled settings, such in designers' everyday work environments. Studies in controlled settings can be beneficial, as they allow common design tasks to be given to participants and allow for the control of variables, including time spent, access to resources, and so on. Although empirical lab studies are commonly employed in design research (e.g., [7, 23]), they differ from realistic design contexts in a number of ways. For instance, lab studies may exclude factors that shape design work in commercial settings, including the effects of organizational culture, project timescales, project management, and workload. Lab studies may also present participants with relatively simple problems over short time periods, which are not often representative of real-world design tasks. As is often the case in experimental research, there is the risk of reducing both ecological and external validity [7]. For these reasons, it is beneficial to conduct studies in both controlled "lab" settings and "in the wild" of real-world practice. Studies can employ a range of methods, including protocol analysis, semi-structured interviews, diary studies, contextual observations, and co-design workshops.

2 Why Study Visualization Design Cognition?

It is easy to appreciate why the cognition of users is an important area of inquiry for data visualization. The many different ways of visualizing data and creating interactive interfaces influence how users interpret, understand, and act. Topics relating to visual marks and channels, color, mental models, uncertainty, biases, sensemaking, and others have received significant attention in the visualization literature. Indeed, the vast majority of literature at the intersection of visualization and psychology is focused on users, whereas the cognition of designers is largely an unexplored topic. The reason for this is unclear, especially since design cognition has been a research topic in multiple design disciplines for decades.

One reason for a lack of inquiry into design cognition may stem from an assumption that is commonly held within scientific fields. The assumption is that design consists largely of the *application* of scientific knowledge to instrumental problems [5, 52]. From this standpoint, the important forms of design knowledge include laws, principles, guidelines, patterns, and other objective forms of knowledge that can be codified. The prototypical role of the designer is to know these and apply them to various design situations, sometimes with the help of models and frameworks. For example, there may exist design guidelines regarding data types, visual marks and channels, color, visualization and interaction techniques, evaluation strategies, and other similar forms of knowledge. From this "application" perspective, designers need to first know about these things, then figure out how to apply them to particular problems. If they have the right training, experience, and access to guidelines when they need them, they can apply this existing knowledge in useful ways.

While this "application" view may be appealing, it has largely abandoned by design scholars, as it has not held up well to empirical scrutiny [11, 27, 29]. While objective, scientific knowledge certainly plays a role in design, and in many cases may be necessary, it is not sufficient for good design [61]. Rather, designers rely on a host of personal and situated factors, along with more formal types of knowledge, to engage with the complexity of design practice [44]. Buchanan [5] articulates how widespread this assumption has been, noting that "each of the sciences that have come into contact with design has tended to regard design as an 'applied' version of its own knowledge", emphasizing the mistake of viewing design as simply a "practical demonstration" of scientific findings. Thus, even if a robust program of research at the intersection of psychology and visualization is developed, if its scope is limited to users only—and especially if design is viewed merely as an application of research findings—we will likely fail to understand and influence design practice effectively.

3 Two Paradigms of Design Cognition

Two dominant paradigms have historically been used to describe the cognitive nature of design. The first was articulated by Herbert Simon in the 1970s and was inspired by information processing psychology and theories of problem solving. The second was articulated by Donald Schön in the 1980s and was inspired by constructivist philosophy and situated views of cognition and professional practice. These two paradigms present different perspectives on the cognitive activities involved in designing, and have different implications for how design is studied and taught. In what follows, I will refer to these views often as the "rational problem solving" view and the "reflective practice" view. These two paradigms will be described next, followed by some of their implications in general and for data visualization in particular.

3.1 Design as Rational Problem Solving

The work of Herbert Simon has been highly influential in the study of design cognition [51], despite there being only a few references directly discussing design cognition among his nearly 1000 articles [66]. In particular, his book *The Sciences of the Artificial* [59], and his article "The Structure of Ill-Structured Problems" [60], serve as the foundation of his work on design cognition. Simon's view of design cognition was unsurprisingly influenced by his pioneering body of work on problem solving (see [43]). Simon was working on design cognition not long after the "cognitive revolution" of the 1950s within psychology (in fact he was a central figure in the movement). The dominant view of mind at the time was as a symbolic information processor. Simon was a proponent of this view, and he saw design cognition as a form of heuristic search carried out by a information processing system. In this view, the logic of design is about *finding* alternatives within a space of possibilities while using certain strategies to manage the complexity of the space.

Simon viewed the "shape of design" as essentially hierarchical. He embraced a systems perspective, in which the way to design for a complex problem is to decompose it into sub-problems or sub-functions. Once the complex problem is decomposed, "the design of each component can then be carried out with some degree of independence of the design of others" [59]. He viewed the design process as involving, "first, the generation of alternatives and, then, the testing of these alternatives against a whole array of requirements and constraints" [59]. He also had ideas for how to sequence activities within the design process that were inspired by how computer programs can engage in top-down programming and resource allocation. His stated inspirations for his theory of design include decision theory, control theory, dynamic programming, heuristic search and means-end analysis, resource allocation, and hierarchical decomposition.

