Journal of the Association for Information Systems
JAIS

Research Article

Generating and Justifying Design Theory

Munir Mandviwalla
Temple University
mandviwa@temple.edu

Abstract

This paper applies Simon's (1996) sciences of the artificial to elaborate a set of structures and processes for developing design theory. Goals, kernel theory, and artifacts inform an inter-related prototyping cycle of design, evaluation, and appropriation / generation to produce strategic design theory. The paper identifies DSR project types to provide signposts for starting and ending the cycle, artifact and evaluation iteration to facilitate the process and provide a chain of evidence, a simplified format for representing design theory iterations, and stopping rules to end the cycle. We use a detailed example to illustrate the ideas, discuss related work, and identify limitations and future research opportunities.

***Keywords**: Socio-technical System, Design Science, Conceptual.*

* Matti Rossi was the accepting senior editor. This article was submitted on 12th January 2012 and went through two revisions.

Volume 16, Issue 5, pp. 314-344, May 2015

Generating and Justifying Design Theory

1. Introduction

The seminal design science research (DSR) papers by Weber (1987), Nunamaker, Chen, and Purdin (1991), Walls, Widmeyer, and El Sawy (1992), and March and Smith (1995) initially defined the role and importance of DSR. More recently, Hevner, March, Park, and Ram (2004), Gregor and Jones (2007), Peffers, Tuunanen, Rothenberger, and Chatterjee (2008), Kuechler, and Vaishnavi (2012), Sein, Henfridsson, Purao, Rossi, and Lindgren (2011), and Gregor and Hevner (2013) have formalized DSR's knowledge contribution, components, cycle, elaboration, forms, and positioning. In this paper, I contribute to this literature by focusing on theorizing in DSR. Specifically, I develop design theory and justificatory knowledge in existing DSR methods. The research is important because even though current approaches address the "what" and the overall meta-cycle of DSR, there is still much to be done on the "how"—the steps needed to produce theories and practical insights. Further, recent research continues to identify design theory generation as an important but problematic task (e.g., Alturki, Gable, & Bandara, 2013; Lukyaneko & Parsons, 2013; Buckl, Matthes, Schneider, & Schweda, 2013; Venable, 2013; Hovorka & Pries-Heje, 2013). This paper helps fill the above gap and focuses on the research question: how do you develop design theory in DSR?

Simon (1996) introduced the notion of a design theory and the need to make it explicit and precise—to make it a *science* of the artificial. Science requires the scientific method; that is, to be systematic, observable, empirical, and measurable. In IS, Gregor and Jones (2007) make the case that, for the field to rise "above the level of a craft" (p. 314), it is important to express design knowledge as theory. Yet, DSR's central focus, the artifact, by itself cannot easily be added to the knowledge base of science. It is hard to systematically show, measure, and justify the scientific contribution of an artifact. Consider a ground breaking lab experiment and compare it to creating an interesting artifact. The experiment is a contribution and should be published; in addition, to add to knowledge in a way that is systematic and measurable, there is also a need to infer the theoretical contribution. In the case of artifacts, the option of (only) publishing the artifact is challenging. Contemporary IT artifacts are built on top of complex hierarchies of components (e.g., user interface components such as buttons) and, unless explicitly elaborated, it can be hard to discern the unique contribution. The signal-to-noise ratio in leaving the contribution embedded inside the artifact is problematic. Therefore, even though artifacts can represent new and scientifically interesting knowledge, they are not the best tool for exposition and representation. The "truth statements" that exemplify scientific propositions are a more precise elaboration and are more in line with Simon's call for a science of the artificial. When the design is elaborated in truth statements, it can then be evaluated and compared to previous and future knowledge. However, as I show in this paper, generating design theory is difficult and expensive, and the results will likely only be interesting if the underlying artifact is novel. Further, it is hard to discern upfront the potential theoretical value and artifact novelty. What is needed is a design theory generation approach that considers theoretical interest and artifact novelty.

Simon (1996), Walls et al. (1992), Gregor and Jones (2007), and Gregor and Hevner (2013) provide a formal roadmap of the key components of design theory and DSR in general, but they don't fully elaborate on how to generate design theory. This paper contributes to the literature by providing a set of processes and structures for developing design theory. I start by reviewing the relevant literature to motivate the research question. Next, I refer to Simon (1996) to propose processes and structures for generating design theory. In Section 4, I apply the ideas in a sample DSR project. In Section 5, I note the paper's contributions and its relationship to DSR and organizational literature, and I also highlight future research opportunities. Finally, in Section 6, I conclude the paper.

2. Relevant Literature

The IS literature has addressed many of the methodological challenges facing DSR. Walls et al. (1992) have defined its key components. Hevner et al. (2004) have established its legitimacy and provided evaluation guidelines. Gregor and Jones (2007) have specified the structure of design theory, which is important because a shared vocabulary will lead to more systematic and measurable knowledge contributions. Peffers et al. (2008) have outlined the overall process, which is important because, now, projects can be placed in an overall context and are thus more observable. Sein et al. (2011) have shown how DSR can operate in and leverage "field" settings to ensure that relevant problems are addressed. Kuechler and Vaishnavi (2012) have focused on predictive and explanatory theory to further expand DSR's knowledge contribution. Gregor and Hevner (2013) have outlined levels of DSR knowledge contribution based on level of abstraction and maturity and how the relationship between descriptive (e.g., behavioral) and predictive (e.g., design theory) knowledge can influence different forms of DSR.

Despite all these gains in DSR overall, some researchers continue to question design theory's role and importance (e.g., March & Smith, 1995; Hooker, 2004; Venable, 2013), and recent research notes that design theory construction, representation, and application is unclear and difficult to achieve (e.g., Alturki et al. 2013, Lukyaneko & Parsons, 2013; Hovorka & Pries-Heje, 2013). Gregor and Hevner (2013) discuss the need to harmonize the "design-theory camp" with the "pragmatic-design camp". The underlying cause of these challenges is that current approaches address only the "what" and the overall meta-cycle of DSR. The steps needed to produce design theories and practical insights (the "how") are still unclear. Specifically, the following "how" questions still remain unresolved in developing design theory:

- Kernel theory: how does kernel theory fit with design theory? When is kernel theory needed? How do you select appropriate kernel theory? What if no kernel theory is available?

- Design theory: how do you generate the components? How do you formulate the constructs, principles of form and function, and the testable propositions?

- Evaluation: how and when will the artifact and design theory be evaluated and how does that influence design theory construction? How do you demonstrate the validity of the design theory? How do you know when to stop the design theory development (and DSR) cycle?

- Relevance and rigor: given contemporary tools, it is much easier today to produce designs at a rapid pace and these designs have many possible implications. How will DSR hone in on the design theories that are rigorous and relevant, the most strategic?

Theorizing in the social and organizational sciences (e.g., Dubin, 1978; Weick, 1989) has had an important influence on DSR. For example, Dubin (1978) is the basis for key concepts in Walls et al. (1992), and Gregor and Hevner's (2013) knowledge contribution framework parallels Colquitt and Zapata-Phelan's (2007) theory classification. Yet, the social and organizational sciences are also sparse on the how. As Weick (1989, p. 517) states: "Unfortunately, the literature on this topic is sparse and uneven, and tends to focus on outcomes and products rather than process". More recently, Suddaby, Hardy, and Huy (2011) have summarized the key ideas on theory construction and conclude that a lot more still needs to done to generate new interesting theories, especially indigenous theories.

The above "how" questions and gaps motivate the research question I address in this paper: how can we develop design theory in DSR for interactive digital artifacts, and especially how can we develop new and interesting theories that illustrate the digital innovation inherent in new and interesting artifacts. Applying Gregor and Hevner (2013), I build on previous DSR and leverage Simon (1996) to identify the processes and structures needed to produce level 2 (nascent design theory) and level 3 (well-developed design theory) DSR contributions in which level 1 (implementation of the artifact) contributions play a key role.

I use the term artifact to "refer to a thing that has, or can be transformed into, a material existence as an artificially made object (e.g., model, instantiation) or process (method, software)" (Gregor & Hevner, 2013, p. 341). This definition is consistent with Simon; however, he places artifacts in a larger context by referring to them as an artificial thing created by humans (Simon, 1996). I further focus on the issues, constraints, and opportunities for developing design theory for the "instantiation" of "interactive" novel digital artifacts in both the organizational *and* consumer spaces. Nevo et al. (2011)

conclude after analyzing more than 1000 IS publications that the literature, in effect, focuses on instantiations. An interactive artifact is one that involves interaction with humans. At least 65 percent of the IT artifact papers that Nevo et al. analyzed focus on interactive systems[1]. Further, given the current prevalence of highly interactive cloud and mobile artifacts for both organizations and consumers, one can argue that the ratio will only increase.

3. Herbert Simon—The Sciences of the Artificial

In his seminal book *The Sciences of the Artificial*, Simon (1996) attempts to make design theory explicit and precise so that it can match natural science theory. The key principles from Simon's work on developing design theory include formalizing the process of generating design theory by (a) following a systematic and observable process and (b) representing the results including the design and the design theory. Further, according to Simon, developing design theory is difficult and expensive and can be enhanced and constrained by (a) sustaining quasi-independence between the design and problem context, (b) leveraging insights from previous versions, and (c) evaluation. In Sections 3.1 to 3.6, I elaborate on Simon's principles.

3.1. The Process for Generating Design Theory

According to Simon (1996, p. 6), "An artifact can be thought of as a meeting point—an 'interface' in today's terms—between an 'inner' environment, the substance and organization of the artifact itself, and an 'outer' environment, the surroundings in which it operates.". There is an inherent tension between this inner and outer environment: Weber (1987, p. 12) quotes the law of requisite variety in general systems theory to make the case that research involving human behavior (outer environment) and systems (inner environment) will inevitably focus more on human concerns because they have a greater range of responses than inanimate objects. As a result, Weber suggests focusing on only system (inner environment) issues. However, ignoring human concerns can lead to technological determinism (Spacapan & Oskamp, 1990). Therefore, other researchers have suggested focusing on the inner and outer environments in phases or cycles or to mediate the interaction (e.g., Hevner, 2007; Bartneck, 2009). However, it is always going to be difficult to separate the two environments. Simon (1996, p. 3) puts the issue in perspective:

> *…for those things we call artifacts are not apart from nature. They have no dispensation to ignore or violate natural law. At the same time they are adapted to human goals and purposes. They are what they are in order to satisfy our desire to fly or to eat well. As our aims change, so too do our artifacts–and vice versa.*

Similarly, according to Iivari (2007), world 1 (nature) relates to and constrains world 3 (artifacts). Therefore, when a designer decides to include a feature, they are explicitly or implicitly hoping to support or encourage an act in the world of the user—the outer environment. So, every design decision in the inner environment is also a decision about impact on the reality of the user in the outer environment. And every problem in the outer environment is an opportunity in the inner environment. In essence, the design of an artifact includes a series of *inter-related* inner and outer environment decisions that reflect DSR's duality challenge. The duality is that computing is both an abstract, mathematical object obeying natural laws and an empirical object that reacts differently to different environmental and organizational stimuli (Simon, 1996; Newell & Simon, 1976). The duality challenge is further compounded in today's pervasive IT environment:

1. The outer environment challenge: Weick (1989) notes that theorists face challenges because organizational problems are wide in scope, lack detail, are vague, and are difficult to represent. Further, in most natural sciences such as physics, human actions do not have to be taken into account. It is feasible to theorize about black

[1] 65% of the papers analyzed by Nevo, Nevo, and Ein-Dor (2011) for the most recent five-year period ending in 2006 belong to categories of systems that are interactive by nature (e.g., DSS, CMC, ERP, websites). This is a conservative estimate since, of the remaining artifacts, 24% belong to the inter-organizational systems and infrastructure services categories and since systems such as electronic markets also typically have an interactive element.

holes in a vacuum and ignore the human. Contemporary interactive digital design, on the other hand, faces a major challenge in creating artifacts for increasingly sophisticated users operating in dynamic and complex organizations and whose actions (that are hard to predict with any reliability) can take on a wide range of possibilities including modifying and re-appropriating the artifact.

