

## Editorial: A Critical Look at Theories in Design Science Research

Juhani Iivari

University of Oulu, Finland, [juhani.iivari@oulu.fi](mailto:juhani.iivari@oulu.fi)

### Abstract

This editorial critically reviews design theories, kernel theories, and substantive technological theories in design science research (DSR) within information systems (IS). The review identifies four different but interrelated interpretations of “design theory” for the design product. In the case of kernel theories, it pays attention to the softening of their scientific status that undermines the status of design theories as well. Substantive technological theories refer to intermediating theories between abstract descriptive/explanatory kernel theories and design. The editorial suggests that DSR in IS (DSRIS) has great potential to contribute design-oriented, explanatory/predictive, substantive technological theories. Overall, the review illustrates excessive use of the word “theory” in DSR. This is in line with the alleged “theory fetish” of information systems.

**Keywords:** Design Science Research, Design Theory, Kernel Theory, Substantive Technological Theory

Suprateek Sarker was the accepting senior editor. This editorial was submitted on October 30, 2018, and underwent one revision.

### 1 Introduction

“Theory” is the most celebrated output of academic research. Therefore, its role in DSRIS—with innovative IT artifacts as its core contributions (Hevner et al. 2004)—continues to arouse considerable interest (Baskerville et al. 2018, Peffers, Tuunanen, & Niehaves, 2018).<sup>1</sup> Baskerville et al.’s editorial (2018) suggests that one should find a balance between design artifacts and design theories in DSR papers, whereas Peffers et al.’s editorial (2018) proposes that we should accept that there are different “genres” of DSR, each with different expected expectations about DSR

contributions, the role of theory therein, and the research method to be applied. This editorial is much more radical than either of its two predecessors. It is critical toward the excessive and even ritualistic use of “theory” and, in particular, “design theory” in the DSRIS literature. Being closer to Baskerville et al. (2018) than Peffers et al. (2018), it discusses the role of theory within the DSR paradigm as a whole rather than separately within various DSR genres.

The concept of “theory” is a nebulous one. Following her review of different views of “theories,” Gregor (2006) characterizes them as “abstract entities that aim to describe, explain, and enhance understanding of the

<sup>1</sup> DSRIS is interpreted here as research with “design as a method of investigation” and not as research with “design as a topic of investigation” (Kuechler & Vaishnavi, 2008a). The latter is behavioral science research by nature (Iivari, 2015), such as the scientific study of the design activity (Cross, 2001),

with descriptive/explanatory theories of design as possible contributions. This editorial does not particularly address these theories of design, even though they may serve as kernel theories in DSRIS.

world and, in some cases, to provide predictions of what will happen in the future and to give a basis for intervention and action” (p. 616). As the major contribution, she introduces a taxonomy of five types of theory: (1) theory for analyzing, (2) theory for explaining, (3) theory for predicting, (4) theory for explaining and predicting, and (5) theory for design and action. This editorial adopts this taxonomy, the special focus being in “theories for design and action,” or “design theories.”<sup>2</sup>

Walls, Widmeyer, & El Sawy’s (1992) concept “design theory” initiated the interest in theory in DSRIS (e.g., Markus, Majchrzak, & Gasser, 2002; Gregor & Jones 2007; Kuechler & Vaishnavi, 2008a; Kuechler & Vaishnavi, 2008b; Baskerville & Pries-Heje, 2010; Gregor & Hevner, 2013; Niehaves & Ortbach, 2015; Baskerville et al., 2018; Peffers et al., 2018). The idea of “design theories” implies that “theories” not only inform the design of various IT artifacts in DSRIS but, as “design theories,” they are central outcomes of DSRIS. Unfortunately, much of the later literature that is related to the seminal works of DSRIS (such as Walls et al., 1992; March & Smith, 1995; Hevner et al., 2004) has been conceptually quite confusing and, in my view, has advocated “more theory” in DSRIS, leading to a “theory fetish” therein.<sup>3</sup>

By “theory fetish,” I mean excessive emphasis on “theory” and “theory building,” as if theory were the only remarkable scientific contribution, as well as the ritualistic and undifferentiated use of the word “theory” when more exact or informative concepts could be used. Such excessive use easily inflates the value of the word “theory” in scientific argumentation. To take an example, if all conceptual frameworks are considered theories (“theories for analyzing”), practically all scholarly papers would be theory grounded, since it is impossible to do scientific research without any conceptual framework. Instead of “theory,” it would be more informative and modest to speak about the specific classification, taxonomy, or conceptual framework that one has in mind.

The purpose of this editorial is to contest the “theory fetish” of DSRIS, to critically review various concepts of theory used in DSRIS, and to clarify the different meanings of “design theory” discernible in DSRIS. Based on the scope of this paper, the DSR literature outside DSRIS will not be covered. This editorial is complementary to Avison and Malaurent (2014) and to Hirschheim (2019) and the associated commentaries, which largely omit DSRIS and related theories.

The editorial starts with a criticism of “theory fetish” in IS and in related disciplines. The purpose of this criticism is by no means to denounce “theory” and “theory building” but to serve as a contextual introduction proposing a more considerate use of theory in DSRIS. After this prelude, the editorial proceeds to the concept of “design theory” as a central intellectual outcome of DSRIS. A detailed analysis shows that the concept is often used ambiguously—“nascent design theory” (Gregor & Hevner, 2013) being a prime example of this. More concretely, the editorial distinguishes four different but interrelated interpretations of “design theory.” Therefore, it advocates a more sparing and careful use of the concept but does not propose that it should be totally abandoned. Some DSR outcomes earn the status of “design theory,” the relational data model (Codd 1970) being a good example of that.

Walls et al. (1992) suggest that “design theories” include “kernel theories” from natural and social sciences and mathematics, which govern design requirements. This editorial emphasizes that it is the soundness of “kernel theories” that is decisive when one assesses whether it is justifiable to speak about “design theory.” The relational data model, for example, is based on mathematical set theory, and on the mathematical concept of relation, and relational algebra in particular. The review below shows that the concept of “kernel theory” has been subjected to dubious refinements that undermine the expected soundness of kernel theories and, consequently, the scientific value of the concept of “design theory.”

More recently, Kuechler and Vaishnavi (2008b, 2012) and Arazy, Kumar, and Shapira (2010) point out that kernel theories from reference disciplines are not (necessarily) concrete enough to guide design and that intermediating theories are needed to bridge the gap. They recommend the use of “design relevant explanatory/predictive theories” (Kuechler & Vaishnavi, 2008b, 2012) and “applied behavioral theories” (Arazy et al., 2010) for that purpose. Bunge (1966) calls such theories “substantive technological theories.” This leads to the suggestion that the philosophy of technology provides a better background than Merton’s (1949) hierarchy of theories for discussing the relationship between kernel theories and design and suggests that DSRIS has great potential to contribute design-oriented, explanatory/predictive, substantive technological theories.

Finally, this editorial contrasts with Baskerville et al.’s (2018) and Peffers et al.’s (2018) editorials and makes a number of remarks and recommendations based on

---

<sup>2</sup> As it includes “design theories” (“theories for design and action”) as a special type of theory, I will bypass the discussion of whether “design theory” is possible or not (see Hooker, 2004; Walls, Widmeyer, & El Sawy, 2004).

<sup>3</sup> However, I do not claim that all premier IS journals and conferences require that a DSR paper proposes a “design theory” to be acceptable; nevertheless, I fear that many authors and reviewers expect good DSR papers to do so.

the analyses conducted. Overall, it advocates *less theory, but better design theory*, i.e., more sparing use of the word “theory” and design theories based on sound kernel theories—their soundness being determined by the research community of the originating discipline (Truex, Holmström, & Keil, 2006).

## 2 Theory Fetish in IS and DSRIS

To understand the current position of “theory” in DSRIS, it is important to be aware of its role in IS research more generally. The editorial statements of leading IS journals, such as *MIS Quarterly*, *Information Systems Research*, and *Journal of the Association for Information Systems*, emphasize that papers should be grounded in theory, following in that respect the example of premier journals in organization and management studies (Hambrick 2007, Corley & Gioia 2011). However, European IS journals such as *European Journal of Information Systems*, *Information Systems Journal*, and *Journal of Information Technology* do not directly express a preference for theory-grounded papers. Nevertheless, Avison and Malaurent (2014) characterize the situation within IS as a whole as a “theory fetish” that encompasses even papers focusing on qualitative IS research.

Taking this larger context into account, it is understandable that DSRIS has fallen into the theory trap. Gregor (2006, p. 613) contends that “developing theory is what we are meant to do as academic researchers and it sets us apart from practitioners and consultants,” and Venable (2013, p. 136) writes: “Since theory is a key output of rigorous academic research, one would expect the production of DT [design theory] to be a key element of DSR.”