Overall Simon's view of design is formal, objective, and computational. He was inspired by utility theory and statistical decision theory as a logical framework for rational choice among alternatives within a design space. He recognized this choice could not be optimized, instead advocating for satisficing [58] as the dominant heuristic. He thought of design problems mechanistically, as systems of interrelated parts that could be broken down, solved, and put back together again. The cognitive acts involved in these processes were essentially a heuristic search process, with the aim of deducing which of the available alternatives satisfies the given design criteria within a set of constraints.

3.2 Design as Reflective Practice

Donald Schön presented the most well-known alternative to Simon's problem solving view of design in his book *The Reflective Practitioner* [52] and a series of subsequent papers. Schön rejected the view of design as an information processing or heuristic search problem, instead positioning it as a form of *making* in which design cognition is fundamentally transactional in nature, unfolding as a conversation with the materials of the design situation [53, 54]. He draws on Nelson Goodman's notion of worldmaking [19], positing that people are continuously making and maintaining the worlds that are matched with their professional knowledge. People have "particular, professional ways of seeing their world and a way of constructing and maintaining the world as they see it" [52]. According to Schön, the designer does not mainly search through a solution space; rather, the designer actively structures the space by framing it, determining which things to attend to, and imposing their view of the world on it—in this view, design is a much more constructive kind of enterprise than the view put forward by Simon.

Schön argued that the problem solving view is accurate only when ends are fixed and clear, which is not typical of design problems. He pushed back against the "technical rationality" of the problem solving view, noting that the central *problem setting* work of the designer is not technical: "It is rather through the non-technical process of framing the problematic situation that we may organize and clarify both the ends to be achieved and the possible means of achieving them" [52]. This kind of framing work falls outside the scope of the problem solving view, yet it constitutes a large part of the cognitive work of designing. Designers "name and frame" the problem, using their professional judgment to assess the particular situation in which they find themselves. These situations tend to be unique, complex, and dynamic—thus not well suited to the simple application of theoretical knowledge to the situation at hand.

One of Schön's critiques of the problem solving view is that it is too narrow and does not capture much of what actually happens in design. Beyond this scope issue, however, there is a subtler distinction in the nature of the cognitive acts being carried out. Simon's view suggests that designers plan and select from alternatives systematically, sometimes iteratively and in parallel, eventually finding a solution that fits the design criteria. Schön's view is that cognitive acts are much more contingent and situated. The designer engages in a process of "seeing-moving-seeing", consisting of action sequences where there are unintended consequences of each move that is made. From this perspective, the design process is essentially a conversational structure, where, as in a conversation between friends, there is no way to predict each turn the conversation will take. The design process is unpredictable yet still disciplined. It does not follow a pre-determined process, but rather reacts to the needs of the situation given the professional knowledge of the designer. Table 1, adapted from a similar table by Dorst & Dijkhuis [17], summarizes the two paradigms and some of their key differences.

Table 1 Two paradigms of design cognition compared. Adapted from Dorst & Dijkhuis [17].

	Rational Problem Solving	Reflection in Action
Designer	information processor	person constructing reality
Design Problem	ill defined, unstructured	essentially unique
Design Process	heuristic search	reflective conversation
Design Knowledge	laws, rules, procedures	precedent, experience
Design Decisions	rational, objective decisions	personal, tacit judgments
Example/Model	optimization theory	professional artistry

4 Attempts at Integration

While design as problem solving and design as reflective practice have been widely acknowledged as two radically different, competing paradigms, there have been multiple attempts to integrate them—or at least see value in each. Dorst suggests that although these two paradigms "are on opposite sides of a deep schism that runs through science and philosophy" [16], they both can be valuable in understanding design. Dorst believed that both Simon and Schön did not pay enough attention to the structure of design problems, and that by doing so we can see how both paradigms can describe a single design process [15]. Dorst posits that rational problem solving "is better for describing the more determined problem stretches of problem solving, and a variant of reflective practice, with its sensitivity to interpretation and situatedness, could be used to pinpoint the structure in the underdetermined episodes of design thinking, especially the moments of 'breakdown' (or 'reframing')" [15]. In one study, Dorst & Dijkhuis [17] attempted to describe an industrial design process using the two paradigms, with a focus on how closely they matched the *experiences* of the designers. They concluded that the rational problem solving view works work well when design problems are clear-cut—but not otherwise. They also concluded that the reflective practice view does not offer as much descriptive precision and rigor, but it more closely matches the actual experiences of designers and provides a better description of both the design process and its content.

Visser has proposed a similar approach, noting that "design involves problem solving, but that design is not (only) problem solving" [66]. However, although Visser acknowledges the value in Simon's perspective, she also suggests that Simon "misrepresented design" in six key ways, including that his position overestimates the importance of problem decomposition, the importance of search, and the importance of means-end analysis as cognitive aspects of design. Visser suggests Simon's misrepresentations are due to his proclivity to view engineering as the prototypical design discipline, and in doing so he neglects the wide diversity of design traditions. Visser believes that situated perspectives on design cognition are essential, as in Schön's view, but also believed that Schön's work lacked some precision that has since been improved by others (e.g., [1]). Visser ultimately notes that while the problem solving approach does have value, the situated approach "has in principle the potential to propose a more appropriate view on design" [66].