2. The inner environment "Moore's law"[2] double-loop challenge: as toolsets improve, the time and resources needed to try out different designs decreases, while the number and capabilities of widgets that allow designers to quickly create new artifacts that build on existing artifacts increases. This "Moore's law" double-loop effect means that contemporary DSR is faced with a vast array of technical paths and possibilities.

For each inner or outer concept, there is variation to consider in its home environment and a set of inter-related variation in the other environment. This duality challenge is not present in organizational theorizing. The duality amplifies Weber's (1987) original concern about behavior; further, the situation is "worse" in that the technical possibilities and people's ability to appropriate those technologies for completely different and new uses have dramatically increased in the last two decades. Yet, the duality is unique to DSR, and ignoring it in design theory risks losing relevance or rigor. For example, the concept of social networks (an organizational theory from the outer environment) is a more insightful way to demonstrate the usefulness of a new social media feature (a new innovation in the inner environment) than a generic attribute (e.g., ease of use). Similarly, there are so many toolsets, software and hardware platforms, networking layers, available libraries, and capabilities that have already been built in the inner environment that it is a challenge to pinpoint the key design properties and architecture and instantiate the design in an artifact (i.e., to even get it to work with all these components). The instantiation is an important event in itself and needs to be recognized as such.

The duality is also an opportunity because each environment can constrain the other. Simon (1996) suggests a "quasi-independence" between the inner and outer environments. The need to consider both does not mean that they are the same or that they need to be integrated. According to Simon, the environments' independence is important because it is time consuming and difficult to understand the details and nuances of both environments and because it affords the opportunity to identify a few key properties of the inner environment (while ignoring others) that influence the outer environment and vice versa. These key properties can lead to simplified models that relate the inner and outer environment and convenient levels of abstraction. That is why we can have safe and reliable bridges by focusing on a key properties (e.g., per Simon) without fully understanding how elementary atomic particles work. These key properties have been termed "strategic hypothesis"—the hypotheses that count (Dubin, 1978; Weber, 1987), while Weick (1989) identifies them as the theoretical conjectures that are "interesting".

Simon's (1996) notion of quasi-independence along with his insight to focus on a few key properties and simplified models provides a path to start addressing the duality challenge by inter-relating design with the outer environment and constraining the theoretical and technical options to a few key properties. It is these few key properties that lead to parsimonious and strategic design theory.

Simon's (1996) concept of the "generator-test cycle" provides a tool to guide the path to design theory. The generator-test cycle manages and leverages the tension between the inner and outer environments:

> *… think of 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 (that come from the inner and outer environments). There need not be merely a single*

[2] Gordon Moore described a key trend in computing: the number of transistors that can be placed inexpensively on an integrated circuit doubles about every two years (Moore, 1965). To highlight a parallel complementary trend, I added the "double loop" qualifier: productivity increases through improved toolsets *and* the availability of new libraries with new capabilities, which creates a self-reinforcing loop that provides DSR with an ever-increasing array of possibilities.

> *generate-test cycle, but there can be a whole nested series of such cycles.* (Simon, 1996, p. 128-129).

The generator-test cycle is what is commonly referred to as prototyping in contemporary software design; it includes the key characteristics of prototyping: intelligence, design, and implementation (Janson & Smith, 1985). Applying prototyping to DSR is not new. Hevner et al. (2004) assert that design science is fundamentally iterative. Markus, Majchrzak, and Gasser (2002) created 70 functional prototypes and outline an iterative "hypothesis development" strategy. Using iteration and feedback is also assumed in the DSR approaches proposed by Peffers et al. (2008) and as a phase in action design research (Sein et al., 2011). Further, scholars see theorizing in general as an iterative process (cf. Shepherd & Sutcliffe, 2011; Weick, 1989; Bourgeois, 1979). Weick (1989) places "trial-and-error" thinking as a key attribute of theorizing and terms each iteration a "thought trial". However, in current DSR, prototyping (or another similar process) is often assumed to operate in the background as an important but taken-for-granted process. More generally, in IS, prototyping has typically been seen as (a) a tool for improving the speed of systems development in organizations and (b) as a tool to communicate with users.

Applying prototyping as a research process[3] to generate design theory is new and requires formal elaboration. Describing the theorizing process more explicitly and using it more self-consciously is an important step to improving theory construction (Weick, 1989). Doing so will allow more self-conscious use and deeper explanation.

According to Naumann and Jenkins (1982) and Janson and Smith (1985), prototyping includes five key processes, which, when combined, serve to separate it from other methods: (a) selection of design attributes, (b) construction, (c) evaluation, (d) a feedback loop that feeds a cycle of frequent iteration, and (e) frequent iteration itself. Figure 1 presents a formalized process based on prototyping to generate design theory.

Goals, existing kernel theory, and existing artifacts serve to constrain the process and provide tools to move it forward. The prototyping cycle (depicted by the dotted circle) includes design, evaluation, and appropriation/generation. The design theory box is the outcome of the process. The arrangement and implied "steps" of Figure 1 are similar in the broad sense to other models of theory construction (e.g., Weick, 1989; Bourgeois, 1979). What is different is (a) the formal use of kernel theory and existing artifacts, (b) the rapid yet formalized concurrent iteration among design, evaluation, and appropriation/generation, and (c) the inclusion of appropriation/generation in the process as a separate activity that is different from design and evaluation.

Sculpting provides a useful analogy for thinking about design theory generation. In this paragraph, I present an idealized process (and its DSR equivalent) that a sculptor may follow to create new art. The process will often start with broad and often divergent *goals* that may include size, type, form, material, location, audience, purpose, and so on. These goals then drive search for relevant concepts—analogous to reviewing *kernel theory*—such as reviewing theories of representation from the literature and researching construction techniques. In parallel, the sculptor may seek inspiration and review existing sculptures—*artifacts*—relevant to their goals, and this may also include collecting objects that are incorporated later into the sculpture. The actual act of sculpting is then an inter-related process in which ideas for the sculpture (the whole and pieces) are *appropriated and/or generated,* which are then instantiated—*design*—in specific prototype artifacts. In turn, these artifacts are evaluated against the goals, and the results of the evaluation lead to a new round of appropriation and generation. The goals are often changed as a result of the constraints and affordances of the

[3] Baskerville and Wood-Harper (1998) include prototyping in action research. Yet, the use is consistent with the "communication" application in which "prototypes surmount the esoteric nature of system design descriptions by presenting a working model of a specification, and allow the user to understand and comment on the design" (p. 99). Prototyping with research goals (which may or may not come from organizational settings) whose purpose is to add to the scientific body of knowledge and in which the process of iteration refines knowledge (as opposed to achieving an intervention) does not fit action research. It is more similar to interpretive research where the hermeneutic circle is analogous to iteration and grounded theory is analogous to appropriating and generating propositions.

material, inspiration from the act of design, and new insights from evaluation. This process iterates until a complete and coherent model—*the design theory*—forms that satisfies the final goals.

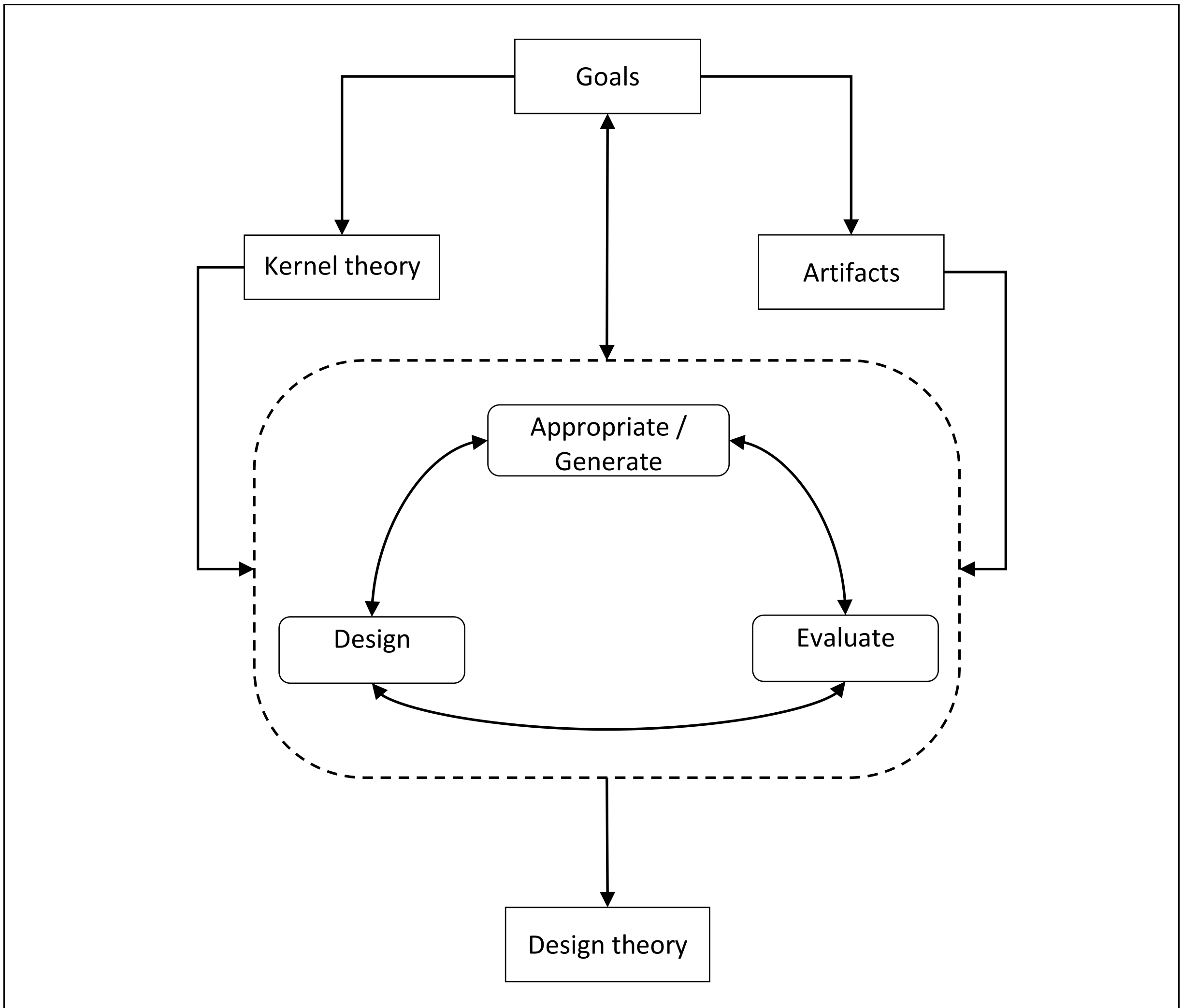

Figure 1. Generating Design Theory

3.2. Goals

Goals are part of the outer environment and represent research goals and the underlying research problem. A goal statement is analogous to the research problem statement in the natural and social sciences but is different in that the orientation is on synthesis (Simon, 1996) (i.e., creating a prescriptive design theory, in contrast to natural science where the orientation is typically on descriptive analysis) (Gregor & Hevner, 2013). Goals, like the goals of a sculptor working on a new piece of art, thus serve as inspiration that directs the process rather than as a set of criteria that may come from a problem statement. The double-sided arrow indicates that goals may change over time as a result of the process. This is a further difference from natural science in that, in synthesis when the focus is on creation, it is more common for the final goal to vary in response to serendipitous discovery. It is less common for the final goal to vary when one is analyzing a well-defined problem.