Among empirical disciplines such as organization and management studies, theories (excluding design theories) may have such a monopoly position but this is not necessarily the case in fields such as the natural sciences or economics. This is evidenced by, for example, the editorial statements of premier physics journals: *Advances in Physics* and *Nature Physics* do not refer to theory and *Physical Review Letters* states that papers *may* “advance new theoretical views.” Further, in the editorial material accompanying *Economic Journal*, there is no mention of theory, and in *Journal of Finance*, theory is mentioned only in a list of keywords. However, *Journal of Economic Theory* does require papers to be “firmly grounded in theory.” Of course, just because a journal’s editorial

statement does not explicitly mention “theory,” it does not follow that the papers published in these journals do not use theories.

One may explain the difference between IS and more mature disciplines such as physics and economics by arguing that the latter have already built solid theories and therefore do not need new theories to the extent that IS does. Furthermore, the central phenomena investigated by IS—i.e., the development and use of information systems and technology for human enterprise, covering IS use by individuals, teams, organizational units, and organizations as well as IS use by communities, markets, industries, and societies (Grover & Lyytinen, 2015)—concern artificial reality, are socially constructed, and are subject to constant change. This largely explains the theoretical diversity of the IS discipline (Benbasat & Zmud, 1996). Lim et al. (2013), for example, identifies 174 distinct theories used in 385 papers published in *MIS Quarterly* and *Information Systems Research* from 1998 to 2006 alone, the majority of them (over 70 % according to Grover & Lyytinen, 2015) being “midrange theories” that lie somewhere between abstract theories borrowed from reference disciplines and concrete data from IS-specific phenomena of interest.<sup>4</sup> Given that IS already has a few hundred such theories in play, it is hard to claim that more are desperately needed, although novel and distinct theories would certainly be welcome (Grover and Lyytinen 2015).

The main point of criticizing the theory fetish is, however, that “theory” should not be given monopoly status in the representation of scientific knowledge as, implied, for instance, by Gregor (2006) and Venable (2013). Hambrick (2007, p. 1346) contends that “theories are not ends in themselves” and that “a blanket insistence on theory ... actually retards our ability to achieve our end: understanding.” Both Avison and Malaurent (2014) and Hirschheim (2019) echo him in this regard. Insistence on theory and theory building prevents research on interesting phenomena for which no theory yet exists (Hambrick, 2007), studies about anomalies that existing theories are not able to explain—barring a theoretical attempt to solve them—and papers reporting the results of replication studies. All these are essential to scientific progress, as argued by Dennis and Valacich (2014) in the context of IS replication studies.

The studies by Gregor (2006) and Venable (2013) implicitly extend the theory discussion to DSRIS. In this context, one can argue that the outputs of DSRIS, including constructs, models, methods, design principles, and technological rules (Gregor & Hevner 2013),<sup>5</sup> also

---

<sup>4</sup> Grover and Lyytinen (2015) make it clear they use “midrange theory” in a more specific meaning than Merton (1949).

<sup>5</sup> I have difficulties considering instantiated IT artifacts as knowledge contributions of DSRIS independently of constructs, models, etc. (Gregor & Hevner, 2013) since it is hard to imagine that a reasonably complex IT artifact can be

represent scientific knowledge independently of whether they have been framed as “design theories” or not. The model of decision support systems elaborated by Keen and Scott Morton (1978) is an example of such representation.

Hambrick (2007) explains how the focus on theory emerged in management studies as a response to the accusations of a lack of academic sophistication in the 1950s. In a similar vein, Avison and Malaurent (2014, p. 328) explain the excessive emphasis on theory as “the way forward to make information systems an ‘acceptable’ discipline.” Hirschheim (2019) describes in length the pressure to make IS more acceptable and legitimate within North American business schools and universities by making it look more scholarly and theory oriented. Thus, it would seem that the real reason for the theory fetish in IS in general and DSRIS in particular has largely been political rather than intellectual—to legitimize IS as a discipline and DSRIS within it rather than to advance the IS/DSRIS body of knowledge.

The concept of “design theory” discussed in the next section is an example of a concept that has been introduced to legitimize DSRIS (Walls et al., 1992, Gregor 2006, Gregor & Jones, 2007). The message has been that *DSR also suggests theories, but of a different type than those of IS researchers engaging in natural/behavioral sciences research*. The concept “nascent design theory” (Gregor & Hevner, 2013) is a prime example of the “theory fetish” in DSRIS: the adjective “nascent” combined with the ambiguity of “design theory” simply makes the whole concept too shaky and permissive, meaning that “anything goes.” Luckily, any discussion of nascent theory can easily be abandoned by simply referring to the design knowledge in question: constructs, methods, models, design principles, technological rules, etc.

### 3 Conceptions of Design Theories in DSRIS

There is wide consensus that the concept of “design theory” in DSRIS has not been fully agreed upon (Venable 2006, Baskerville & Pries-Heje 2010, Venable 2013, Niehaves & Ortbach 2015). When introducing different conceptualizations, it has been customary to extend the concept to all sorts of “IT artifacts” (e.g., the critical success factors approach to requirements determination in Walls et al., 1992; a framework for risk management in software development in Gregor & Jones, 2007) and even

beyond (e.g., portfolio theory in Baskerville & Pries-Heje, 2010). Although this generalization is of value, it often forces the authors to stretch and adapt the concepts in a way that makes it difficult to understand whether the examples are fully consistent with the “theoretical” conceptions and with each other. In order to avoid such confusion, as far as possible, I will use information systems and/or software artifacts as examples in my argumentation since they are archetypical examples of IT artifacts.

#### 3.1 Review of the Major Conceptions

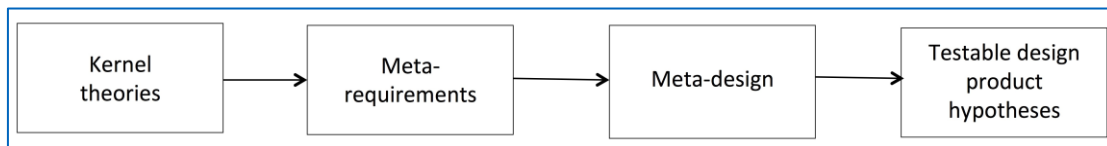
As far as I know, Walls et al. (1992), Venable (2006), Gregor and Jones (2007), Baskerville and Pries-Heje (2010), and Niehaves and Ortbach (2015) comprise the most significant set of DSRIS papers that introduced their own conceptualizations of “design theory.” For example, Walls et al. (1992) proposed the concept of IS design theory, identifying it in two parts: design product and design process. This editorial focuses only on design theory for the design product, which consists of *metarequirements* and *metadesign* for the IS design product, *kernel theories* governing the design requirements, and *testable design product hypotheses* used to test that the metadesign satisfies the metarequirements. Figure 1 illustrates the resultant structure.<sup>6</sup>

My analysis below shows that, especially the concept of metarequirements, is widely interpreted in a way that does not correspond to what Walls et al. (1992) exactly mean. Venable (2006) refines Nunamaker, Chen, and Purdin’s (1990) framework, placing theory building at the center of the DSR activity. As for design theory, he suggests that it should be in the form of utility theories, which relate improvements expected from applying a particular type or types of technologies (“metadesigns” in Walls et al. 1992) to a particular type of problem. Venable (2006) does not provide any concrete examples of what he means by “problem” in the IS context, but—misleadingly in my opinion—associates it with Walls et al.’s notion of “metarequirements” or in Markus et al.’s (2002) “user requirements.” Venable’s interpretation is understandable, given that Walls et al. define metarequirements as “the class goals to which the theory applies.” Their most concrete example—metarequirements for vigilant information systems—indicates, however, that their “metarequirements” actually specify the functionality of the system (e.g., MR1: “A VIS should support issue representation in the form on triggers, templates, and twitches,” Walls et al. 1992, p. 51).

---

instantiated without any instantiation of design knowledge (constructs, models, etc.). Furthermore, the knowledge embedded in the instantiated artifact must be communicated in some way to the DSRIS community and is likely done so through constructs, models and design principles.

<sup>6</sup> Design theory for the design process comprises the *design method* of constructing the artifact, *kernel theories* governing the design process, and *testable design process hypotheses* used to verify that the design process results in an artifact that is consistent with the metadesign (Walls et al. 1992).