More recently, Hatchuel has argued that Simon's attempts to develop a theory of design cognition were left unfinished [22]. Inspired by Simon's famous concept of bounded rationality, Hatchuel has proposed the concept of "expandable rationality" as a paradigm that addresses some of Simon's shortcomings. This topic will be addressed in more detail in Section 5.2 in the context of automated visualization design.

4.1 Philosophical Considerations

Despite attempts to integrate the two paradigms in pragmatically useful ways, there are still fundamental differences between them that are not easy to rectify. These issues are related to much bigger philosophical arguments in cognitive science about the nature of symbolic information processing in complex real-world situations [63, 64], and to debates about epistemology in science and design. Each of these paradigms is built on an underlying philosophical view of knowledge and of the world. Simon's perspective is predominately positivist, as evidenced by his emphasis on empiricism, logic, and objectivity. He was influenced by computational metaphors of information processing psychology, which assume the mind is essentially a disembodied information processor. Simon was either not aware of or dismissed contemporary work arguing that knowledge in general is essentially personal and tacit [47, 48], and that design knowledge in particular relies on social and political judgments [50]. Schön explicitly rejected the positivist epistemology of Simon, and instead built his perspective from a constructivist orientation [25]. Schön embraced the personal and tacit nature of knowledge, and explicitly rejected the idea that design knowledge could be fully codified in any objective manner [54].

There is no hope to rectify these epistemological issues here, and the philosophical debates are somewhat removed from data visualization research and practice. However, these differences do have implications for data visualization. For instance, one difference deals with how designers make decisions as they move through their process. I have previously interviewed data visualization practitioners to understand design practice as described in their own terms (see [44, 45]). It was abundantly clear from their descriptions that judgments were essential cognitive acts performed continually throughout their design process. Practitioners described engaging in judgements all the time—often in overlapping, layered ways—and engaging in logical decision making processes and search strategies very rarely if at all.

Simon's view of design cognition rejects judgment as a legitimate cognitive act. He instead views the cognitive acts of design as involving formal logic and rational search strategies, that in theory and practice can be fully codified. He wrote in praise of computer programs that could fully represent complex design processes, where "there is no question . . . of the design process hiding behind the cloak of 'judgment' or 'experience"'. Simon viewed judgment with suspicion, as if any decision making processes that are rigorous should essentially be separable from the decision maker, and thus be amenable to codification in formal language.

I have also written about how a concept like chartjunk is used by practitioners in personal, contextually relevant ways [45]. It appears unlikely that the use of such a concept could be codified in a rule-based, prescriptive manner. There is strong evidence that good design relies on personal, tacit, situated knowledge—the kind of knowledge that cannot be articulated and codified. This claim is not originally my own—it has been demonstrated previously in multiple design fields, including instructional design [21] and interaction design [61].

On the role of judgment in design cognition, it appears that Simon was mistaken. Yet the positivist orientation in general, and the rational problem solving view in particular, are still very popular in STEM fields. Meyer and Dykes [36] have recently written about the positivist leanings of the visualization and computer science communities. Schön [53] pointed out the dominance of positivism in academia and in the professional schools, referring to the dominant epistemology as *technical rationality*. From this standpoint, the kinds of knowledge that are valued are general, objective, and abstract. Many academics in science and technology fields learn that these characteristics are the hallmarks of good research. Herein lies the consequence of adopting, even unconsciously, one of these underlying philosophical orientations. If researchers are trained in an environment embracing technical rationality, they will likely strive to generate knowledge that is abstract, objective, and general. Even while doing empirical work, they may only see that which is abstract, objective, and general, because that is what they have learned to recognize as valuable. The philosophical orientation can be reinforced even through empirical investigation because it acts a a lens through which the researcher interprets the world.

The issue of philosophical orientation is of course a very general one, so what does it mean for visualization researchers? If the field is primarily positivist in orientation, researchers are likely to gravitate towards the rational problem solving view because it claims to be objective and systematic. They will then view visualization design as a primarily as a heuristic search problem carried out by designers as information processors. If researchers do not recognize design cognition as comprising personal and tacit forms of knowledge, they will not develop a sufficient understanding of how design is actually practiced, and may not be able to train visualization designers effectively. While these are some generic implications of adopting one paradigm—most likely to be the rational problem solving paradigm in the visualization community—I attempt to elaborate some more specific implications in the following section.

5 Implications for Data Visualization

The way that design cognition is construed is not simply a matter of abstract intellectual debate, and has many—often subtle—consequences for visualization research, practice, and training. In the following sections I discuss some implications for defining, automating, modeling, and teaching data visualization design.