Organizational oriented routine design goals (i.e., "requirements") are excluded (Hevner et al., 2004). For such goals, the problem domain is usually well understood, artifacts are available, and it is clear how they should be created; there is no need for design theory. In contrast, the goal "improve the academic peer review process" (the research goal of the example presented in this paper) requires new design theory because we do not fully understand how to best change and improve the peer review process.

3.3. Kernel Theory

Kernel theory refers to formal theories in the natural and behavioral sciences, design theory, or normative heuristics from practice that can provide relevant constructs and measures from the outer environment. Scholars have identified such theories as important contributing elements of DSR (e.g., Walls et al., 1992; Hevner et al., 2004; Gregor & Jones, 2007; Iivari, 2007). Kernel theories are important because they are existing (e.g., they are documented and accessible in the knowledge base), valid (they've gone through a scientific validation process as part of being adding to the knowledge base), and relevant (the selected theory will have some self-evident role to play in the research). For the sculptor, kernel theories will provide insights into how to build something or how their object will be perceived.

Kernel theories can influence the design properties (e.g., speech act theory was directly implemented into the Coordinator system) (Flores, Graves, Hartfield, & Winograd, 1988) or evaluation (e.g., social network analysis has been used to study social media). Each role is different in timing (e.g., when the kernel theory is needed / applied) and importance (e.g., how critical the kernel theory is for a project's success). When a kernel theory is applied to guide the design, it significantly reduces the challenge of the outer environment because it can influence every aspect of the process including the goals, specific design properties, and evaluation metrics. Thus, the important research goal is less about identifying the fundamental design properties and more about the specific design theory (analogous to specifying "mid-level" constructs; see Weber, 2012) needed to realize the vision provided by the kernel theory. In practice, it is unlikely that there are many kernel theories available that can be successfully applied in such a "top-down" manner.

Conversely, kernel theories could have little or no role in the design but have a major role in evaluation (e.g., using social capital theory to evaluate knowledge management systems). For completely novel artifacts, which represent goals that have no grounding in current knowledge, it is possible that no kernel theory will be relevant or available and new concepts will be required to fully assess the artifact's usefulness.

In both the above situations, inventing the key design properties will follow the creative vision and synthetic process of the designer and the inner environment will only sporadically sample the outer environment to remain relevant. The more realistic scenario is that kernel theories will influence a portion of the project (either the design or the evaluation or both), and further multiple kernel theories may be required (Pries-Heje & Baskerville, 2008). For instance, group decision making system (GDSS) research followed this path in which specific design and evaluation concepts were, over time, appropriated from kernel theory (e.g., social psychology literature on anonymity). Similarly, in the example in this paper, business process theory is one of several kernel theories that provide the basis to improve the peer-review process.

Explicitly using the body of knowledge that goes beyond the normal research requirement of "reviewing previous literature" is also gaining ground in organizational theorizing (see Hassard, Wolfram Cox, & Rowlinson, 2013). For example, Shepherd and Sutcliff (2011) outline a top-down approach that applies induction and abduction to iteratively compare representations of interesting tensions, oppositions, and contradictions in the "literature" to develop new theory. In Shepherd and Sutcliff (2011), the literature is the "data". In design theorizing, there are multiple sources of data, including kernel theory and each successive version of the prototype and its associated evaluations. Existing artifacts also serve as a source of "data" in design theorizing.

3.4. Artifacts

Artifacts represent existing, feasible, and relevant knowledge about the inner environment. Existing artifacts provide tools and widgets that support the formulation of design concepts, which, in turn, lead to design theory. For instance, including an accelerometer and associated libraries in the iPhone allowed the envisioning of completely new motion-based software designs. It is likely very difficult and impractical today to build an artifact completely from the ground-up without reusing existing artifacts. Similarly, a sculptor will likely not create their own materials or baseline components. Further, building new artifacts on top of existing artifacts constrains design to what is feasible and it demonstrates rigor in that the new design properties are feasible. Selecting and using existing artifacts is one-step toward addressing the Moore's law double-loop challenge described earlier in which design scientists face an ever-increasing range of possible technological options. Using existing artifacts allows the researcher to leverage previous work and more precisely focus in on the design properties and corresponding design theory that are new and interesting. In rare situations, it is possible that a completely new artifact will stand on its own and not leverage existing artifacts.

The above encompasses the computational, tool, proxy, and ensemble views of artifacts in the literature (Orlikowski & Iacono, 2001). However, consistent with the implication in Orlikowski and Iacono (2001) and the formal explication of Purao, Henfridsson, Rossi, and Sein (2013), I assume that all artifacts are inherently "ensemble artifacts". I consider the issue of when and how the design of artifacts takes on an ensemble perspective in Section 3.5.3.

There is no direct analogue in the organizational literature of leveraging artifacts into the theorizing process. According to Boxenbaum and Rouleau (2011), empirical material, theoretical concepts, and more recently metaphors are the building blocks of new theories. In DSR, these building blocks come in with kernel theory; however, existing artifacts are closely aligned with metaphors in that they represent a coherent representation of technical knowledge arranged into a scaffolding (the design) that can be referenced. Suddaby et al. (2011) note that assembling multiple metaphors from a wide range of literature enables theoretical creativity. Similarly, in DSR, assembling a wide range of design artifacts that are relevant to one's goals will enhance the novelty of the resulting design theory. Boxenbaum and Rouleau (2011) also add the idea of epistemic scripts, which categorize the way scholars persuade others that they have produced something new. Epistemic scripts include evolution (new insight added to the existing body of knowledge), differentiation (a new paradigm), and bricolage (assembly of available elements). The bricolage concept is similar to the idea of assembling and integrating existing artifacts (and kernel theory) to form a coherent new whole—a design theory. I discuss the process of arriving at the design theory in Section 3.5.

3.5. The Prototyping Cycle

The prototyping cycle is Simon's (1996) "generate-test" cycle where the iteration occurs in the constraints provided by the goals, existing kernel theory, and existing artifacts, and it is through generating alternatives that the design theory is eventually developed. The sculptor building and discarding mock-ups or actual artifacts follows a similar process. Further, the "thought trials" in organizational theory development (Weick, 1989) are similar except for two major differences: (a) the duality challenge of DSR increases the range of possibilities, which, while increasing complexity, also likely increases the frequency of and heterogeneity among cycles (an important requirement in organizational theorizing; Weick 1989; Astley & Van de Ven 1983); and (b) the process is much more formalized and includes considering representation, frequency, selection, and stopping rules. In addition, unlike organizational theory development, a formalized prototyping cycle can manage and leverage the tension between the inner and outer environments to generate relevant and feasible theory via three inter-related activities: design, evaluation, and appropriation/generation of (new or revised) designs and evaluation schemes.

Even though the terminology and mechanics of prototyping may seem familiar, the application here is different. Using the terms from Nunamaker et al. (1991), the act of prototype construction is the vehicle (as opposed to the end result) from which the "conceptual framework" is created, the "system architecture" is specified, and the overall system is designed. Further, the evaluation process is inter-related into the construction of the prototype, and the results of evaluation drive the construction of each successive design and refinements to the evaluation itself. This is important because iteration supported by structure and constraints becomes the driver for successively improved versions of the artifact *and* evaluation. This is the key toward arriving at the "strategic hypothesis". Design, evaluation, and appropriate /generate are key elements of the prototyping cycle and drive convergence toward the strategic hypothesis.

3.5.1. Design

Design involves quickly creating prototypes that, in each iteration, get closer to meeting the goals. In other words, it is analogous to the act of sculpting. The act of design in DSR is different from organizational theorizing because it blends top-down deduction with bottom-up induction; it is deductive in that it often starts with a vision that is applied to a specific instance, and it is inductive in that each successive design and the results of successive evaluation provide a corpus to induce new and divergent visions. Ultimately, the design process is about "finding alternatives" (Simon, 1996, p. 121). These alternate prototypes include mock-ups created with storyboarding or prototyping tools, screen designs, simulations, and traditional programming. Drawings are useful when they are constrained to the context (e.g., drawing an iPhone app on 4x6 index cards to mimic the form factor restrictions of a smartphone and then linking them together with tape to mimic interactivity). In relation to organizational theorizing, the prototypes are analogous to "representations" (Shepherd & Sutcliffe, 2011) of what is currently theoretically interesting. The design process will likely produce multiple versions of these representations and these versions will also be useful in demonstrating convergence to the strategic hypothesis (see below).

Design prototypes are a key unique characteristic of DSR: the act of successfully creating the prototype demonstrates rigor and feasibility because it is in the context of physical and natural laws (Simon, 1996). The prototype is much more tangible than the "imaginary experiments" that most organizational theorists (e.g., Weick, 1989) suggest. The tangible elements include the laws of mathematics and properties of electrical currents, manufacturing constraints, and availability of previous artifacts. For example, a button on a screen is constrained by what can be programmed, which is constrained by the programming language, which, in turn, is constrained by the available application programming interface (API) and selected ecosystem, the underlying operating system and network stack, and, ultimately, by the hardware's physical properties. When designers overcome these constraints, they have already made a significant step toward showing rigor in demonstrating the feasibility of solving the research problem. It is what Nunamaker et al. (1991) terms "proof-by-demonstration". The proof by demonstration is important because it shows that a) the ideas in the inner environment are feasible and (b) the accumulation of a set of proven feasible ideas converges the process of generating design theory. Proof by explicit demonstration in DSR is an important difference with other forms of theorizing; the resulting artifacts and associated design concepts are signposts that "forcibly carve out" (Chia, 2000) the implicit embedded concepts and assumptions that are a challenge in organizational theorizing (see Suddaby et al., 2011). Thus, design theorizing provides a new toolkit for organizational theorists who are struggling to identify important concepts in the context of technology use. After all, it is through information technology that many new organizational innovations and changes are enacted. Yet, Lukyanenko and Parsons (2013) caution that such constraints can be an anchor that reduces abstractness and produces (only) mid-range theories. However, the level of the resulting design theory is also significantly influenced by how the artifact fits into the environment and by its evaluation.

3.5.2. Evaluate

Evaluation is integral to generating design theory and eventually producing "justificatory knowledge". Evaluating involves the sculptor stepping back and reviewing the work to see how it meets the goals and considering lighting, the viewing context, and angles. Evaluation in the prototyping cycle is not the same as formally validating a theory. It would be premature to rush each iteration into a formal validation process. Rather, evaluation's purpose is to answer the question: are the design concepts feasible and interesting? Thus, evaluation's purpose is to converge the prototyping cycle into a design theory that can later be validated in a new cycle of research.

Feasibility is a valuable natural constraint of DSR, and it serves to ground and direct future theorizing. If, given current technology and use context, a potential design concept is not feasible, then the evaluation is negative and the researcher will move to a different idea or retry the idea using a different approach. Thus, feasibility provides a specific and tangible constraint in DSR in contrast to organizational theorizing where the constraints are limited to mental experiments. Consequently, in design theorizing, the need for frequency and diversity of evaluation criteria is moderated by the availability of this natural constraint. According to Weick (1989, p. 516):

> *theorizing is viewed as disciplined imagination, where the "discipline" in theorizing comes from consistent application of selection criteria to trial-and-error thinking and the "imagination" in theorizing comes from deliberate diversity introduced into the problem statements, thought trials, and selection criteria that comprise that thinking.*

In DSR, problem statements relate to goals, thought trials to design, and selection criteria to evaluation.