**Figure 1. Components of Design Theory for the Design Product (Walls et al., 1992)**

So, they use the term “requirements” in the sense used in software engineering, where requirements are traditionally interpreted to be part of the software artifact (Freeman 1987). Venable, in contrast, views problems as something “external” to solution technologies, analogous to “business problems.”<sup>7</sup>

Gregor and Jones (2007) expand the concept of design theory suggested by Walls et al. (1992), proposing that it consists of eight components, the first six being mandatory, and the last two optional: (1) *purpose and scope*, describing “what the system is for”; (2) *constructs* of “theory”; (3) *principle of form and function*, i.e., the abstract blueprint or architecture of the IS artifact (either the IS product or IS development method); (4) *artifact mutability*, i.e., the changes in state of the artifact anticipated in the “theory,” (5) *testable propositions*, i.e., the truth statements about the design theory; (6) *justificatory knowledge*, i.e., the underlying knowledge or theory from the natural or social or design sciences that provides a basis and explanation for the design (e.g., kernel theories in Walls et al. 1992); (7) *principles of implementation*, i.e., a description of processes for implementing the “theory” (either product or method) in specific contexts (design method in Walls et al. 1992); and (8) *expository instantiation*, i.e., a physical implementation of the artifact.

Gregor and Jones (2007) interpret *purpose and scope* as corresponding to the “metarequirements” of Walls et al. (1992) and *principles of form and function* as corresponding to “metadesign.” Their example of human productivity as the purpose of the relational model for databases (see Table 1 in Gregor and Jones 2007) implies that the *purpose and scope* (may) cover potential effects of the artifact.<sup>8</sup> Referring to the above discussion in the context of Venable (2006), I claim that Gregor and Jones (2007) also do not recognize that metarequirements in Walls et al. (1992) specify functional (and possibly nonfunctional) requirements for the artifact, i.e., they correspond to *principles of*

*function* rather than *purpose and scope* in Gregor and Jones (2007).

Gregor and Jones (2007) construe that their *testable propositions* cover “design product hypotheses” and “testable design process hypotheses” (Walls et al. 1992). As explained above, Gregor and Jones understand “metarequirements” in Walls et al. (1992) as corresponding to the *purpose and scope* component, which covers the potential effects of the artifact. Therefore, Gregor and Jones’s *testable propositions* concerning the design product may include truth statements about how the IS artifact performs relative to its possible effects (e.g., how the relational database technology affects human productivity). This means that IS design theory in the sense of Gregor and Jones (2007) covers but is not limited to the relationship between IS artifacts and their effects.

Baskerville and Pries-Heje (2010) suggest that one can distinguish two types of design theory: explanatory design theory and practice design theory. I will focus on explanatory design theory (EDT) only, which “explains why a generalized set of requirements is satisfied by a generalized set of object features” (p. 273). Practice design theory (PDT) concerns the process side (just as design theory for the design process in Walls et al. 1992), prescribing in a practical way how to design or construct an IT artifact. Baskerville and Pries-Heje do not recognize the difference in the meaning of metarequirements (or generalized requirements) in Walls et al. (1992) versus Gregor and Jones’ use (2007), but adopt the definition of “requirements” on the IEEE Std 610.12.-1990 for software, which defines requirements as a condition or capability met or possessed by a system rather than something external to it. Thus, their characterization of “metarequirements” seems to be in line with that of Walls et al. (1992). As such, their notion of explanatory design theory (EDT) addresses the relationship between metarequirements and metadesign, internal to the IT artifact.<sup>9</sup>

<sup>7</sup> Venable (2006) illustrates his ideas most concretely in the context of medical analogy, where he sees diseases as problems and treatments as solution technologies. Assuming that the causes (biological mechanisms) are known, I would claim that “metarequirements” for a drug (treatment) specify that the drug affects the causes of the disease (i.e., the underlying biological mechanisms) in a specific way. It is a separate question whether affecting the causes by using a drug has the desired effect on the disease.

<sup>8</sup> As an additional piece of evidence, applying the “design theory” of Gregor and Jones (2007), Giessmann and Legner (2015) propose a design principle for the platform-as-a-service (PaaS) business model to *ensure mid- to long-term profitability*.

<sup>9</sup> Unfortunately, many of the examples in Baskerville and Pries-Heje (2010) are quite poorly formulated. Concerning, for instance, vigilant information systems (p. 279), which could be the most concrete example, it is difficult to figure



Niehaves and Ortbach (2015) continue the work of Baskerville and Pries-Heje (2010). Because they are influenced by structural equation modeling, they introduce a fairly complex framework for EDT. Based on their article, it is not entirely clear whether they have metarequirements in the sense of Walls et al. (1992) and Baskerville and Pries-Heje (2010) in mind when they use the term “effects” of the IT artifact. I interpret their example of a feedback system with “social presence” as the dependent variable (effect) as suggesting that the effect is a quality of the system rather than something external to it (for example, the effect of the feedback system on employees’ work performance). However, Peffers et al. (2018) state that explanatory design theory is “a type of design theory that emphasizes design features and their effects on the environment, e.g., on users” (p. 134). This differs from my reading of Baskerville and Pries-Heje (2010). Despite that, in this situation, I rely upon Peffers et al.<sup>10</sup>

When compared with Baskerville and Pries-Heje (2010), another difference is that Niehaves and Ortbach (2015) introduce kernel theories to explain the relationship between metarequirements and metadesign. Even though they suggest that their model is an EDT, I am not convinced that it is adequate to address the relationship between metadesign and metarequirements when interpreted in the sense of Walls et al. (1992). The reason for this is that there are typically a number of metarequirements (as the authors themselves remark in passing) and the relationship between metadesign and metarequirements is not n:1, nor is it even n:m; rather, some of the metarequirements (especially nonfunctional requirements such as security) are emergent properties determined by the metadesign as a whole and not by individual design items.

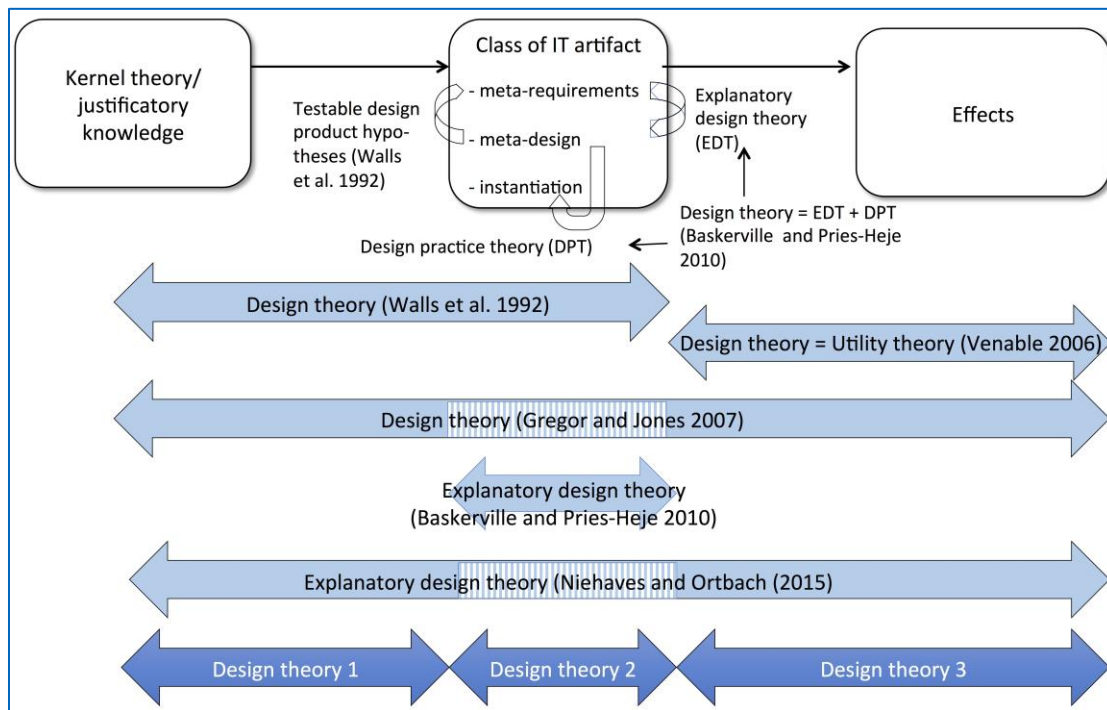


Figure 2. Different Conceptions of Design Theory

out how the “capability” statement (“Executives have to deal with issues from a consistent vision, while managers must deal with issues while avoiding culture discrepancies”) is a generalized requirement related to the system. Their example of patterns as an explanatory design theory, however, supports my interpretation.