5.1 Defining Design for Data Visualization

Addressing many of the challenges and differences discussed in this chapter rests on determining what kind of activity design is considered to be. The question of what constitutes design is inevitable but not easy to answer. There are the well-known generic answers, like Simon's devising "courses of action aimed at changing existing situations into preferred ones" [59]. Nelson & Stolterman promote design as a "third way", distinct from science and art, describing it as "the ability to imagine that-which-does-not-yet-exist, to make it appear in concrete form as a new, purposeful addition to the real world" [42]. Regarding visualization specifically, van de Moere and Purchase draw from multiple definitions of design to emphasize the creative aspects of design and the personal role of the designer in shaping the design process [39].

Providing a commonly-agreed on definition of design for data visualization is likely an impossible goal. One possible way to achieve consensus is to focus on the guiding values or ideals of the discipline, as has been done recently for interaction design [24]. I will not attempt to do that here, but will suggest that any agreement about the nature of design cognition needs to rest on some degree of consensus about the nature of design broadly construed. For instance, if design is limited to the selection of visualization and interaction techniques from a known set of possibilities, implications are different from the case where design includes the creation of new techniques or the managing of clients, software tools, deadlines, collaborations, and other facets of real-world practice.

Any definition of design should be informed by the cognitive and other considerations that go into the work of doing design. Whether design cognition is fundamentally rational problem solving or reflective practice necessarily influences the definition of design. Is design primarily about discovery, as Simon's view might suggest, or about constructing the reality of the situation, as Schön's view suggests? Nelson & Stolterman reject the problem solving view of design, noting its focus being limited to "that-which-is (description and explanation)" instead of "that-which-ought-to-be (ethics and morality), and . . . that-which-is-desired (desiderata)" [42]. Adopting any view of visualization design in general, and design cognition in particular, will influence the view of other important topics like automated design, design models, and design education, each of which will be discussed in the following sections.

5.2 Automated Visualization Design

One topic for which design cognition has deep implications—in ways that may not be immediately obvious—is in automated visualization design. Since the early days of AI, researchers have had a desire to model human cognition computationally, with the goal of either fully or partially automating cognitive processes and human knowledge. One particular instance of this is in modeling design knowledge and processes with the goal of automating design. However, there are implications for the success of this vision based on the nature of design cognition. For instance, if human cognition operates fundamentally in a symbolic information processing mode—which is the basic thrust of information processing psychology and Simon's theory of problem solving—then it should be possible to replicate cognitive processes computationally. However, if symbolic information processing is not the right metaphor, or is at least not the right one for design cognition, it may not be possible to fully automate the relevant knowledge and cognitive processes for design.

Within the visualization literature, the goal of automating design has a long history. The most well-known early work was from Mackinlay on his presentation tool APT [31], which viewed the design of graphical representations as fundamentally a search problem aiming to optimize effectiveness and expressiveness. Subsequent research in this space has informed the design of systems like Tableau [30], SAGE [38], Voyager [67], Draco [40], and numerous others in recent years.

These automated tools appear to be framed around two visions. One vision is to assist analysts in understanding their data. Because analysts do not have expertise in visualization design, it can be difficult for them to create useful visualizations. As a result, they will often turn to default charting options in tools like Excel, which may be unhelpful or even misleading. If instead of this, visualizations can be recommended to them based on characteristics of the data and the analyst's goals, the analysis situation can be greatly improved. This is the stated intention behind techniques like Tableau's "Show Me" [30] and more recent work on query languages like CompassQL [68]. Another vision for these automated tools is to assist designers in creating visualizations based on codified design knowledge. This is a perspective taken by recent systems like Draco [40], in which the aim is to formally model design knowledge with hard and soft constraints over logical facts. This vision embraces the "application" view described previously in Section 2—researchers conduct empirical studies to generate design knowledge, which then gets applied in practice. One stated motivation for creating these systems is that there is a gap that needs to be filled between the researcher-generated knowledge and its application in practice.

There is no question that that these automated visualization design tools are valuable, especially as aids to analysts who do not have adequate visualization design knowledge. In such cases, however, the "design" work that is being done is narrow with respect to the whole range of considerations that go into the real-world design practice. The extent to which visualization design can be automated depends in part on the nature of design cognition, design knowledge, and design practice. If the complexity, uncertainty, and messiness of real-world design is not considered, computational tools can automate only small parts of the design process. Here we see again implications of which paradigm(s) of design cognition are embraced. If design cognition is fundamentally symbolic information processing, as in Simon's view, design knowledge and processes are separable from the designer and can be codified as such. Based on the reflective practice view, however, an expectation for fully automated design is unrealistic even in principle. Unless artificial intelligence can develop the subtle appreciative and imaginative abilities of humans, there is no hope for design to be fully automated. On this matter Schön [54] argues that computers would need to achieve *phenomenological and functional equivalence* with humans to be able to reproduce essential aspects of design cognition, including the continual, subjective appreciation of a situation and the envisioning of future design worlds.