The “interesting” question can be addressed by both empirical and conceptual techniques. Empirical evaluation in DSR is typically less rigorous than formal testing and can leverage multiple techniques such as observation, user interviews, and short surveys (see Hevner et al., 2004, for a comprehensive list of evaluation techniques). However, empirical evaluation is expensive. Therefore, conceptual evaluation such as leveraging the researcher’s intuition about what is interesting (especially when the research and development team self-use the design) can serve as an initial filter on whether a design concept is even worth empirically evaluating. Weick (1989) outlines selection criteria for assessing the plausibility of conjectures. In the DSR context, these criteria are:

- Interesting: is the design new in relation to previous kernel theory and artifacts?
- Obvious: has this been done before? Is it an embedded design facet that has been overlooked? Is it new in a different context?
- Connected: is it similar to a different problem, artifact, and kernel theory issue? How is this design concept related to a previous design concept?
- Believable: does the design concept fit the context? The back story?
- Beautiful: is the design elegant? Does it solve a problem with parsimony?
- Real: will it solve a specific real world problem? Will it integrate with existing artifacts?

Overall, we can expect that the evaluation process will iterate side-by-side with design and appropriation/generation (see below) and produce multiple results and that these results will be useful in demonstrating convergence to the strategic hypothesis (see below).

Each evaluation step moves the project closer to relevance[4] in that it becomes clearer where the artifact fits into the outer environment. To do that, the evaluation will also need to go beyond testing performance or ease of use; these concepts are important baseline measures, but they will not fully explain *how and why* a particular design property influences the outer environment; they will not explain the “value proposition” of a new design. According to Hovorka and Pries-Heje (2013), it is important to surface a more complete justification of design theory. To do that, often new or newly appropriated concepts are needed to fully understand why a design is useful (e.g., the concept of social capital is a more insightful way to understand the use of social media sites in business than just “ease of use”). So substantive evaluation is tied to identifying an appropriate kernel theory in the natural and behavioral sciences, design theory, or normative heuristics from practice that can provide relevant constructs and measures. If a kernel theory is not available, then the process will rely on the designer’s intuition to identify the key evaluation concepts (see Section 3.5.3).

[4] Relevance in this paper is an outer environment issue and refers to usefulness, applicability, and utility, i.e., the “practical significance” (Straub & Ang, 2009) of an artifact. Rigor, on the other hand, is an inner environment issue and successfully creating an artifact is a major step in demonstrating rigor. Software engineering techniques that assure whether a piece of code works and meets specifications, can contribute to further demonstrating rigor. Evaluation and relevance as applied here, however require consideration of the outer environment and testing. Further, as per Straub and Ang, involving practitioners is one tool to improve practical significance but it is not necessary or sufficient.

In relation to organizational theorizing, evaluation is a “test” of how well the current conceptualization addresses (“fits”) the goals (e.g., Shepherd & Sutcliffe, 2011). The greater the gap (Folger & Turillo, 1999), the more likely it will trigger further DSR “appropriate/generate” cycles.

3.5.3. Appropriate/Generate

Appropriate/generate is the last of the three inter-related activities of the “generate-test” cycle of Simon (1996)—what Simon calls assembling and searching. This involves searching and assembling—appropriating—existing designs and kernel theories (e.g., using an existing software library or an existing theory that can explain the proposed design). Given the challenges of synthesis, it is likely that the first set of selected design widgets or kernel theories will not fully satisfy the research goals. This does not mean that the widgets and kernel theories are poor or irrelevant: they are necessary first, second, third, steps for the process to move forward. It is also possible that existing designs and kernel theories may prove unsatisfactory. At this point, the researcher will leverage their informed intuition to generate new design and/or explanatory ideas, and these new concepts will likely also go through a period of refinement and iteration. For example, an identified kernel theory may need to be adapted to fit the context (Arazy, Kumar, & Shapira, 2010; Kuechler & Vaishnavi, 2012).

The act of appropriating/generating artifacts and explanatory concepts about the artifact “closes the loop” and inter-relates design (inner environment) with evaluation (outer environment) while maintaining quasi-independence. The process allows the research to iteratively evolve towards posing the right question: The imperative to build the next version of the artifact steps away from human concerns and toward design, while the imperative to evaluate that artifact steps away from design and toward human concerns. The continuous iteration of this dance away and toward human concerns is the path toward the most relevant design theory. It is also similar to the sculptor intensely focusing on sculpting, stepping back to review, appropriating/generating ideas, sculpting again, and so on.

Table 1 illustrates this process from the perspective of appropriation/generation and provides DSR examples. The rows show three scenarios with respect to kernel theories: unknown/unavailable refers to situations where no applicable kernel theory is available (or has been identified) to explain and evaluate a particular artifact, appropriate refers to situations where existing kernel theory is identified and then appropriated to the new technology context, and generate refers to situations where new or extended explanatory concepts are proposed. The columns show three scenarios with respect to artifacts: whether the associated artifact has been identified as interesting from the research perspective (i.e., not every artifact will be deemed interesting), situations in which an artifact is appropriated into an interesting purpose (i.e., which is different from its original purpose), or whether the artifact must be created (or extended).

Table 1. DSR Projects

		Artifact		
		Identify	**Appropriate**	**Generate**
Kernel theory	**Unknown/ unavailable**	Type I *Exploratory research to identify what is important about an artifact* (e.g., ethnographic study on how firms use Facebook)	Type IV *Apply interesting artifacts to specific problems and look for explanations later* (e.g., apply consumer social media inside a business)	Type VII *Invent new artifacts and look for explanations later* (e.g., invent a new kind of group support system)
	Appropriate	Type II *Formal evaluation of artifacts using specific theories* (e.g., apply cognitive learning theories to evaluate the ease of use of Windows 8)	Type V *Investigate the appropriation of specific artifacts and associated explanations* (e.g., customize social media to higher education and explain using cognitive learning theory)	Type VIII *Create artifacts that apply specific theories* (e.g., create a new kind of social networking platform by applying and adapting social capital theory)
	Generate	Type III *Develop concepts to explain specific artifacts* (e.g., explain the impact of blogging on politics)	Type VI *Adapt artifacts and create explanations* (e.g., propose how social media can be appropriated to higher education; propose concepts to explain use)	Type IX *Create new artifacts and explanations* (e.g., propose a new system for academic peer review; propose new concepts to explain the process changes)

Going across the columns, types I, II, and III are likely not applicable to DSR and are listed for comparison and completeness. In types IV, V, and VI, the focus is on appropriating existing artifacts (e.g., customizing an artifact to a new purpose), while in types VII, VIII, and IX, the focus is on generating new artifacts (e.g., developing a new system). The distinction between appropriation and generation is of intent: is the intent to reuse or is the intent to start afresh? Both sets of cells could involve substantial development effort.

Going down the rows, in types IV and VII, the focus is on innovation, which means more of a focus on the inner environment and less concern about the outer environment. The design theory task is more focused on identifying the major design concepts needed to instantiate an intuition (e.g., the design principle of columns and rows to manipulate numerical data in spreadsheets). Such projects will likely result in level 1 design theory (Gregor & Hevner, 2013) and later require a new cycle of research to explain use and relevance (e.g., types I, II, or III). In types V and VIII, the problem space is likely not new and existing kernel theory is available to guide the explanation. Further, in type V, existing artifacts are also available. Such projects suggest collaboration with experienced researchers versed in the relevant literature, problem space, and available artifacts. In types VI and IX, the focus on both innovation and explanation will likely require substantial resources for researching the inner and outer environments (e.g., developers working with social scientists in a field setting (Sein et al., 2011)).

The intent in types VI and IX (and to a lesser extent in type VIII) is to generate new design theory and explanatory concepts. Beyond generating the design concepts, there is an opportunity for the researcher to leverage their close-to-the-phenomena understanding to improve the evaluation by explicitly identifying intuitions ("design claims"; Kraut & Resnick, 2012) about the outer environment. These intuitions can be represented as candidate outer environment propositions similar in style to the theoretical propositions common in social science research. The candidate propositions do not replace the later development of kernel theory; however, they will significantly reduce the search cost of identifying relevant concepts that later go into formal theory development that can become kernel theory.

Table 1 provides signposts for starting, evolving, and ending appropriation/generation. I do not include it to limit DSR to specific boxes or to value one type of DSR over another. Table 1 encompasses the tool, proxy, ensemble, and computational views of artifacts and ignores the nominal (Orlikowski & Iacono, 2001). For a given unit of research, the computational view is more aligned with types IV and VII, the tool and proxy view with types V and VIII, and the ensemble aligns with VI and IX. I base these associations on the assumption that the ensemble view is the most socially nuanced and comprehensive, and full realization may likely require new kernel theory. However, since it is possible that a comprehensive theory could exist to fully realize the ensemble view in for example type VII, the above characterizations are potential starting signposts as opposed to normative end points. A specific DSR project may start in one cell and, over time and through maturity of the problem context and artifacts, move to another cell (see Figure 3 in Gregor & Hevner, 2013). All artifacts are ensemble artifacts; that is, they carry "traces of the organizational and social domain" (Purao et al., 2013, p. 76), but units of research will focus on different views and, over time, the knowledgebase will evolve to the ensemble view, especially for the most interesting and enduring artifacts. The problem context is also important (e.g., complex problems with no obvious solution or kernel theory may need to go through several computational cycles in type VII). Finally, the setting is also an important constraint. In a lab, the project will likely have access to a greater set of technical resources and freedom to explore, while, in an organizational setting, there may be more specific business value and temporal constraints but more opportunity to evaluate the full social context of the project (e.g., Sein et al., 2011).

Suddaby et al. (2011) map extant organizational theorizing approaches into a 2x2 matrix in which the rows represent implicit assumptions versus explicit (known) constructs, while the columns represent theorizing in a literature domain versus across multiple domains. In Table 1 above, appropriating kernel theory relates to working with explicit constructs (types V and VIII), while generating kernel theory most closely corresponds to working with implicit assumptions (types VIII and IX). DSR can use the Suddaby et al. matrix to identify additional appropriation and generation strategies. For example, Suddaby et al. suggest applying Tsang and Ellsaesser's (2011) contrastive strategy when operating in a literature domain and when the constructs are explicit. The contrastive strategy is based on comparing the explanatory power of current concepts with alternative concepts. In DSR, this could apply to a type VIII project in which more than one theory is applicable and multiple design artifacts are generated and contrasted to assess the relevance and utility of each theory. In addition, Lewis and Grimes (1999) summarize approaches when operating across paradigms. One such approach—bracketing—is particularly relevant when the researcher is faced with appropriating or generating new theory from completely different perspectives (e.g., type VIII appropriation in social media research where the goal is to apply both economic and altruism theories).

Langley (1999) and Langley, Smallman, Tsoukas, and Van De Ven (2013) summarize strategies for analyzing process data. The design theorizing approach I present in this paper produces process data at each iteration, but, unlike real-world temporal contexts (e.g., stock prices), the relationship between iterations is only important for validating the scientific method, they are not analogous to events in the goal and problem space. The goals and problem space driving DSR may lend themselves to process or variance theorizing or, more likely, to aspects of both. Langley also notes how process theorizing can lead to variance theorizing. Therefore, I present Langley's strategies below in the context of DSR goals, and they can be applied to further enhance appropriation and generation:

- Types IV and VII, where the intent is to innovate will benefit from organizing strategies such as visual mapping (screens) and narratives (storyboards) that can make sense of many different potentially divergent ideas.

- Types V and VIII, where the intent is to apply existing ideas will benefit from replicating strategies such as temporal bracketing (compare versions and ideas) and quantification (create benchmarks).

- Types VI and IX, where the intent is to both invent and explain will benefit from grounding strategies such as grounded theory (induce relevant design concepts) and alternate templates (compare competing design metaphors).

The sculpture project is now nearing a major milestone, the goals have been refined, both existing and new ideas have been appropriated and generated and tried out and refined in prototypes that change through successive cycle of design and evaluation. It is now time to put forward a coherent design theory that satisfies the goals.