<sup>10</sup> Inspired by the example of “social presence” (Lowenthal 2010), I could conceive of illusions created by an IT artifact being examples of effects on the users that could serve as metarequirements in explanatory design theories as introduced by Baskerville and Pries-Heje (2010). However, Peffers et al. (2018) do not constrain the effects on the environment in any way.

### 3.2 Synthesis

As a synthesis, Figure 2 depicts different interpretations of “design theory” in DSRIS. Appendix A provides a more concrete conceptual mapping of the vocabulary used by the five major proponents of design theory, including an artificial example of “nascent design theory” employing spoons as exemplary artifacts.

Figure 2 shows that there are widely different views of “design theory.” Walls et al. (1992) focus on the relationship between kernel theory and the IT artifact and on the relationship between metarequirements and metadesign internal to the IT artifact. “Design theory” in Venable (2006) concentrates solely on the relationship between the IT artifact and its effects (utility). “Design theory” in Gregor and Jones (2007) combines, to some extent, both Walls et al. (1992) and Venable (2006) but neglects the relationship between metarequirements ( $\approx$  principles of function) and metadesign ( $\approx$  principles of form) internal to the IT artifact. Baskerville and Pries-Heje’s paper (2010), on the other hand, is limited to the relationship between metarequirements and metadesign internal to the IT artifact. Finally, after reading Peffers et al. (2018), I conclude that the scope of the explanatory design theory of Niehaves and Ortbach (2015) corresponds to that of Gregor and Jones (2007).<sup>11</sup>

Based on Figure 2, one can identify four different conceptions: “Design Theory 1,” which is a slightly modified version of Walls et al. (1992)<sup>12</sup>, “Design Theory 2” à la Baskerville and Pries-Heje (2010), “Design Theory 3” à la Venable (2006), and “Design Theory 4,” which is a union of Design Theories 1-3. A clear conclusion from Figure 2 is that anybody who talks about “design theory” should carefully indicate in what sense he or she uses the concept—as exemplified by a recent article by Peffers et al. (2018) that fails to recognize the differences between “design theory” in Walls et al. (1992) and Gregor and Jones, (2007) and even the differences between Baskerville and Pries-Heje (2010) and Niehaves and Ortbach (2015).

<sup>11</sup> If I had not consulted Peffers et al. (2018), I would have understood the scope of “design theory” in Niehaves and Ortbach (2015) in Figure 2 as corresponding to that of Walls et al. (1992), with a special focus on explanatory design theory, as in Baskerville and Pries-Heje (2010).

<sup>12</sup> In other words, IS design theory for the design product in Walls et al. 1992, excluding testable product hypotheses.

<sup>13</sup> According to Walls et al. (1992) kernel theories govern metarequirements but not metadesign in the case of the “design product.” I assume that by this they limit kernel theories to those that govern distinctive aspects of the “design product” and exclude various fairly standard “design

### 4 Conceptions of Kernel Theories in DSRIS

Kernel theories are central constituents of design theories. It is notable, however, that the concept of “kernel theory” has evolved over the years in a direction that is not necessarily desirable. According to Walls et al. (1992), kernel theories derive from natural and social sciences and mathematics.<sup>13</sup> I guess that they implicitly assume that theories from natural and social science are “theories for explaining,” “theories for predicting,” and “theories for explaining and predicting” as per Gregor (2006). Walls et al. seem to introduce kernel theories as mandatory constituents of design theories, even though in their example—the software development life cycle model as a “IS design theory”—they fail to specify its kernel theory.

Based on Walls et al. (1992), Markus et al. (2002) suggest a simplified design theory for emergent knowledge processes, consisting of kernel theories, metarequirements for the system, design principles for the system, and principles for its development. From the viewpoint of theory, it is noteworthy that they soften the concept of kernel theory so that it may also be a practitioner theory-in-use, i.e., not only an academic theory.

Gregor and Jones (2007) substitute “justificatory knowledge” for “kernel theory.” They make it explicit that justificatory knowledge may include existing design theories and also practical theories-in-use (as do Markus et al., 2002). Compared with “kernel theories” the concept of justificatory knowledge allows for a more eclectic use of existing research as justification for the design ideas. For example, Giessmann and Legner (2016) suggest four design principles that each have a distinct source (in terms of references) of justificatory knowledge. This is an acceptable practice for justificatory knowledge, but it does not necessarily constitute theory (e.g., see “references are not theory” in Sutton & Staw, 1995).

Practical theories-in-use further degrade the idea of justificatory knowledge as a mandatory constituent of design theory. For example, in the case of medicine, justificatory knowledge could include all kinds of

theories” that are used in metadesign. For example, one can claim that the metadesign of an IT artifact comprising computing is always, at least implicitly, based on a number of mathematical theories, starting with Boolean algebra. When one includes existing “design theories” among kernel theories (Gregor & Jones, 2007), it is clear that metadesign may adopt numerous existing design theories, such as the relational model, object-orientation, modular design, software patterns, etc., if these are considered to be design theories. Usually these do not imply anything distinctive into the design product but are widely used methods and techniques in technical design and implementation.

pseudoscientific theories, such as those relating to alternative medicine. Iivari (2007) suggests that in order to be qualified as design theory, the core artifact (e.g., medical treatment) should be backed by kernel theories that are considered to be scientifically justified.

As a parallel to kernel theory rather than kernel theory per se, Kuechler and Vaishnavi (2012, p. 404) introduce the concept of “tacit theory” i.e., “insights or evidence/experience-based justifications for pursuing a novel design. This theory type is informal and is frequently not explicitly stated, but is very important to DSRIS in that such theories provide design science research with the ability to explore areas where formal theory is sparse or non-existent.” While I have no problem with this definition, I do question why they call this “tacit theory” instead of tacit knowledge?

If eclectic justificatory knowledge, practical theories-in-use, pseudoscientific theories, and possibly “tacit theories” are regarded as acceptable kernel theories, the concept of “design theory” is, indeed, very loosely circumscribed. When one adds the idea of “nascent design theory” (Gregor and Hevner 2013), design theory becomes almost anything. If it is “nascent,” why is it called “theory”? Why not speak about the artifacts typical of such “nascent design theories”: e.g., specific constructs, methods, models, design principles, and technological rules in question, without any mention of “nascent design theory”?

Referring to Figure 2, it is important to keep in mind that Design Theories 1-3 likely have at least partially different kernel theories. It is hard to imagine that there could be a serious Design Theory 3 without kernel theories that take into consideration the context in which the IT artifacts (instantiations) are supposed to be used (e.g., relevant characteristics of users, groups, organizations, communities, or societies). Design Theory 2, addressing the relationship between metadesign and metarequirements, is a central problem in software testing. Since it focuses on the relationship internal to IT artifacts, it is unlikely that explanatory/predictive theories from natural sciences or social sciences would be relevant in this case. However, existing methods and techniques in software engineering might be available as kernel theories. Design Theory 1 may or may not include contextual aspects. As an early example of the latter, Flores et al. (1988) used speech act theory (Searle, 1979) to generate the general requirements of their email system, called The Coordinator. One could imagine, however, that Design Theory 1 could include kernel

theory to justify a priori the effectiveness of the class IT artifact suggested in Design Theory 1. This would require that, similar to Design Theory 3, the kernel theory included contextual factors and effectiveness criteria so that it would be possible to predict the effectiveness of the class of IT artifacts.

## 5 Substantive Technological Theories in DSRIS

### 5.1 Need for Substantive Technologies Theories

Arazy et al. (2010) and Kuechler and Vaishnavi (2008b, 2012) point out that kernel theories are not (necessarily) concrete enough to guide design so that an intermediating theory is needed to bridge the gap. Arazy et al. (2010) suggest “applied behavioral theory” for that purpose and Kuechler and Vaishnavi (2012), “design relevant explanatory/predictive theory” (DREPT). The idea of an intermediating theory is an important contribution to DSRIS. Philosophers of technology actually suggested similar ideas a long time ago. Bunge (1966), for example, makes a distinction between “scientific theories” and two kinds of “technological theories”: substantive theories and operative theories. The former provide knowledge about objects of action (technology), whereas the latter are concerned with the operations of men and man-technology complexes. Bunge uses airplanes as an example: there is a substantive technological theory of flight that is an application of a scientific theory of fluid dynamics.<sup>14</sup> An optimization model of flight booking is an example of operative technological theory. Substantive technological theories are “essentially applications, to nearly real situations, of scientific theories,” whereas operative ones apply method(s) of science and make little, if any use, of substantive scientific knowledge (Bunge, 1966, p. 331-332).