A perspective that may be helpful here is Hatchuel's concept of "expanded rationality" [22], briefly described previously in Section 4. Hatchuel argues that true design problems involve infinite and non-countable sets, for which heuristic search his not an appropriate strategy. Bounded rationality does not help with these kinds of situations, because true design situations are infinitely expandable. For instance, when a client approaches a designer and says "help me see something useful in my sales data", there is no bounded problem space that can be computationally exhausted. The concept of "useful" is infinitely expandable, and it is the task of the designer to frame the problem space and make the design task manageable. The designer may interview the client and other stakeholders, for instance, and determine that what is useful is to see the growth of certain market segments in relation to political or natural events. This is very much in line with Schön's emphasis on the active, constructive nature of framing the problem to addressed. Even in principle, infinite time and computing resources could not explore the space. Hatchuel's theory argues that a situation is a "real design problem" only if the initial concepts allow for unexpected expansion. The designer makes use of their creative, imaginative, and appreciative abilities to do this work. If there is no opportunity for expansion, Hatchuel argues there is no real *design* problem—only a regular problem to be solved. This view may provide conceptual support for the issue of automation in design, as it offers a useful description of what distinguishes design from typical problem solving.

Aside from the cognitive processes involved in design, a central topic for understanding the possibilities of automation in design is *design knowledge*. If design involves knowledge that is—even in part—fundamentally irreducible, it is not possible to codify. There is much evidence that design knowledge is holistic and personal in the ways that Polanyi described of scientific knowledge [47]. Certainly some design knowledge is objective and able to be codified, but if critical pieces of knowledge are personal and tacit, there is little hope to fully automate design. The nature of design knowledge has been written about extensively by others and is beyond the scope of this chapter.

When it comes to the topic of automated visualization design, there are at least two key topics that must be encountered. First, if design cognition is expansive in nature, computational search processes are not sufficient for fully automating design—even in principle. Second, if design knowledge is personal and tacit, it cannot be fully codified. Some aspects of design can be automated, especially those that deal with heuristic search through spaces of known solutions, and those that deal with objective kinds of knowledge like laws and principles. It is important to recognize that these cover only a portion of what is involved in real-world design practice, however. The important question is not whether design can be fully automated, but rather: what are the cognitive processes and types of knowledge that can be codified, and how should tools work in concert with designers to leverage computational and human strengths in design?

5.3 Visualization Design Models and Frameworks

Similar to the topic of automated visualization design is the view of design models and frameworks in the visualization literature. Numerous frameworks and models have been proposed to describe the design process and to provide researchers and designers with advice and guidance. For instance, popular decision models include the Nested Model [41] and its Blocks and Guidelines extension [37]. Popular process models include the nine-stage framework in the Design Study Methodology [56], the Design Activity Framework [34], and others [20, 33, 55, 57].

Here we can again ask what role the underlying paradigms of design cognition have played in the development of these models and frameworks. Most appear to be closer in spirit to Simon's view of design cognition—supporting decision making and a rational search process through a space of possibilities. They do not appear to align as much with Schön's reflective practice view, in which we would expect to see descriptions of the situated, transactional nature of design—e.g., problem framing and setting, imagination, judgment, and contingent means of navigating the design situation. We would also expect to see descriptions of reflection-in-action taking place within the design situation. These existing models generally do value reflection, but mainly as a mechanism for making contributions back to the research community and not as an essential cognitive act that aids movement through the design process. For instance, the nine-stage framework [56] has as its 8th stage "reflection". By this is meant reflection after the design process, on what was done and how it relates to the research landscape. In their paper "Reflection on Reflection in Applied Visualization Research", Meyer and Dykes [35] have noted how there is a "bias towards post-study reflection", recommending "a more structured and purposeful approach to reflection throughout the entire design process". Additionally, many of these models and frameworks acknowledge the ill-structured and wicked nature of design problems. However, Simon acknowledged this as well [60], but still promoted the rational problem solving view. Referring to design as ill-structured, wicked, messy, complex, or iterative does not indicate which paradigm is being adopted. The goal here is not to criticize these frameworks, but rather to raise the question of how they relate to paradigms of design cognition.

The models and frameworks discussed above have not explicitly engaged with paradigms of design cognition to motivate their development, perhaps due to a lack of awareness, as it is largely an unknown topic within the visualization literature. However, these models and frameworks rely on assumptions about cognitive aspects of design, whether they are explicitly acknowledged or not. These assumptions are of course influenced by the primary philosophical paradigms within the visualization field, which Meyer and Dykes [36] have recently argued is dominated by positivism. Furthermore, these models have largely been developed for the academic research community. Here there may be somewhat of a self-fulfilling prophecy taking place. If the research community adopts one paradigm of design cognition—unconsciously or not—they will value models that conform to that paradigm. For example, a positivist community will value objective, empirical, abstract knowledge; because that is what is valued, researchers will attempt to develop abstract, general models;

these models will then be evaluated on the same positivist criteria that led to their development. Finally, it is important to note that these models and frameworks have been generated and used within the research community, and their applicability to practice is uncertain. The cognitive acts of researchers doing design work and practitioners doing design work likely have some differences, but also likely share foundational elements. However, the nature of these relationships has not been investigated and could be a topic for future inquiry.