3.6. Design Theory

Gregor and Jones (2007) argue that the formal knowledge that follows from specifying and elaborating a design theory is important because it (a) raises the IS field above a craft, (b) adds to rigor and legitimacy, and (c) supports the cumulative building of knowledge that can only come if new insights and ideas are elaborated as knowledge that can then be classified and compared. Gregor and Jones provide the structure for reporting the resultant design theory; the prototyping cycle and associated structured and processes I describe above provides the means to generate the design theory. Therefore, at this point in the sculpture project, the researcher can apply Gregor and Jones to specify a candidate design theory. What is still unclear is how to identify the "best" candidate design theory and stop the iteration that is inherent in the prototyping cycle. What is a complete unit of research that is publishable and that can legitimately be added to the body of knowledge?

The prototyping cycle is important because it allows evolution toward posing the right question that is relevant to the outer environment rather than solving an obvious or uninteresting problem. Moreover, the successive refinement and identification leads to the most novel and useful design properties in the inner environment. The narrowing down is important because DSR has to contend with the duality challenge I discuss earlier: the pot of possibilities of both behavior and technology keeps growing. Thus, it is likely that the first version of any design or associated design theory will require further refinement. The iteration, evaluation, and refinement process provides the basis to stop the cycle, clues on how to generate the next alternative (if needed), and clues on when a complete unit of research has been achieved (the "strategic hypotheses") (Dubin, 1978; Weber, 1987).

In DSR, two potential indicators of achieving the strategic hypothesis are (a) the quality of the design concepts and (b) the validity and reliability of the design theory.

The design concepts are typically presented as design propositions including truth statements (e.g., "a bulletin board can represent structured communication"), annotated screen prints that show new kinds of widgets or representations, algorithms (e.g., the grammar of SQL), or design principles (e.g., the Web 2.0 principle of interactivity). Design concepts are similar to "constructs" for which Suddaby (2010) suggests four quality criteria: 1) definitions that are comprehensive, precise, and parsimonious; 2) good scope so that the constructs, their relationships, and where, when, and to whom they apply to is clear; 3) Logical consistency so that the rhetoric underlying the constructs is consistent and integrative; and 4) Understandable relationship among constructs so that the lineage and position in relation to related concepts is clear. Good design propositions should also exhibit the above criteria. In addition, a fifth criteria unique to DSR is feasibility given the constraints of technology.

Another indicator of quality and of achieving the strategic hypothesis is attaining a certain baseline level of validity and reliability. However, the very essence of prototyping promotes exploration at the expense of the formal controls that come with making a process more systematic, observable, valid, and reliable. Yin (1989) outlines three principles for increasing the validity and reliability of case studies. Case study research and DSR share one key attribute: they both operate in situations where the researcher often has little control. Table 2 lists each of Yin's principles and DSR examples. One of the challenges of technical research is that developers often find it difficult to elaborate the rationale for why the artifact works a particular way. In the proposed approach described in Table 2, previous versions of prototypes and results of evaluations show why and how a particular design was discarded or selected. This is important for two reasons: 1) when portions of the design are discarded and other portions are successively reused, the reuse demonstrates logical convergence toward the strategic hypothesis; and 2) the artifact and evaluation databases provide a knowledge base for refinement and convergence of the current (and future) research.

In sum, following Yin's (1989) principles for increasing validity and reliability mean that the design theory will include the final set of design and explanatory propositions and the justificatory knowledge demonstrating and validating each of the major previous iterations. Large projects following formal project management techniques already apply aspects of Table 2; however, their focus is limited to establishing management controls and completing requirements. In relation to organizational theorizing, Table 2 represents a more structured application of Eisenhardt's (1989) "field notes", promotes the consistency that Weick calls for in thought trials and selection criteria (1989), and places the application of Suddaby's (2010) suggestions for improving construct quality in a larger system of strategies.

Table 2. DSR Validity and Reliability

Principles for collecting case study data (Yin, 1989)	DSR principles	Examples
Use multiple sources of evidence (e.g., archival records, direct observation, interviews)	Generate multiple versions of prototypes. Use multiple forms and sources of evaluation.	Use a rapid prototyping tool to create new versions. Observe use of each new version and supplement with interviews.
Create a case study database (e.g., self-notes, documents, tabular data, narratives)	Maintain an artifact database to archive each interesting version of the prototype (or relevant portion). Maintain an evaluation database to archive interesting evaluation methods and results.	Use a software library tool to save each iteration. Save and time stamp key screen prints. Use time stamped documents to save evaluation results.
Maintain a chain of evidence (e.g., use cross referencing so that conclusions can be traced to specific data elements)	Record the reasons why specific designs are discarded in the artifact database. Record the reasons that led to specific improvements and link to specific evaluation results in the evaluation database.	Annotate screen prints or evaluation results to record reasons for changes to the previous design or evaluation model.

Applying Yin's (1989) principles and Suddaby's (2010) criteria increase rigor and support convergence toward the strategic hypothesis, but what constitutes a complete cycle? How do researchers know that they have a candidate strategic hypothesis? In routine development, the obvious solution is to stop when the requirements are met. However, this does not work for research projects in which goals are understandably vague and involve envisioning new artifacts. Recognizing this dilemma, Simon (1996) proposes three strategies: optimization (applying techniques such as linear programming and control theory[5]), satisficing (accepting a satisfactory solution), and cost benefit analysis.

Kuhn's (1970) notion of anomalies provides an approach to satisfice. Kuhn outlines the process of scientific revolution as being sparked by the perception of an anomaly. The normal cycle of iteration and collision of inner and outer environment interests can be a key ingredient in perceiving these "anomalies". Normal science plays a critical role in this process because working with standard tests

[5] Optimization is not applicable for interactive digital artifacts that integrate a multitude of difficult-to-specify human concerns.

and procedures leads to a level of detail and to a precision that prepares the researcher to know what to expect; this person is best equipped to know when something has gone “wrong” (or “right”) and best equipped to identify true and interesting anomalies. In other words, the generate-test cycle continues to operate in normal science mode by revising prototypes and fine tuning evaluation until an anomaly is perceived (i.e., an important new design concept and/or the crossing of an important evaluation threshold). That is a signal to pause the cycle of prototyping. This pause will represent the project’s or one of its important milestone’s (“version”) completion. Finally, intellectual and pragmatic cost/benefit resource concerns will govern if enough anomalies of sufficient interest have been identified to label the project as one complete unit of research.

Anomalies, doubt, and tension have always been used in organizational theorizing as a mechanism to trigger the identification of “interesting” concepts and relationships (e.g., Weick, 1996; Locke, Golden-Biddle, & Feldman, 2008; Shepherd & Sutcliffe, 2011).

Figure 2 provides a simple structure to represent the iterations and, as Hovorka and Pries-Heje (2013) suggest, “over-specify” the design theory. The design and explanation columns represent the inner and outer environments, respectively, and the evaluation and decision columns document the relationship among the environments. As rows are added and the prototyping process unfolds, the relationships among the cells are based on forms of reasoning including abduction, induction, triangulation, and reflection (see Kuechler & Vaishnavi, 2012, for a complete list), and the relationships increase validity (shown by the grey dotted box) and leads to a strategic hypothesis. Improvements in evaluation increase falsifiability and the decisions provide evidence of the process. As the N increases, the value of contrastive approaches to compare the iterations (see Tsang & Ellsaesser, 2011) and the ability to abstract from the particular to the general—to see “what is this a case of?” (Tsoukas, 2009, p. 298)—increases.

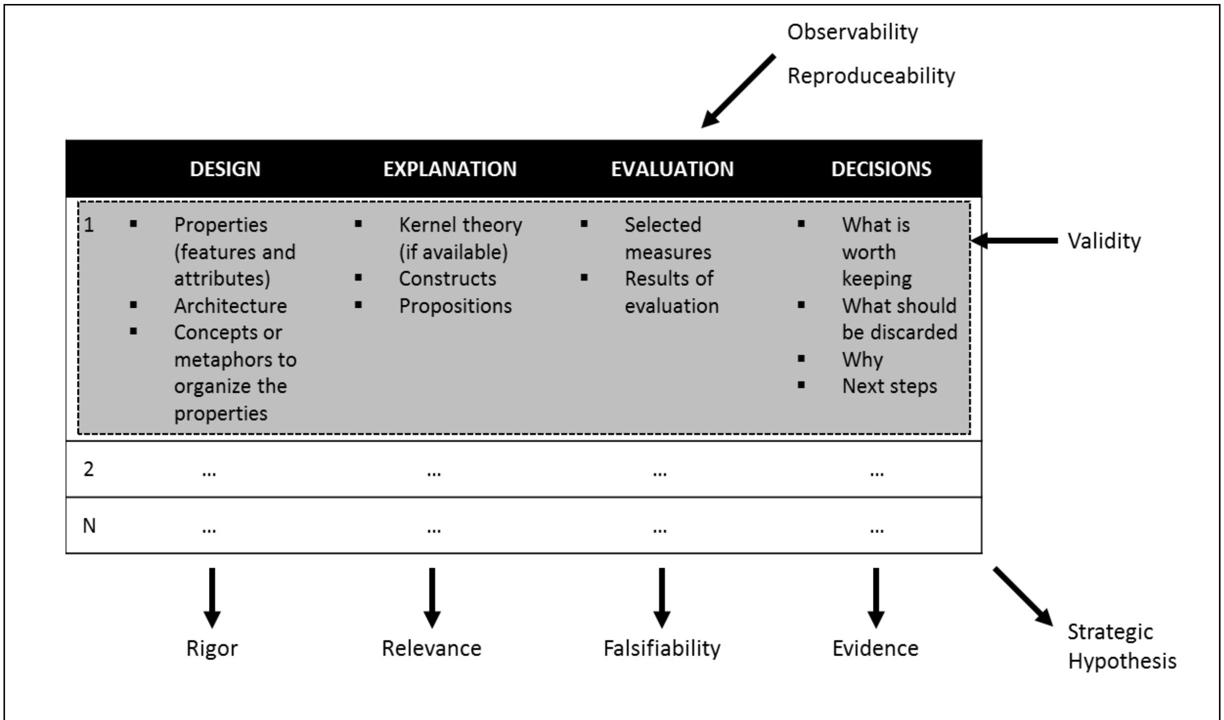

Figure 1. Design Theory Iterations

The annotations in the Figure 1 apply Weber’s (2012) theory quality criteria to show observability, reproducibility, validity, rigor, relevance, falsifiability, evidence, and strategic hypothesis. Specifically, each iteration (showed by the N column) of design increases rigor, while the associated iteration in explanation increases relevance. The easy-to-understand structure increases observability and reproducibility. In sum, the figure describes the design theory and includes the iterations including the purpose and scope, constructs, principles of form and function, testable propositions, and justificatory knowledge (Gregor & Jones, 2007).

To summarize, goals, existing artifacts, and existing kernel theory constrain and support the prototyping cycle, which, through inter-related activities of design, evaluation, and appropriation/generation, balances the inner and outer environments and goes through a constrained cycle of search and iteration. When the resulting design and evaluation propositions and concepts are of high quality and they reach the status of a "strategic hypothesis" based on the rules of satisficing, perception of anomalies, and cost benefit analysis, the process is complete. Completing and identifying a strategic hypothesis is aided in the above process by (a) identifying key properties, (b) iterating and generating multiple versions to converge on the key properties, (c) iterating and evaluating provide clues for the need for, and the direction of the next iteration, and finally (d) accumulating and codifying provide the chain of evidence needed to justify the results. A design theory is produced that includes the instantiation of the artifact and the justification of the final design and evaluation concepts. In Section, 4, I show a specific application of the above ideas.