It seems that both Arazy et al. (2010) and Kuechler and Vaishnavi (2012) interpret kernel theories to be something like “scientific theories,” even though they are not confined to the natural sciences. As noted above, Walls et al. (1992) mention theories of social sciences as possible kernel theories and Gregor and Jones (2007) include even existing design theories as justificatory knowledge ( $\approx$  kernel theories). Gregor and Hevner (2013) remark that, currently, kernel theories are typically implicitly assumed to arise from reference disciplines. This assumption would exclude

<sup>14</sup> There is a danger of terminological confusion here, since the literature on grounded theory uses the term “substantive theory” in a different meaning as Bunge (1966)—as the opposite of “formal theory” (Urquhart, Lehman, & Myers, 2010). Therefore, I use the term “substantive technological

theory” rather than “substantive theory” in the sense of Bunge (1966). Some DSRIS literature also uses “substantive theory” referring to grounded theory (e.g., Kuechler and Vaishnavi 2012).



the “theories for analyzing,” “theories for explaining,” “theories for predicting,” “theories for explaining and predicting,” and even “theories for design and action” (Gregor 2006) that have originated in IS from the set of possible kernel theories. I do not think that we, as IS researchers, should accept or reinforce this contention. Therefore, this editorial does not assume that kernel theories are necessarily derived from reference disciplines external to IS. On the contrary, I suggest below that IS and DSRIS, in particular, have great potential to contribute design-oriented substantive technological theories as kernel theories.

## 5.2 The Nature of Substantive Technologies Theories

Kuechler and Vaishnavi (2012, p. 398) introduce DREPTs as midrange theories, as “conceptual intermediaries between the highly abstract space of potential problem solutions suggested by kernel theories or insights and the concrete problem solution of the implemented artifact.” According to Merton, (1949, p. 448) midrange theories “lie between the minor but necessary working hypotheses that evolve in abundance during day-to-day research and the all-inclusive systematic efforts to develop a unified theory that will explain all the observed uniformities of social behavior, social organization, and social change.” This quote clearly shows that Merton introduces midrange theories in the context of “theories for explaining,” “theories for predicting” and “theories for explaining and predicting” (Gregor 2006). Since DREPTs are predictive/explanatory theories, characterizing them as midrange theories is not misleading as such, but it is not effective, since, currently, practically all empirically testable theories in social sciences are midrange theories in Merton’s sense.<sup>15</sup> Thus, my conclusion is that the characterization of DREPTs as midrange theories just distracts the reader. The argument that DREPT is not only theory, but a special type of theory, a midrange theory, sounds convincing: if DREPT is midrange theory, it must be theory too!

Furthermore, Gregor (2006) and Kuechler and Vaishnavi (2012), as well as Merton (1949) himself, tend to confuse the theoretical scope (e.g., individual hypotheses vs. “all observed uniformities of social behavior, social organization, and social change” (Merton 1949, p. 448) in the case of sociology) and the generality of theory. I see the essence of Merton’s argumentation as the theoretical scope. Increasing the theoretical scope tends to decrease the generality of theory, i.e., the cases in which the theory is valid among its units of interest. When theories with a wide theoretical scope are made more abstract in order to

increase their generality, they tend to lose their empirical testability so that, ultimately, one cannot decisively “prove” that they are general.

I believe that the philosophy of technology provides a fertile background—superior to Merton’s hierarchy of theories—for discussing the relationship between kernel theories and design ( $\approx$  engineering). Much of the philosophy of technology focuses on the relationship between natural sciences and “hard” technologies and has the orthodoxy of *technology as applied science*, as exemplified by Bunge (1966). This orthodoxy has been seriously challenged in the philosophy of technology (Layton 1974, Gardner, 1994, Gardner, 1995, Boon, 2006). Despite this, much DSRIS still follows a similar orthodoxy when claiming that design in DSRIS (or the resultant “design theory”) should necessarily be based on kernel theories (Walls et al., 1992; Gregor & Jones, 2007; Kuechler & Vaishnavi, 2012).

March and Smith (1995) point out, on the contrary, that DSRIS artifacts are often invented without any kernel theory. Iivari (2007) and Hevner (2007) echo their view. Iivari (2010) claims that the necessity of recognizable kernel theories may considerably limit the innovativeness of DSRIS and questions whether Berners-Lee could have published his ideas on WWW in a top IS journal, for example. Therefore, researchers such as March, Smith, and Hevner—who are recognized within the DSRIS community as representing the orthodox constructive view (Kuechler & Vaishnavi, 2008a), according to which the artifact is the distinctive contribution in DSR and is not necessarily based on descriptive/explanatory kernel theory—are more modern in the wider context of the philosophy of technology than are the critics of the constructive view discussed above. The philosophy of technology might lead DSRIS to critically reflect the assumption of *technology as applied science* in their research.

A recent paper of Baskerville et al. (2018, p. 364) positions DSR in the interplay between science and technology and pinpoints that there “should not necessarily be a requirement to show that the DSR is based on theory in the natural and social sciences as new designs may arise by processes including trial-and-error and creative insights, rather than logical deduction from prior theory.” I understand this quote to mean that they reject the idea of *technology as applied science* but, at the same time, fail to recognize the role of substantive technological theories in cases

<sup>15</sup> “Midrange design theories” and “grand design theories” (Gregor & Hevner, 2013) are more problematic as concepts because Gregor and Hevner (2013) remove the distinction

between “midrange” and “grand” from their original context of explanatory/predictive theories in their “design theories” context.

in which science happens to inform technology.<sup>16</sup>

Of course, it is wonderful when descriptive/explanatory theories—either directly or indirectly via a substantive technological theory—serve as a source of ideas that inspire the building of an innovative IT artifact. But, in line with Baskerville et al. (2018), I do not think that it should be required that each and every IT artifact contributed by DSRIS is theory ingrained (Sein et al., 2011) or inscribed by constructs of some kernel theory (Hayes & Carroll, 2010). Such a requirement would easily lead to the superficial listing of kernel theories and, above all, would decrease the innovativeness of DSRIS. The next section argues that the temporal relationship between the kernel theories and the artifact may, in fact, be reversed, especially in the case of innovative IT artifacts.

### 5.3 Substantive Technologies Theories as IS Contributions

Kuechler and Vaishnavi (2012) essentially contend that every DSRIS contribution has DREPT, claiming that the same technique they used in the case of one DSRIS paper to identify its underlying DREPT can be applied to any published example of DSRIS (p. 411). They also implicitly recommend DREPT (1) as means of capturing design knowledge that would otherwise remain tacit in the design artifact, and (2) as a formalism that helps to explain the nature of design theory and its relationship to both the artifact and kernel theories (p. 415). I agree with them that something like DREPT might be useful for representing design knowledge at approximately the same conceptual level as kernel theories.<sup>17</sup>

Arazy et al. (2010) suggest the use of applied behavioral theory as a heuristic means in their theory-driven method for design to bridge kernel theories and design. It is a separate question whether applied behavioral theories and its equivalents such as DREPTs have intellectual value independent of their use as such heuristics. To my knowledge, there are not many good examples of this. Kuechler and Vaishnavi (2012) outline two examples in such coarse terms that it is difficult to assess their value. The example of DREPT in their earlier paper (Kuechler & Vaishnavi, 2008b) identifies seven propositions. Even though they may have been useful in the transition from the selected kernel theories to design, it is difficult to figure out the unifying idea of the propositions. In line with Sutton and Staw's (1995, p. 376) claim that “hypotheses (or predictions) are not

theory,” I would claim that a list of propositions is also not theory. Kuechler and Vaishnavi (2012) also admit that DREPTs are more or less based on “tacit theory.” I agree with the spirit of this idea but would prefer to characterize this as tacit knowledge, creativity, or intuition.

Perhaps, the applied behavioral model in Arazy et al. (2010) is the most convincing example of a DREPT that could be of interest outside its DSRIS context. One could imagine that it could have been published in a respected IS journal as an ordinary explanatory/predictive model (theory). It is interesting that the model does not include any designable qualities of a recommender system or IT artifacts more generally (see design items and independent and dependent variables in Niehaves & Ortbach, 2015). Thus, while Arazy et al. (2010) do not directly derive the metarequirements from their applied behavioral theory, there still is a creative leap between the two.