5.4 Visualization Education

The view of design cognition that is adopted by educators has significant implications for data visualization pedagogy. For instance, consider the “application” view described previously in Section 2. Within this perspective, there is an implicit hierarchy of knowledge. The foundation is basic, scientific knowledge generated from research settings, which then gets applied in practice. The way that many educators teach in universities essentially follows this model. First, students attend lectures that cover the theory, then they subsequently “apply” the knowledge through labs or assignments. This model makes sense if design really is an application of basic knowledge to specific problems. It fits squarely within the positivist landscape of the majority of STEM disciplines. The view of the cognitive work that is taking place also aligns with Simon’s view of design, where designers are engaging in heuristic search activities in an attempt to select and apply a good combination of components from the solution space. If this model is not accurate, however, the pedagogical approach may not be so effective. Plenty has been written on the topic of removing lectures and making classes more “active”, often by using “flipped” modes of instruction and various “hands-on” activities within the classroom. These approaches may make instructors feel modern, but if they are not appropriate for the intended activity the students are training for, they are ultimately not very useful. Sometimes lectures are the appropriate mode of instruction, especially if people really do simply need to apply abstract knowledge. The important question is what kind of activity is design? Is it essentially a form of application, or is there more to it? And what is the appropriate mode of instruction for training designers?

I believe the contents of this chapter have demonstrated that visualization design is not primarily about application (although it may involve application), and it is not only a rational problem solving activity (although it may involve it too). Explicating a pedagogy of data visualization is much beyond the scope of this chapter, but a few points can be made here. If data visualization design is not simply about application; if its cognitive aspects involve making, expansive thinking, and the envisioning of future worlds; if their activity is fundamentally transactional in nature; then students need to be supported in these activities with the right instructional models. One such pedagogical model is that of the design studio, a model that has a long history in art and design. Design studios position the role of an instructor as more of a coach than a lecturer, where students learn by doing, and the coaches provide demonstrations, critiques, and just-in-time instruction as means of formative feedback. Studio pedagogy tends to be more constructionist-oriented, aligning with the reflective practice view of design and not so well with the application view. The goal of studio pedagogy is to prepare students to handle the complexity, uncertainty, and messiness of real-world practice rather than providing them with prescriptive procedures to follow or abstract theory to apply. In Schön's view, design students need tools for reflection that allow them to appropriately face each unique design situation with all its complexity and richness under consideration. Based on this, Stolterman [61] suggests that design education should focus on training designers to be "prepared-for-action" and not "guided-in-action."

Here the implications of adopting one paradigm of design cognition should again be apparent. If the cognitive acts of designing are akin to rational problem solving, where logic and systematic thinking are valued, the design studio and all of its messiness may not be as effective as a traditional lecture and lab format. But if the cognitive acts of designing are more transactional in nature, where designers need to "converse" with the materials of a particular situation, and receive just-in-time instruction and critique, the design studio appears to be a much better fit. Much more can be said about design cognition and visualization pedagogy, and I have only scratched the surface of the issue here. Hopefully these ideas inspire others engage in future investigation of these topics.

6 Summary

Design cognition has not received much attention thus far in the visualization literature. In this chapter, I have introduced some aspects of design cognition and described their relevance for data visualization. The discussion has been built largely on the two historically dominant paradigms of rational problem solving and reflective practice. These paradigms provide different pictures of what design cognition is, and I have attempted to describe some implications for four data visualization topics: defining, automating, modeling, and teaching data visualization design. I posit that the visualization community embraces the rational problem solving view of design more than the reflective practice one, although future research can investigate this claim in more detail. Embracing the reflective practice paradigm may lead to more comprehensive knowledge of data visualization design and open new opportunities for research and teaching.

References

1. R. S. Adams, J. Turns, and C. J. Atman. Educating effective engineering designers: The role of reflective practice. *Design Studies*, 24(3):275–294, 2003.
2. O. Akin. Variants in design cognition. In C. Eastman, W. Michael McCracken, and W. C. Newstetter, editors, *Design Knowing and Learning: Cognition in Design Education*, page