4. Example—An application

This section provides an example that applies the proposed processes and structures for generating design theory in DSR. Mandviwalla, Patnayakuni, & Schuff (2008) applied a combined behavioral and design approach to improve the academic peer-review process using IT. Table 3 analyzes four different iterations of their project. The table illustrates how the prototype evolved and how each of the four iterations provided new insights. There were many more iterations; the table only focuses on important milestones—the "anomalies". The key design concepts from each iteration were saved for reference purposes. Screen prints of key user interface elements and the underlying code and database were also archived. The evaluation included: 1) an ad-hoc group of interested academics that provided feedback on ideas and tested specific designs, 2) system use data in four mini-tracks, three major conferences, and one online journal with more than five hundred "beta" users over a two-year period. The data include log files were as a form of observational data since the log captures each click, structured interviews, suggestions for improvements from "beta" users, and results of short questionnaires. At each major iteration, the usage data was analyzed and archived. The iterations and results of evaluations led to successive improvements in the design and explanatory concepts. I discuss four key iterations below.

1. Initially the designers focused on what now seems a simplistic view of peer review and automated the existing process. The strategic hypothesis was that peer review is transactional in nature and the focus was on improving the efficiency of transactions. Therefore, the key processes of peer review were analyzed and automated. Informal testing showed that the "features" were easy to use, and the project was thought to be complete. However, subsequent evaluation showed that these features were not compelling enough for the casual user. This led to a new cycle of prototyping.

2. In the next major iteration, the designers applied the discussion forum / bulletin board design concept to peer review and the results showed some promise. The strategic hypothesis was that peer review was really about facilitating communication (as opposed to conducting transactions). However, the evaluation showed that the design had the unintended consequence of increasing the perceived workload of one set of stakeholders without providing compelling advantages. So, again, the designers went back to the drawing board.

3. The next iteration was more radical in that the designers envisioned a complete change to the entire process of publishing based on dis-intermediating the knowledge producers and consumers from the publishers. For example, they envisioned and started building a "meta" site that would allow any conference or journal to interactively set up their peer-review site without any outside support or resources. However, the

design concept was difficult to implement and it was also hard for prospective users to understand. This led to increased need for technical support, which was not consistent with the original vision.

4. In this iteration, the designers went back to the bulletin board metaphor but used newly identified kernel theories of “mutual understanding” and “procedural justice” to constrain the interaction to optimize issues of fairness and value. The strategic hypothesis was a “structured communication process” that included user interface and management techniques to constrain and direct the communication among and between authors, reviewers, and editors. The concept of value and fairness came from iteration 2 when users expressed concern about workload, but were also excited about the potential to equalize perceived power differentials. Further, iteration 4 was able to use the transactional and customization design concepts from iterations 1 and 3 to improve the overall experience of using the prototype.

The iteration stopped when the propositions and associated concepts were stable and met the “strategic hypothesis” test (i.e., they represented interesting and novel explanations of how the new design met the goal of improving the peer-review process). While I show the iterations above in retrospective logical order, the actual sequence was less orderly. At one time, both iteration 2 and 3 were under parallel development and evaluation. Iteration 3 branched into a separate project that was later abandoned, while iteration 2 evolved into iteration 4. The concepts and evaluation measures also evolved in each iteration. Iteration 1 focused on business process redesign as a kernel theory, iteration 2 focused on social psychology and management theory (specifically theories on computer mediated communication), iteration 3 did not use a particular theory base, except for referencing various editorial proponents of new forms of “e-publishing”, and iteration 4 focused on appropriating procedural justice theory and economics for specific constructs and on communications to provide the overall guiding concept. The final list of propositions from iteration 4 are available in Mandviwalla et al. (2008).

Table 3. Design Theory Iterations

N	Design	Explanation	Evaluation	Decisions
1	• Transaction processing • Workflow management	Peer review as a business process that can be redesigned	• Key measures: usage, convenience, responsiveness, satisfaction • Positive response on the concept of improving the process. • Inconclusive on the specific prototype	• Process improvements were insufficient incentive to change a periodic and infrequent process
2	• Bulletin board metaphor (papers submitted and reviewed on a bulletin board) • Open (non-anonymous) communication between authors and reviewers and among reviewers	Publishing as a system of communication and control. • Reduce controls and increase communications to increase understanding which will increase inter-reviewer agreement, equalize power differentials • Above will encourage innovative research and increase quality	• Key measures: Ease of use, accountability, quality of review, quality of papers • Authors positive, reviewers and editors neutral. • Concern about workload	• Increasing communication is good for authors but for reviewers/editors the benefits are less obvious • Increase in workload is an obstacle • Increasing the power of authors is not congruent with the worldview of reviewers/editors • Publishers not interested in changing status quo

3	Adaptation
	• Web-based "setup wizard" allows creation of online journals and review process
	• Journals customize:
	  • Roles and workflow (e.g., associate editor, senior editor)
	  • Reviewing process (open vs. traditional)
		Disintermediation
		• "Skywriting", "e-publishing"
		• Publishing as an industry that can be improved by allowing actors to create and experiment with new journals and new reviewing models
			• Key measures: interest in using
			• Design concepts difficult to implement
			• Prospective users had difficulty in understanding the concepts
			• A web-based tool increases the need for help-desk style support
				• The goals were too utopian
				• Did not consider the "eco-system" and "business model" needed to sustain a new innovation
				• "Too much too quickly"

4	Structured communication process
	  • Common space (bulletin board)
	  • Interactive and parallel
	  • Hierarchical
	  • Stored
	  • Citation
		Peer review as a form of goal-directed communication (mutual understanding)
		• Procedural justice (fairness)
		• Knowledge production function (value)
			• Key measures: management, fairness, effort
			• Authors positive, reviewers slightly positive
			• Need to incorporate "learning" as an additional behavioral concept
				• Ready for formal evaluation

Overall, in the context of design theory, the project considered existing artifacts and kernel theory and leveraged the prototyping cycle to formally search for and create a series of artifacts that (a) created a new artifact, and (b) extended kernel theories to explain the underlying phenomena, and (c) produced a design theory. The design theory iterations were challenging in terms of time and effort involved; however, the tabular format was an important tool in capturing the underlying complexities and tensions between the inner and outer environment and in encouraging simplicity and clarity in communicating and reflecting on the tensions. Table 4 elaborates the design theory. Only the final version is included. For example, the initial purpose and scope was conceptualizing peer review as a business process, then as a form of communication and control, next as a hierarchy that needs to be disinter-mediated, and finally the broader view of goal-directed communication (see below).

Table 4. Design Theory (Based on Gregor & Jones, 2007)

Type	Component examples
Purpose and scope	The focus is peer review, the process by which new knowledge is legitimized by acceptance and dissemination to the wider (academic) community. Peer review is conceptualized as a form of goal directed communication.
Constructs	Structured communication to represent the scripted communication that occurs in peer review, fairness to represent procedural justice achieved by increasing transparency and accountability, mutual understanding to represent communication among actors, and knowledge production to refer to the quality of the final papers and learning.
Principles of form and function	The structured communication model links authors, referees, and editors in the cycle of submission, refereeing, editing, presentation, and discussion. The model uses common (open) spaces to increase transparency, a bulletin board style discussion forum supplemented by various structures (e.g., email reminders) to increase communication, and storing and referencing comments to increase accountability.
Artifact mutability	The evaluation found that “learning” (new ideas) was a useful unintended concept that led to proposed changes in the design. There was also a discussion to add the concept of document workflow.
Testable propositions	The design propositions include truth statements about implementing common spaces as a discussion forum, interactivity through a Web-based posting and commenting, visual hierarchy of posts and comments to represent the structured communication, storage, and the ability to reference (cite) conversations using hyperlinks.
Justificatory knowledge	The explanatory propositions include statements about increasing mutual understanding, procedural justice, and knowledge production.
Principles of implementation	The paper discusses the pros and cons of modifying an existing discussion forum to achieve the goals. The paper also discusses adoption challenges.
Expository instantiation	The system was implemented initially as a set of screen designs, next as snippets of functional software, and finally as a functional system by modifying the capabilities of an existing platform.

The design theory development also led to an interactive novel digital artifact that was put into practice at four major conferences and one online journal with more than five hundred users over a two-year period. This demonstrates the value of design theorizing in DSR and how it can lead to novel and practically relevant artifacts while also producing new justificatory knowledge (i.e., increasing understanding of the peer review process).

5. Discussion and Implications

The paper examines how to develop design theory in DSR by introducing a set of processes and structures for developing design theory based on *The Sciences of the Artificial* (Simon, 1996) for contemporary interactive digital artifacts. This section highlights the contributions by (a) showing how the processes and structures address DSR challenges by applying, elaborating, and extending Simon's ideas, (b) connecting the processes and structures to existing DSR approaches, (c) showing how the processes and structures relate to and how DSR can play a unique role in organizational theorizing, and (d) identifying interesting implications for further research.

5.1. Design Science Research Implications

Table 5 summarizes the paper's contributions: the first column lists the DSR challenges identified in the literature review (see Section 2), the second lists key ideas from Simon (1996), and the third column outlines the proposed structures and processes. I elaborate on the contribution of these structures and processes after Table 5.

Table 5. Summary of Contributions

DSR challenges	Simon (1996)	Proposed processes and structures
• Generating design theory is difficult • Challenges to the scientific legitimacy of DSR • Challenges to the need for artifact construction and/or design theory	Systematic and observable process	1. Formalize process 2. Categorize DSR types 3. Create chain of evidence
• It is unclear whether to emphasize design theory or to emphasize the artifact	Respect the quasi-independence between the inner and outer environment	4. Structure and leverage the dependency
• What are the steps of design theory development?	Leverage the benefits of iteration	5. Formally apply prototyping 6. Formally apply knowledge base
• What is a complete unit of research in DSR? • What are the stopping rules for the design process?	Search process is expensive and must be constrained	7. Relate to previous iteration 8. Constrain the process 9. Tools to identify strategic hypothesis
• Role and importance of kernel theory is unclear	Evaluation is integral to the design process	10. Integrate evaluation and Inter-relate evaluation with design
• DSR is under-specified • Design theory is too complicated	Design representation is important to clarify the problem and solution	11. Specify the structure and process 12. Represent each iteration simply

The paper introduces a parsimonious and easy-to-understand approach to generate design theory. The structures include (a) showing how goals, kernel theory, artifacts feed design, evaluation, and appropriation/generation, which leads to design theory; (b) structuring design, evaluation, and appropriation/generation into an inter-related cycle; (c) identifying DSR project types to provide signposts for starting and ending the cycle; (d) conceptualizing artifact and evaluation archives; and (e) arranging design, explanation, evaluation, and decisions into a structure that represents and leverages the iterations of design theory development. The processes include (a) applying prototyping to design theory, (b) elaborating multiple options to appropriate/generate theory for design and evaluation (see Table 1), (c) applying Yin (1989) to increase validity and reliability and elaborate a chain of evidence, (d) applying Kuhn's (1970) notion of anomalies as a stopping rule, and (e) showing how the processes and iteration can achieve the strategic hypothesis. The paper also demonstrates the proposed structures and processes in the context of a specific DSR project.

First, the paper impacts DSR by making design theory generation much more explicit and addressing the "how" (Weick, 1989). According to Weick (1989), theory construction can be improved by focusing on (a) the problem statement (i.e., make the assumptions more explicit, make the representation more accurate, and make the representation more detailed), (b) thought trials (i.e., increase number and heterogeneity), and (c) the criteria to select among the thought trials (apply criteria more consistently, apply more criteria simultaneously, apply more-diverse criteria). In DSR, problem statements relate to goals, thought trials to design, and selection criteria to evaluation. Second, the paper improves the quality of DSR theory by increasing observability, reproducibility, validity, rigor, relevance, falsifiability, evidence, and identification of the strategic hypothesis (Weber, 2012). Below, I integrate Weick's (1989) suggestions for improving theory construction and Weber's criteria into twelve processes and structures that summarize this paper's contribution of this paper.