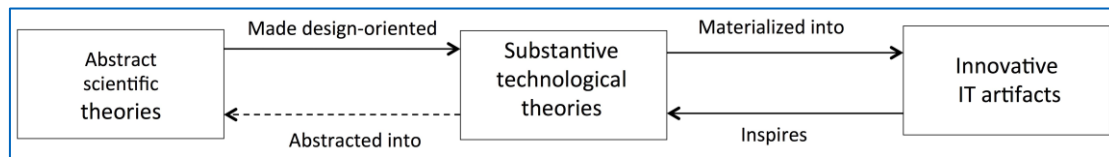
Quite curiously, Kuechler and Vaishnavi (2012) seem to reduce the novelty of design (for the class of IT artifact) to DREPTs: “By linking effects, the causes of which are explained by the kernel theory, with design features, the DREPT explains how and why a design based on the DREPT achieves its desirable novelty” (p. 411). Since novelty is relative to the state of the art (i.e., the existing alternative artifacts), I do not follow this line of argumentation. Rather, I would conjecture that the more innovative and useful (relative to alternatives) the design for a class of IT artifacts turns out to be, the more likely it is that it has novel design assumptions and/or design characteristics (such as design decisions, design qualities, or design features), which may or may not be formalizable into a substantive technological theory.

Figure 3 illustrates this situation. The arrows leading to the right follow the reasoning of Arazy et al. (2010) and Kuechler and Vaishnavi (2012), who view their substantive technological theories as design-oriented concretizations of abstract kernel theories. I suggest an optional reverse process in which an innovative artifact (technology) with its innovative design may inspire the development of new substantive technological theories, which possibly may further be abstracted into scientific theories without design/technology orientation. The research question could be: *What design assumptions and/or characteristics make the IT artifact successful in what contexts?*

<sup>16</sup> As a more minor comment, in their Figure 1, Baskerville et al. (2018) claim that “technology informs science via deeper understanding and generalization (Design Theory).” I do not follow the sense in which they are using “design theory” here. In my opinion, technology informs science without any design theory distinct from the artifact (technology). As an example, information technology has opened new research avenues for behavioral science research in different

disciplines (e.g., psychology, sociology, economics, organization studies, and also information systems).

<sup>17</sup> There was intensive research on design rationales in the 1980s and 1990s (Lee & Lai, 1990) in order to document design rationales ( $\approx$  knowledge) in software engineering. I believe that this idea failed, since it easily led documentation that was more complex than the software code itself.



**Figure 3. The Interplay Between Abstract Scientific Theories, Substantive Technological Theories, and an Innovative IT Artifact**

As noted by Gregor and Hevner (2013), kernel theories are often borrowed from reference disciplines. If borrowed from disciplines outside computing (e.g., computer science, software engineering, information systems), they typically do not have any IT substance. As a consequence, I believe that innovative and successful (e.g., useful) artifacts provide an excellent opportunity for IS and DSRIS to make theoretical contributions—in the sense of “theories for explaining,” “theories for predicting” and “theories for explaining and predicting”—at the level of substantive technological theories. Therefore, theorizing (March & Smith 1995), in terms of substantive technological theories is important—either during the DSR project or after its conclusion, by the initial researchers or by outsiders. It may be a matter of taste whether this theorizing should be considered behavioral science research, especially when conducted afterwards by outsiders, or part of a wider DSR program when conducted afterwards by the initial researchers. If it is considered DSR, an innovative artifact should serve as the focus of theorizing so that the research effort is not wasted on theorizing about all manner of IT artifacts (e.g., spoons—see Appendix A).

## 6 Discussion and Concluding Comments

This editorial is based on two general principles. First, I believe that scientific discourse should be conceptually as clear as possible. My analysis above indicates that the DSRIS literature is not strong in this respect. Second, I believe that scholars should be strongly committed to promoting truth and trust in science (Sztompka 2007). Truth implies sincerity, honesty, and modesty in scholarly argumentation. Unfortunately, truth and trust are under threat in modern science because of academic capitalism, fierce competition for research funding, commodification of research results, and the bureaucratization of science (Sztompka 2007, Münch 2016, Chubb & Watermeyer 2017). In comparison to these great threats to academic life, uses of “theory” and “design theory” in DSRIS would seem to be quite

innocent violations of scientific honesty and modesty; nevertheless, they direct scholars away from the ideal of truth rather than toward it.

In contrast to editorials by Baskerville et al. (2018) and Peffers et al. (2018) in which design theory is frequently referred to as if it were an uncontested or clear concept, I demonstrate that the concept of “design theory” is very ambiguous. Therefore, I contrast my ideas here at a more general level.

I agree with Baskerville et al. (2018) on the issues of science-technology interaction and evolution. I also agree with them that in the DSR context, IS research has been too occupied with “nondesign” (kernel) theories at the expense of the novelty of the design knowledge. Nonetheless, their editorial is, in spirit, very design theory-oriented; in contrast, I suggest that the whole concept should be used with care. They see design theory as the most mature form of representing design knowledge, and their discussion of research impacts gives the impression that all design knowledge constitutes “some form of design theory” (p. 367). However, most conceptions of design theory comprise kernel theories. So, the very idea of design theory pushes authors, reviewers, and editors to look for kernel theories as an indicator of the quality of the paper, which is contrary to Baskerville et al.’s opinion above regarding attention to “nondesign” kernel theories.<sup>18</sup>

I think that Baskerville et al. (2018) correctly point out that the development of “design theory” is a long-term goal of DSR. At the same time, they emphasize—under the label “design theory”—that some degree of design theorizing should be expected in every DSR paper (p. 368). Again, this easily pushes the authors to conduct design theorizing with the hope of ending up with acceptable “nascent design theory.” I would claim that design theorizing—both in its interior mode and in its exterior mode (Baskerville et al., 2012)—can be done without any purpose suggestive of design theory. The interior mode involves producing “prescriptive statements about how artifacts can be designed, implemented, and evaluated” (p. 363). In the IS context, this is usually referred to as IS development

<sup>18</sup> It is not clear to me that metarequirements and metadesign describing a class of IT artifacts are less well-developed than “nascent design theories” such as design principles.

According to Walls et al. (1992) metarequirements and metadesign are the core constituents of design theories.

methods or techniques.<sup>19</sup> It is unnecessary to pretend that they are any special “design theories.” The exterior mode aims “at analyzing, describing, and predicting what happens as artifacts exist and are used in their external environments” (p. 363). The explicit purpose of theorizing is not necessarily to produce a coherent explanation that could be reasonably considered a kernel theory (or a set of kernel theories) by most conceptions of design theory.

It is also challenging to combine the long-term DSR goal of developing “design theory” with the rapid development of technology. This would obviously require the identification of technological invariants that surpass the temporary technologies in fashion at any given time (Iivari, 2014). Furthermore, the significance of technological ideas may only be realized years later. For example, the object-oriented features of Simula 67 (Dahl, Myhrhaug, & Nygaard, 1968) were recognized about 10 years later when Alan Kay and his colleagues had adopted them in Smalltalk language and introduced them to a wider audience. Therefore, it might be wiser to consider “design theories” as some sort of theoretical ideals (Hall of Fame achievements), normally up to the next generation of researchers to propose if the technologies are still of interest. Proposing design theory concurrently with an innovative artifact is a great accomplishment but should not be expected or required.

Peffer et al. (2018) provide another interesting contrast to this editorial. They note that while DSR has become a well-accepted research orientation within IS, there are different views among authors, reviewers and editors regarding the expected outcomes of DSR submissions, the research method to be applied, and the role of theory in argumentation. To address such issues, they propose five prototype genres, two that are design theory-oriented (Walls et al., 1992 and Gregor & Jones, 2007) and (Baskerville & Pries Heje, 2010 and Niehaves & Ortbach, 2015), two that suggest specific DSR methods (Peffer, Tuunanen, & Rothenberger, 2007; Sein et al., 2011), and one that is a version of DSR that is applied in German-speaking countries. They propose that each DSR submission could identify its genre and that it should be reviewed and evaluated using the standards of the genre in

question.

Within this framework of these five genres, this editorial addresses the two design-oriented genres. Peffer et al. (2018) do not recognize the inconsistency between “design theory” in Walls et al. (1992) and “design theory” in Gregor and Jones (2007), nor do they recognize this inconsistency in Baskerville and Pries-Heje (2010) versus Niehaves and Ortbach (2015). They also seem to interpret “design theory” in a surprising way: “An IS design theory (ISDT) can be understood as similar to a behavioral science theory. It enables the IS design researcher to communicate design theory, independent of the applied science, from whence it was derived” (Peffer et al., 2018, p. 131). I have always read Walls et al. (1992), Gregor (2006), and Gregor and Jones (2007) as suggesting that design theories form a special type of theory that differs from explanatory/predictive theories typical to natural and behavioral sciences. Furthermore, I argue that the core of design theory is the IT artifact (Walls et al., 1992), which likely reveals something about the “applied science” from which design theory derives.<sup>20</sup>

Taken more broadly, I would claim that the two *design theory-oriented* genres, on the one hand, and the two *research method-oriented* genres, on the other hand, are logically orthogonal to each other, meaning that papers could, at least in principle, combine ideas belonging to both pairs of genres. At the time of writing (September 2019), Google Scholar returned approximately 500 hits for design theory and ADR (action design research, Sein et al. 2011). My guess is that many of these papers apply ADR and propose design theories (e.g., Giessmann & Legner, 2016). Thus, I would expect that the authors would have difficulties in identifying one and only one genre for their submissions. Forcing them to do so would be problematic since DSR outcomes and DSR methods are actually orthogonal to each other.