105–124. Elsevier, 2001.
3. C. J. Atman, R. S. Adams, M. E. Cardella, J. Turns, S. Mosborg, and J. Saleem. Engineering design processes: A comparison of students and expert practitioners. *Journal of Engineering Education*, 96(4):359–379, 2007.
4. L. J. Ball and B. T. Christensen. Advancing an understanding of design cognition and design metacognition: Progress and prospects. *Design Studies*, 65:35–59, Nov 2019.
5. R. Buchanan. Wicked problems in design thinking. *Design Issues*, 8(2):5–21, 1992.
6. J. M. Carroll. Scenarios and design cognition. In *Proceedings IEEE Joint International Conference on Requirements Engineering*, page 3–5, 2002.
7. P. J. Cash, B. J. Hicks, and S. J. Culley. A comparison of designer activity using core design situations in the laboratory and practice. *Design Studies*, 34(5):575–611, Sep 2013.
8. C. L. Cramer-Petersen, B. T. Christensen, and S. Ahmed-Kristensen. Empirically analysing design reasoning patterns: Abductive-deductive reasoning patterns dominate design idea generation. *Design Studies*, 60:39–70, 2019.
9. N. Crilly. Creativity and fixation in the real world: A literature review of case study research. *Design Studies*, 2019.
10. D. P. Crismond. *Investigate-and-Redesign Tasks as a Context for Learning and Doing Science and Technology: A study of naive, novice and expert high school and adult designers doing product comparisons and redesign tasks*. PhD thesis, Harvard University, 1997.
11. N. Cross. Science and design methodology: A review. *Research in Engineering Design*, 5(2):63–69, June 1993.
12. N. Cross. Designerly ways of knowing: Design discipline versus design science. *Design Issues*, 17(3):49–55, 2001.
13. A. Damle and P. J. Smith. Biasing cognitive processes during design: the effects of color. *Design Studies*, 30(5):521–540, Sep 2009.
14. A. Dong, D. Lovallo, and R. Mounarath. The effect of abductive reasoning on concept selection decisions. *Design Studies*, 37:37–58, Mar 2015.
15. K. Dorst. Exploring the structure of design problems. In *International Conference on Engineering Design (ICED)*, page 10, 2003.
16. K. Dorst. *Frame Innovation: Create New Thinking by Design*. MIT Press, Mar 2015.
17. K. Dorst and J. Dijkhuis. Comparing paradigms for describing design activity. *Design Studies*, 16(2):261–274, Apr 1995.
18. C. M. Eastman. New directions in design cognition: Studies of representation and recall. In *Design Knowing and Learning*, page 147–198, 2001.
19. N. Goodman. *Ways of Worldmaking*. Hackett Publishing, 1978.
20. 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, 2013.
21. C. M. Gray, C. Dagli, M. Demiral-Uzan, F. Ergulec, V. Tan, A. A. Altuwaijri, K. Gyabak, M. Hilligoss, R. Kizilboga, K. Tomita, and et al. Judgment and instructional design: How ID practitioners work in practice. *Performance Improvement Quarterly*, 28(3):25–49, 2015.
22. A. Hatchuel. Towards design theory and expandable rationality: The unfinished program of Herbert Simon. *Journal of Management and Governance*, 5(3–4):260–273, 2002.
23. N. V. Hernandez, J. J. Shah, and S. M. Smith. Understanding design ideation mechanisms through multilevel aligned empirical studies. *Design Studies*, 31(4):382–410, Jul 2010.
24. K. Höök and J. Löwgren. Characterizing interaction design by its ideals: A discipline in transition. *She Ji: The Journal of Design, Economics, and Innovation*, 7(1):24–40, 2021.
25. E. A. Kinsella. Constructivist underpinnings in Donald Schön's theory of reflective practice: echoes of Nelson Goodman. *Reflective Practice*, 7(3):277–286, 2006.
26. B. Lawson. The context of mind. *Designing in context*, page 133–148, 2001.
27. B. Lawson. *How Designers Think*. Routledge, 2006.
28. B. R. Lawson. Cognitive strategies in architectural design. *Ergonomics*, 22(1):59–68, 1979.
29. J. Löwgren. Applying design methodology to software development. In *Proceedings of the conference on Designing interactive systems processes, practices, methods, & techniques - DIS '95*, pages 87–95, Ann Arbor, Michigan, United States, 1995. ACM Press.