1. **Formalization**: the paper proposes an easy-to-understand explicit process with which goals, kernel theory, artifacts, design, appropriation/generation, and evaluation leads to design theory (see Figure 1).

2. **Categorization**: the paper specifies the different types of DSR projects and their starting and ending points (Table 1 summarizes).

3. **Chain of evidence**: the paper provides an approach to capture the chain of evidence that underlies design theory (Table 2 outlines).

4. **Structured dependency**: the appropriate/generate process described in the paper ties the inner and outer environments together while keeping them separate (see the dotted box in Figure 1).

5. **Apply iteration**: the paper formalizes the application of prototyping as a research process (see dotted box in Figure 1)

6. **Apply knowledge base**: the paper formalizes the process for including and leveraging previous artifacts and kernel theories in generating design theory (see the relevant boxes in Figure 1).

7. **Leverage relationships**: the paper shows how previous designs and evaluations provide clues for the need for and direction of the next iteration.

8. **Use constraints**: these previous designs and evaluations provide the chain of evidence that constrains and ultimately converges the process.

9. **Focus on strategic**: Anomalies, satisficing, and cost-benefit analysis further assist in identifying the "best" strategic hypothesis.

10. **Integrate evaluation**: the paper elaborates on the multifaceted role of kernel theory in evaluation (see Table 1) and inter-relates evaluation with design.

11. **Over-specification**: the paper describes the key properties represented by Design, Explanation, Evaluation, and Decision concepts (see Figure 2).

12. **Represent**: finally, the simple structure described in Figure 2 allows one to iteratively develop and compare design theories.

The ideas in this paper build on previous DSR literature. The processes and structures are a further detail of “develop/build” and “justify/evaluate” in Hevner et al. (2004, p. 88) and Peffers et al. (2008, p. 54). The paper also builds on the design theory formalization and knowledge contribution ideas of Gregor and Jones (2007) and Gregor and Hevner (2013) and applies Kuechler and Vaishnavi’s (2012) ideas on the importance of kernel theory and the forms of reasoning in DSR. The processes and structures reflect the key principles of Sein et al. (2011): principle 2 (theory ingrained artifact), principle 3 (reciprocal shaping), principle 5 (authentic and concurrent evaluation), and principle 6 (guided emergence). Generating design theory as described here fits as a phase of an overall ADR project (Sein et al., 2011). Further, even though iteration is implied or suggested in all the above DSR approaches, I formulate it here into a specific prototyping process.

The paper also complements more-specific DSR approaches. Offermann, Levina, Schonherr, and Bub (2009) propose an iterative DSR process that leverages practitioners, combines quantitative and qualitative methods, includes the development of hypotheses, and incorporates specific evaluation mechanisms. Buckl et al. (2013) propose an iterative pattern-based DSR method that applies the concept of patterns to document and engage with practitioners and create a design theory nexus (Pries-Heje & Baskerville, 2008). Overall, the Buckl et al. (2013) and Offermann et al. (2009) DSR process are particularly well suited for large projects focusing on complex problems conducted in R&D environments (e.g., type V or VIII in Table 1). The structures and processes can be used to generate the design theories in each of these approaches.

5.2. Design Theory and Organizational Theory

According to Brown and Duguid (1991, p. 51), the discovering organization is “the archetype of the conventional innovative organization, one which responds—often with great efficiency—to changes it detects in its environment”. Enacting organizations on, the other hand, “gather information by trying new behaviors and seeing what happens. They experiment, test, and stimulate, and they ignore precedent, rules, and traditional expectations.” (Daft & Weick, 1984, p. 288). The need to discover and/or enact new technological innovations is a strategic necessity for most organizations in today’s hyper-competitive environment. DSR and particularly design theorizing provides organizations with a rigorous toolset to discover (appropriate) type V and type VI or enact (generate) type VIII or type IX new and innovative technological artifacts. DSR can discover the valuable application of a new artifact (i.e., identify or create a new behavioral theory) and it can enact artifacts in response to potential applications (i.e., identify a new use or create a new artifact), which is the essence of what Simon (1996) proposes in the *Sciences of the Artificial*. The structures and processes I propose here for design theorizing ensure that organizations will better understand and assess the impact of these innovations. Furthermore, organizational theorizing in partnership with design theorizing can now gain new insights and influence upfront about how new technological artifacts change today’s complex and dynamic organizational processes and structures. Design theorizing identifies the specific technological attributes and behaviors that the new artifacts enable, especially for enacting organizations that build out, integrate, and evolve with new technological innovations. The artifacts will ultimately mirror and make explicit in their technological features the implicit assumptions and practices of organizations; these implicit characteristics have always confounded organizational theorists (Weick, 1989). In today’s organizations, most transactional, interactional, and document centric tasks are technologically enabled (and constrained). Thus, design theorizing can help address what organizational theorists claim is a “growing chasm” between theory and practice (Suddaby et al., 2011, p. 237) and provide new insights into the structures and processes of the new digital business models that are driving hyper-competition.

Organizational scholars generate theory using various techniques involving combinations of induction, deduction, abduction, inferential statistics, and mathematical modeling (Langley, 1999; Suddaby et al., 2011; Weick, 1989). Design theorizing provides a new toolkit in which (a) the constraints of feasibility and iteration accelerate the identification of concepts and relationships that are practical and theoretically interesting, and (b) instantiation of the artifact provides a new “x-ray vision” into previously implicit organizational processes and structures.

Finally, the structures and processes I introduce in this paper can inform future improvements in organizational theorizing. For example, thought trials are discussed generally by many theorists (e.g., Weick, 1989; Langley, 1999) but without full consideration of issues of iteration, representation, and the stopping rule.

5.3. Implications for Future Research

This paper addresses the research question of how to develop design theory in DSR by elaborating a set of structures and processes. The results reveal new important gaps in design theorizing and DSR in general. I discuss these gaps along and present suggestions for future research below.

5.3.1. Gaps & Suggestions for Future Research

Representation: according to Simon (1996), design representation is important because it can help clarify the problem and solution and includes spatial, natural language, models, and actual artifacts. Yet, Simon notes that, for representation, “we are still far from a systematic theory of the subject—in particular, a theory that would tell us how to generate effective problem representations” (1996, p. 131). Therefore, we need representation schemes for (a) the problem, (b) proposed solutions, and (c) related design theory about the problem and solution. These representation schemes will need to facilitate translation into specific actionable guidelines (Lukyanenko & Parsons, 2013).

The arrangement of design, explanation, evaluation, and decisions into a structure in Figure 2 is one step toward addressing the representation problem. However, there is still much to be done. For example, which form (e.g., truth statements vs. screen prints) is better for which situation? More importantly, if there is no standardized representation scheme, how will we compare in and across projects? Gregor and Jones’ (2007) design theory structure does represent one standard, but it is perceived to be hard to implement (e.g., Venable, 2013). It may be more accessible to apply the structure of Figure 2 for level 1 and level 2 design theories and to reserve complete formalization for level 3 theories (see Gregor & Hevner, 2013). Using a standardized design ontology may represent another solution (see Alturki et al., 2013). However, all ontologies require cognitive translation, and it is unclear if DSR is evolved enough to settle on any particular ontology.

Modeling design theory construction: an underlying simplifying assumption in this paper is that design theory generation follows a purposeful, complete, closed-ended process with few if any resource or structural constraints and disruptions. However, most DSR projects involve multiple actors such as researchers, developers, sociologists, and most cross organizational boundaries with unique worldviews, constraints, and skills. Further, resource and intellectual property concerns may limit the time and sequencing of activities. In all, different DSR projects (see Table 1) will face differing levels of the above challenges. An interesting area for future research would be to empirically analyze and model the requirements of different DSR projects.

Retrospective design theory: designers are typically not trained or accustomed to elaborate design and explanation at the level suggested in this paper. This is a practical challenge to generating design theory, especially in commercial R&D environments. However, this creates an opportunity for the IS and organizational researcher to bridge boundaries and act as a scribe by participating in the process (see Sein et al., 2011; Lindgren, Henfridsson, & Schultze, 2004; Markus et al., 2002). The researcher can practice a form of “retrospective” design theory generation in which the strategic hypothesis, key iterations, artifact and evaluation databases are inferred and organized retrospectively at different points in the evolution of the project.

Design theory generation for all types of DSR: the focus of this paper is on generating design theory for interactive digital artifacts. Such artifacts make up the bulk of recent IS research and the IT industry. However, research on new design methodologies, frameworks, and algorithms will likely require a different approach. For instance, an algorithm is (typically) a mathematical artifact that is governed by the laws and constraints of mathematics. It is unclear if the iterative and evaluative approach of this paper applies to algorithms, for which the more important justificatory knowledge may be a mathematical proof rather than user evaluation. An important topic of future research is, thus, to delineate the role and form of design theory construction for additional types of DSR.

Strategic hypothesis and the stopping rule: Simon's (1996) notion of satisficing and Kuhn's (1970) anomalies provide important clues on how to stop the generate-test cycle of prototyping. Yet, we still don't have guidelines on how many iterations are best. One option is to apply Simon's (1996) work on modeling the logic of design where "parameters" are the issues that influence the construction of the artifact and "constraints" are the environmental issues that constrain the artifact. Thus, parameters represent the number of existing, available, and related artifacts and constraints the number of related and known kernel theories. Let *N* denote the number of iterations needed to perceive an anomaly, let X denote the number of related artifacts that already exist (parameters), and Y denote the number of related kernel theories (constraints), so that:

$$N = f(x, y)$$

In a DSR project. N > 1 and the value of N is a function of available related artifacts and kernel theories, where:

$$f_x < 0 \text{ and } f_y < 0$$

In reference to Table 1, when X is high, it will be easier to borrow ideas and modules from similar artifacts. When Y is high, it will be easier to identify, extend, or create kernel theories. The above leads to interesting future research questions such as: when both X and Y are high, then, on average, can we expect that the number of iterations required to complete the cycle of prototyping will be lower than when X and Y are low? Is there an optimal X or Y or N? What is the minimum of X and Y needed? Buckl et al. (2013) note the "rule of three" (based on Coplien, 1996) for patterns. Patterns relate to anomalies, and an important future research question is to derive a heuristic or optimum value of N. Such research could yield new insights on the cycles of iteration needed to reach the strategic hypothesis.

6. Conclusion

The design of novel interactive artifacts - especially artifacts that are fundamentally changing complex and dynamic organizational business processes - can generate interesting design theory and relevant justificatory knowledge. Like sculpting, generating interesting knowledge can take multiple paths. When it proceeds on bouts of insight and intuition alone, the researcher will need to retrospectively infer what is interesting. Alternatively, as Simon (1996) envisioned and when appropriate, the design can proceed more deliberatively to become a science of the artificial.

A key contribution of this paper is its elaborating a research approach that integrates Simon's (1996) ideas with Kuhn (1970) and Yin (1989) to inter-relate design, kernel theory, and evaluation with the concept of the strategic hypothesis. The paper goes beyond vocabularies to detail the process of design theory generation, responds to the challenges and opportunities of contemporary tools, provides a roadmap for appropriating and modifying kernel theories and artifacts to create design theory, and includes a process to stop the DSR cycle to produce complete units of research. This paper builds on previous work that elaborates the "what" and overall meta-cycle of DSR to provide a detailed exposition of the "how"—the specific structures and processes needed to generate scientifically interesting and valid design theory. I achieve this by leveraging and combining two very different modes of scientific inquiry: discovery (the social science worldview) and design (the engineering worldview; i.e., the inner and outer environments). The paper applies Simon's (1996) *Sciences of the Artificial* to elaborate structures and processes for developing design theory (engineering) and justificatory knowledge (social science) in existing DSR methods. The approach focuses on interactive digital artifacts that include goals, kernel theory, and existing artifacts as key ingredients to the prototyping cycle of inter-related appropriation/generation, design, and evaluation. The outcome is a strategic design theory.