In conclusion, I suggest that DSRIS should avoid the fetishistic use of “theory” and, in particular, “design theory,” as exemplified by, for example, “nascent design theory.” In general, it is possible to simply talk about specific classes of IT artifacts in question and related “design knowledge” without losing anything essential. If we accept that new design knowledge is valid as a DSR contribution, then the purpose of DSR

<sup>19</sup> Recent DSRIS papers often propose a set of design principles, characterizing them “nascent design theory.” Unfortunately, it is often quite difficult to determine their completeness and consistency (Prat, Comyn-Wattiau, & Akoka, 2015). The completeness and consistency of models and methods (Hevner et al. 2004) are actually easier to evaluate since they are not simple lists and there are reference models (e.g., Olle et al. 1988, Wand and Weber 1990) and metamodeling formalisms (Tolvanen, Rossi, & Lui, 1996) that can be used to analyze them.

<sup>20</sup> Contrary to Baskerville et al. (2018), Peffer et al. (2018) do not particularly emphasize that DSR attempts to develop new, innovative artifacts. Does this imply that, in their opinion, DSR contributions may consist of “design theories” developed to describe existing artifacts such as spoons or anything else? Is it even DSR anymore if it does not involve the building of innovative artifacts at all? And, how would such an interpretation affect the practical relevance of DSR?



is to produce *new design knowledge—both design product knowledge and/or design process knowledge—about novel IT artifacts with practical utility*. As such, authors and reviewers should focus on the newness of the design knowledge and the novelty of the artifact and its practical utility when authoring and reviewing DSR submissions.

I am not categorically against “theory” and “design theory” in DSRIS but, in my opinion, they should be used more sparingly and with more care than is currently done. It is time for collective reflection and we are not particularly late with it. Even though more than 25 years have gone by since Walls et al. (1992) published their seminal work, the real boom of “design theory” started about ten years ago. And, compared to management studies scholars—who have repeatedly found it necessary to go back to the basics to remind themselves “What constitutes a theoretical contribution?” (Whetten, 1989), “What theory is not?” (Sutton & Staw, 1995), and even to express concerns about a theory fetish (Hambrick, 2007) 30-40 years after the “theory-turn” in their discipline—we are well in time.

In conclusion, I would like to sum up the major points of this editorial in eight guidelines.

1. It is justified to use the concept “design theory” only when it is grounded on sound kernel theories (Iivari, 2007).
2. It is up to the research community of the discipline where the kernel theory was originally proposed to determine the soundness of kernel theories (Truex et al., 2006).
3. When writing/speaking about “design theories,” make the meaning in which you use the term clear:
  - a) Design Theory 1: theoretical origin of metarequirements and metadesign for the IT artifact (Walls et al., 1992)
  - b) Design Theory 2: the relationship between metadesign and metarequirements of the IT artifact (Baskerville & Pries-Heje, 2010)
  - c) Design Theory 3: the relationship between

the class artifacts (as defined by metarequirements and metadesign) and the effectiveness of the artifacts (Venable, 2006)

- d) Design Theory 4: a union of Design Theories 1-3.
4. Each of the three design theories (1-3) may have their own kernel theories.
5. Existing kernel theories are not necessarily concrete enough to be directly applied as justification of Design Theories 1-3 but must be concretized into substantive technological theories (Aryzi et al., 2010; Kuechler & Vaishnavi, 2012).
6. Substantive technological theories are also kernel theories and, consequently, should be sound.
7. Innovative design for a class of IT artifacts in DSRIS is not necessarily based on any sound kernel theories (March & Smith, 1995; Baskerville et al., 2018).
8. The more innovative and useful (relative to alternatives) the design for a class of IT artifacts turns out to be, the more likely it has novel design assumptions and/or design characteristics that may or may not be elaborated into a substantive technological theory.

I believe that following these guidelines would make DSRIS more considerate, more transparent, more truthful, and more scholarly. Or, should scholars be opposed to following these guidelines, I hope that, at the very least, they stimulate healthy debate.

## Acknowledgments

I wish to express my gratitude to Professor Suprateek Sarker, past editor-in-chief of *Journal for the Association of Information Systems*, and to the three reviewers for their comments on earlier versions of this editorial. Many of their critical comments forced me to clarify, sharpen, and sometimes tone down my points of criticism. Finally, I wish to thank Monica Birth for the excellent editing of my text.

## References

- Arazy, O., Kumar, N., & Shapira, N. (2010). A theory-driven design framework for social recommender systems. *Journal of the Association for Information Systems*, 11(9), 455-490
- Avison, D., & Malaurent, J. (2014). Is theory king?: Questioning the theory fetish in information systems. *Journal of Information Technology*, 29, 327-336
- Baskerville, R. L., & Pries-Heje, J. (2010). Explanatory design theory. *Business & Information Systems Engineering*, 2(5), 2010, 271-282
- Baskerville, R., Baiyere, A., Gregor, S., Hevner, A., & Rossi, M. (2018). Design science research contributions: Finding a balance between artifact and theory. *Journal of the Association for Information Systems*, 19(5), 358-376
- Benbasat, I., & Weber, R. (1996). Research Commentary: Rethinking “diversity” in information systems research. *Information Systems Research*, 7(4), 1996, 389-399
- Boon, M. (2006). How science is applied in technology. *International Studies in Philosophy of Science*, 20(1), 2006, 27-48
- Bunge, M. (1966). Technology as applied science. *Technology and Culture*, 7(3), 329-347
- Chubb, J., & Watermeyer, R. (2017). Artifice or integrity in the marketization of research impact? Investigating the moral economy of (pathways to) impact statements within research funding proposals in the UK and Australia. *Studies in Higher Education*, 42(12), 2017, 2360-2372
- Codd, E. F. (1970). A relational model of data for large shared data banks. *Communications of the ACM*, 13, 377-387
- Corley, K. G., & Gioia, D. A. (2011). Building theory about theory building: what constitutes a theoretical contribution? *Academy of Management Review*, 36(1), 12-32
- Cross, N. (2001). Designerly ways of knowing: Design discipline versus design science. *Design Issues*, 17(3), 2001, 49-55
- Dahl, O.-J., Myhrhaug, B., & Nygaard, K., Some features of the Simula 67 language, *Proceedings of the Second Conference on Applications of Simulations*, 1968, 29-31
- Dennis, A. R., & Valacich, J. S. (2014). A replication manifesto. *Transactions on Replication Research*, 1(1), 1-4
- 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
- Freeman, P. (1987). *Software perspectives, the system is the message*. Addison-Wesley,
- Gardner, P. L. (1994). The relationship between technology and science: Some historical and philosophical reflections: Part 1. *International Journal of Technology and Design Education*, 4(2), 1994, 123-153
- Gardner, P. L. (1995). The relationship between technology and science: Some historical and philosophical reflections: Part 2. *International Journal of Technology and Design Education*, 5(1), 1-33
- Giessmann, A., & Legner, C. (2016). Designing business models for cloud platforms. *Information Systems Journal*, 26, 551-579
- Gregor, S., (2006). The nature of theory in information systems. *MIS Quarterly*, 30(3), 611-642
- 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 for Information Systems*, 8(5), 312-335
- Grover, V., & Lyytinen, K. (2015). New state of play in information systems research: The push to the edges. *MIS Quarterly*, 39(2), 271-296
- Hambrick, D. C. (2007). The field of management’s devotion to theory: Too much of a good thing? *Academy of Management Journal*, 50(6), 1346-1352
- Haynes, S. R., & Carroll, J. M. (2010). The range and role of theory in information systems design research: from concepts to construction. *ICIS Proceedings*.
- Hevner, A. R., March, S. T., Park, J., & Ram, S. (2004). Design science in information systems research. *MIS Quarterly*, 28(1), 75-105
- Hevner, A. (2007). A three cycle view of design science research. *Scandinavian Journal of Information Systems*, 19(2), 89-92
- Hirschheim, R. (2019). Against theory: With apologies to Feyerabend, *Journal of the Association for Information Systems*, 20(9), 1340-1357
- Hooker, J. N. (2004). Is design theory possible? *Journal of Information Technology Theory and Application*, 5(2), 73-82