30. J. Mackinlay, P. Hanrahan, and C. Stolte. Show me: Automatic presentation for visual analysis. *IEEE Transactions on Visualization and Computer Graphics*, 13(6):1137–1144, 2007.
31. J. D. Mackinlay. Automating the design of graphical presentations of relational information. *ACM Transactions on Graphics*, 5(2):110–141, 1987.
32. X. Mao, O. Galil, Q. Parrish, and C. Sen. Evidence of cognitive chunking in freehand sketching during design ideation. *Design Studies*, 67:1–26, Mar 2020.
33. N. McCurdy, J. Dykes, and M. Meyer. Action design research and visualization design. In *Proceedings of the 6th Biannual Workshop on evaluation and BEyond - methodoLogIcal approaches for Visualization (BELIV)*, page 10–18. ACM Press, 2016.
34. S. McKenna, D. Mazur, J. Agutter, and M. Meyer. Design activity framework for visualization design. *IEEE Transactions on Visualization and Computer Graphics*, 20(12):2191–2200, 2014.
35. M. Meyer and J. Dykes. Reflection on reflection in applied visualization research. *IEEE Computer Graphics and Applications*, 38(6):9–16, 2018.
36. M. Meyer and J. Dykes. Criteria for rigor in visualization design study. *IEEE Transactions on Visualization and Computer Graphics*, 26(1):87–97, 2020.
37. M. Meyer, M. Sedlmair, P. S. Quinan, and T. Munzner. The nested blocks and guidelines model. *Information Visualization*, 14(3):234–249, 2015.
38. V. O. Mittal, J. D. Moore, G. Carenini, and S. Roth. Describing complex charts in natural language: A caption generation system. *Computational Linguistics*, 24(3):431–467, 1998.
39. A. V. Moere and H. Purchase. On the role of design in information visualization. *Information Visualization*, 10(4):356–371, Oct 2011.
40. D. Moritz, C. Wang, G. L. Nelson, H. Lin, A. M. Smith, B. Howe, and J. Heer. Formalizing visualization design knowledge as constraints: Actionable and extensible models in draco. *IEEE Transactions on Visualization and Computer Graphics*, 25(1):438–448, Jan 2019.
41. T. Munzner. A nested model for visualization design and validation. In *IEEE Transactions on Visualization and Computer Graphics*, volume 15, page 921–928, 2009.
42. H. G. Nelson and E. Stolterman. *The Design Way: Intentional Change in an Unpredictable World*. The MIT Press, second edition edition, 2012.
43. A. Newell and H. A. Simon. *Human Problem Solving*. Prentice-Hall Englewood Cliffs, NJ, 1972.
44. P. Parsons, C. M. Gray, A. Baigelenov, and I. Carr. Design judgment in data visualization practice. In *IEEE Visualization Conference (VIS), short papers*, Sep 2020.
45. P. Parsons and P. Shukla. Data visualization practitioners' perspectives on chartjunk. In *IEEE Visualization Conference (VIS), short papers*, Sep 2020.
46. R. S. Perez and C. D. Emery. Designer thinking: How novices and experts think about instructional design. *Performance Improvement Quarterly*, 8(3):80–95, 1995.
47. M. Polanyi. *Personal knowledge: Towards a post-critical philosophy*. University of Chicago Press, 1958.
48. M. Polanyi. *The Tacit Dimension*. University of Chicago Press, 1966.
49. A. Purcell and J. S. Gero. Design and other types of fixation. *Design Studies*, 17(4):363–383, Oct 1996.
50. H. W. Rittel and M. M. Webber. Dilemmas in a general theory of planning. *Policy sciences*, 4(2):155–169, 1973.
51. N. F. Roozenburg and K. Dorst. Describing design as a reflective practice: Observations on schön's theory of practice. In *Designers*, pages 29–41. Springer, 1998.
52. D. A. Schön. *The Reflective Practitioner: How Professionals Think In Action*. Basic Books, 1983.
53. D. A. Schön. Designing: Rules, types and worlds. *Design Studies*, 9(3):10, 1988.
54. D. A. Schön. Designing as a reflective conversation with the materials of a design situation. *Research and Engineering Design*, 3(3), 1992.
55. K. Sedig and P. Parsons. Design of visualizations for human-information interaction: A pattern-based framework. *Synthesis Lectures on Visualization*, 4(1):1–185, 2016.

56. 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.
57. B. Shneiderman and C. Plaisant. Strategies for evaluating information visualization tools: multi-dimensional in-depth long-term case studies. In *Proceedings of the 2006 AVI workshop on BEyond time and errors: novel evaluation methods for information visualization*, pages 1–7, 2006.
58. H. A. Simon. *Administrative Behavior*. Macmillan Inc., 1947.
59. H. A. Simon. *The Sciences of the Artificial*. MIT Press, 1969.
60. H. A. Simon. The structure of ill structured problems. *Artificial intelligence*, 4(3-4):181–201, 1973.
61. E. Stolterman. The nature of design practice and implications for interaction design research. *International Journal of Design*, 2(1):55–65, 2008.
62. C. Stones and T. Cassidy. Seeing and discovering: how do student designers reinterpret sketches and digital marks during graphic design ideation? *Design Studies*, 31(5):439–460, Sep 2010.
63. L. A. Suchman. *Plans and Situated Actions: The Problem of Human-Machine Communication*. Cambridge University Press, 1987.
64. A. H. Vera and H. A. Simon. Situated action: A symbolic interpretation. *Cognitive science*, 17(1):7–48, 1993.
65. W. Visser. Two functions of analogical reasoning in design: a cognitive-psychology approach. *Design Studies*, 17(4):417–434, 1996.
66. W. Visser. Designing as construction of representations: A dynamic viewpoint in cognitive design research. *Human–Computer Interaction*, 21(1):103–152, Mar 2006.
67. K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Voyager: Exploratory analysis via faceted browsing of visualization recommendations. *IEEE Transactions on Visualization and Computer Graphics*, 22(1):649–658, 2015.
68. K. Wongsuphasawat, D. Moritz, A. Anand, J. Mackinlay, B. Howe, and J. Heer. Towards a general-purpose query language for visualization recommendation. In *Proceedings of the Workshop on Human-In-the-Loop Data Analytics*, HILDA ’16. ACM, 2016.