Acknowledgements

Thanks to Lorne Olfman, Omar El Sawy, Ola Henfridsson, Carol Saunders, Richard Baskerville, Youngjin Yoo, and David Schuff for feedback on earlier versions of this paper. Thanks to Matti Rossi for his excellent guidance and the reviewers for their helpful reviews.

References

Alturki, A., Gable, G., & W. Bandara. (2013). BWW ontology as a lens on IS design theory: Extending the design science research roadmap. In J. vom Brocke, R. Hekkala, S. Ram, & M. Rossi (Eds.), *Proceedings of DESRIST 2013* (pp. 258-277). Berlin: Springer.

Arazy, O., Kumar, N., & Shapira, B. (2010). A theory-driven design framework for social recommender systems. *Journal of the Association for Information Systems*, *11*(9), 455-490

Astley, W. G., & Van de Ven, A. H. (1983). Central perspectives and debates in organization theory. *Administrative Science Quarterly*, *28*(2), 245-273.

Bartneck, C. (2009). Using the metaphysics of quality to define design science. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology* (p. 6). ACM.

Baskerville, R. L., & Wood-Harper, A. T. (1998). Diversity in information systems action research methods. *European Journal of Information Systems*, *7*(2), 90-107.

Bourgeois, L. J. (1979). Toward a method of middle-range theorizing. *Academy of Management Review, 4*(3), 443-447.

Boxenbaum, E., & Rouleau, L. (2011). New knowledge products as bricolage: Metaphors and scripts in organizational theory. *Academy of Management Review*, *36*(2), 272-296.

Brown, J. S., & Duguid, P. (1991). Organizational learning and communities-of-practice: Toward a unified view of working, learning, and innovation author. *Organization Science*, *2*(1), 40-57.

Buckl, S., Matthes, F., Schneider, A., & Schweda, C. (2013). Pattern-based design research—an iterative research method balancing rigor and relevance. In J. vom Brocke, R. Hekkala, S. Ram, & M. Rossi (Eds.), *Proceedings of DESRIST 2013* (pp. 73-87). Berlin: Springer.

Chia, R. (2000). Discourse analysis as organizational analysis. *Organization*, *7*(3), 513-518.

Coplien, J. O. (1996). *Software patterns: Management briefs*. Cambridge, UK: Cambridge University Press.

Colquitt, J. A., & Zapata-Phelan, C. P. (2007). Trends in theory building and theory testing: A five-decade study of the academy of management journal. *Academy of Management Journal*, *50*(6), 1281-1303.

Daft, R. L., & Weick, K. E. (1984). Toward a model of organizations as interpretation systems. *Academy of Management Review*, 9(2), 284-295.

Dubin, R. (1978). *Theory Building*. New York: Free Press.

Eisenhardt, K. M. (1989). Building theories from case study research. *Academy of Management Review*, *14*, 532-550.

Folger, R., & Turillo, C. J. (1999). Theorizing as the thickness of thin abstraction. *Academy of Management Review*, *24*(4), 742-758.

Flores, F., Graves, M., Hartfield, B., & Winograd, T. (1988). Computer systems and the design of organizational interaction. *ACM Transactions on Office Information Systems*, *6*(2), 153-172.

Gregor, S., & Hevner, A. (2013). Positioning and presenting design science research for maximum impact. *MIS Quarterly*, *37*(2), 337-355.

Gregor, S. &, Jones, D. (2007). The anatomy of a design theory. *Journal of the Association of Information Systems*, *8*(5), 312-335.

Hassard, J., Wolfram Cox, J. R., & Rowlinson, M. (2013). Where are the old theories of organization? Prospects for retrospection in organization theory. *Academy of Management Review*, *38*(2), 309-313.

Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, *28*(1), 77-105.

Hevner, A. R. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems*, *19*(2), 87-92.

Hooker, J. N. (2004). Is design theory possible? *Journal of Information Technology Theory and Application*, *5*(2), 73-82.

Hovorka, D., & Pries-Heje, J. (2013). Don't ignore the iceberg: Timely revelation of justification in DSR. In J. vom Brocke, R. Hekkala, S. Ram, & M. Rossi (Eds.), *Proceedings of DESRIST 2013* (pp. 228–241). Berlin: Springer.

Iivari, J. (2007). A paradigmatic analysis of information systems as a design science. *Scandinavian Journal of Information Systems*, *19*(2), 39-64.

Janson, M., & Smith, L. D. (1985). Prototyping for systems development: A critical appraisal. *MIS Quarterly*, *9*(4), 305-316.

Kraut, R. E. & Resnick, P. (2012). *Building successful online communities: Evidence-based social design.* Cambridge, MA: MIT Press.

Kuechler, W., & Vaishnavi, V. (2012). A framework for theory development in design science research: Multiple perspectives. *Journal of the Association for Information Systems*, *13*(6), 395-423.

Kuhn. T. (1970). *The structure of scientific revolutions* (2nd ed.). Chicago, IL: University of Chicago Press.

Langley, A. (1999). Strategies for theorizing from process data. *Academy of Management Review*, *24*(4), 691-710.

Langley. A., Smallman, C., Tsoukas, H., Van De Ven, A. H. (2013). Process studies of change in organization and management: Unveiling temporality, activity, and flow. *Academy of Management Journal*, *56*(1), 1-13.

Lewis, M. W., & Grimes, A. I. (1999). Metatriangulation: Building theory from multiple paradigms. *Academy of Management Review*, *24*(4), 672-690.

Lindgren, R., Henfridsson, O., & Schultze, U. (2004). Design principles for competence management systems: A synthesis of an action research study. *MIS Quarterly*, 28(3), 435-472.

Locke, K., Golden-Biddle, K., & Feldman, M. S. (2008). Making doubt generative: Rethinking the role of doubt in the research process. *Organization Science*, *19*(6), 907-918.

Lukyanenko, R., & Parsons, J. (2013). Reconciling theories with design choices in design science research. In J. vom Brocke, R. Hekkala, S. Ram, & M. Rossi (Eds.), *Proceedings of DESRIST 2013* (pp. 165-180). Berlin: Springer.

Mandviwalla, M., Patnayakuni, R., & Schuff, D. (2008). Improving the peer review process with information technology. *Decision Support Systems*, *46*(1), 29-40.

March, S. T., & Smith, G. (1995). Design and natural science research on information technology. *Decision Support Systems*, *15*(4), 251-266.

Markus, M. L., Majchrzak, A., & Gasser, L. (2002). A design theory for systems that support emergent knowledge processes. *MIS Quarterly*, 26(3), 179-212.

Moore, G. E. (1965). Cramming more components onto integrated circuits. *Electronics Magazine*, *38*(8), 114-117.

Naumann, J. D., & Jenkins, A. M. (1982). Prototyping: The new paradigm for systems development. *MIS Quarterly*, *6*(3), 29-44.

Newell, A., & Simon, H. (1976). Computer science as empirical inquiry: Symbols and search. *Communications of the ACM, 19*(3), 113-126.

Nevo, S., Nevo, D., & Ein-Dor, P. (2009). Thirty years of IS research: Core artifacts and academic identity. *Communications of the Association for Information Systems*, *25*, 2009.

Nunamaker, J., Chen, M., & Purdin, T. D. M. (1991). Systems development in information systems research. *Journal of Management Information Systems*, *7*(3), 89-106.

Offermann, P., Levina, O., Schönherr, M., & Bub, U. (2009, May). Outline of a design science research process. In *Proceedings of the 4th International Conference on Design Science Research in Information Systems and Technology* (p. 7). ACM.

Orlikowski, W. J., & Iacono, C. S. (2001). Desperately seeking the "IT" in IT research—a call to theorizing the IT artifact. *Information Systems Research*, *12*(2), 121-134.

Peffers, K., Tuunanen, T. Rothenberger, M., & Chatterjee, S. (2008). A design science research methodology for information systems research. *Journal of Management Information Systems*, *24*(3), 45-77.

Purao, S., Henfridsson, O., Rossi, M., & Sein, M. (2013). Ensemble artifacts: From viewing to designing in action design research. *Systems, Signs & Actions*, *7*(1), 73-81.

Shepherd, D. A., & Sutcliffe, K. M. (2011). Inductive top-down theorizing: A source of new theories of organization. *Academy of Management Review*, *36*(2), 361-380.

Sein, M., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011). Action design research. *MIS Quarterly*, *35*(1), 37-56.

Simon, H. A. (1996). *The sciences of the artificial* (3rd ed.). Cambridge, MA: MIT Press.

Spacapan, S., & Oskamp, S. (1990). People's reaction to technology. In S. Oskamp & S. Spacapan (Eds.), *Peoples reaction to technology: The Claremont symposium on applied social psychology* (pp. 9-29). Newbury Park: Sage.

Straub, D., & Ang, S. (2008). Readability and the relevance versus rigor debate. *MIS Quarterly*, 32(4), iii-xii.

Suddaby, R. (2010). Editor's comments: Construct clarity in theories of management and organization. *Academy of Management Review*, *35*(3), 346-357.

Suddaby, R., Hardy, C., & Huy, Q. N. (2011). Where are the new theories of organization? *Academy of Management Review*, *36*(2), 236-246.

Tsang, E. W. K., & Ellsaesser, F. (2011). How contrastive explanation facilitates theory building. *Academy of Management Review*, *36*(2), 404–441.

Tsoukas, H. (2009). Craving for generality and small-n studies: A Wittgensteinian approach towards the epistemology of the particular in organization and management studies. In D. Buchanan and A. Bryman (Eds.), *The Sage handbook of organizational research methods* (pp. 285-301). London: Sage.

Venable, J. (2013). Rethinking design theory in information systems. In J. vom Brocke, R. Hekkala, S. Ram, & M. Rossi (Eds.), *Proceedings of DESRIST 2013* (pp. 136–149). Berlin: Springer.

Walls, J., Widmeyer, G., & El Sawy, O. (1992). Building an information system design theory for vigilant EIS. *Information Systems Research*, *3*(1), 36-59.

Weber. R. (1987). Toward a theory of artifacts: A paradigmatic base for information systems research. *Journal of Information Systems*, *1*(2), 3-19.

Weber, R. (2012). Evaluating and developing theories in the information systems discipline. *Journal of the Association for Information Systems*, *13*(1), 1-30.

Weick, K. (1989). Theory construction as disciplined imagination. *Academy of Management Review*, *14*(4), 516-531.

Yin, R. (1989). *Case study research: Design and methods*. Beverly Hills, CA: Sage.

About the Authors

Munir MANDVIWALLA is Associate Professor and chair of the Management Information Systems department and Executive Director, Institute for Business and Information Technology, Fox School of Business, Temple University. He founded the Fox School's MIS department in 2000 and led the creation of AIS student chapters. He is currently leading a nationwide study on the IS job market and designing a new type of learning management system. His research interests include design science, social media and broadband policy. The research has appeared in major journals including *Management Information Systems Quarterly*, *Information Systems Research*, *ACM Transactions on Computer Human Interaction*, *Journal of Management Information Systems*, *Decision Support Systems*, *Small Group Research*, *Public Administration Review*, and *Information Systems Journal*. His work has been supported by grants from the National Science Foundation (NSF), SIM Advanced Practices Council, Bell Atlantic, IBM, Microsoft Corporation, CIGNA Corporation, Advanta Corporation, Lotus Development Corporation, and Lilly Endowment, Inc.