- Iivari, J. (2007). A paradigmatic analysis of information systems as a design science. *Scandinavian Journal of Information Systems*, 19(2), 39-63
- Iivari, J. (2010). Twelve theses on design science research in information systems. In A. Hevner, & A. Chatterjee (Eds.), *Design science a research paradigm for information* (pp. 43-62). Springer.
- Iivari, J. (2014). Information systems as a practical discipline. In H. Topi & A. Tucker (Eds.), *Computing handbook: Information systems and information technology* (3rd ed., pp. 3.1-3.14). CRC Press.
- Iivari, J. (2015). Distinguishing and contrasting two strategies for design science research. *European Journal of Information Systems*, 24(1), 107-115
- Keen, P. G. W., & Scott Morton, M. (1978). *Decision support systems: An organizational perspective*. Addison-Wesley.
- Kuechler, W., & Vaishnavi, V. (2008a). The emergence of design research in information systems in North America. *Journal of Design Research*, 7(1), 1- 16
- Kuechler, W., & Vaishnavi, V. (2008b). On theory development in design science research: Anatomy of a research project. *European Journal of Information Systems*, 17(5), 489-504
- Kuechler, W., & Vaishnavi, V. (2012). A framework for theory development in design science research: multiple perspectives science research. *Journal of the Association for Information Systems*, 13(6), 2012, 395-423
- Layton, E. T. Jr. (1974). Technology as knowledge. *Technology and Culture*, 15, 31-41
- Lee, J., & Lai, K.-Y. (1991). What's in design rationale? *Human-Computer Interaction*, 6, 251-280
- Lim, S., Saldanha, T. J. V., Malladi, S., & Melville, N. P. (2013). Theories used in information systems research: Insights from complex network analysis. *JITTA: Journal of Information Technology and Application*, 14(2), 5-44.
- Lowenthal, P. R. (2010). *Social computing: concepts, methodologies, tools and applications*. IGI Global.
- March, S. T., & Smith, G. F. (1995). Design and natural science research on information technology. *Decision Support Systems*, 15, 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.
- Merton, R. K. (1949). On sociological theories of the middle range. In *Social theory and social structure* (pp. 39-53). The Free Press
- Münch, R. (2016). Academic capitalism. In *Oxford research encyclopedia of politics*. Oxford University Press.
- Niehaves, B., & Ortbach, K. (2016). The inner and the outer model in explanatory design theory: The case of designing electronic feedback systems, *European Journal of Information Systems*, 25, 303-316.
- Nunamaker, J. F., Chen, M., & Purdin, T. D. M. (1990). System development in information systems research. *Journal of Management Information Systems*, 7(3), 99-106
- Olle, T. W., Hagelstein, J., Macdonald, I. G., Rolland, C., Sol, H. K., Van Assche, F. J. M., & Verrijn-Stuart, A. A. (1988). *Information systems methodologies: A framework for understanding*, Addison-Wesley.
- Peffer, K., Tuunanen, T., Rothenberger, M.A., & Chatterjee, S. (2007). A design science research methodology for information systems research. *Journal of Management Information Systems*, 24(3), 45-77
- Peffer, K., Tuunanen, T., & Niehaves, B. (2018). Design science research genres: Introduction to the special issue on exemplars and criteria for applicable design science research, *European Journal of Information Systems*, 27(2), 129-139
- Prat, N., Comyn-Wattiau, I., & Akoka, J. (2015). A taxonomy of evaluation methods for information systems artifacts. *Journal of Management Information Systems*, 32, 229-267
- Searle, J. R. (1979). *Expression and meaning: Studies in the theory of speech acts*, Cambridge University Press.
- Sein, M. K., Henfridsson, O., Purao, S., Rossi, M., & Lindgren, R. (2011) Action design research. *MIS Quarterly*, 35(1), 37-56
- Sztomka, P. (2007). Trust in science: Robert K. Merton's inspirations. *Journal of Classical Sociology*, 7(2), 2007, 211-220.
- Sutton, R. I., & Staw, B. M. (1995). What theory is not. *Administrative Science Quarterly*, 40(3), 371-384
- Tolvanen, J.-P., Rossi, M., & Liu, H. (1996). Method engineering: Current research directions and implications for future research. In S. Brinkkemper, J. Lyytinen, & R. Welke (Eds.), *Method engineering: principles of method construction and tool support* (pp. 296-317). Springer.

- Truex, D., Holmström, J., & Keil, M. (2006). Theorizing in information systems research: A reflexive analysis of the adaptation of theory in information systems research. *Journal of the Association for Information Systems*, 7(12), 797-821
- Urquhart, C., Lehman, H., & Myers, M. D. (2010). Putting the “theory” back into grounded theory: Guidelines for grounded theory studies in information systems. *Information Systems Journal*, 20, 357-381.
- Van Aken, J. E. (2004). Management research based on the paradigm of the design sciences: The quest for field-tested and grounded technological rules. *The Journal of Management Studies*, 41(2), 219-246.
- Venable, J. R. (2006). The role of theory and theorising in design science research. *Proceedings of the First International Conference on Design Science Research in Information Systems and Technology*.
- Venable, J. R. (2013). Rethinking design theory in information systems. *Proceedings of the International Conference on Design Science Research in Information Systems and Technology*.
- Walls, J., Widmeyer, G. R., & El Sawy, O. A. (1992). Building an information system design theory for vigilant EIS. *Information Systems Research*, 3(1), 36-59
- Walls, J. G., Widmeyer, G. R., & El Sawy, O. A. (2004). Assessing information system design theory in perspective: How useful was our 1992 initial rendition? *JITTA: Journal of Information Technology Theory and Application*, 6(2), 43-58.
- Wand, Y., & Weber, R. (1990). An ontological model of an information system. *IEEE Transactions on Software Engineering*, 16(11), 1282-1292.
- Whetten, D. A. (1989). What constitutes a theoretical contribution? *The Academy of Management Review*, 14(4), 490-495



## Appendix A

**Table A1. Conceptual Mapping Between Major Conceptions of Alternative Design Theories**

Example	Justificatory knowledge	Metarequirements	Metadesign	Effects
Design theory for spoons	Physics (mechanics) Human physiology	Must enable moving liquid food to the human mouth. Must be convenient to handle. Etc.	Has a concave bowl and a handle. The size of bowl is such that it fits in the human mouth (breath 5.0 cm at the maximum). The bowl is symmetrical so that it can be divided into symmetrical halves by a straight line. The handle is connected to the bowl in the direction of the symmetry line at the narrower end of the bowl. The length of the handle is 8.5 cm at the minimum and 2.2 times the length of the bowl at the maximum. The handle is headed 15 degrees up when compared with bowl in the horizontal position (the concave side upwards) The handle has a grasping area, the width of which is 10-15 % of the breath of the bowl. Etc.	Efficiency of moving liquid food to the mouth. Percentage of wasted liquid food due to spilling. User satisfaction with a spoon.
Walls et al. (1992)	Kernel theories	Metarequirements	Metadesign	
Venable (2006)			Solution technologies	Problem understanding
Gregor & Jones (2007)	Justificatory knowledge	Principles of function	Principles of form	Purpose and scope
Baskerville & Pries-Heje (2010)		General requirements	General components	
Niehaves & Ortbach	Kernel theories		Design model	Effects

## About the Authors

**Juhani Iivari** is a professor emeritus in the Department of Information Processing Science, University of Oulu, Finland. During his career he has served as a professor at the University of Jyväskylä and at the University of Oulu. Before his retirement, he also worked for ten years as a part-time scientific head of INFWEST/INFORTE programs, which are joint efforts of a number of Finnish universities to support doctoral studies in IT. Juhani has also served in various editorial positions for IS journals including *Communications of the Association for Information Systems*, *European Journal of Information Systems*, *Information Systems Journal*, *Information Systems and e-Business Management*, *Information Technology and People*, *Journal of the Association for Information Systems*, *MIS Quarterly*, and *Scandinavian Journal of Information Systems*. His research has broadly focused on the theoretical foundations of information systems, IS development methods and approaches, organizational analysis, implementation and acceptance of information systems, and design science research in IS.

Copyright © 2020 by the Association for Information Systems. Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and full citation on the first page. Copyright for components of this work owned by others than the Association for Information Systems must be honored. Abstracting with credit is permitted. To copy otherwise, to republish, to post on servers, or to redistribute to lists requires prior specific permission and/or fee. Request permission to publish from: AIS Administrative Office, P.O. Box 2712 Atlanta, GA, 30301-2712 Attn: Reprints, or via email from [publications@aisnet.org](mailto:publications@aisnet.org).